Digitalisierung eines Pen-&-Paper-Rollenspiels mit Übertragung von Interaktionen in die reale Welt

Antony Konstantinidis
Nicolas Kopp

Publisher: Dean Prof. Dr. Wolfgang Heiden

University of Applied Sciences Bonn-Rhein-Sieg,
Department of Computer Science

Sankt Augustin, Germany

December 2014

Technical Report 05-2014

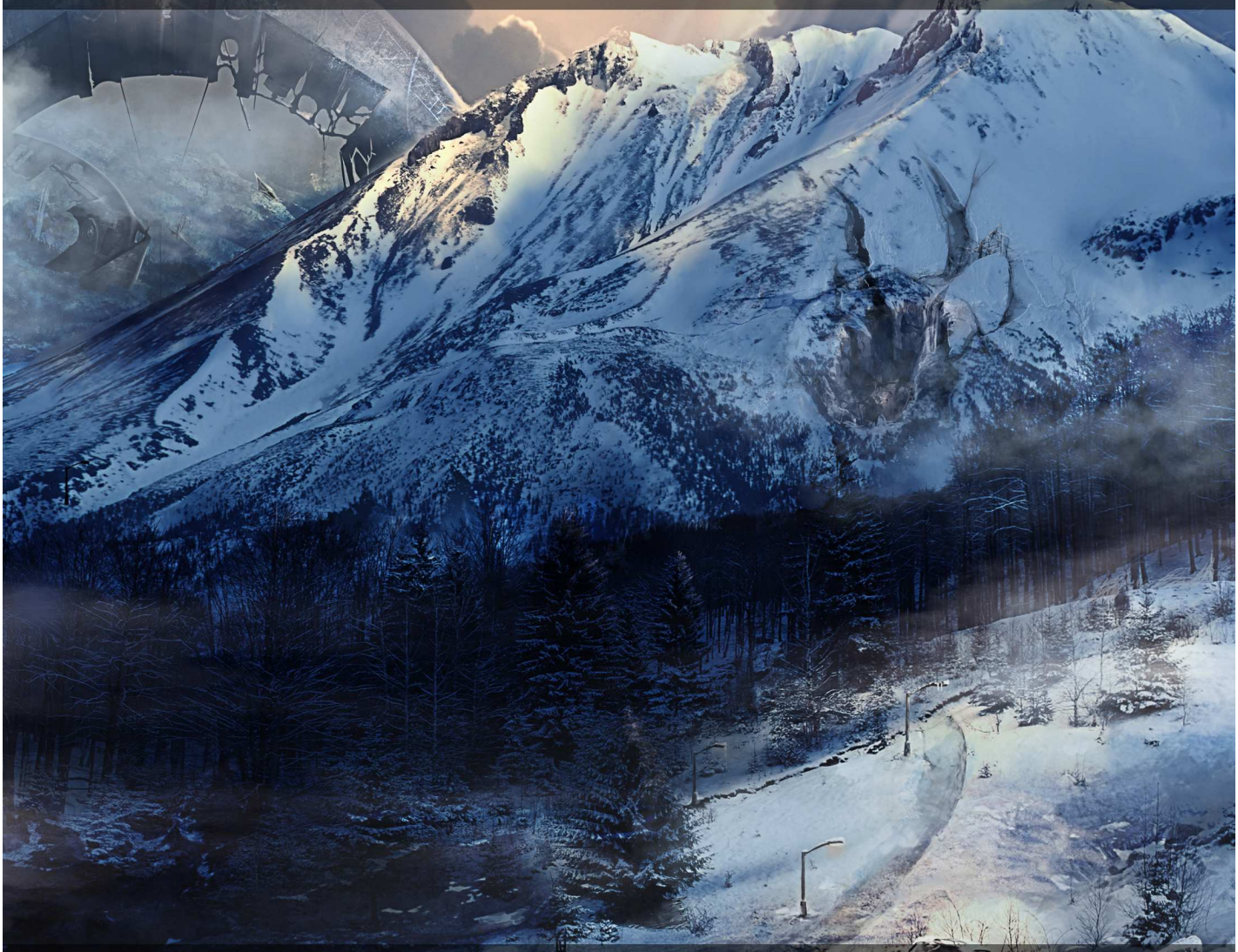
Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

ISSN 1869-5272

Copyright © 2014, by the author(s). All rights reserved. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Das Urheberrecht des Autors bzw. der Autoren ist unveräußerlich. Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Das Werk kann innerhalb der engen Grenzen des Urheberrechtsgesetzes (UrhG), *German copyright law*, genutzt werden. Jede weitergehende Nutzung regelt obiger englischsprachiger Copyright-Vermerk. Die Nutzung des Werkes außerhalb des UrhG und des obigen Copyright-Vermerks ist unzulässig und strafbar.

Digitalisierung eines Pen-&-Paper-Rollenspiels mit Übertragung von Interaktionen in die reale Welt



Antony Konstantinidis
Nicolas Kopp
Masterarbeit / Masterprojekt
2013-2014

thinklabs
thinklabs.de

Inhaltsverzeichnis

Abkürzungsverzeichnis IX

Abbildungsverzeichnis XV

Tabellenverzeichnis XXI

Verzeichnis der Quelltexte XXIII

Vorwort zur Werkszusammenfassung XXV

Vorwort zur Ausarbeitung XXVII

Masterprojekt 1

1. Einleitung 3

1.1. Inhalt des Projektes und Erläuterung des Hintergrundes . . . 5

1.1.1. Das Konzept des Pen-&-Paper-Rollenspiels . . . 6

1.2. Motivation, Konzept und Zielsetzung . . . 7

1.3. Stand der Wissenschaft und Forschung . . . 8

1.3.1. Projektabgrenzung . . . 11

1.4. Aufbau der Arbeit . . . 12

2. Grundlagen 13

2.1. Charakterisierung eines Rollenspiels . . . 14

2.1.1. Game Master . . . 15

2.1.2. Regelwerk . . . 15

2.1.3. Spieldesign . . . 16

2.1.4. Hilfsmittel . . . 17

2.1.5. Identifizierte Kernelemente eines Pen-&-Paper-Rollenspiels . . . 18

2.2. Game-Engines . . . 19
2.2.1. Cry Engine SDK . . . 23
2.2.2. Unreal Development Kit . . . 24
2.2.2.1. UnrealScript . . . 26
2.3. Netzwerkprogrammierung . . . 27
2.3.1. Definition und Kontextualisierung von Netzwerken . . . 27
2.3.2. Netzwerkprotokolle . . . 30
2.3.2.1. Transmission Control Protocol (TCP) . . . 32
2.3.2.2. User Datagram Protocol (UDP) . . . 34
2.3.3. Netzwerkspielmodelle . . . 35
2.3.4. Bandbreitenvergleich: Peer-2-Peer - Client-Server . . . 37
2.3.5. Netzwerkprogrammierung in Java . . . 38
2.3.6. Netzwerkkommunikation . . . 40
2.3.6.1. Blockierende Sockets . . . 41
2.3.6.2. Polling-Ansatz . . . 41
2.3.6.3. Asynchrone Sockets . . . 41
2.3.6.4. Threadbasierte Verfahren . . . 42
2.4. Threads . . . 42
2.5. Android . . . 43
2.5.1. Architektur . . . 46
2.5.2. Komponentenmodell . . . 47
2.5.3. Threads in Android . . . 48

3. Realisierung **51**
3.1. Designentscheidungen . . . 51
3.1.1. Digitalisierung von Regelwerk und Spieldesign . . . 52
3.1.2. Auswahl der Game-Engine . . . 53
3.1.3. Digitalisierung des Game Masters . . . 55
3.1.4. Digitalisierung von Spielern und Hilfsmitteln zur Visualisierung . . . 58
3.1.5. Wahl des passenden Netzwerkprotokolls . . . 62
3.1.6. Wahl des Netzwerkspielmodells . . . 64
3.1.7. Wahl des mobilen Betriebssystems . . . 65
3.1.8. Wahl der Programmiersprachen . . . 66
3.2. Konzept . . . 67
3.2.1. Smartphone-Clients . . . 69
3.2.2. Workstation . . . 72

3.2.3. Java-Server . . . 72
3.2.3.1. Threadproblematik . . . 75
3.2.4. WLAN-Router . . . 77
3.2.5. Game Master - Schnittstellenkonzeption . . . 78
3.2.5.1. Anwendungsfall des Game Masters und der Decision Engine . . . 84
3.2.6. Game-Engine-Client . . . 86
3.2.6.1. Realisierung mehrerer Spieler . . . 87
3.2.6.2. Kalkulation der Kamera . . . 89
3.2.7. Kommunikationsprotokoll . . . 90
3.2.7.1. Byte-basierter Ansatz . . . 91
3.2.7.2. JSON-basierter Ansatz . . . 91

4. Zusammenfassung und Fazit **95**

Masterarbeit **99**

5. Einleitung **101**
5.1. Zielsetzung und Motivation der Arbeit . . . 104
5.1.1. Motivation der Themenwahl . . . 104
5.1.1.1. Exkurs: Das Pen-&-Paper-Rollenspiel . . . 105
5.2. Stand der Wissenschaft und Forschung . . . 107
5.2.1. Hintergrund und zur Verfügung stehende Basis . . . 107
5.2.2. Ausgangspunkt und Ziele der Masterarbeit . . . 110
5.2.2.1. Umsetzung einer Mixed Reality-Anwendung . . . 110
5.2.2.2. Modulares Storytelling . . . 110
5.2.2.3. Verfeinerung des Netzwerkmodells . . . 111
5.2.2.4. Entscheidungsmaschine . . . 112
5.2.3. Vergleichbare Arbeiten . . . 112
5.3. Aufbau der Arbeit . . . 115

I. Grundlagen **119**

6. Interaktionen **121**
6.1. Metapherhaftigkeit von Interaktionen . . . 124

6.2. Der allgegenwärtige Computer . . . 124
6.2.1. Formen interaktiver Systeme . . . 126
6.3. Vom Touchscreen zum Smartphone als Interaktionsgerät . . . 129
6.4. Kooperative und kollaborative Interaktionen . . . 134

7. Mixed Reality **139**
7.1. Taxonomie der Mixed Reality . . . 141
7.1.1. Zusammenfassung . . . 144
7.2. Das Prinzip von DIBA-PIBA . . . 145
7.3. PIGA - „Physical Is Good At“ . . . 146
7.4. DIGA - „Digital Is Good At“ . . . 150
7.5. Zusammenfassung . . . 153

8. Storytelling in Spielen **155**
8.0.1. Definitionsabgrenzung . . . 156
8.1. Storytelling in verschiedenen Medien . . . 157
8.2. Erzählstrukturen . . . 161
8.2.1. Lineares Storytelling . . . 161
8.2.2. Verzweigungen (in linearer Ausführung) . . . 162
8.2.3. Netzform (Simple Form der Nichtlinearität) . . . 164
8.2.3.1. Die Hypermedia Novel als Anwendungsbeispiel der Netzform . . . 166
8.3. Modulares Storytelling . . . 168
8.4. Zusammenfassung . . . 172

9. Künstliche Intelligenz **175**
9.1. Künstliche Intelligenz im Bereich der Spieleprogrammierung . . . 176
9.2. Entscheidungsfindung . . . 177
9.3. Ansätze zur Entscheidungsfindung . . . 178
9.3.1. Skripting . . . 179
9.3.2. Finite State Machines . . . 180
9.3.3. Spiel-/Entscheidungsbäume . . . 182
9.3.4. Fuzzy-Logik . . . 183
9.3.5. Regelbasierte Systeme . . . 186
9.3.6. Neuronale Netze . . . 189
9.3.7. Zielorientierte Ansätze . . . 189
9.4. Zusammenfassung . . . 190

10. Kamerasysteme in Spielen 191
10.1. Third-Person und First-Person . . . 191
10.2. Top-Down und Side-Scrolling . . . 193
10.3. Isometrische Kamera . . . 194
10.4. Zusammenfassung . . . 195

11. Netzwerkgrundlagen 197
11.1. Netzwerke in Computerspielen . . . 197
11.1.1. Probleme . . . 200
11.2. Netzwerkprotokolle . . . 201
11.3. Transmission Control Protocol (TCP) . . . 204
11.3.1. TCP in Netzwerkspielen . . . 208
11.4. Zusammenfassung . . . 210

II. Verwandte Arbeiten und Konzept 213

12. Verwandte Arbeiten 215
12.1. Mixed Reality und Interaktionen . . . 215
12.1.1. Virtuelle Kollaborationsplattform . . . 215
12.1.2. iGameFloor . . . 216
12.1.3. QuakeRunner . . . 217
12.1.4. CREWSpace . . . 217
12.1.4.1. Abgrenzung zur Arbeit . . . 218
12.2. Pervasive Games . . . 219
12.2.1. Abgrenzung zur Arbeit . . . 221
12.3. Tangible User Interfaces . . . 221
12.3.1. Golem Arcana . . . 223
12.3.1.1. Abgrenzung zur Arbeit . . . 224
12.3.2. KnightMage . . . 224
12.3.2.1. Abgrenzung zur Arbeit . . . 226

13. Mixed Reality-Konzeption 227
13.1. Die vierte Wand als Volumen . . . 229
13.2. Das Smartphone als Mixed Reality-Interaktionsgerät . . . 230
13.3. MR-Definition . . . 232

14. Storytelling- und Gameplay-Konzeption **235**
14.1. Storytelling-Konzeption . . . 235
14.1.1. Modulares Storytelling in Computerspielen . . . 237
14.1.2. Verschachtelte Module . . . 238
14.2. Hintergrundgeschichte . . . 240
14.2.1. Shadowrun . . . 241
14.2.2. Hintergrundgeschichte der Implementierung . . . 243
14.3. Realisiertes Storytelling und Gameplay . . . 244
14.3.1. Interaktionsformen . . . 248

15. Interaktionskonzeption **251**
15.1. Kopplung und Metapherhaftigkeit von Interaktionen . . . 252

16. Konzeption der künstlichen Intelligenz **257**
16.1. Wahl des passenden Entscheidungsfindungsansatzes . . . 257
16.2. Konzeption des Kamerasystems . . . 261

17. Netzwerkkonzeption **265**

III. Realisierung **267**

18. Übersicht der Anwendungsarchitektur **269**
18.1. Kommunikationsprotokoll . . . 271
18.2. Trigger-Objekte . . . 273

19. Nachrichtenagenten **275**
19.1. Interaction-Agent . . . 277
19.1.1. Smartphone- und Game-Engine-Clients . . . 281
19.2. MR-Agent . . . 282
19.3. Game Master . . . 284
19.3.1. Agent . . . 285
19.3.2. Decision Engine . . . 285

20. Server **287**
20.0.3. Beliebige Verbindungsreihenfolge . . . 287
20.0.4. Lobbyfunktionalität . . . 289

21. Smartphone-Clients **291**
21.0.5. Kommunikation innerhalb der Anwendung . . . 292
21.0.6. Interaktionsansichten . . . 293
21.0.7. Erweiterung des Anmeldemechanismus . . . 294

22. Game-Engine-Client **297**
22.1. Grundlegende Implementierung des Gameplays . . . 298
22.2. Realisierung der Navigation . . . 301

23. Realisierung des gewählten Kamera-Systems **305**
23.1. Berechnung des Kamera-Bezugspunkts . . . 307
23.2. Zoom und Kamera-Kollision . . . 311
23.3. Verbindung von Interaktionen, Story und Gameplay . . . 312

IV. Evaluation und Konklusion **317**

24. Evaluation **319**
24.1. Eingesetzte Netzwerktopologie . . . 319
24.1.1. Vorgehen . . . 320
24.1.2. Ergebnisse . . . 322
24.2. Infrastrukturelle Problemstellungen . . . 325
24.3. Mixed Reality-Anwendung . . . 328
24.3.1. Vorgehen . . . 328
24.3.2. Ergebnisse . . . 331
24.3.3. Fazit . . . 335

25. Zusammenfassung **339**

26. Ausblick **345**

Glossar **349**

Literaturverzeichnis **363**

A. Bilderserie der realisierten Anwendung **i**

B. Illustrationen **xvii**

C. Weiterführende Informationen **xxix**

C.1. Arten von Pervasive Games xxix

C.2. Das Projekt „Blinkenlights“ xxxi

Abkürzungsverzeichnis

Das Abkürzungsverzeichnis beinhaltet eine Auflistung von wichtigen Abkürzungen der Arbeit mit entsprechenden Seitenangaben, auf denen die Abkürzungen unter anderem verwendet werden. Zu vielen der Begriffe existieren darüber hinaus Glossareinträge, welche weiterführende Erklärungen beinhalten. Das Glossar ist am Ende dieser Ausarbeitung zu finden.

AR [xxix, 219, 220]

Augmented Reality

BCI [xxix]

Brain-Computer-Interface(s)

BITKOM [230]

Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V.

BIU [102]

Bundesverband Interaktive Unterhaltungssoftware e.V.

CCC [xxxi]

Chaos Computer Club

CGI [142, 143, 191]

Computer Generated Imagery

CSCW [140, 218]

Computer Supported Cooperative Work

DE [7, 8, 109, 273, 275, 276, 284–286, 342]

Decision Engine

DIBA-PIBA [145, 153]

Digital Is Better At - Physical Is Better At

DVM [47]

Dalvik Virtual Machine

EJB [47]

Enterprise JavaBeans

FOV [306]

Field of View

FSM [180, 260, 302]

Finite State Machine

GM [6, 15, 78, 105, 275, 284, 359]

Game Master

GPS [219]

Global Positioning System

GUI [140]

Graphical User Interface

HCI [116, 121, 123, 124, 126, 129, 140, 251, 339, 346]

Human-Computer Interaction

HMD [xxix]

Head-Mounted Display(s)

IP [30, 202]

Internet Protocol

ISP [38]

Internet Service Provider

JSON [84, 271, 294]

JavaScript Object Notation

KI [114, 175, 257, 280, 347]

Künstliche Intelligenz

LAN [35, 198, 199]

Local Area Network

LARP [221]

Live Action Role Playing (Game)

MABA-MABA [145]

Machines Are Better At - Men Are Better At

MMO [18]

Massive Multiplayer Online (-Role Playing Game, in dem Falle MMORPG)

MMORPG [33, 158, 208]

Massive Multiplayer Online Role Playing Game

MR [61, 84, 95, 104, 107, 116, 138, 140, 144, 219, 224, 230, 248, 264, 282, 305, 336, 341, 346]

Mixed Reality

MTCP [327]

Multipath TCP

MTU [33, 207, 265]

Maximum Transmission Unit

NIO [40]

New I/O

NPC [15, 88, 170, 172, 177, 179, 190, 299, 349, 359]

Non Player Character

OSGi [47]

Open Services Gateway initiative

OSI [202]

Open Systems Interconnection Model

PIBA [147]

Physical Is Better At

PIGA-DIGA [116, 145, 153]

Physical Is Good At - Digital Is Good At

POI [307]

Point of Interest

RMI [281]

Remote Method Invocation

RPC [109, 270, 297, 342, 343]

Remote Procedure Call

RPG [175]

Role Playing Game

RTO [205, 206, 358]

Retransmission Timeout

RTT [323]

Round Trip Time

SDK [23, 24]

Software Development Kit

TCP [32, 117, 202, 265, 320, 325, 326]

Transmission Control Protocol

TUI [140, 221, 224, 226]

Tangible User Interface

UDK [i, 24, 86, 88, 107, 111, 241, 270, 284, 297, 298, 300, 301, 306, 315, 341, 347]

Unreal Development Kit

UDP [32, 202, 203, 320, 326]

User Datagram Protocol

VoIP [216]

Voice over IP

WIMP [126, 140, 221]

Windows, Icons, Menus und Pointer

Abbildungsverzeichnis

1.1. Anzahl der aktiven Spieler bei Steam am 02. August 2013. 5
1.2. Schematische Skizze des geplanten Systems 8
1.3. Parallelen zwischen Pen-&-Paper- und Konsolenspielen 10

2.1. Architektur eines 2D-Spiels, 1994 21
2.2. Architektur eines 3D-Spiels, 2004 22
2.3. Cry Engine Sandbox 23
2.4. UDK Editor 25
2.5. Evolution der Gerätevernetzung 30
2.6. TCP/IP Vierebenenmodelle 31
2.7. Eine Übersicht verschiedener Netzwerktopologien 36
2.8. In Gebrauch befindliche Geräte 2009-2011 (USA) 45
2.9. Eine Übersicht über die Architektur von Android. 46
2.10. Explizite Intents 49

3.1. Verschachtelte Module Beim Storytelling 55
3.2. Game-Master Flowchart 58
3.3. Spielszene aus *Watchdogs* 60
3.4. Anwendungskonzept 68
3.5. Start-Activity der Android-App 69
3.6. App-Architektur 70
3.7. Hovern eines Gegenstands 71
3.8. Multi-threaded Server 73
3.9. Aufbau des Servers 74
3.10. nichtlineares Gameplay 79
3.11. Simples nichtlineares Storytelling 81
3.12. Module einer Story 82
3.13. Schema der verwirklichten Kommunikation mit der GM-Schnittstelle 83
3.14. Aufbau des Anwendungsfalls 85

3.15. Verschlossene Tür . . . 86
3.16. Prototyp - kollaborative Interaktion . . . 87
3.17. Kamerakalkulation in Aktion . . . 89

5.1. Anzahl der weltweit aktiven Spieler bei Steam . . . 103
5.2. Schematische Skizze des bestehenden Systems . . . 108
5.3. Prototyp - kollaborative Interaktion . . . 109
5.4. Parallelen zwischen Pen-&-Paper- und Konsolenspielen . . . 114
5.5. *Golem Arcana*-Spielsituation . . . 116

6.1. Ansicht verschiedener Interaktionssysteme . . . 127
6.2. Aufbau eines Interaktionssystems . . . 128
6.3. Spielszene aus dem Titel *Watch Dogs* . . . 132
6.4. Touch-Gesten . . . 133

7.1. Virtuelles Kontinuum . . . 139
7.2. Uncanny Valley . . . 142
7.3. Renderpipeline . . . 150

8.1. Lineare Spielerfahrung im nichtlinearen Raum . . . 160
8.2. Lineares Storytelling . . . 161
8.3. Verzweigungen (In linearer Ausführung) . . . 163
8.4. Netzform (Simple Form der Nichtlinearität) . . . 165
8.5. Visualisierung des erzeugten Erzählgraphen einer Hypermedia Novel . . . 167
8.6. Modulares Storytelling . . . 168
8.7. Modulares Storytelling – Beispiel . . . 170

9.1. Decision Making . . . 178
9.2. Formale Definition eines endlichen Automaten . . . 181
9.3. Zustandsdiagramm einer FSM . . . 182
9.4. Entscheidungsbaum . . . 184
9.5. Fuzzy Logic . . . 185
9.6. Regelbasierte Systeme . . . 188

10.1. Third-Person-Kamera und First-Person-Kamera . . . 192
10.2. Top-Down-Kamera und Side-Scrolling-Kamera . . . 193
10.3. Isometrische Kamera . . . 194

11.1. Eine eingeschränkte Übersicht verschiedener Netzwerktopologien . . . 199
11.2. OSI-Modell . . . 203
11.3. *Send-/ Receive Window* . . . 205

12.1. *iGameFloor* . . . 216
12.2. *CREWSpace* . . . 218
12.3. *SandScape* . . . 222
12.4. *Golem Arcana* Spielaufbau . . . 223
12.5. *KnightMage* . . . 225

14.1. Fakultätsgleichung möglicher Story-Durchläufe . . . 237
14.2. Verschachtelte Module . . . 239
14.3. Artwork aus *Shadowrun* . . . 242
14.4. Echtezeitansicht der eingestürzten Höhle des Spielanfangs . . . 244
14.5. Aufbau des in dieser Arbeit realisierten Erzählkonstrukts . . . 247

16.1. Spielszene der Konsolenversion von *Diablo 3* . . . 263

18.1. Anwendungsarchitektur . . . 269
18.2. Nachrichtenschema . . . 271
18.3. Trigger-Objekt . . . 274

19.1. Eingehende Game-Engine-Client-Nachrichten . . . 276
19.2. Interaction-Agent . . . 279
19.3. Maximale Zeitdifferenz von Zeitpunkten in der Menge $X_0, \ldots, X_n$ in Sekunden. . . . 280
19.4. Bewegungsansicht mit Möglichkeit zur Interaktion . . . 281
19.5. Sound-Trigger . . . 283
19.6. Berechnung der Lautstärke einer Soundausgabe auf einem Smartphone. . . . 284

20.1. Serverseitige Ausführungsschleife des Game-Engine-Client-Sockets . . . 288
20.2. Multi-threaded Server . . . 290

21.1. App-Architektur . . . 292
21.2. Beispiel einer Interaktionsansicht auf dem Smartphone . . . 293
21.3. Verbindungsaufbau der Smartphone-Clients . . . 295

22.1. Realisierung der Spieler im UDK . . . 300
22.2. Wegpunktnetz im Anfangsraum des Spiels . . . 301

23.1. Zusammenwirken der Klassen zu Berechnung der Kamera im UDK . . . 305
23.2. Berechnung des absoluten Mittelpunkts einer finiten Menge von k Punkten $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$ in $\mathbb{R}^n$. . . . 308
23.3. Unterschiede in der Berechnung eines Kamera-Bezugspunkts . . . 308
23.4. Berechnung des geometrischen Schwerpunkts eines Polygons . . . 309
23.5. Sortiertes versus unsortiertes Polygon . . . 310
23.6. Sortierung der Polygon-Eckpunkte . . . 311
23.7. Visualisierung des berechneten Kamera-Schwerpunkts im Spiel . . . 313
23.8. Gameplay-Ansicht mit photovoltaischer Anlage . . . 314
23.9. Gameplay-Ansicht mit Wand-Interaktion . . . 315

24.1. Netzwerktopologie-Vergleich . . . 320
24.2. Evaluationsmap . . . 321
24.3. Jittervergleich des Prototypen . . . 324
24.4. Ping-Ergebnisse localhost . . . 325
24.5. Evaluation Sterndiagramm . . . 331
24.6. Gruppenmeinungen im Hinblick auf verschiedene Bewertungskategorien . . . 333

25.1. Anwendungsfall Bewegungssteuerung . . . 341
25.2. Anwendungsfall Interaktionsdurchführung . . . 343

26.1. Splitscreen . . . 359
26.2. Whiteboxing . . . 361

B.1. Historische Entwicklung sozialer Spiele . . . xix
B.2. Steam-Satellitenbild . . . xix
B.3. Marktverteilung der mobilen Betriebssysteme in den vergangenen sieben Jahren . . . xx
B.4. 11*ms* Sendeintervall, 40 Events . . . xxii
B.5. 21*ms* Sendeintervall, 20 Events . . . xxii
B.6. 42*ms* Sendeintervall, 10 Events . . . xxiii
B.7. Stanley Parable Ending Tree . . . xxv
B.8. Schaubild von Jon Radoff zur historischen Entwicklung sozialer Spiele. . . . xxvi
B.9. Der deutsche Markt für Computer- und Videospiele in der Übersicht. . . . xxvii
B.10. Cry Engine Abstammungsbaum . . . xxviii

C.1. Smart Toy von *Zowie* . . . xxx
C.2. *Brainball* . . . xxx

C.3. Projekt Blinkenlights – Pong xxxi

Tabellenverzeichnis

13.1. Unterstützung menschlicher Bedürfnisse durch Smartphones 231

14.1. Anzahl möglicher Story-Durchläufe in Abhängigkeit der Anzahl von Modulen 238
14.2. Anzahl möglicher Story-Durchläufe in Abhängigkeit der Anzahl von Modulen bei Verschachtlung 240

24.1. Tabellarischer Vergleich verschiedener Sendeintervalle 322

B.1. Spiele-Definitionen xxi

Verzeichnis der Quelltexte

2.1. Typischer Aufbau einer UnrealScript-Datei 26

3.1. Threads können zu Inkonsistenzen im Programmablauf führen 76
3.2. Verwendung von Monitoren um Inkonsistenzen zu vermeiden 77
3.3. Auswertung eines RPC im Game-Egine-Client 84
3.4. Ausschnitt der Kamera-Kalkulation 90
3.5. Beispiel der Verwendung eines JSON-Objektes 92
3.6. Verwendete JSON-Struktur 92

18.1. Builder-Pattern für die Kreierung von Nachrichtenobjekten 272

Vorwort zur Werkszusammenfassung

Das hier vorliegende Werk ist eine Zusammenführung des Masterprojekts und der darauf aufbauenden Masterarbeit von Antony Konstantinidis und Nicolas Kopp. Diese Arbeiten sind in den Jahren 2013 und 2014 entstanden und ergeben zusammen ein umfassendes Bild der Software- und Spielenentwicklung, der Konzeption von Echtzeitanwendungen und vermitteln Hintergründe aus den verschiedensten Bereichen der Mixed Reality, des Storytelling, der Netzwerkkonzeption und der künstlichen Intelligenz.

Als Ergebnis dieser Arbeiten stand nicht nur ein wissenschaftlicher Abschluss, sondern ein fertiges Produkt, dass ab dem Herbst 2014 als kommerzielles Produkt am Markt lanciert wird (oder bereits wurde[1]). Aufgrund der kaskadierenden Rückbezüge innerhalb der Arbeiten haben wir uns dazu entschlossen, die Ausarbeitungen unverändert in der Reihenfolge abzudrucken, in der die ursprünglichen Werke verfasst wurden. Das hat den Vorteil, die Gedankengänge, welche zum Entstehen dieses Projekts geführt haben, kleinschrittig nachzuvollziehen zu können. Der Nachteil besteht darin, dass manche Themengebiete sich überschneiden und in thematisch doppelter Ausführung zu finden sind, was sich zum Zeitpunkt der Ausarbeitungen jedoch nicht vermeiden lies.

Ausdrücklich hervorheben möchten wir an dieser Stelle unsere Betreuer **Herr Prof. Dr. Wolfgang Heiden** und **Herr Dr. Gernot Heisenberg**, die uns durch alle Höhen und Tiefen dieser Ausarbeitungen geführt sowie jederzeit zur Seite gestanden haben und letztlich dazu ermutigt haben, dieses Werk einer breiteren Öffentlichkeit zugänglich zu machen.

Sollten Sie Fragen oder Anmerkungen zu diesem Werk haben, so erreichen Sie die Autorenschaft jederzeit und gerne unter der E-Mail-Adresse info@thinklabs.de.

Wir wünschen nun viel Spaß bei der Lektüre!

Köln, im September 2014
Nicolas Kopp & Antony Konstantinidis

[1] Informationen hierzu sind auf der Internetseite `http://www.fermi-paradox.de/` zu finden.

Artwork zum entstanden Produkt Fermi-Paradox Konzept und Design der Autoren, Ausarbeitung i.A. Ulrike Kleinert [Kleinert, 2014].

„Every adventure requires a first step. Trite, but true, even here.“

(„The Cheshire Cat“ aus „Alice in Wonderland (1951)“ [IMDb.com, Inc., 2014]*)*

Vorwort zur Ausarbeitung

Von Spielerinnen und Spielern - ein Wort zur Genderneutralität in dieser Arbeit

Mit Ausarbeitung dieser Arbeiten ist uns aufgefallen, dass es in der IT viele Begriffe gibt, die eine *scheinbare* Wertung des Geschlechts vornehmen. Allen voran wäre da „the user“ - vom Englischen übersetzt meist als „der Nutzer“. Sollten nicht auch explizit alle „Nutzerinnen“ angesprochen werden?

Die gleiche Frage stellen sich wiederholt auch Universitäten, die nicht mehr von Studentinnen und Studenten reden, sondern, politisch korrekt, von den „Studierenden“. Max Gold verfasste hierzu das folgende Zitat:

> *„Wie lächerlich der Begriff Studierende ist, wird deutlich, wenn man ihn mit einem Partizip Präsens verbindet. Man kann nicht sagen: In der Kneipe sitzen biertrinkende Studierende. Oder nach einem Massaker an einer Universität: Die Bevölkerung beweint die sterbenden Studierenden. Niemand kann gleichzeitig sterben und studieren.“*

[Goldt, 2002, S. 56]

Der politische Wille zur Etablierung einer genderneutraleren Sprache führte hier zu einer Wortverfremdung, bei der das Wort nicht mehr die Wortdeutung des Status, sondern die Tätigkeit im Partizip Präsenz beschreibt [Zentrum für Arbeitsbeziehungen und Arbeitsrecht, 2014].

Wenn „die Nutzerin“ demnach explizit angesprochen wird, so ist das höflich, weiter gedacht müsste es jedoch auch heißen: „Der Endbenutzerinnenlizenzvertrag“ - und Betriebssysteme hätten dann nicht nur einen Eintrag dafür, einen neuen Benutzer hinzuzufügen, sondern auch eine Benutzerin. Wir sind daher übereingekommen, diese das Problem nicht behebenden und erst genderisierenden Bezeichnungen an geeigneten Stellen zwar durch neutrale Begriffe im Sinne einer beidgeschlechtlichen Nennung zu ersetzen, jedoch nur, wenn diese den Textfluss nicht stören und den Sinn nicht verfälschen. Bei Auflistungen legen wir uns deshalb auf eine Form fest. Um es allen recht zu machen sprechen wir dabei auch mal von der Nutzerin - meinen damit jedoch ebenfalls ausdrücklich sowohl die weibliche als auch die männliche Leserschaft!

Legende

Diese Arbeit weist einige spezielle Formatierungen auf, die an dieser Stelle kurz erläutert werden sollen.

Kursiv geschriebene Texte → dienen der Erklärung oder Hervorhebung von Begriffen. Kursiv geschrieben Wörter werden auch für Eigennamen genutzt, wie beispielsweise *XBox One*.

„Begriffe in Anführungszeichen“ → dienen der Verbildlichung eines meist umgangssprachlichen Wortcharakters, die teilweise benutzt werden, um Zusammenhänge an besonders komplizierten Stellen zu vereinfachen und eine alternative Erklärung anzubieten.

Die in diesem Dokument enthaltene Masterarbeit baut auf dem vorangegangen Masterprojekt auf, in dem einige behandelten Aspekte und Konzepte der Masterarbeit insbesondere im Grundlagenteil des Masterprojekts erklärt wurden. Da sich der Stand der Forschung hier jedoch nicht verändern hat, wurden einige Passagen zum besseren Verständnis wortgetreu übernommen und in dieser Zusammenführung beibehalten, um einen höheren Textfluss zu gewährleisten. Zur erhöhten Lesbarkeit sind diese längeren Zitate zum einen mit einer veränderten Schriftart gekennzeichnet und zum anderen werden diese Absätze von zwei blauen Linien eingerahmt, die den Start und das Ende der entnommenen Passage kennzeichnen:

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Wortwörtliche Zitate → werden eingerückt, mit Anführungszeichen versehen und kursiv dargestellt:

> *„Lipsum“*

Direkte Zitate im Textfluss → *„werden ebenfalls kursiv formatiert und mit Anführungszeichen versehen“*.

Danksagung

Diese Arbeit ist nicht nur das Produkt der Zusammenarbeit zweier Menschen, sondern das Ergebnis eines Lebensabschnitts und wäre ohne die Hilfe, Liebe und Unterstützung vieler nahestehender Personen nicht möglich gewesen. Wir danken insbesondere unseren Familien, Freunden und Partnern

für die Unterstützung, auch durch schwierige Zeiten. Wir danken ganz herzlich unseren Betreuern, die uns diese Arbeit ermöglicht haben und uns mit Rat und Tat zur Seite standen. Wir verstehen die vorliegende Arbeit als ein Ergebnis hervorragender, gemeinschaftlicher Zusammenarbeit.

Wir danken ferner insbesondere, in alphabetischer Reihenfolge:

- Christopher
- Lee
- Méline
- Stefan
- Stephan
- Sunayana

Weiterhin danken wir allen Menschen, Freundinnen und Freunden, die mit uns ein spannendes, sehr langes Evaluationswochenende verbracht haben. Vielen Dank!

Wir wünschen allen *Leserinnen und Lesern* eine angenehme Lektüre bei der nun folgenden Ausarbeitung.

Digitalisierung eines Pen-&-Paper-Rollenspiels mit Übertragung von Interaktionen in die reale Welt

MASTERPROJEKT

KAPITEL 1

Einleitung

„Games and stories have much in common. Both deal with how we handle fear. Both can teach us about the world and ourselves. Both can challenge us, move us to laughter or tears.“ [Sheldon, 2004, S. 3]

Menschen sind zutiefst sozial veranlagt. Ein hohes Bedürfnis, sich dem Gegenüber mitzuteilen, Meinungen zu hören, Kritik, Lob, Leid und Freude zu teilen, scheint sich durch alle Zeitalter der Menschheitsgeschichte zu ziehen. Der zuletzt großflächig wahrnehmbare Gipfel dieser Entwicklung spiegelt sich in der Entstehung und dem rasanten Wachstum der sozialen Netzwerke im Internet[2] wider. Derer weltweiter Zulauf ermöglicht es, die Globalisierung selbst bei sich zu Hause zu spüren: Mit einem Klick oder einem Anruf können Freunde, Familie und Bekannte herbeigeholt werden. Somit scheint die Distanz zwischen zwei Kommunikationspartnern keine Rolle mehr zu spielen respektive wird diese durch technische Raffinesse immer weniger greifbar.

Wer das Internet in gemeinsamer Interaktion mit anderen Menschen nutzt, teilt eine Art Raum mit Ihnen. Der Begriff „Raum“ muss an dieser Stelle genauer ausdifferenziert werden. Das Internet kann keinen tatsächlich *physikalisch* vorhandenen Ort ersetzen, sondern lediglich einen *sozialen* Raum zum Austausch untereinander bilden - obwohl die Bewohner nicht denselben physikalischen Raum teilen. Durch Kommunikation und soziale Interaktion der Gruppenmitglieder wird der virtuelle Raum in seiner Gestalt definiert und entsteht erst durch diese Interaktion [Dringenberg u. a., 2002, S. 33]. Verschiedenste Möglichkeiten der globalen Vernetzung dienen dabei als Hülle zur Bildung eines sozialen Interessenverbandes, beispielsweise in Form von Internetforen, Chaträumen oder auch Kommunikationsmöglichkeiten in modernen Computer- sowie Videospielen [Dringenberg u. a., 2002, S. 34, S. 78]. Die Zugehörigkeit zu einem solchen Raum birgt Chancen und Risiken. Ähnlich wie in der realen Welt entstehen Gefühlsbeziehungen der Menschen untereinander. Diese Beziehungen bilden

[2] Zum Ende des Jahres 2012 verzeichnete beispielsweise das soziale Netzwerk „Facebook“ eine Milliarde regelmäßig aktive Nutzer. Damit ist mit Blick auf die Gesamtbevölkerung der Erde zum Jahreswechsel 2012/2013 ein Siebtel aller lebenden Menschen alleine bei diesem Netzwerk angemeldet [Stallmeister, 2012; Wilkens, 2012].

sich im Vergleich zur realen Welt sogar schneller aus und oftmals fällt die Kommunikation leichter, um nur ein Beispiel zu nennen [Dringenberg u. a., 2002, S. 37]. Auf der anderen Seite findet auch eine Entfremdung statt: Menschen werden füreinander parzellierbar[3] und „Freundschaften" lassen sich beispielsweise mit einem Klick herstellen oder auch beenden.

Eine besondere Art der zwischenmenschlichen Interaktion stellt das gemeinsame Spielen dar. Doch was bedeutet Spielen? Fine definiert Spiele mit Bezug auf Avedon u. Sutton-Smith als:

> *„[…] an exercise of voluntary control systems in which there is an opposition between forces, confined by a procedure and rules in order to produce a disequilibrial outcome."*

[Fine, 2002; Avedon u. Sutton-Smith, 1971, S.7]

Insbesondere die Elemente der freiwilligen Teilnahme, des Vorhandenseins von Regeln und des unprädiktierbaren Ausgangs eines Spiels werden für diese Definition als essentiell betrachtet [Fine, 2002, S.7].

Schon das altägyptische Spiel *Senet* zeugt mit seinem Entstehungsdatum vor mehr als 5000 Jahren von einem dem menschlichen Wesen innewohnendem Trieb, sich spielerisch miteinander und mit der umgebenden Welt zu beschäftigen [Piccione, 2010]. Spiele sind keine Erfindung der Neuzeit: Weit und groß gefächert ist das Repertoire der sozialen Interaktion[4].

Mit fortschreitender Technik ist es möglich, weltweit und jederzeit aus Millionen verschiedenster Spiele auszuwählen und miteinander unbekannte Welten zu erforschen und zu erleben oder in kompetitiver Manier ein Kräftemessen zu veranstalten. Durchschnittlich, wie in Abbildung 1.1 zu sehen, sind derzeit[5] etwa fünf Millionen Spieler weltweit gleichzeitig auf einer der größten[6] Online-Plattformen online und spielen mit- oder gegeneinander [Valve Corporation, 2013b].

Mehr als jeder dritte Bundesbürger spielt am Computer und in der Altersgruppe bis zu dreißig Jahren spielen sogar acht von zehn Menschen[7]. Computer- und Videospiele sowie die Utilisierung moderner Medien zur Vernetzung und zum sozialen Austausch untereinander sind in der Mitte der Gesellschaft angekommen [Brehm, 2013]. Hybride Teilableger, wie beispielsweise *Second Life*, stellen darüber hinaus einen Verbund beider Welten dar: Soziale Interaktion und Spiel verschmelzen hier zu einem Gesamtkonzept. Auch wenn die Begegnung der Menschen untereinander in diesem Beispiel rein virtuell bleibt, lassen sich hier mit realem Geld virtuelle Waren handeln. Auf diese Weise diffundieren Effekte

[3] Menschen können in Gruppen eingeteilt werden, dies wird beispielsweise beim sozialen Netzwerk *Google+* explizit als Feature genannt. Kurze Zeit später wurde dieses Konzept auch für das soziale Netzwerk *Facebook* adaptiert [Lippert, 2011]

[4] Ein Schaubild zur Illustration der Geschichte menschlicher Interaktion in Form von sozialen Spielen befindet sich im Anhang in Abbildung B.8

[5] Zur Entstehungszeit dieser Arbeit

[6] Siehe [Leschnikowski, 2012]

[7] Eine erläuternde Grafik hierzu befindet Im Anhang unter Abbildung B.9

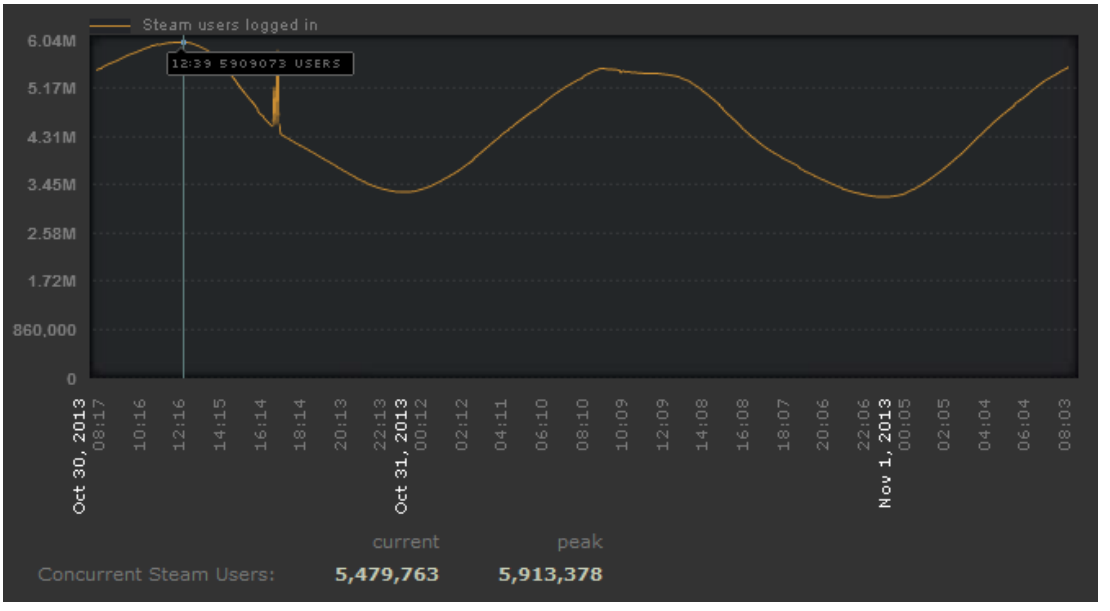


Abbildung 1.1.: Anzahl der aktiven Spieler bei Steam am 02. August 2013.

Aus [Valve Corporation, 2013b].

der realen und virtuellen Welt ineinander, so dass beispielsweise virtuell verkauftes Land in eine regulierte Währung umgetauscht werden kann, um damit reale Güter zu finanzieren [Casati u. a., 2007].

1.1. Inhalt des Projektes und Erläuterung des Hintergrundes

Das in der vorliegenden Arbeit dokumentierte Projekt zielt auf die Verbindung der realen und virtuellen Interaktion in einer bis zu diesem Zeitpunkt noch nicht dagewesenen Art und Weise. Motiviert wurde dieses Projekt durch die Idee, dass es möglich sein soll, Menschen digital und sozial miteinander spielen zu lassen, ohne ein autopoietisches[8] System in einem der beiden Habitate (virtuell oder real) zu erschaffen. Mögliche Grenzen zwischen realer und virtueller Interaktion sollen in der Art aufgehoben werden, dass die Räume fließend ineinander diffundieren.

Ausgehend von dieser Prämisse wurde das Projektziel formuliert, das Konzept des Pen-&-Paper-Rollenspiels auf die digitale Welt zu übertragen. Diese Übertragung erfolgt, indem Mitspieler ihre Rollen im Sinne des Spiels in digitaler Form wahrnehmen. Gleichzeitig soll die digitale Präsentation des Spiels die

[8] Sich selbst erhaltendes.

Grenzen zwischen Realität und Virtualität aufweichen. Somit soll eine Balance zwischen echter sozialer Interaktion und digitalisierter Spielwelt erschaffen werden.

1.1.1. Das Konzept des Pen-&-Paper-Rollenspiels

Gesellschafts- respektive Brettspiele haben zumeist einen kompetitiven Charakter. Prominente Beispiele im deutschen Sprachraum dazu sind *Monopoly* oder *Mensch ärgere Dich nicht*. Im Gegensatz dazu kann das kooperative Konzept eines Pen-&-Paper-Rollenspiels folgendermaßen vereinfacht skizziert werden: Die Spieler und Spielerinnen sitzen gemeinsam um einen Tisch herum, welcher meist mit einer Karte der Spielwelt, diversen Würfeln, Attributkarten und ähnlichem Zubehör belegt ist. Ein Mitspieler übernimmt die Rolle eines „Spielleiters“[9] und konfrontiert die anderen Spielenden mit Spielsituationen, die Entscheidungen herbeiführen sollen und letztlich in einer individuellen Geschichte und Spielerfahrung münden. Das Spiel läuft in der Phantasie der Teilnehmer ab: *„Sie betreten nun eine dunkle, große Höhle, dort steht ein Schatz, bewacht von drei Trollen. Was tun Sie?“*

Als Definition von Pen-&-Paper-Rollenspielen im Rahmen dieser Arbeit soll gelten[10]:

> *„A role-playing game must consist of quantified, interactive storytelling.“* [Schick, 1991, S.10].

Oben stehendes Beispiel ist eine vom Game Master herbeigeführte *Entscheidungssituation* im Rahmen einer Geschichte. Als Reaktion darauf können die Spieler nun mit unterschiedlichen Aktionen agieren („*interactive*“). Sollte die Gruppe der Spielenden beispielsweise über einen Zauberer[11] verfügen, so könnte dieser die Trolle mit einem hellen Licht ablenken, während der Dieb der Gruppe sich derweil zum Schatz schleicht. Der Game Master kann nun, je nach Regelwerk, mittels Würfeln oder auch durch Eigeninitiative über Erfolg und Misserfolg der Entscheidungen urteilen. Sollten die Aktionen erfolgreich sein, so werden die wertvollen Gegenstände „aufgeteilt“, was bedeutet, dass jeder der Mitstreitenden nun einen Anteil der Beute erhält, welcher auch schriftlich im sogenannten Charakterbogen festgehalten wird. Nach der erfolgreichen Aktion erhalten die Spielenden Erfahrungspunkte, mit denen ihre Spielfigur bessere Eigenschaften erhalten kann. Besser gerüstet geht die Reise der Gruppe anschließend weiter.

[9] engl. Game Master (GM) [Schick, 1991, S.10]. Im Fortang der Arbeit wird der englische Begriff oder die Abkürzung desselben verwendet.

[10] Genauer ist dies eine Zuschreibung von Attributen. Schick bezeichnet diese Attributisierung jedoch als „working definition“, was im Rahmen dieser Arbeit als hinreichend angesehen wird [Schick, 1991, S.10].

[11] Spieler und Spielerinnen können in klassischen Fantasy-Pen-&-Paper-Rollenspielen die Identität von in solchen Welten vorkommenden Wesen annehmen oder fantastische Eigenschaften erhalten. Das hier angebrachte Beispiel geht von einer solchen Fantasy-Welt aus, in der es beispielsweise Zauberer geben kann. Ein Beispiel für eine klassische Fantasy-Welt wäre *Dungeons and Dragons*, welches seit 1974 bis heute entwickelt wird [Wizards of the Coast, Inc., 2003].

1.2. Motivation, Konzept und Zielsetzung

Das kurz angerissene Konzept der Pen-&-Paper Rollenspiele ist deswegen so interessant, weil sich durch eine Digitalisierung viele Konzepte der Informatik vereinigen und anwenden lassen.

Der Grundgedanke bei der Projektkonzeption war, dass Menschen mit modernen Ein- und Ausgabemethoden eine Bereicherung des Spielerlebnisses erhalten. Gleichzeitig soll jedoch der Einsatz der Technik keine Barriere darstellen, welche eine reale soziale Interaktion verhindert: Vielmehr soll genau diese echte menschliche Interaktion erhalten bleiben. Mit Bezug auf das vorherig näher erläuterte Konzept[12] des sozialen Raums wird das Ziel der *erlebten Vereinigung* des realen und virtuellen Raums angestrebt. Am ehesten sind derartige Konzepte momentan bei modernen Spielkonsolen zu finden, wo sich mehrere Menschen mit Controllern an ein- und dieselbe Konsole anschließen und gemeinsam ein Spiel spielen. Das Projekt soll von diesen Konzepten jedoch wirksam abgegrenzt werden; eine nähere Erläuterung hierzu befindet sich im Abschnitt 1.3.

Der im Rahmen dieses Projektes konzeptionierte und realisierte Prototyp setzt dort an, wo andere, kommerzielle Lösungen derzeit noch enge Grenzen setzen: Eingabemedien sollen gleichzeitig als Ausgabemedien dienen, die das Spielerlebnis nicht nur unterstützen, sondern essentiell beeinflussen. Eine virtuelle Aufgabe, wie beispielsweise das Aufschieben einer Tür, soll dabei mit Hilfe des Eingabegerätes simuliert werden. Dabei soll das digitale Resultat der realen Interaktion in einem zirkulären Prozess zu den Nutzern zurück geführt werden: Ständig ist ein Fluss zwischen dem primären Ausgabegerät und den Eingabegeräten vorhanden. Zusätzlich beeinflusst der Mensch durch sein Verhalten respektive *Gruppenverhalten* das Spiel in seiner Linearität. Letztlich wird damit auch die erzählte und erlebte Geschichte laufend verändert.

Zu diesem Zweck muss eine Schnittstelle konzipiert und prototypisch implementiert werden. Diese reagiert auf Eingaben durch die Spieler, stößt einen Entscheidungsprozess an und lässt die Ergebnisse zurück in die Spielwelt fließen. Die Schnittstelle wird zu diesem Zweck mit einer primitiven Entscheidungsmaschine[13] verbunden, welche die Funktions- und Ausbaufähigkeit demonstriert. Die Schnittstelle fungiert somit als virtueller Game Master, während die dahinter stehende Decision Engine das „Gehirn“ dieses Game Masters darstellt.

Für die Ein- und Ausgabe der Mitspieler sollen Smartphones auf *Android*-Basis verwendet werden. Smartphones sind durch die Vielzahl eingebauter Sensoren sowie Aktoren und des Touchscreens gerade zu prädestiniert dazu, die benötigten Funktionen abzudecken. Dem Gegenüber steht zum Beispiel

[12] Siehe Abschnitt 1 auf Seite 3.
[13] Später: Decision Engine (DE).

die *Nintendo Wiimote*, welche häufig zum Zweck der virtuellen Manipulation eingesetzt wird. Nachteilig hierbei ist, dass komplexe Funktionen auf die wenigen zur Verfügung stehenden Tasten abgebildet werden und Benutzer diese Zuordnung zunächst lernen müssen. Smartphones dagegen sind weit verbreitet und vergleichsweise leicht zu bedienen. Auch komplexe Bedienkonzepte, welche durch die zusätzlichen Möglichkeiten in Form von virtuellen Interaktionselementen gegeben sind, lassen sich auf diesem Weg nutzen.

Diese Smartphones kommunizieren drahtlos mit einem Computer, welcher wiederum die eigentliche Spielwelt präsentiert und die Eingaben der Spieler durch die erwähnte Decision Engine auswertet. Da die Kommunikation der Endgeräte untereinander über drahtlose Netzwerke verläuft, muss auch ein passendes Netzwerkprotokoll entwickelt und implementiert werden, welches die Kommunikation mehrerer Eingabegeräte und der Repräsentationsschicht sicherstellt. Das Konzept ist nochmals grob in Abbildung 1.2 skizziert.

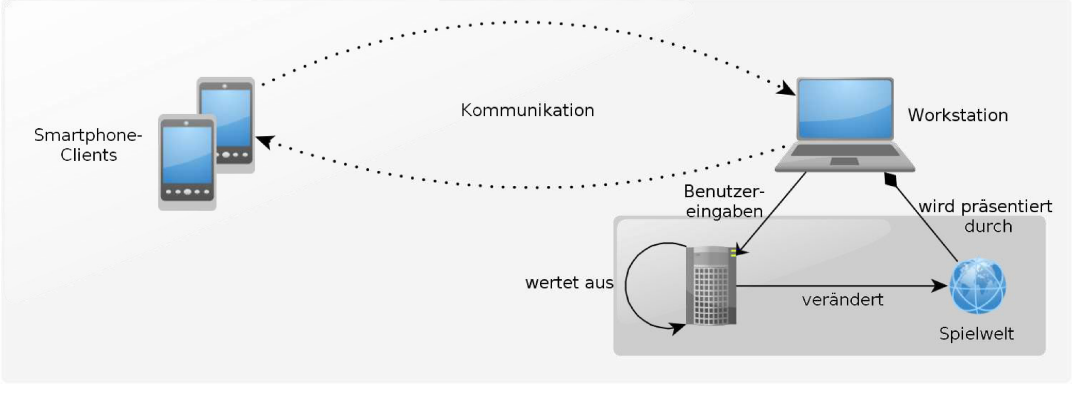


Abbildung 1.2.: Die Abbildung zeigt eine grobe Skizze des geplanten Systems. Die Spielenden sind mit Hilfe von Eingabegeräten in Form von Smartphones drahtlos mit einer Workstation verbunden. Diese Workstation beinhaltet die Logik zur Koordination des Netzwerkverkehrs, zur Visualisierung und Auswertung der Spielsituation.

1.3. Stand der Wissenschaft und Forschung

Menschen miteinander spielen zu lassen, ohne sie sozial und räumlich voneinander zu trennen, ist Gegenstand aktueller Forschung und eine wiederkehrende Fragestellung. Insbesondere, nicht jedoch ausschließlich, wird im Bereich der Videospielkonsolen eine Annäherung an diese Thematik vorangetrieben. So beschäftigte sich eine Gruppe von Forschern des *Institute of Technology* aus Carlow/Irland

(ITC) im Jahre 2012 mit der Fragestellung, wie Senioren zum gemeinsamen Fitnesstraining animiert werden können. Gleichzeitig wurde untersucht, auf welche Art und Weise ein intergenerativ soziales Spielen förderbar ist. Hierbei wurde der von der Firma Microsoft entwickelte *Kinect*-Sensor für Bewegungssteuerung ausgewählt und mit einem HTML5-fähigen Browser angesteuert, um Senioren gemeinsam vor dem Bildschirm einen virtuellen Hindernislauf absolvieren zu lassen [Cantwell u. a., 2012, S.2]. Dieser Sensor, in Zusammenhang mit der dazugehörigen Videospielkonsole *Xbox360*, ermöglicht seit der Markteinführung im Jahr 2010[14] hauptsächlich eine Projektion sozialer Interaktion in Bereichen, die auch im realen Leben von enger sozialer Interaktion geprägt sind . Dazu gehören beispielsweise kooperative oder kompetitive Tanz- und Sportspiele, bei denen sich mehrere Personen im gleichen Raum aufhalten und die entsprechenden Aktivitäten nachahmen [Goad, 2010]. Die jüngste Entwicklung ermöglicht die Verbindung der Kinect mit beliebigen Projektionsflächen als Touch-Sensor, sodass auf diesem Weg neue Anwendungsfälle erschließbar sind. Beispielsweise können mehrere Spieler die Spielwelt direkt durch Berührung manipulieren [Hüfner, 2013].

Konsolen im Allgemeinen bieten in ihrer klassischen Verwendungsform einen Aufbau, welcher dem der Pen-&-Paper-Rollenspiele vom Konzept her im weitesten Sinne ähnelt. Die Videospielkonsole (abstrakt in dieser Analogie zu verstehen als „Game Master und Decision Engine“) wird dabei meist an ein zentrales Ausgabemedium angeschlossen, beispielsweise einen Fernseher oder einen Beamer (übertragbar als „Spieltisch“). Die Spieler kontrollieren ihre Eingaben über mit der Konsole verbundene Controller und teilen dem System auf diese Weise ihre Aktionen und Entscheidungen mit. Das Ergebnis dieser Eingaben wird daraufhin durch das Ausgabemedium visualisiert. Diese Parallele ist in Abbildung 1.3 schematisch festgehalten.

Einen Schritt weiter in Richtung der in diesem Projekt angestrebten Realisierung geht der Hersteller Nintendo mit seiner Spielkonsole *Wii U*. Das als „asymmetrische Spielmechanik“ bezeichnete Konzept besteht darin, dass der Controller neben dem von der Konsole angesteuerten Haupt-Ausgabemedium über einen Touchscreen verfügt. Mit dessen Hilfe kann der Spieler weitere Informationen abrufen und durch die Eingabefähigkeit des Bildschirms Befehle an die Konsole übermitteln [Coffin, 2012]. Die explizit als „mit sozialem Fokus“ vermarktete Videospielkonsole setzt den Schwerpunkt vornehmlich in der Erweiterung sozialer Räume über weite Distanzen. So kann mit Hilfe des Controllers, welcher auch mit einer Kamera ausgestattet ist, eine Videotelefonie-Verbindung zu einem Gesprächspartner über das Internet hergestellt werden [Humphries, 2012]. Die *Wii U* konnte bis dato keinen marktwirtschaftlichen Erfolg verbuchen, da viele Hersteller von Videospielen keine Unterstützung zusagten und es daher an Spielen für diese Konsole mangelt [Karmali, 2013].

[14] Vgl. [Volker Zota, 2010].

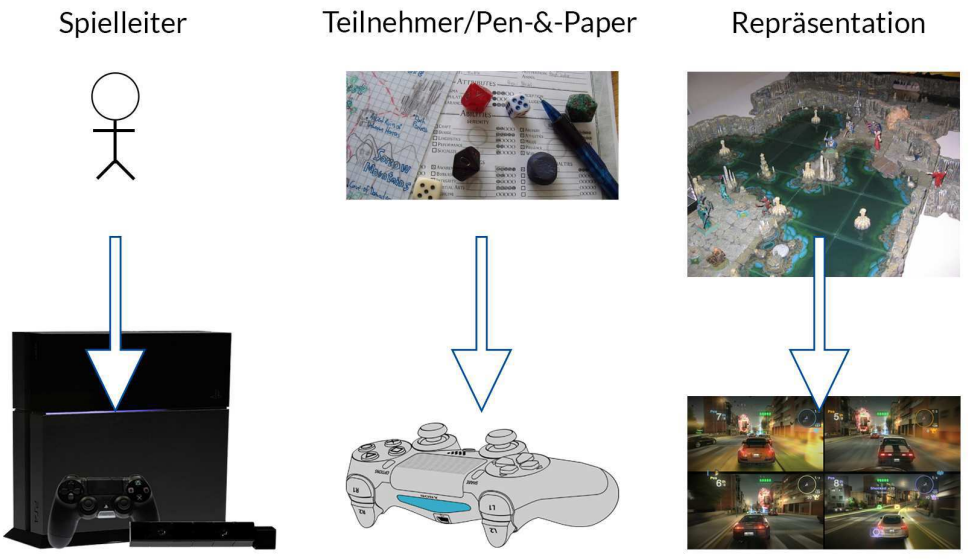


Abbildung 1.3.: In diesem vereinfachten Schema werden die Parallelen zwischen Pen-&-Paper- und Konsolenspielen verdeutlicht. Die Figur und der Aufgabenbereich des (menschlichen) Game Masters fallen weg und werden durch die Konsole respektive die dem jeweiligen Spiel innewohnende Logik substituiert. Die Teilnehmer tätigen Ihre „Eingaben“ und Entscheidungen nicht mit Stift und Papier, sondern über elektronische Eingabegeräte. Die Repräsentation findet bei traditionellen Pen-&-Paper-Rollenspielen hauptsächlich in der Phantasie der Spielenden statt, kann jedoch ebenfalls durch den Spieltisch unterstützt werden. Bei einer Digitalisierung hingegen wird die gesamte Repräsentation als direkte Folge von Aktionen und Entscheidungen visualisiert (rechts im Bild).

mit Material von [tofuprod, 2013; Bearl, 2012; PC Games, 2010; Tokyoship, 2013].

Die *Playstation 3* sowie die im Jahre 2013 annoncierte *Playstation 4* von Sony wird eine Portierung des ursprünglich für den Computer entwickelten Titels *Diablo 3* erhalten. Bemerkenswert hieran ist, dass dieses Spiel zumindest in Grundzügen einen Rollenspielcharakter aufweist und von bis zu vier Spielern gleichzeitig gespielt werden kann. Der Digitalisierung eines Pen-&-Paper-Rollenspiels kommt diese Umsetzung damit recht nah, da alle Spielerinnen und Spieler ihre Charaktere und die Spielwelt auf einem gemeinsamen Ausgabemedium wahrnehmen und der Bildschirm nicht wie oft üblich für jeden Spieler einen einzelnen Bereich vorhält[15]. Dies kommt dem Erlebnis eines Pen-&-Paper-Rollenspiels näher, da alle Spielenden gemeinsam „auf eine Welt schauen“ [Steinlechner, 2013].

[15] Auch bekannt als „Splitscreen-Verfahren“.

1.3.1. Projektabgrenzung

Im vorherigen Abschnitt wurde kurz umrissen, dass durchaus Bestrebungen zu verzeichnen sind, welche auf ein soziales Spielerlebnis abzielen. Zumindest steht Technologie bereit, welche eine Umsetzung im Sinne dieses Projektes ermöglichen könnte. Was bestehenden Konzepten bis zu diesem Zeitpunkt jedoch fehlt oder bisher nur in Teilen umgesetzt wurde[16], ist ein einheitlicher Ansatz, welcher folgende Prämissen vereinigt:

- Die Spieler sollen sowohl räumlich als auch virtuell nicht getrennt werden. Soziale Interaktion soll sowohl real als auch virtuell stattfinden. Eine virtuelle Trennung, wie beispielsweise durch ein Splitscreen-Verfahren, ist hierfür ungeeignet, da in diesem Fall für jeden Spieler respektive jede Spielerin ein eigener Visualisierungsbereich geschaffen wird. In diesem Fall würde eine virtuelle Trennung stattfinden und auf diese Weise einen Bruch in der sozialen Interaktionsebene entstehen. Eine räumliche Trennung ist nicht geeignet[17], da die direkte soziale Interaktion unterbunden wird. Es ist explizites Projektziel, diese zu erhalten.
- Interaktion soll in bidirektionaler Art und Weise zugelassen werden: Eingaben sollen nicht nur getätigt, sondern auch Ausgaben direkt zum Spieler zurückgeführt werden. Eine reine Eingabemethode, wie die eines üblichen Konsolen-Controllers in Form eines Gamepads, bietet solche Möglichkeiten nur eingeschränkt (Ausgaben sind oftmals nur in Form von Vibration möglich). Die auf einem Bildschirm gemeinsam betrachtete, virtuelle Welt wird durch einen eigenen Bildschirm im Eingabegerät personalisiert und erweitert. Dadurch verschiebt sich das Erlebnis vom Bildschirm weg und zu den Nutzern hin.
- Der unter Punkt zwei angesprochene Effekt der Durchmischung von Realität und Virtualität soll dadurch verstärkt werden, dass Aktionen der Spielenden mit Hilfe der Eingabegeräte gemeinsam durchgeführt werden sollen. Dazu ist reale Kommunikation und kollaboratives Spielen notwendig. Hier besteht die Notwendigkeit eines Bedienkonzeptes, welches Eingaben der Spielenden sowohl real als auch virtuell zusammenführt.

Der im Rahmen dieser Arbeit zu entwickelnde Prototyp soll ein technisches Grundgerüst bereitstellen, auf dessen Basis die Digitalisierung eines Pen-&-Paper-Rollenspiels möglich ist. Der Prototyp soll jedoch keine Aussagen darüber treffen, welche Arten der sozialen Interaktion für eine Digitalisierung geeignet sind. Weiterhin wird im Rahmen dieser Ausarbeitung die Fragestellung nicht behandelt, ob und wie die erzählte Geschichte und Umsetzung eines virtuellen Spielleiters ein Gemeinschaftserlebnis intensivieren kann.

[16] Bestehende Konzepte, die für den Prototypen in diesem Projekt relevant sind, werden im Abschnitt 3.1.4 vorgestellt und einander gegenüber gestellt.
[17] Beispielsweise wäre eine solche Trennung mittels Videotelefonie bei einer Digitalisierung denkbar.

1.4. Aufbau der Arbeit

Die Gliederung der Arbeit gestaltet sich wie folgt:

In Kapitel 2 wird ein Überblick über Rollenspiele im Allgemeinen gegeben, woraufhin derer inhärente Elemente identifiziert werden. Hinzu kommt des weiteren ein Vergleich zweier moderner Systeme für Echtzeitgrafikanwendungen. Ferner wird die Netzwerkprogrammierung unter Berücksichtigung verschiedener Eigenschaften wie Protokolle sowie Netzwerkspielmodelle betrachtet. Eine besondere Beachtung erfährt hierbei die Programmiersprache Java, auf die im Einzelnen ebenfalls eingegangen wird. Darüber hinaus wird in diesem Kapitel noch kurz auf das mobile Betriebssystem Android und dessen Komponenten eingegangen.

Im Kapitel 3 werden daraufhin zunächst die identifizierten inhärenten Elemente eines Rollenspiels für die Begründung verschiedener Designentscheidungen (Abschnitt 3.1) herangezogen, gefolgt von der Beschreibung der eigentlichen Konzeptionierung und Implementierung des Prototypen in Abschnitt 3.2. Begonnen wird hierbei mit einem Gesamtüberblick über die Anwendung. Im Anschluss daran wird auf die Umsetzung der einzelnen Komponenten eingegangen, aufgetretene Probleme sowie zusammenhängenden Lösungen.

Den Abschluss der Arbeit bildet eine Zusammenfassung der Arbeitsergebnisse, sowie ein Ausblick auf mögliche nachfolgende Arbeiten, in Kapitel 4.

KAPITEL 2

Grundlagen

An dieser Stelle werden technische Konzepte erläutert, deren Verständnis für die spätere Beschreibung der Prototyp-Realisierung notwendig ist. Das technische Grundkonzept des im Rahmen dieser Arbeit entstehenden Prototyps wird dabei in weiten Teilen durch sukzessive Analyse des Leitgedankens festgelegt, welcher die Umsetzung induziert. Dieser Leitgedanke kann im Wesentlichen auf folgende Formulierung herunter gebrochen werden:

> *„Wie kann das Konzept[18] eines Pen-&-Paper-Rollenspiels auf die digitale Welt übertragen werden?“*

Aus dieser Formulierung ergibt sich in direkter Folge, dass es für eine differenzierte Betrachtung möglicher Realisierungen unumgänglich ist, die einem Pen-&-Paper-Rollenspiel innewohnenden Mechaniken zu identifizieren. Zu diesem Zweck müssen diese in einem iterativem Prozess auf die atomaren Kernelemente reduziert werden. Mit Hilfe weitergehender Betrachtung dieser Bestandteile wird darauf folgend eine mögliche Digitalisierung der einzelnen Elemente konzeptioniert. In diesem Kapitel geht es daher zunächst darum, die inhärenten Elemente eines Pen-&-Paper-Rollenspiel zu identifizieren. Im Kapitel Realisierung werden verschiedene Möglichkeiten erörtert, diese Elemente in die digitale Welt zu übertragen und die gewählten Herangehensweise des jeweiligen Aspekts begründet.

Im zweiten Teil dieses Kapitels befindet sich eine Einführung sowohl in die Grundlagen der Netzwerktechnik als auch in die Konzepte des mobilen Betriebssystems *Android*, da dieses, wie in der Einführung bereits erwähnt[19], im Rahmen der Digitalisierung als Betriebssystem der Benutzerschnittstelle zum Einsatz kommt. Die Rolle der Smartphones wird dabei ebenso angesprochen wie eine vergleichende Darstellung moderner Game-Engines, welche als Kandidaten für eine Visualisierung der gemeinsamen Repräsentationsschicht dienen.

[18] Das Konzept ist in ausführlicherer Formulierung in Abschnitt 1.2 zu finden.
[19] Vgl. Abschnitt 1.2.

2.1. Charakterisierung eines Rollenspiels

Zum Zweck einer Charakterisierung von Pen-&-Paper-Rollenspielen wird die in Abschnitt 1.1.1 bereits vorgestellte Definition herangezogen:

> „*A role-playing game must consist of quantified, interactive storytelling.*“ [Schick, 1991, S.10].

Diese Definition bezieht sich explizit auf Pen-&-Paper Rollenspiele und lässt eine Unterteilung und nähere Betrachtung zu [Schick, 1991, S.7]. Es werden drei Elemente genannt, die zur Einordnung eines Spieles in die Kategorie des „Rollenspiels“ notwendig sind. Diese lassen sich auf implizierte Aussagen untersuchen:

„Quantified“ im Sinne eines Rollenspiels bedeutet, dass die Charakterentwicklung sowie die Fähigkeit, auf bestimmte Situationen zu reagieren, einem expliziten Regelwerk unterliegt. Dies ist abzugrenzen von „offen formulierten“ Spielzielen. Dort ist durchaus möglich, dass eine Spielwelt (das Regelwerk und grundlegendes Design) in Grundzügen vorformuliert und den Spielern die weitere Ausgestaltung und Sinngebung überlassen wird. Diese Herangehensweise fällt jedoch in den Bereich der Fiktion und Dramen [Schick, 1991, S.10].

„Interactive“ bedeutet, dass die Spielgeschichte von den Spielern entscheidungsbasiert vorangetrieben wird. Spieler dürfen dabei jedoch nicht aus dem Regelwerk ausbrechen, sondern müssen sich jederzeit im Bereich der überprüfbaren Quantifikation des Regelwerks bewegen [Schick, 1991, S.10].

„Storytelling“ im Sinne eines Rollenspiels bedeutet, dass eine Geschichte erzählt wird, in welcher die Spielenden eine tragende Rolle einnehmen. Diese Geschichte ist nicht grundsätzlich offen, sondern wird oft von einem bestimmten Ziel getragen und ist in Grundzügen vorgefertigt [Tychsen u. a., 2005, S.1-2]. Der Game Master gibt den Spielern dabei Hinweise, in welche Richtung sich die Story weiterentwickelt; die eigentlichen Details der Geschichte werden durch die Handlungen der Spielenden bestimmt. Eine Geschichte oder Ausgangssituation kann grundsätzlich auch mehrmals gespielt werden, wobei sich die jeweiligen Spieldurchläufe in Hergang und Ende erheblich voneinander unterscheiden können [Schick, 1991, S.10].

Pen-&-Paper-Rollenspiele haben jedoch noch weitere Eigenschaften, die sie von anderen Spielformen abgrenzen. Diese Eigenschaften respektive Elemente werden im Folgenden näher vorgestellt.

2.1.1. Game Master

Ein zentrales Element der Pen-&-Paper-Rollenspiele ist das Konzept des Game Masters. Die Aufgabe des Game Masters besteht darin, das Regelwerk zu kennen und anzuwenden, die Aktionen und Entscheidungen der Spieler auszuwerten und sowohl die Spielwelt als auch alle darin vorkommenden Charaktere zu kontrollieren, welche nicht von Spielern kontrolliert werden[20] [Schick, 1991, S.10]. Der Game Master ist dabei nicht als Antagonist der Spieler zu verstehen, sondern als Schnittstelle und Feedbackmöglichkeit, mit dessen Beschreibungen und Ausführungen die Spielerinnen und Spieler die Spielwelt wahrnehmen. Die Beschreibung der Spielwelt kann dabei durchaus bidirektional und auch auf Anforderung erfolgen. Beispielsweise könnten Spieler einen Raum betreten und den Game Master fragen: „Was kann ich in diesem Raum alles sehen?“. In einem Rollenspiel arbeiten die Spieler miteinander auf ein gemeinsames Ziel hin, niemals gegeneinander (obwohl es zulässig ist, dass sich Spieler innerhalb der Gruppe eigene Vorteile verschaffen, allerdings müssen Sie dabei auch mit den Konsequenzen umgehen).

Gemeinsame Entscheidungen und Herangehensweisen der Spieler und Spielerinnen werden in einem sozialen Kommunikationsprozess aufeinander abgestimmt und dem Game Master mitgeteilt. Der Game Master wiederum entscheidet nach Regelwerk über Erfolg und Misserfolg der Entscheidungen und präsentiert den Spielenden die Ergebnisse und die daraufhin veränderte, neue Situation [Schick, 1991, S.11]. Er ist damit zuständig für die Fortentwicklung der Spielgeschichte, wodurch sich im ständigen Austausch mit den Spielern eine dynamische, narrative Feedback-Schleife ergibt [Tychsen u. a., 2005, S.1]. Eine Besonderheit des Game Masters im Bereich der Pen-&-Paper-Rollenspiele liegt darin, dass er in weiten Teilen dafür zuständig ist, das Engagement der Spielenden aufrechtzuerhalten. Durch die Ausführungen und Beschreibungen des Game Masters erhalten die Spieler Zugang zur Welt. Die erlebte, subjektive Qualität des Spiels steht und fällt mit der Narrationsfähigkeit des Game Masters [Tychsen u. a., 2005, S.2]. Es ist ebenfalls erwähnenswert, dass nur der Game Master mit dem Regelwerk vertraut sein respektive jederzeit darauf Zugriff haben muss. Die restlichen Spielenden benötigen nur begrenztes Wissen darüber, welche Regeln speziell auf ihre jeweiligen Charaktere zutreffen [Schick, 1991, S.13].

2.1.2. Regelwerk

Das Regelwerk und dessen Einhaltung erfüllt mitunter den wichtigen Zweck, Willkür und Chaos zu verhindern. Beim Rollenspiel ist alles erlaubt, allerdings nur innerhalb der Grenzen des zugrundeliegenden Regelwerks [Schick, 1991, S.13f.]. Das Regelwerk kann auch als eine Sammlung von Randbedin-

[20] So genannte NPCs.

gungen aufgefasst werden, welche in der jeweiligen Welt zu gelten haben, wobei zwischen „harten“ und „weichen“ Regeln unterschieden werden kann. *Harte* Regeln legen dabei genau fest, wie Spielmechaniken funktionieren, beispielsweise wie genau ein Angriff oder Dialog ausgeführt werden muss. *Weiche* Regeln sind solche, die als Randbedingungen oder „Naturgesetze“ der Welt bezeichnet werden können. Eine weiche Regel könnte beispielsweise sein: „Man kann nicht fliegen, es gelten die Gesetze der Schwerkraft“ [Tychsen u. a., 2005, S.2].

Es werden jedoch auch feine Details festgelegt, welche sich teilweise mit dem *Design* des Spieluniversums vermischen. Beispielsweise kann als Hintergrundgeschichte eines Universums gelten, dass eine Splittergruppe von Anarchisten sich seit mehreren Jahrhunderten der etablierten, macht-habenden Politik widersetzt. Eine daraus folgende, „weiche“ Regel könnte sein: Angehörige des Staatsapparates sind grundsätzlich misstrauisch gegenüber der Opposition. Wenn ein Charakter nun einen Anarchisten spielt und beispielsweise versucht, einen Regierungsbeamten zu bestechen, ist es sinnvoll, diese Hintergrundinformation in die Berechnungen des Erfolges der Bechstechungsaktion mit einzubeziehen. Regelwerk und Spieldesign sind daher in praktischer Anwendung oft als Einheit zu begreifen.

Das Regelwerk stellt demnach Informationen darüber zur Verfügung, welche Gesetze und Gegebenheiten in einer Welt herrschen respektive was einem Spieler erlaubt ist und was nicht. Das Spiel*design* wiederum gibt Aufschluss darüber, wie diese Gesetze zustande kommen und welche jeweilige Intention dahinter steht. Wie zuvor im Text schon erwähnt, ist es Aufgabe des Game Masters, darauf zu achten, dass das Regelwerk eingehalten wird beziehungsweise durch diesen zur Anwendung gebracht wird.

Im Sinne einer Digitalisierung ist das Verständnis des Regelwerks besonders relevant. Das Regelwerk bildet die Grundlage der erlaubten Wirkungsweise der Decision Engine und legt fest, wie Entscheidungen getroffen werden können.

2.1.3. Spieldesign

Regelwerk und Spieldesign sind meist eng miteinander verknüpft und bilden im Regelfall durch ihre Einheit die *Spielwelt* oder das *Universum*. Es ist möglich, Regelwerk und Spieldesign zu trennen. Ein bekanntes Beispiel für ein Regelwerk respektive ein Spiel-Universum ist das seit 1974 entwickelte klassische Fantasy-Rollenspiel *Dungeons and Dragons* [Wizards of the Coast, Inc., 2003].

Dungeons & Dragons ist das erste und damit auch älteste Rollenspiel und hat damit eine große Palette an Ablegern und Umsetzungen vorzuweisen. Das Regelwerk ist flexibel genug, Änderungen zu erlauben, wie beispielsweise das Hinzufügen mythologischer Figuren, Änderungen des sozialen Gefüges et

cetera. Es wird jedoch empfohlen, sich an einer durch das Mittelalter angelehnten Welt zu orientieren [Fine, 2002, S.16]. Klassisches Pen-&-Paper-Rollenspiel im Dungeons & Dragons Universum sieht vor, dass die Spielenden eine Gruppe[21] bilden und einen Dungeon erforschen. Die wenig verankerte Hintergrundgeschichte lässt viel Spielraum für Interpretationen und mittelwerile gibt es 115 offizielle Umsetzungen des Dungeon-&-Dragon-Regelwerks in verschiedenen Computerspielen [MobyGames, 2013]. „Offiziell" bedeutet in diesem Fall, dass das zugrunde liegende Regelwerk strikt eingehalten wird und eine Lizenz erteilt wurde. Manche dieser Spiele haben eine isometrische Ansicht, manche sind dreidimensional. Einige konzentrieren sich auf Handlung, anderer wiederum auf Kämpfe, wiederum andere haben den Aufbau einer Ritterburg als Spielinhalt. Alle diese Spiele beinhalten demnach völlig voneinander verschiedene *Designs* und Spielprinzipien, obwohl alle das selbe Regelwerk einhalten.

Dieses Beispiel macht deutlich, dass sich innerhalb eines Universums durchaus verschiedene Designentscheidungen und Hintergrundgeschichten entwickeln können. Diese müssen sich nicht zwangsläufig widersprechen, sofern das Regelwerk in seiner Integrität nicht verletzt wird.

2.1.4. Hilfsmittel

Es gibt weitere Werkzeuge, mit denen sowohl der Game Master als auch die Spieler arbeiten. Namensgebend ist hier als erstes „Pen-&-Paper" zu nennen, was darauf zurückzuführen ist, dass jeder Spieler Stift und Papier vor sich liegen hat, um Eckpunkte zum eigenen Charakter festzuhalten, das eigene Inventar zu katalogisieren oder auch Strategien zu skizzieren. Eine Persistierung von Charaktereigenschaften ist alleine schon aus dem Grund sinnvoll, weil Charaktere teilweise über Jahre gespielt- und ständig weiterentwickelt werden.

Der Game Master verfügt zusätzlich über Regelwerk-spezifische Werkzeuge, allen voran den Spielwürfeln. Würfel spielen bei Pen-&-Paper-Rollenspielen eine wichtige Rolle in den Situationen, in denen das Prinzip des Zufalls (oder „Glücks") zum Tragen kommen soll. Sie werden beispielsweise benutzt, um Erfolg und Misserfolg von Spieler-Aktionen „auszuwürfeln" und unterliegen dabei individuellen Berechnungsregeln, welche im jeweiligen Regelwerk festgelegt sind. Um zu bestimmen, ob beispielsweise der Schwerthieb eines Charakters gegen einen Gegner erfolgreich ist, kann unter anderem die Stärke des Spielers, die Attribuierung des Schwertes, die generelle Kampferfahrung und auch die Verteidigung des Gegners mit eingerechnet werden. Die Augenzahl des Würfels bringt in diese Rechnung einen Zufallswert mit ein, woraufhin das Endergebnis bestimmt wird. Ebenso häufig anzutreffen ist eine Spielkarte oder *Miniaturwelt*, die es den Mitspielerinnen und Mitspielern ermöglicht, die eigene

[21] Im Fachjargon eine sogenannte „Party".

Position innerhalb der Welt zu visualisieren oder generell das „Flair“ der Spielwelt einfängt [Schick, 1991, S.11].

Klassische Pen-&-Paper-Rollenspiele werden üblicherweise in einer kleinen Gruppengröße gespielt, sinnvoll sind hierbei eine Anzahl von zwei Spielern plus einem Game Master und übersteigen selten sieben bis acht Spieler (exklusive Game Master). Diese Restriktion ist nicht nur im Aufbau begründet, bei dem die Spieler meist um einen Tisch herum sitzen, sondern auch in einem kritischen Verhältnis, ab dem der Game Master an Einfluss verliert. Je mehr Spieler dem Game Master gegenüber stehen, desto geringer wird sein Einfluss gegenüber dem individuellen Spieler und desto generischer fällt die erzählte Geschichte aus, da sämtliche Aktionen aller Spieler berücksichtigt werden müssen [Tychsen u. a., 2005, S.7f.]. Besonders klein ist dieses Verhältnis in so genannten MMOs, bei denen mehrere tausend Spieler gleichzeitig in einer Welt spielen. Da hier alleine auch aus physikalischen Gründen der direkte Kontakt vom Game Master zu den Spielern fehlt, geht ein hohes Maß an sozialer Interaktivität verloren [Tychsen u. a., 2005, S.6]. Da es implizites Ziel des Projektes ist, die soziale Interaktion eines Pen-&-Paper-Rollenspiels zu erhalten, muss ein Weg gefunden werden, mehrere Spieler real miteinander interagieren zu lassen und bidirektionale Kommunikation mit dem virtuellen Game Master zu ermöglichen. Die spätere Wahl der richtigen Hilfsmittel zur Eingabe der Befehle ist daher ein entscheidender Faktor bei der Prototypkonzeption.

2.1.5. Identifizierte Kernelemente eines Pen-&-Paper-Rollenspiels

Identifiziert als die unabdingbaren, inhärenten Elemente, die jedes Pen-&-Paper-Rollenspiel aufweisen muss, sind somit die Folgenden:

1. Ein Regelwerk, dessen Regeln sowohl Spieler, Game Master als auch Spieldesign unterworfen werden. Das Regelwerk zeigt Grenzen auf, in denen sich Geschichte, Spieler und Game Master bewegen können und gilt gleichsam als „Naturgesetz“ des Spieluniversums.

2. Spieldesign respektive Spieldesigner. Der Spieldesigner ist verantwortlich für das generelle Setting des Universums. Es ist nicht zu vernachlässigen, dass ein großer Teil der Atmosphäre und auch der Charakterentwicklung der Spielenden letztendlich durch das Design des Spiels beeinflusst wird. Der Spieldesigner liefert einen Bausatz der Welt und (Hintergrund-)Geschichte, welche dann durch Game Master und Spieler verwirklicht werden.

3. Ein Game Master, welcher die zu erzählende Geschichte den Regeln und Mechaniken des Regelwerks unterwirft und diese Mechaniken als ausführendes Organ vertritt (beispielsweise durch Würfeln, Konsistenzprüfung der Spielerentscheidungen mit dem Regelwerk). Der Game Master

ist ebenfalls dafür verantwortlich die Spielwelt zu beschreiben und Entscheidungssituationen zu forcieren.

4. Spieler, welche die Geschichte vorantreiben, indem sie in die Rollen ihrer jeweiligen Charaktere schlüpfen und im Einklang mit dem Regelwerk gemeinsam auf vom Game Master vorgegebene Situationen reagieren. Die Reaktionen definieren dabei die Folgesituation.

5. Hilfsmittel zur Visualisierung der Spielwelt und/oder der Spielsituation, sowie Hilfsmittel zur Persistierung der Charaktereigenschaften und spielbestimmender Variablen. Dazu gehören beispielsweise Würfel, Spielfiguren und Charakterbögen[22].

6. Eine Geschichte, die von den Spielern vorangetrieben wird. Ausgangssituation und grobe Eckpunkte der Geschichte sind meist vorgegeben. Details, Verlauf und Ende werden von den Spielenden bestimmt und nur vom Regelwerk der Spielwelt eingegrenzt [Bowman, 2010, S.25].

Diese identifizierten Elemente werden im Kapitel „Realisierung“ als Referenz herangezogen, um auf diese Weise eine sukzessive Analyse durchführen zu können. Die Ergebnisse dieser Analyse werden darauf folgend als Entscheidungsbasis zum jeweiligen Digitalisierungskonzept herangezogen.

2.2. Game-Engines

Im Kapitel „Realisierung“ in Abschnitt 3.1.1 wird verdeutlicht, dass für die Digitalisierung der inhärenten Elemente eines Rollenspiels unter anderem eine Game-Engine[23] herangezogen werden soll, welche dreidimensionale Visualisierungen ermöglicht. Die Game-Engine dient als Werkzeug, um eine Visualisierung zu realisieren, und kann ebenfalls Ereignisse innerhalb der virtuellen Welt registrieren und an die Spieler zurückführen. An dieser Stelle werden deshalb, neben einer allgemeinen kurzen Einführung in Game-Engines, exemplarisch zwei der leistungsstärksten Vertreter vorgestellt. In Abschnitt 3.1.2 wird schließlich begründet, welche Faktoren zur Festlegung auf die im Prototyp verwendete Game-Engine geführt haben.

Eine Game-Engine kann laut Amresh u. Okita folgendermaßen definiert werden:

> *„A game engine is a piece of software or code written by programmers to facilitate the game production pipeline. It can serve various needs specific to a particular game or in general to all games. [...] Some of the [...] common elements include camera systems, user interface, animation, physics and artificial intelligence.“* [Amresh u. Okita, 2010, S. 353]

[22] Eine Sammlung verschiedener Hilfsmittel ist unter [Games Workshop Deutschland GmbH, 2013] zu finden.
[23] Auch „Spiel-Engine“

Im Gegensatz zur rein textbasierten Systemen respektive solchen zum Zweck der rein zweidimensionalen Darstellung (siehe Abbildung 2.1), verfügen moderne Echtzeitgrafiksysteme für 3D-Inhalte über einen nur schwer überschaubaren und komplexen Aufbau (siehe Abbildung 2.2). Diese Komplexität resultiert unter anderem, nicht jedoch ausschließlich, aus dem Anspruch der namensgebenden *Echtzeitfähigkeit*. Im Gegensatz zu Animationsfilmen beispielsweise, deren Berechnungen prinzipiell lange dauern dürfen, sind Echtzeitgrafiksysteme zeitkritisch [Helman, 1995, S.2f.]. Dabei stehen der Anspruch einer möglichst wirklichkeitsgetreuen Darstellung im Kostenverhältnis zur Rendering-Geschwindigkeit. Diese Abhängigkeit ergibt sich aus dem nontrivialen Unterfangen, die sogenannte *Rendergleichung* kostengünstig zu lösen. Die aus der Computergrafik bekannte Gleichung bietet ein mathematisches Modell zur Berechnung von Lichtreflektionen auf Objektoberflächen [Kajiya, 1986, S.1f.].

Ebenfalls nicht zu unterschätzen ist die Zuständigkeit einer Game-Engine im Bereich des Ressourcen-Managements, allen voran einem schnellen und sinnvollen I/O. Da nicht alle Bestandteile eines Computers einen gleich hohen Datendurchsatz haben, ist das intelligente Bereitstellen und akquirieren respektive transferieren von Inhalten in schnellere Bereiche[24] des Computers von essentieller Wichtigkeit. Dazu gehört ebenfalls ein ausgeklügeltes Netzwerkmodell.

Ein harmonisches System zu entwickeln, welches Eingaben schnell verarbeitet und in ansprechender Qualität visuelle Rückgaben liefert, erfordert neben der Kenntnis verschiedenster Algorithmen auch eine hohe mathematische Abstraktionsebene. Ebenfalls wird ein hohes Verständnis der Wechselwirkung von inkludierten Komponenten untereinander vorausgesetzt [Blow, 2004, S.7ff.]. Anspruch einer modernen Spielentwicklung ist es daher, diese Komplexität beherrschbar zu halten und soweit zu abstrahieren, dass der Fokus der eigentlichen Entwicklung auf hohem Level erfolgen kann [Blow, 2004, S.1].

Eine solche Abstraktion erfolgt generell durch die Verwendung und teilweise auch Zweckentfremdung zahlreicher Hilfsprogramme, welche ursprünglich nicht für die Entwicklung von Echtzeit-Inhalten vorgesehen waren. So erfolgt die Erstellung von dreidimensionalen Objekten in spezialisierten Programmen wie *Maya* oder *3D Studio Max*. Der eigentliche Programmcode wird wiederum in einer anderen Entwicklungsumgebung geschrieben, wie beispielsweise dem *Microsoft Visual Studio*, während das Erstellen von Texturen in spezialisierten Bildbearbeitungsprogrammen wie *Photoshop* erledigt wird [Blow, 2004, S.4]. Dazu kommen noch Aufgaben wie Shader-Programmierung, Animationen, ein Werkzeug zur Erstellung von Partikeleffekten und einiges mehr.

Diese Dezentralisierung von Arbeitsabläufen und der damit einhergehende ständige Wechsel zwischen verschiedenen Programmen ist in der praktischen Entwicklung hinderlich. Ein Lösungsansatz

[24] Wie Texturen, die für schnellen Zugriff in den Grafikkarten-RAM überführt werden, gegenübergestellt zu Festplattenzugriffen.

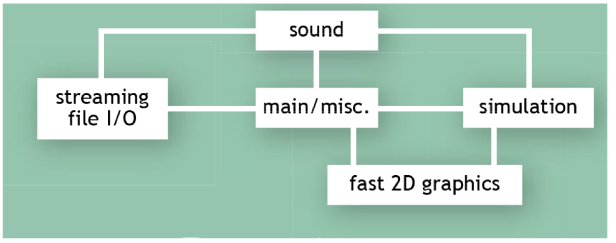


Abbildung 2.1.: Die Abbildung zeigt die Bestandteile respektive Architektur eines generischen 2D-Spiels, wie sie im Jahre 1994 üblicherweise angewandt wurde. Einzelne Genres von 2D-Spielen können unterschiedliche Diagramme aufweisen, der Kerngedanke ist jedoch der, dass faktisch nur Speicherzugriffe mit Spiellogik, Grafik- und Tonausgaben auf sehr einfacher Ebene miteinander kombiniert wurden.

Aus [Blow, 2004, S.3].

dieses Problems resultierte in der Konzeption von integrierten Entwicklungsumgebungen oder „Frameworks“, welche ein zentralisiertes Arbeitsumfeld ermöglichen sollen [Blow, 2004, S.4]. Hier wird insbesondere Wert auf einen erleichterten Entwicklungsprozess im Sinne der Kontinuität gelegt: Viele wiederkehrende Aufgaben lassen sich erledigen, ohne das gesamte Projekt berechnen zu müssen[25]. Insbesondere um dieses letztgenannte Problem zu adressieren, teilen moderne Echtzeitgrafiksysteme den Programmcode in Einheiten auf: Der Basicode („Engine“-Code) wird logisch vom „Gameplay-Code“ oder auch „Script-Code“ getrennt. Um einen Spieler beispielsweise zehn statt fünf Einheiten hoch springen zu lassen, ist keine Änderung an der Engine des Spiels nötig, sondern lediglich eine Änderung des Parameters für die Höhe, welcher der Engine als Eingabe übergeben wird.

Im Folgenden werden zwei moderne Vertreter für Echtzeitgrafiksysteme verglichen und insbesondere daraufhin untersucht, ob sie sich für die Umsetzung des geplanten Projekts eignen. Dabei ist die Handhabung des generellen Workflows im Hinblick auf eine Erarbeitung im Team genau so wichtig wie eine möglichst hohe Modularität und die Fähigkeit der Engine, Inhalte flexibel und zur Laufzeit an mögliche Story-Entscheidungen anpassen zu können. Letzteres ist ein wichtiger Punkt im Hinblick auf die Verwendung der Engine als Plattform zur Digitalisierung eines Game Masters und der dahinter liegenden Decision Engine, um mit veränderten Situationen auf die Entscheidungen und Aktionen der Spieler reagieren zu können (siehe Abschnitt 2.1.1).

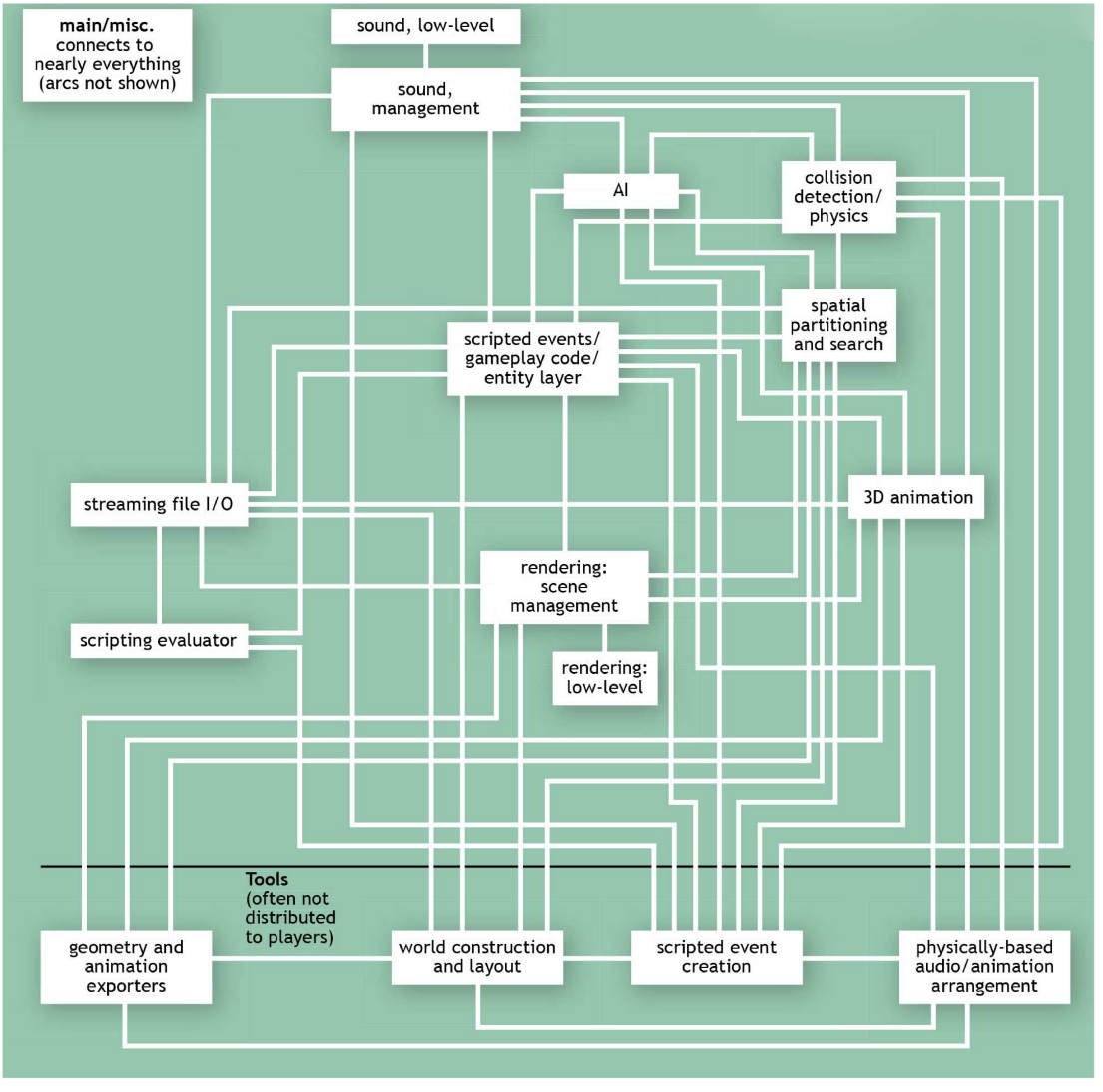


Abbildung 2.2.: Bereits 1996 fing die Spielindustrie in großen Maße an, 3D-Spiele zu produzieren [Blow, 2004, S.3]. Diese Abbildung zeigt die generische Verflechtung der Komponenten untereinander, wie sie in einem modernen 3D-Spiel mit Entstehungsdatum etwa Mitte der 2000er Jahre vorzufinden sind.

Aus [Blow, 2004, S.6].

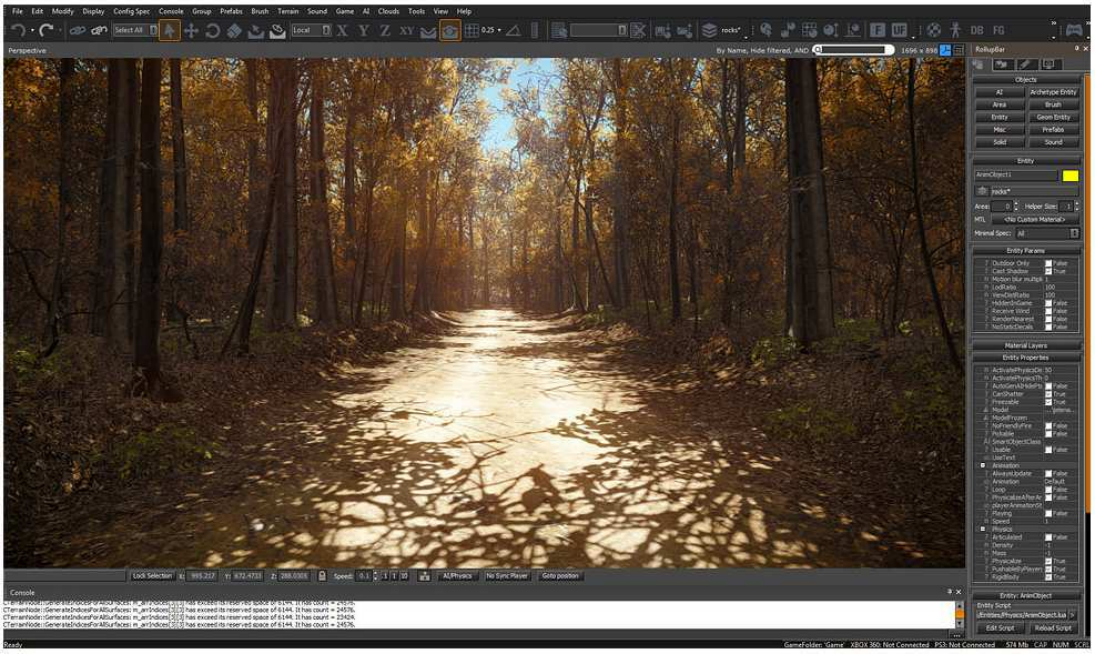


Abbildung 2.3.: Die Abbildung zeigt die sogenannte „Sandbox“ des Cry Engine SDK, in welcher Inhalte in Echtzeit bearbeitet werden können.

Aus [Crytek GmbH, 2013b].

2.2.1. Cry Engine SDK

Das Cry Engine Software Development Kit (SDK) wurde von der deutschen Firma „Crytek“ im August 2011 kostenfrei für die nicht-kommerzielle Nutzung zur Verfügung gestellt [Crytek UK Ltd., 2011]. Die namensgebende „Cry Engine“ wurde erstmals im Jahre 2004 im First-Person-Shooter „Far Cry“ verwendet und gelangte, insbesondere durch besonders hohe Grafikqualität, schnell zu weltweiter Popularität [CBS Interactive Inc., 2004]. Durch den hohen Erfolg konnte die Engine ständig weiterentwickelt werden und wird inzwischen erfolgreich an weitere Lizenznehmer vermarktet[26]. Zur Lizenzierung stehen dabei verschiedene Preismodelle zur Verfügung, welche je nach Einsatzzweck des SDKs unterschieden werden [Crytek GmbH, 2013c]. Das Cry Engine SDK unterstützt die Entwicklung für PC und neuere Konsolengenerationen.

[25] Insbesondere Die Entwicklung von Programmcode erforderte in der Vergangenheit auch bei kleinsten Änderungen eine vollständige Neuübersetzung in Maschinencode. Dieser Prozess kann bei großen Spielen in den Bereich von Stunden gehen [Blow, 2004, S.4].

[26] Vgl. dazu Abbildung B.10 im Anhang

Neben modernen grafischen Effekten wie beispielsweise „Tesselation“ und „Parallax Occlusion Mapping“ unterstützt das SDK Werkzeuge zur Erstellung und Programmierung von künstlicher Intelligenz und verfügt darüber hinaus über einen Editor zur Erstellung plattformübergreifender Spielwelten (siehe Abbildung 2.3) [Crytek GmbH, 2013a]. Plattformübergreifend ist in diesem Zusammenhang so zu verstehen, dass Inhalte zeitgleich und in Echtzeit für mehrere Plattformen entwickelt werden und das Ergebnis sofort nach Änderung auf allen Plattformen gleichermaßen sichtbar ist, sofern diese Plattformen an den Entwicklungsrechner angeschlossen sind [Crytek GmbH, 2013d].

Besonders erwähnenswert ist die volle Unterstützung dreidimensionaler Bildinhalte bis hin zu einer Auflösung von 4096 mal 2304 Bildpunkten, sogenanntes „4K“ [Austinat, 2010]. Die Spielwelt kann dabei in Echtzeit in stereoskopischem 3D modelliert und betrachtet werden [Crytek GmbH, 2013e].

Das Cry Engine SDK scheint keine besonderen Werkzeuge zur Team-Entwicklung bereitzustellen, vielmehr liegt der Fokus, wie bereits beschrieben, auf der Entwicklung von einem Arbeitsplatz heraus, dafür jedoch gleichzeitig für mehrere Endgeräte [Crytek GmbH, 2013d]. Ein Aspekt ist jedoch erwähnenswert, wenn es darum geht, das Cry Engine SDK für die Visualisierung von Welten im Sinne eines Pen-&-Paper-Rollenspiels zu prüfen. Ein so genanntes „Level-Streaming“, das heisst die verschachtelte Darstellung mehrerer Welten-„Einheiten“ ineinander wird nicht unterstützt. Zur Laufzeit können daher keine kompletten Umgebungen ineinander verzahnt oder beliebig durch andere Umgebungen ausgetauscht werden [thedesigner u. Zephbane, 2012; Crytek GmbH, 2012]. Im Cry Engine SDK wird mit der freien Scriptsprache *Lua* programmiert [Mathieu u. a., 2013].

2.2.2. Unreal Development Kit

Das Unreal Development Kit (UDK) leitet seinen Namen aus der „Unreal“-Engine ab, welche im Jahre 1998 erstmals zusammen mit dem gleichlautenden PC-Spieletitel vorgestellt wurde [Chan, 2013]. In der Funktion als Entwicklungsumgebung für Echtzeitanwendungen erschien das Unreal Development Kit im November 2009 kostenlos für nichtkommerzielle Anwendungen, wobei die Unreal Engine in der 3. Version enthalten war [Hirsch, 2009]. Die Version 4 der Unreal Engine befindet sich zur Entstehungszeit dieses Dokumentes in der Endphase der Entwicklung und wird zusammen mit einem neuen SDK erscheinen [Epic Games, Inc., 2013f].

Das UDK ist zu Entwicklungszwecken nur unter Windows lauffähig, allerdings können resultierende Anwendungen nativ auf sämtliche (mobil-)Plattformen portiert werden [Epic Games, Inc., 2013h]. Die grafischen Fähigkeiten sind mit denen der Cry Engine vergleichbar; ebenso wie das Cry Engine SDK verfügt das UDK über fortgeschrittene Unterstützung moderner grafischer Funktionen, sowie

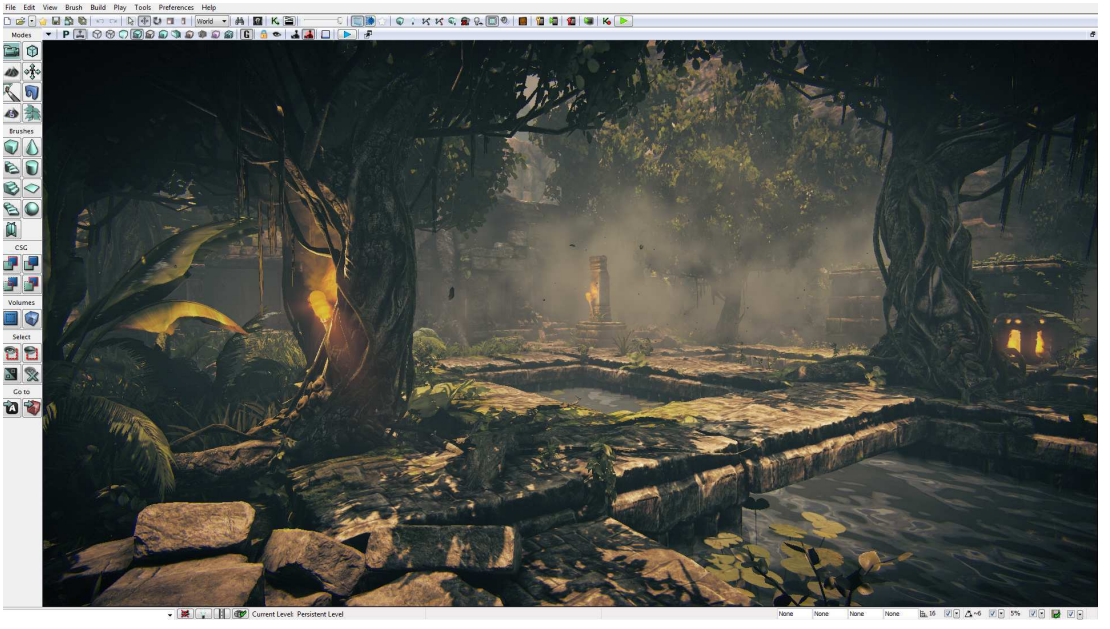


Abbildung 2.4.: Die Abbildung zeigt das Hauptfenster des Unreal Devleopment Kits mit einer geöffneten Naturszene.

über Werkzeuge zur Erstellung von Animationen, Audio-Positionierung, einen Level- sowie Terraineditor[27].

Im Gegensatz zum Cry Engine SDK verfügt das Unreal Devleopment Kit explizit über Werkzeuge zur verteilten Entwicklung in Teams. Aufwendige Berechnungen wie Licht- und Schattenberechnungen können verteilt über das Netzwerk auch von mehreren Computern gleichzeitig durchgeführt werden, wodurch der Entwicklungsprozess linear beschleunigt werden kann [Epic Games, Inc., 2013a]. Ebenfalls verfügt das UDK über eine Möglichkeit, verschiedenen Szenen ineinander zu streamen. Dies bedeutet einerseits, dass auch verteilt an mehreren Ebenen oder Ausprägungen einer sonst linearen Welt gearbeitet werden kann, und andererseits, dass die Spielwelt zur Laufzeit modular zusammengesetzt werden kann [Epic Games, Inc., 2013c].

Neben einer eigenen Scriptsprache namens „UnrealScript“ stellt das UDK zusätzlich ein Werkzeug zur visuellen Programmierung namens „Kismet“ zur Verfügung [Epic Games, Inc., 2013g]. Da diese Programmiersprache einige Besonderheiten aufweist, die auch für das Verständnis der Funktionsweise

[27] Abbildung 2.4 demonstriert einige der grafischen Fähigkeiten im Kontext einer organischen Umgebung.

von Game-Engines im Allgemeinen von weiterer Relevanz sind, wird diese hier kurz näher beschrieben.

2.2.2.1. UnrealScript

Trotz der Bezeichnung als „Scriptsprache“ kann UnrealScript als Hochsprache angesehen werden, welche auch erweiterte Konzepte wie Vererbung, Bibliotheksanbindung und Objektorientierung bietet [Epic Games, Inc., 2013d] [Flynt u. Caviness, 2006, S.22]. Dennoch lassen sich mit Hilfe von UnrealScript keine eigenen, logisch von der Unreal Engine getrennten Programme entwickeln: Ziel der Sprache ist es alleine, Performance-kritische Engine-Funktionen von der Anwendungslogik zu entkoppeln. Dieses Konzept wurde bereits in Abschnitt 2.2 ausführlicher erläutert und kommt ebenfalls durch die Wahl der Scriptsprache *Lua* im Cry Engine SDK zur Geltung.

Eine simple UnrealScript-Datei ist wie in Quelltext 2.1 dargestellt aufgebaut.

```
1  class MyClass extends ParentClass;
2          var bool myClassVariable;
3          function aFunctionName(int myParameter){
4                  local int aLocalVariable;
5          }
6          /* Initialize default values */
7          defaultproperties
8          {
9                  myClassVariable = false;
10         }
```

Quelltext 2.1: Typischer Aufbau einer UnrealScript-Datei

Eine Klasse wird hierbei immer von einer Elternklasse abgeleitet, sofern diese nicht vom Typ *Object* ist. Daraus resultiert, dass grundsätzlich jede Klasse ein Objekt ist. Klassen werden immer am Anfang der Datei deklariert und müssen eine Namensgleichheit zur tatsächlich physikalisch vorhandenen Datei aufweisen, in welcher diese deklariert wurden, was als Besonderheit anzusehen ist. Auf diese Weise kann der Compiler schnell bereits auf Dateiebene erkennen, ob eine Veränderung am Code stattgefunden hat und gegebenenfalls modifizierte oder neue Dateien kompilieren.

Performance-kritische Funktionen sind innerhalb der Unreal-Engine weiterhin in nativem C/C++ implementiert, während UnrealScript insbesondere erleichterten High-Level-Zugriff auf Schnittstellen der Engine-Funktionen bietet, beispielsweise solche zur Netzwerkprogrammierung oder zur Anwendung künstlicher Intelligenz [Epic Games, Inc., 2013e]. UnrealScript arbeitet demnach auf operativem

Level und ordnet Ausführungsgeschwindigkeit einer Simplifizierung von Anwendungslogik unter [Epic Games, Inc., 2013e].

UnrealScript erlaubt es, vom Konzept der sogenannten *Ticks* Gebrauch zu machen. Ticks sind kein exklusives Feature der Unreal Engine im Speziellen, jedoch auch dort vertreten. Die Dauer eines Ticks ist abhängig von der Geschwindigkeit des Host-Systems und gibt an, wie lange der Rechner für die Durchführung eines kompletten Zyklus benötigt (alle Zustände und Variablen auslesen und das Bild neu zeichnen). Wenn dieser Faktor in bestimmte Berechnungen mit einbezogen wird, können beispielsweise Animationen erzeugt werden, die auf jedem Host-System gleich lange dauern, obwohl möglicherweise eine andere Bildwiederholrate vorliegt [Amresh u. Okita, 2010, S.355f.]. Die Wichtigkeit des Konzeptes der *Ticks* wird im späteren Verlauf der Arbeit in Abschnitt 3.2.6 anhand eines konkreten Beispiels verdeutlicht.

2.3. Netzwerkprogrammierung

Beim Spielen von Pen-&-Paper-Rollenspielen in der realen Welt können Menschen frei miteinander kommunizieren. In diesem Zusammenhang wird auch von einem *Kommunikationsnetzwerk* gesprochen, allerdings befindet sich dieses lediglich auf sozialer Ebene. Die Spieler reden und interagieren dabei sowohl miteinander als auch mit dem Game Master.

Da es Ziel des Projektes ist, digitale und reale Welt ineinander diffundieren zu lassen und Spieler ihre Rollen in digitaler Form wahrnehmen sollen, ist innerhalb des Prototypen zusätzlich die Implementierung eines *technischen* Netzwerks notwendig. Aus diesem Anspruch heraus werden im Folgenden die grundlegenden technischen Gegebenheiten von Netzwerken behandelt, wobei der Netzwerkprogrammierung in der Programmiersprache Java eine besondere Geltung zukommt, da unter anderem die Androidprogrammierung[28] in Java erfolgt. Eine ausführlichere Begründung der Auswahlentscheidung zugunsten der Programmiersprache Java wird in den Abschnitten 3.1.8 und 3.1.7 nachgeliefert.

2.3.1. Definition und Kontextualisierung von Netzwerken

Ein Netzwerk kommt laut Harold folgendermaßen zu Stande:

> *„Eine Sammlung von Computern oder anderen Geräten, welche Daten in mehr oder weniger Echtzeit voneinander empfangen und einander senden können“*[29] [Harold, 2004, S. 19]

[28] Wie in Abschnitt 1.2 erwähnt wurde, sollen für die Umsetzung des Prototypen Smartphones auf Android-Basis benutzt werden.
[29] übersetzt aus dem Englischen

Im heutigen Technologiezeitalter spielen Netzwerke eine entscheidende Rolle, da sie den Datenaustausch über Länder- und Zeitgrenzen hinweg ermöglichen [Boonsri Dickinson, 2013]. Insbesondere durch die Vielzahl der Kommunikationsmöglichkeiten, welche durch das Internet geboten werden, findet eine stetig steigende weltweite Vernetzung verschiedenster Geräte untereinander statt (siehe Abb. 2.5). Netzwerke manifestieren sich in unterschiedlichen Formen und Größen. So können kleinere Netzwerke bereits durch die Übertragung von Daten eines Mobiltelefons auf den Computer entstehen. Ein Beispiel für ein Netzwerk im globalen Zusammenhang ist die Verbindung von Nutzern Facebooks untereinander [Jeffrey Van Camp, 2010].

Innerhalb eines Netzwerkes werden einzelne Geräte als Knoten (*nodes*) bezeichnet. Vollwertige Computer werden weiter konkretisiert und als Host benannt. Aus der programmiertechnischen Sicht macht diese namentliche Distinktion jedoch keinen Unterschied. Gleiches gilt für die Art der Codierung der Daten. Unabhängig davon, ob es sich um eine Kabelverbindung handelt und digitale Daten in elektromagnetische Wellen transformiert oder ob Daten in kabellosen Netzwerken über Mikrowellen übertragen werden, bietet zum Beispiel die Programmiersprache Java laut Plüss für Netzwerkanwendungen eine gut ausgebaute, einfach zu benutzende Klassenbibliothek.

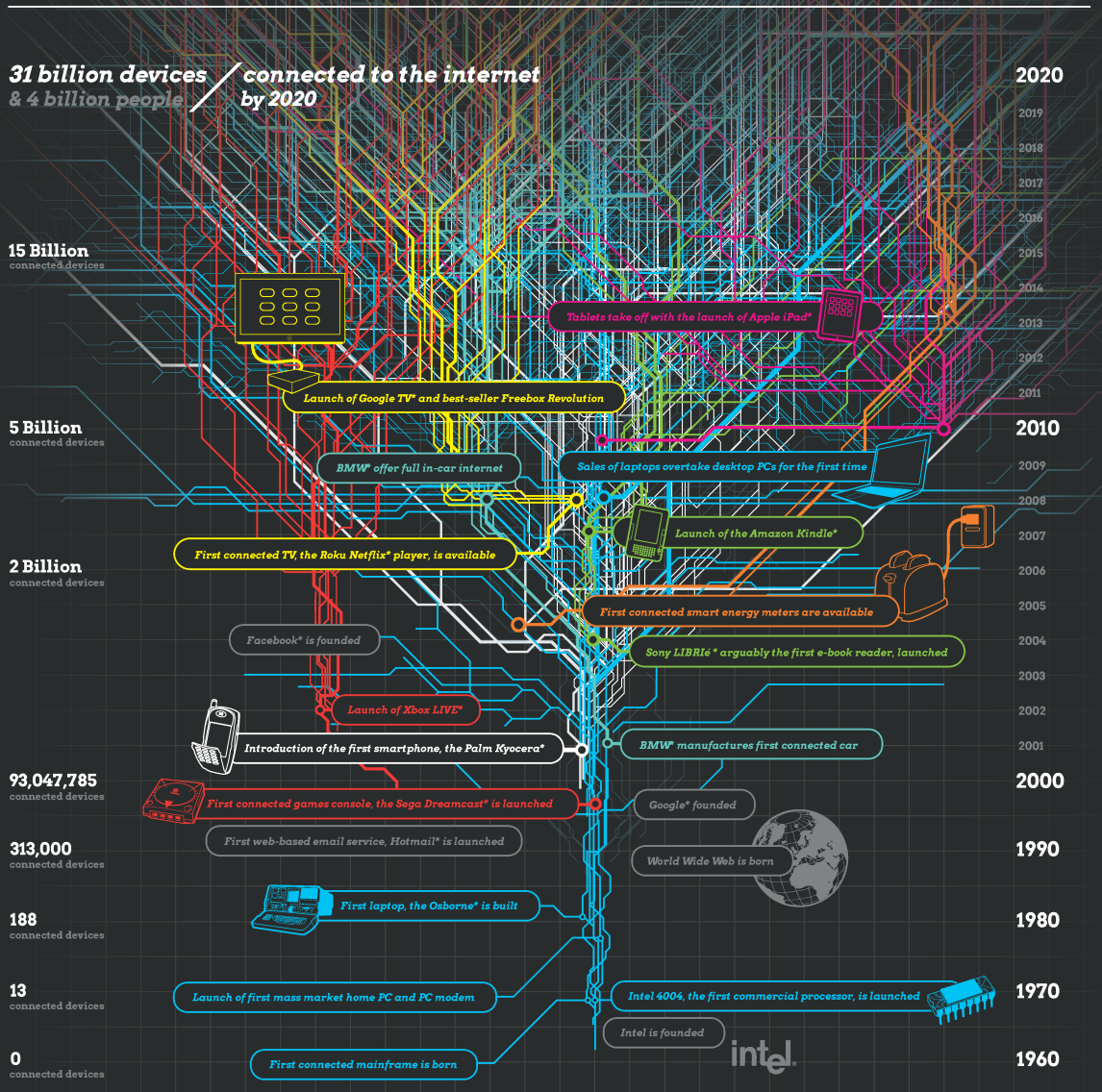
Jedes im Netzwerk befindliche Gerät besitzt eine eindeutige Adresse (IP-Adresse), über welche dieses identifiziert und zur Kommunikation verwendet werden kann. Damit Daten über das Netzwerk übertragen und somit zwischen Netzwerkteilnehmern ausgetauscht werden können, wird ein Netzwerkprotokoll benötigt.

The Internet of Things

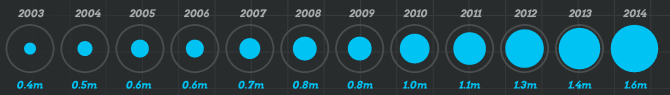
The Internet is evolving, again. Every day, billions of people connect to the Internet through billions of devices – PCs, smartphones and TVs to name just a few. While the PC remains at the centre of this evolution, Internet connectivity is now embedded into cars, fitness equipment, factory robots and vending machines. This smarter, connected world has the potential to change how we live.

Here, Intel has produced a quick snapshot of how the number of connected devices has exploded since the birth of the Internet and the PC, as well as a glimpse forward to 2020. The Internet may already be huge, but it's about to get a lot bigger.

- Mainframes, PCs & Laptops
- Smart TVs
- Tablets
- Smartphones
- E-book Readers
- Smart Energy Meters
- Games Consoles
- Automotive



More than one million PCs sold every day



80% of all PCs shipped today have Intel® Inside

The data referenced in this document came from a variety of sources. For a full list please visit: www.intel.co.uk/internetofthings

Intel is a trademark of Intel Corporation in the United States and other countries. *Other names and brands may be claimed as the property of others.

Abbildung 2.5.: Die Vollbildgrafik zeit die Evolution der Gerätevernetzung seit den 1960er Jahren mit einem Ausblick bis auf das Jahr 2020.

Aus [Absolute Media, 2011].

2.3.2. Netzwerkprotokolle

Ein Netzwerkprotokoll ist ein vereinbartes Kommunikationsprotokoll, unter dessen Verwendung in einem Netzwerk verbundene Geräte Daten austauschen können [Siegmund, 2002]. Dafür existiert eine fest definierte Syntax, welche die Semantik der Kommunikation bestimmt. An einem solchen Kommunikationsvorgang ist selten nur exakt ein Protokoll beteiligt, viel mehr handelt es sich um ein komplexes Zusammenspiel mehrerer, auf unterschiedlichen Ebenen befindlicher Protokolle. Da jedes Protokoll einen anderen Zweck erfüllt, wird - wie in der Informatik üblich - ein Schichtenmodell zur Einordnung der Protokolle verwendet. Durch die Verwendung eines solchen kann die Komplexität verringert und übersichtlicher gestaltet werden. Komplexe low-level Funktionalitäten werden durch die darüberliegende Schicht abstrahiert und dem Programmierer in Form eines einfachen Interfaces zur Verfügung gestellt.

Neben dem standardisierten ISO/OSI-Schichtenmodell, existiert ein weiteres, für die meisten Anwendungen ausreichendes, Vierschichtenmodell (siehe Abb. 2.6), welches lediglich das Internet Protocol (IP) berücksichtigt [Plüss, 2004; Harold, 2004].

Auf der Applikationsebene befinden sich Netzwerkanwendungen, welche mit anderen Applikationen kommunizieren. Aus logischer Sicht kommunizieren diese direkt miteinander (horizontale Verbindung). Physikalisch betrachtet ist eine Netzwerkkommunikation viel komplexer (vertikale Verbindung). In absteigender Sicht betrachtet, nimmt der technische Anteil eines Schichtenmodells immer mehr zu. Laut Harold arbeitet der Code einer Netzwerkapplikation zu 90% der Zeit in der Applikationsebene und muss lediglich mit der Transportebene kommunizieren.

Netzwerkprotokolle können anhand diverser Merkmale voneinander unterschieden werden. Die Art einer Netzwerkkommunikation kann sowohl symmetrisch als auch asymmetrisch funktionieren. Ersteres wäre ein klassisches Peer-2-Peer-Modell, in welchem alle Teilnehmer gleichberechtigt sind. Das am häufigsten eingesetzte asymmetrische Modell ist das Client-Server-Modell [Andrew Davison, 2005]. Hierbei ist der Server allen anderen Teilnehmern übergeordnet, da diese sich zum Server verbinden, um zum Beispiel bestimmte Services in Anspruch zu nehmen. Darüber hinaus kann eine Unterscheidung auch durch die Art der Verbindung vorgenommen werden, namentlich durch verbindungsorientierte und verbindungslose Protokolle. Bei ersteren muss vor dem Übertragen von Daten zunächst eine Verbindung explizit angefragt und aufgebaut werden. Der Pfad der Datenübertragung im Netz

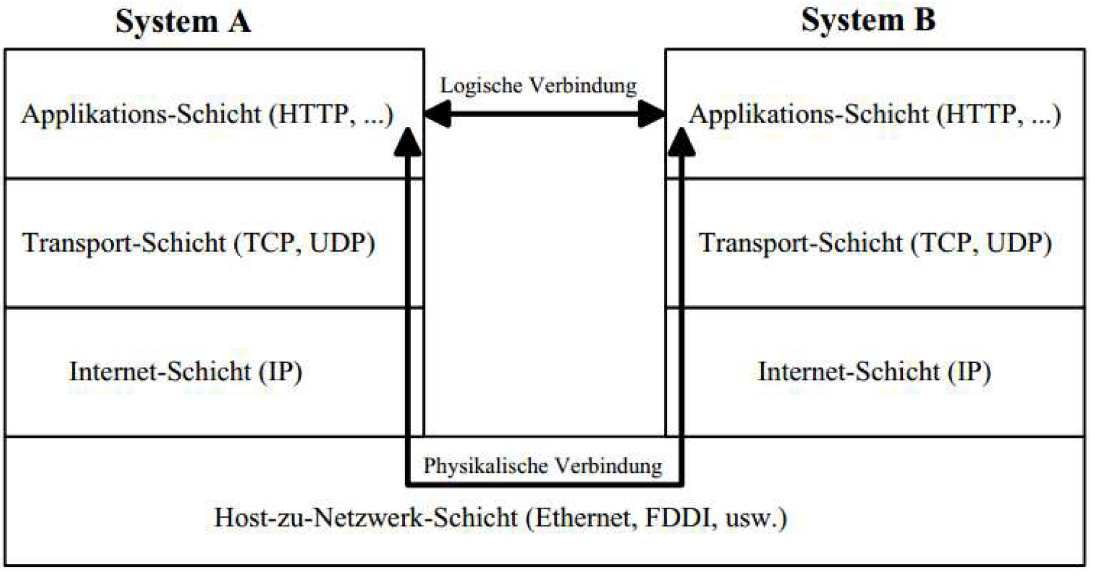


Abbildung 2.6.: Das standardisierte TCP/IP Vier-Ebenen-Modell, speziell für Internetanwendungen. Das ISO/OSI-Modell ist für Internetanwendungen zu generisch und daher für nicht TCP Anwendungen geeignet [Harold, 2004]. Die Anwendung übergibt die zu sendenden Daten an die Transportschicht, welche aus den Daten einzelne Pakete generiert und mit TCP- bzw. UDP-spezifischen Headern versieht und an die Internetschicht weitergibt. Die Pakete werden hier wiederum fragmentiert und den Bestimmungen dieser Schicht entsprechend modifiziert. Die letzte Schicht übersetzt die digitalen in analoge Daten und verwendet das physikalische Medium (Kabel, Strahlen, …) für den Sendevorgang. Auf der anderen Seite findet die Verarbeitungskette umgekehrt statt, wobei zusätzliche Verifikationen durchgeführt werden. Falls ein Paket fehlerhaft übertragen wurde, wird dieses unter der Verwendung von TCP automatisch ein weiteres Mal übertragen (siehe Abschnitt 2.3.2.1).

Aus [Plüss, 2004, Abb. 33.1].

wird zu diesem Zeitpunkt festgelegt und ändert sich nicht mehr. Bei verbindungslosen Protokollen existiert weder eine Verbindung noch ein fester Pfad im Netzwerk. Daten werden in Pakete unterteilt und ins Netz an einen Empfänger entsendet. Da keine Verbindung existiert, kann ein Empfang der Daten beispielsweise nicht quittiert werden. Eine ausführlichere Behandlung dieser Thematik ist in den Abschnitten 2.3.2.1 und 2.3.2.2 zu finden.

Durch das Internet Protocol wird ein paketorientierter, verbindungsloser Ende-zu-Ende-Transport ermöglicht. Daten werden in Pakete unterteilt und einzeln übertragen. Es wird weder eine korrekte Reihenfolge, noch Nachrichtenübertragung garantiert. Ein IP-Paket besteht aus einem Header- und

Datenbereich. Der genaue Aufbau wird vom Betriebssystem abstrahiert und findet daher im Rahmen dieser Projektarbeit keine weitere Betrachtung. Für eine ausführliche Beschreibung sei daher auf [Lindner, 2012, Kap. 30 ff] und [Postel, 1981] verwiesen.

IP unterstützt keine Dienstadressierung, wodurch kein Multiplexing möglich ist. Auf der Transportebene (siehe Abb. 2.6) existieren zwei wichtige Protokolle, welche auf dem IP-Protokoll aufbauen und dieses erweitern: Das Transmission Control Protocol (TCP) und das User Datagram Protocol (UDP). Da diese Ports anbieten, was eine Form der Dienstadressierung darstellt [Kipp, 2002], ist es möglich, mehrere Verbindungen gleichzeitig aufrechtzuerhalten. Ein Port ist keine physikalische Einschränkung, sondern auf dem entsprechenden Gerät über Konfigurationsdateien implementiert. Im Folgenden werden die beiden Protokolle näher beschrieben.

2.3.2.1. Transmission Control Protocol (TCP)

TCP ist ein verbindungsorientiertes Protokoll, dessen Phasen in der Literatur häufig mit dem eines Telefongesprächs verglichen werden [Kipp, 2002; Wikibooks, 2013]. Der Anruf wird von einem Teilnehmer initialisiert; sobald das Telefonat entgegengenommen wird, steht eine Verbindung (vollduplex - zwei entgegengesetzte, unidirektionale Verbindungen), bis einer von beiden das Telefonat und damit die Verbindung beendet. TCP ist mittlerweile im Internet das am häufigsten benutzte Protokoll [Erik Rodriguez, 2012] und wird von Harold als „lingua franca“ der Netzwerkapplikationen bezeichnet. Laut Hidenari Sawashima u. a. sind mittlerweile mehr als 80% der WAN-Ressourcen durch TCP-Verkehr belegt. Seitdem Vinton Cerf Im Jahr 1978 TCP erfand [Cerf u. Kahn, 1974; Massachusetts Institute of Technology, 2000], wurde es stetig weiterentwickelt und war ein intensives Forschungsthema, sodass es mittlerweile eine 40-jährige Evolution vorweisen kann.

Der TCP/IP Stack bietet eine automatische Fehlerkorrektur, Flusskontrolle sowie Übertragungszuverlässigkeit und eine Abstraktion der Lese-/Schreibfunktion (siehe Abschnitt 2.3.5). Häufig wird auch von einer „garantierten Lieferung“ gesprochen [Erik Rodriguez, 2012]. Die Übertragungszuverlässigkeit garantiert, das die gesendeten Daten in der richtigen Reihenfolge beim Empfänger ankommen. Bei Übertragungsfehlern ist ein automatisches Neusenden (Re-Transmission) vorgesehen. Sobald ein Paket gesendet wurde, wird mittels eines Timers[30] überprüft, ob der Empfang quittiert wurde. Sofern dies nicht geschehen ist, sendet TCP dieses Paket automatisch erneut. Zwischenzeitlich gesendete Pakete werden so lange zurückgehalten, bis das verlorengegangene Paket angekommen ist.

Ein großer Vorteil von TCP ist die Flusskontrolle (Blockierungskontrolle). Es ist problemlos möglich, mehrere (100+) TCP-Verbindungen über einen DSL-Link zu betreiben, wobei jede produktiv arbeitet.

[30] zur genauen Funktionsweise und Berechnung des Timers wird auf [Wikibooks, 2013] verwiesen

Dies liegt daran, dass eine TCP-Verbindung u.a. die zur Verfügung stehende Bandbreite sowie die Netzwerkkapazitäten berücksichtigt und nicht willkürlich Daten in das Netzwerk entsendet[31]. Beim Einsatz eines kabellosen Netzwerkes kann sich dies allerdings auch negativ auswirken. Die Wahrscheinlichkeit, dass in einem solchen Netzwerk ein Paket verlorengeht, ist viel höher und führt demnach häufiger zu einer merklichen Empfangsverzögerung, welche in Spielen auch als Latenz[32] bekannt ist. Bei vergleichsweise vielen UDP-Verbindungen käme es zu einem so genannten „verstopfen“ [Jacobson, 1988] der Leitung, da hier eine solche Überprüfung nicht existiert.

TCP/IP implementiert darüber hinaus einen von John Nagle [Nagle, 1984; Braden, 1989a] eingeführten Algorithmus, um den Durchsatz einer Verbindung zu erhöhen. Zu sendende Daten werden von TCP-Anwendungen so lange zwischengespeichert, bis ausreichend viele Daten vorhanden sind, um diese über das Netzwerk zu versenden. Das Senden kleiner Pakete wird dabei dadurch verhindert, dass diese bis zur Maximum Transmission Unit (MTU) zusammengefasst werden. Jedes Paket besitzt einen Header in fester Größe (20 Byte IP- und 20 Byte TCP-Header), welcher bei kleinen Paketen einen unnötigen Overhead generiert. Durch die Zusammenfassung dieser zu einem großen Paket wird das Netz entlastet und eine mögliche Verstopfung verhindert. Besonders bei geringen Bandbreiten kann dadurch der Datendurchsatz erhöht werden [Kyle Kingsbury, 2012]. Latenz-kritische Anwendungen, in denen viele kleinere Nachrichten gesendet werden, können durch das Ausschalten dieses Algorithmus in ihrer Reaktionszeit erheblich verbessert werden.

Seit Jahren existiert eine ausführliche und immer noch anhaltende Diskussion darüber, ob TCP alleine als Protokoll für Spiele angebracht und ausreichend ist, da die zusätzlichen Funktionen die Performance beeinflussen. Der eigentliche Vorteil TCPs, welcher in der Übertragungszuverlässigkeit liegt, kann besonders bei Spielen zum Hauptproblem werden. Sobald ein Paket verloren geht, stoppt die gesamte Netzwerkkommunikation so lange, bis dieses erneut gesendet und empfangen wurde. Die Tendenz der weitaus größten Spieleentwicklercommunity [Betable Tyler, 2012] geht dahin, dass TCP für die meisten Spiele vollkommen zufriedenstellend ist [Watte u. a., 2005]. Laut Harold hat TCP/IP mittlerweile nahezu alle anderen universellen Netzwerkprotokolle effektiv ersetzt. Allerdings gibt es auch Studien [Chen u. a., 2006], welche zu dem Ergebnis gelangen, dass TCP speziell für Massive Multiplayer Online Role Playing Games (MMORPGs) ungeeignet ist. Die Zuverlässigkeit in Netzwerken führt zwangsläufig immer zu einer Erhöhung der Latenz [Andrew Davison, 2005, S. 773]. Bei einer Verwendung der Programmiersprache Flash steht beispielsweise lediglich TCP zur Verfügung.

[31] für weitere Informationen wird u.a. auf [Kipp, 2002] verwiesen.
[32] Latenz ist die Zeit die gebraucht wird bis die gesendeten Daten vom Adressaten empfangen werden. Das Hauptziel bei Mehrspieler-Onlinespielen ist nahezu immer die Latenz zu minimieren.

Ein Echtzeiterlebnis konnte allerdings auch damit geschaffen werden, wie bei folgenden Spielen unter anderem zu erkennen ist: *Stick Arena Dimensions*[33], *Everybody Edits*[34] und *2D Space MMO*[35].

2.3.2.2. User Datagram Protocol (UDP)

UDP ist ein paketorientiertes und verbindungsloses Protokoll. Das Internet ist grundsätzlich in der Lage, Daten zwischen zwei Punkten auf unterschiedlichen Wegen zu transportieren. Im Umkehrschluss bedeutet dies, dass ein später gesendetes Paket den Empfänger früher erreichen kann. Im Gegensatz zu TCP besitzt UDP keine Übertragungszuverlässigkeit oder Flusskontrolle. Pakete können sowohl in falscher Reihenfolge als auch gar nicht beim Empfänger eintreffen. Dieser verminderte Funktionsumfang mündet in einer besseren Übertragungsgeschwindigkeit. UDP bildet damit nur eine sehr schmale Ebene über IP, welche lediglich zusichert, dass gesendete Pakete, wenn diese empfangen werden, tatsächlich vollständig sind.

Laut Erik Rodriguez wird aus diesen Gründen UDP niemals verwendet, um wichtige Daten, wie beispielsweise Datenbankinformationen, zu übertragen. Hauptsächlich wird UDP aufgrund seiner Geschwindigkeit im Bereich des Streamings eingesetzt oder, allgemeiner formuliert, in Anwendungen, in denen eine zuverlässige Datenübertragung auf Transportschicht nicht erforderlich ist und etwaige Fehlerroutinen auf Applikationsschicht vorgesehen sind [Kipp, 2002].

Aufgrund der Tatsache, dass UDP-Ports meistens explizit in der Firewall freigegeben werden müssen, bieten insbesondere Onlinespiele einen Fallback auf die meist problemlos einsetzbaren TCP-Verbindungen an. Einige Spiele ([Griwodz u. Halvorsen, 2006; SWTOR Kundendienst; Guild Wars, 2006; Watte u. a., 2005]) verzichten aufgrund der damit einhergehenden aufwendigeren Netzwerkprogrammierung komplett auf UDP.

Bei UDP liegt der Paketverlust nach Glenn Fiedler durchschnittlich bei 1-5%. Darüber hinaus existieren auch Zeiträume, in denen gar keine Pakete empfangen werden. Insofern muss bei Anwendungen, welche eine gewisse Zuverlässigkeit benötigen und dieses Protokoll verwenden, zusätzlich eine weitere Logikschicht implementiert werden. In deren Aufgabenbereich fallen beispielsweise die folgenden beiden Punkte:

- Aufspaltung der Daten in Pakete (so genannte Datagramme)
- Quittierungssystem mit automatischer Neusendung der Daten.

[33] http://www.xgenstudios.com/play/stickarena, bit.ly/1bTKWJp
[34] http://everybodyedits.com/, http://bit.ly/llkXr4
[35] http://2dspacemmo.wildbunny.co.uk/

Da dieses Protokoll verbindungslos arbeitet, können, im Gegensatz zu TCP, Multicast-Nachrichten versendet werden. Es existieren keine Punkt-zu-Punkt-Verbindungen, sodass ein Teilnehmer Daten an eine Gruppe senden kann, ohne dass sich die Datenübertragungsrate erhöht.

2.3.3. Netzwerkspielmodelle

Bevor in den 1990er Jahren Spiele auf Basis des Internets (TCP/IP) aufkamen und sich ein Wechsel zum *echten* Onlinespiel vollzog, existierten bereits Mehrspielerspiele, welche über das Local Area Network (LAN) gespielt wurden [Glenn Fiedler, 2010; China Daily, 2011]. Das verwendete Netzwerkmodell glich dabei einem Peer-2-Peer-Modell, welches sich dadurch auszeichnet, dass die Topologie ein vollvermaschtes Netz darstellt (siehe Abbildung 2.7), bei dem jeder Netzwerkteilnehmer mit jedem weiteren Teilnehmer verbunden ist. Ein Peer steht dabei für einen verbundenen Spieler. Ein Spiel, welches auf Basis dieses Modells implementiert werden soll, muss dahingehend abstrahiert werden, dass es lediglich aus einer bestimmten Anzahl an vorgegebenen Zuständen und Überführungen besteht, welche während des Spielverlaufs einen Initialzustand deterministisch verändern. Ein konkreter Spielzug besteht aus dem Ausführen von Aktionen (Zustandsüberführungen), welche den Status respektive Zustand des Spiels (und somit auch die Spielwelt), modifizieren.

In einem Mehrspielerszenario startet jeder Spieler im selben Initialzustand und sendet daraufhin jede stattfindende Zustandsüberführung an alle Mitspieler. Eine Schwierigkeit bei der Implementierung eines solchen Modells stellt hierbei der geforderte Determinismus dar. Jede durchgeführte Aktion muss auf jedem Rechner vollkommen identisch ablaufen. Kleinere Ungenauigkeiten häufen sich ansonsten über die Zeit zu einer großen Asynchronität an [Glenn Fiedler, 2010]. Hinzu kommt, dass ein Spielzug erst ausgeführt werden kann, wenn alle Aktionen der anderen Spieler eingetroffen sind, was dazu führt, dass zum Beispiel der am weitesten entfernteste Spieler respektive ein solcher mit der langsamsten Verbindung, die Gesamtlatenz vorgibt[36].

Die Entwickler des Spiels *Doom* bemerkten dieses Problem mit der Verbreitung des Internets, da alle Spielbewegungen an jeden Peer gesendet wurden und erst beim Empfang aller Daten konnte der nächste Frame berechnet werden [Jay Cotton u. Scott Coleman, 1995]. Jeder Router kann außerdem ebenfalls einen zusätzlichen Delay von hunderten Millisekunden hinzufügen, da häufig Daten zwischengespeichert werden, welche den Speicher zum Überlaufen bringen und noch nicht gesendete Daten dadurch zerstört werden, was zu einer erneuten Sendung führt [Andrew Davison, 2005, S. 772]. Ebenfalls darf laut Andrew Davison nicht vernachlässigt werden, dass auch Zeitzonenwechsel zu einem zusätzlichen Delay führen können [Andrew Davison, 2005, S. 772f].

[36] Empfehlenswert als weitere Lektüre ist eine Zusammenfassung der Age of Empires Entwickler [Mark Terrano u. Paul Bettner, 2001a]

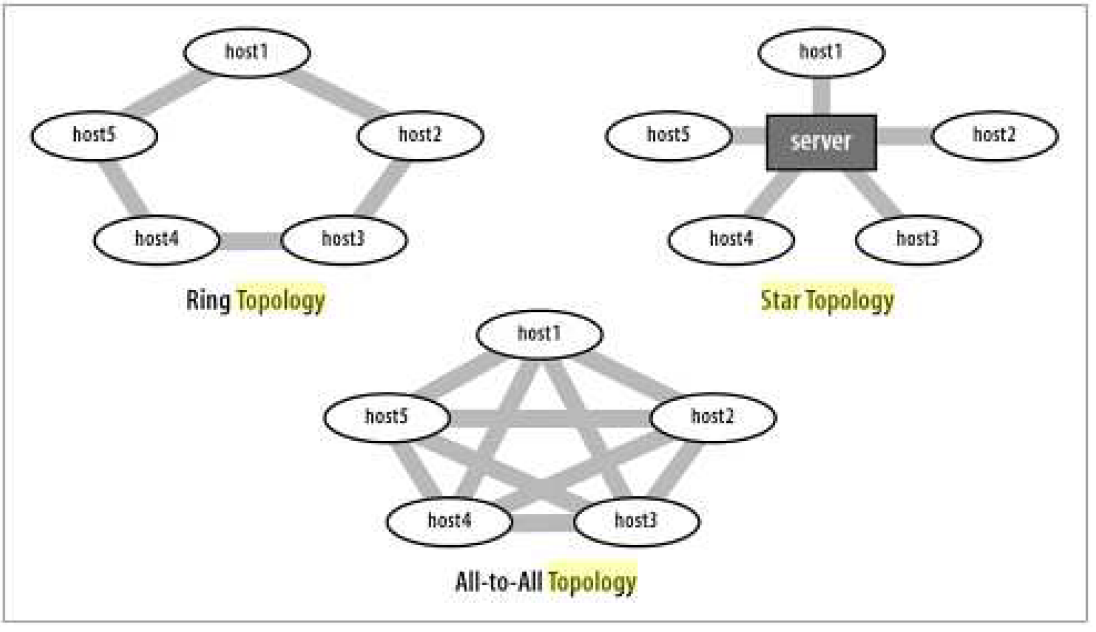


Abbildung 2.7.: Eine Übersicht verschiedener Netzwerktopologien. All-to-All Topology stellt das typische Peer-2-Peer-Modell dar. Das Client-Server-Modell wird durch die Star Topology beschrieben.

Aus [Andrew Davison, 2005, Figure 29-1].

Um diese Probleme zu umgehen, entwickelte und veröffentlichte John Carmack[37] 1996 mit *Quake* das erste Spiel, welches ein anderes Netzwerkspielmodell als Grundlage verwendete: das Client-Server-Modell. Jeder Client (Spieler) fungierte dabei als eine Art Terminal, welcher das Spiel anzeigte und durch einen zentralen Server, mit welchem kommuniziert wurde, aktualisierte [Bringezu, 2001].

In diesem Modell implementiert der Server meist einen Service, welcher von Clients genutzt werden kann, sobald diese zum Server verbunden sind. Häufige Aufgaben eines Services sind: den Clients Daten bereitzustellen oder Daten der Clients entgegennehmen und weiterzuverarbeiten. Die Vorteile eines Servers lassen sich unter anderem wie folgt definieren:

- Eine zentrale Stelle für die Client- und Datenverarbeitung macht es einfacher Benutzer zu überwachen und diese zu administrieren.
- Hauptberechnungen auf einem leistungsstarken Server ausführen zu lassen ermöglicht es, weniger effiziente Clients, wie mobile Endgeräte, zu verwenden.

[37] bekannt als Hauptentwickler für Spiele wie Doom und Quake, heute als CTO der Oculus Rift: http://www.oculusvr.com/

- Der Server kann nicht ausschließlich von Spielclients verwendet werden, sondern zum Beispiel auch von einem Überwachungsclient, welcher aktuelle Statistiken und Geschehnisse anzeigt.

Bei zu vielen verbundenen Clients und Kommunikationsvorgängen kann der Server allerdings auch zum Flaschenhals der Netzwerkanwendung werden und somit Charakteristiken, wie beispielsweise die Latenz, bis zur Unverwendbarkeit verschlechtern.

Nach weiteren Evolutionsstufen und Verbesserungen durch Carmack, ebenso wie durch Tim Sweeney (Epic Games) [Epic Games, Inc., 2013i] oder Yahn Bernier (Valve) [Yahn W. Bernier, 2001], existieren in der heutigen Zeit Ansätze wie die *Client-Side-Prediction* und *Latency Compensation*[38], wodurch der Server lediglich zum Datenabgleich, Schutz vor Betrügern (ein so genannter „Cheatschutz“) und dergleichen eingesetzt wird und die Spiele direkt auf den Clients laufen [Glenn Fiedler, 2010].

2.3.4. Bandbreitenvergleich: Peer-2-Peer - Client-Server

Der Bandbreitenunterschied zwischen diesen Modellen ist hauptsächlich begründet durch die Existenz oder Nichtexistenz eines Servers. Dadurch, dass in einem Peer-2-Peer-Netzwerk alle Daten an jeden im Netzwerk befindlichen Peer gesendet werden, ist die benötigte Bandbreite um ein Vielfaches höher als bei einem Client-Server-Modell, wie folgende Formeln zeigen:

Anzahl der Spieler: $\mathbf{P}$ (2.1)

Bytes pro Spieler pro Update: $\mathbf{B}$ (2.2)

Updates pro Spieler pro Sekunde: $\mathbf{U}$ (2.3)

Header (Overhead) pro Paket: $\mathbf{O}$ (28 Byte bei UDP; 40 Byte bei TCP.) (2.4)

Peer-2-Peer:

Upload: $(P-1) \times U \times (B+O)$ (2.5)

Download: $(P-1) \times U \times (B+O)$ (2.6)

[38] zur weiteren Lektüre wird auf [Glenn Fiedler, 2006] verwiesen

Client-Server:

$$\text{Client Upload: } U \times (B + O) \tag{2.7}$$

$$\text{Client Download: } U \times ((P-1) \times B + O) \tag{2.8}$$

$$\text{Server Upload: } (P-1) \times U \times (B + O) \text{ (Server sendet)} \tag{2.9}$$

$$\text{Server Download: } (P-1) \times U \times ((P-1) \times B + O) \text{ (Server empfängt)} \tag{2.10}$$

aus [Jon Watte, 2008]

Innerhalb eines Client-Server-Modells ist die benötigte Uploadrate (2.7) der Clients viel geringer als deren Downloadrate (2.8). Das ist aus dem Grund gut, da die meisten Internet Service Provider (ISP) eine höhere Download- als Uploadrate zur Verfügung stellen [Gilbert Dietrich, 2011; PCWelt, 2013]. Da bei diesem Modell der Server die Spielqualität bestimmt, ist es wichtig, dass dieser über einen ausreichend hohen Upload verfügt. Ein einzelner Knoten mit genügend hoher Bandbreite kann einfacherer garantiert werden, als dass jeder Knoten diese Anforderungen erfüllt. Ebenfalls ist der gesamte Netzwerkverkehr in Client-Server- im Vergleich zu Peer-2-Peer-Anwendungen viel geringer.

Laut Andrew Davison meiden Spieleprogrammierer Broadcast- respektive Multicastmodelle, welche an die Framerate gekoppelt sind und versuchen die zu übertragenen Daten so gering wie möglich zu halten. Bevorzugt werden Client-Server-Modelle, wobei jeder Client seine Daten lediglich an den Server sendet und sich dieser um die Verteilung kümmert.

2.3.5. Netzwerkprogrammierung in Java

Das UNIX I/O System bildet die Wurzel der Netzwerkkommunikation aller heutigen Implementierungen [Qusay H. Mahmoud, 1996]. Das dahinterstehende Paradigma ist bekannt als *Open-Read-Write-Close*, wobei jeder dieser Begriffe eine Phase bei der Verwendung von Eingabe- bzw- Ausgabeoperationen darstellt.

In Java wurde dieses Paradigma als Ebene über TCP und UDP implementiert, wodurch ein Highlevel-Zugriff auf die Funktionalität der Netzwerkkommunikation geschaffen wurde. Bei der Verwendung von TCP verfügt jeder Kommunikationsteilnehmer über einen eigenen Socket, welcher die Verbindungsdaten enthält. Dieser bildet ein Ende einer Zwei-Wege-Kommunikationsverbindung zwischen Teilnehmern, welche über das Netzwerk miteinander verbunden sind. Es handelt sich um eine komplexe und umfangreiche Datenstruktur, welche die darunterliegende Netzwerkfunktionalität vollständig kapselt und in Form einer vereinfachten Schnittstelle zur Verfügung stellt. Durch die Verwendung des

„Open-Read-Write-Close“-Paradigmas ähnelt die Funktionsweise eines Sockets in Java dem eines üblichen Dateizugriffs. Ferner besitzt ein Socket einen Eingabe- und Ausgabestream, welche eine Kommunikation zwischen den beiden Teilnehmern ermöglichen. Ersterer wird zum Lesen von empfangenen Daten verwendet, wohingegen letzterer zum Schreiben von Daten an den Kommunikationspartner verwendet wird. Java bietet je nach Einsatzzweck viele verschiedene Ausprägungen solcher Sockets an. Des Weiteren existieren über so genannte Filter Streams, welche die Funktionalität der Standardstreams erweitern, zusätzliche Möglichkeiten wie: Übertragung von Java-Primitiven im Binärformat, De-/Kompression und Ver-/Entschlüsselung.[39].

Des Weiteren unterscheidet Java in der Art der Sockets und bietet diese in zwei verschiedenen Ausprägungen an. Ein Server, welcher auf Verbindungsanfragen wartet, implementiert einen *ServerSocket*. Ein Client hingegen, welcher eine Verbindung initiiert, implementiert einen einfachen *Socket*. Ersterer wird an eine bestimmte IP-/Port-Kombination gebunden, über welche dieser für Clients erreichbar ist.

Der Gesamtablauf einer Client-Server-Kommunikation lässt sich nach [Becker, 2012, Kap. 6] in folgende Schritte unterteilen:

1. Der Server erzeugt einen ServerSocket, der an einen vordefinierten Port gebunden ist.
2. Der Server wartet auf eingehende Verbindungsanfragen.
3. Der Client erzeugt einen Socket mit der IP-Adresse und der Portnummer des Servers.
4. Beim Server wird die Anfrage angenommen und liefert als Resultat einen Socket, welcher eine Verbindung zum Client darstellt. Dieser wird in der Literatur auch als „rendezvous socket“ benannt [Andrew Davison, 2005]. Der ursprüngliche Port des Server steht weiterhin für Verbindungsanfragen zur Verfügung, für jede aufgebaute Verbindung wird ein eigener Port vergeben.
5. Client und Server erzeugen Stream-Objekte für die weitere Kommunikation.
6. Daten können mit gewöhnlichen Lese- und Schreiboperation zwischen den Prozessen ausgetauscht werden.

Java bietet ferner weitere Methoden und Einstellungsmöglichkeiten an, wodurch es beispielsweise möglich ist, das Zusammenfassen von TCP-Paketen auszuschalten (Nagle Algorithmus) oder die Art des Services in Form von drei Kriterien (Latenz, Verbindungszeit, Bandbreite) anzugeben, woraufhin die Arbeitsweise des Sockets angepasst wird.

[39] als weitergehende Literatur wird [Harold, 2004, Kap. 4, Kap. 11] empfohlen

Neben den Stream-Sockets, welche bei der Verwendung von TCP verwendet werden, existieren auch *DatagramSockets*. Dadurch wird ein verbindungsloser Pakettransport auf Basis von UDP ermöglicht. Im Rahmen dieses Projektes wurde auf den Einsatz von UDP verzichtet (siehe Abschnitt 3.1.5), sodass für eine detaillierte Beschreibung der DatagramSockets auf weiterführende Literatur, wie beispielsweise [Harold, 2004; Ullenboom, 2012], verwiesen wird. Ferner existiert in Java seit 2009 die so genannte New I/O (NIO)-API, welche mehrere Verbindungen in einem einzigen Thread behandelt. Dies geschieht mittels Multiplexing, wobei sequentiell immer die Verbindung bearbeitet wird, welche neue Daten bereithält. Laut [Harold, 2004] ist die damit zusammenhängende Komplexität für die meisten Anwendungen allerdings unkontrollierbar, weshalb an dieser Stelle ebenfalls auf weitergehende Literatur verwiesen wird.

2.3.6. Netzwerkkommunikation

Die Zugriffszeiten auf Ressourcen im Netzwerk sind im Vergleich zu denen auf Festplatten respektive den Speicher um ein Vielfaches langsamer [Mikito Takada, 2011]. Daher kann eine Netzwerkverbindung sehr schnell zum Flaschenhals der gesamten Anwendung werden. Zur Kompensation existieren daher verschiedene Möglichkeiten der Netzwerkkommunikation, welche häufig auch als *I/O-Strategy* bezeichnet wird.

Nach [Young, 2009] existieren neben anderen, für diese Arbeit irrelevanten Methoden, die Folgenden:

Blockierende Sockets: die Standardimplementierung aller Sockets

Polling: in vorgegebenen Intervallen wird versucht ein Lesezugriff auszuführen; sofern dies aufgrund fehlender Daten fehlschlägt, wird der Socket geschlossen

Asynchrone Sockets: nicht blockierende Sockets, welche automatisch eine Funktion aufrufen, sobald neue lesbare Daten vorhanden sind

Threadbasierte Verfahren: ermöglichen die Blockierung anderweitig zu umgehen.

Die ersten drei Varianten können einerseits isoliert, andererseits jedoch auch in Verbindung mit Threads verwendet werden.

2.3.6.1. Blockierende Sockets

Die Standardimplementierung aller Sockets beinhaltet den so genannten blockierenden Modus. Sobald beispielsweise ein Lesezugriff ausgeführt wird, blockiert der Socket so lange, bis Daten zum Lesen vorhanden sind. Bei Hintergrundprozessen, welche keine eigene Benutzeroberfläche besitzen, stellt sich diese Funktionsweise als weniger problematisch heraus. Bei Anwendungen mit echzeitkritischer Ausgabe hingegen kann dieser Ansatz nicht verwendet werden.

2.3.6.2. Polling-Ansatz

Um den blockierenden Modus zu umgehen, lässt sich ein so genanntes Polling (zyklisches Abfragen) einsetzen. Hierbei werden Methoden auf den Sockets häufig in eigenen Threads ausgeführt, welche mit einem Timer verbunden sind. Sobald dieser abgelaufen ist und der Thread nicht beendet wurde, wird eine Beendigung forciert. Der Aufruf wird intervallweise so oft wiederholt, bis dieser erfolgreich abgeschlossen wurde. Winsockets auf Betriebssystem-Ebene verfügen über eine hardwareseitige Implementierung dieses Polling-Ansatzes, sodass bei systemnaher Umsetzung diese Funktionsweise unmittelbar verwendet werden kann.

Das Polling ist allerdings sehr ineffizent. Die Anwendung führt in regelmäßigen Abständen Methodenaufrufe aus und wartet auf Ergebnisse, welche eventuell nur in unregelmäßigen Abständen eintreffen. Damit werden Ressourcen und CPU-Zeit unnötigerweise in Anspruch genommen. Hinzu kommt zusätzlich die komplexe Logik, welche benötigt wird um diesen Ansatz zu implementieren.

2.3.6.3. Asynchrone Sockets

Asynchrone Sockets verwenden Threads zur Umgehung der Blockierung. Dies geschieht vollkommen automatisiert, d.h. Entwickler müssen sich nicht zusätzlich um eine Threadprogrammierung kümmern, da asynchrone Sockets die Funktionsweise bereits nativ zur Verfügung stellen. Beim Initialisieren eines Lesezugriffes wird zusätzlich eine Callbackfunktion mit übergeben, welche bei Beendigung desselben aufgerufen wird. Speziell in Java existiert jedoch das Problem, dass auch nicht vollständige Nachrichten bereits zurückgemeldet werden [Andrew Davison, 2005]. Dies führt dazu, dass zusätzlich ein Caching-Algorithmus implementiert werden muss, welcher überprüft, ob das Ende einer Nachricht auch tatsächlich erreicht wurde. Es ist jedoch auch möglich, dass ein Callback der Leseoperation bereits einen Teil der neuen Nachricht enthält, sodass das Ende der vorangegangenen Nachricht übersehen wird.

Laut Young erweitern asynchrone Sockets die Anwendung um eine zusätzliche Komplexitätsstufe, sodass diese nicht empfehlenswert sind.

2.3.6.4. Threadbasierte Verfahren

Alle zuvor beschriebenen Verfahren umgehen zwar den Aspekt der Blockierung, können allerdings nicht mehrere Verbindungen verwalten. Für dieses Vorhaben werden zwangsläufig Threads benötigt.

Das Entgegennehmen von Verbindungen, sowie Schreib- und Leseoperationen, werden in eigenständigen Threads absolviert. Jeder Thread besitzt somit eine fest definierte Aufgabe (Kapselung) und ist darüber hinaus an einer zentralen Stelle registriert. Dies erleichtert die Wartbarkeit sowie Kontrolle dieser. Ebenfalls laufen alle Threads unabhängig voneinander ab, sodass keine Interprozess- bzw. Interthreadkommunikation notwendig ist.

Das größte Problem, welches mit der Verwendung von Threads einhergeht, ist der Datenaustausch untereinander respektive die Synchronizität der Daten. Häufig auftretende Phänomene sind Deadlocks oder Race Conditions.

2.4. Threads

Aufgrund der im Vorfeld kurz angesprochenen Problematik beim Einsatz von Threads, wird im Folgenden eine ausführlichere Darstellung dieser vorgenommen. Dies ist deshalb notwendig, da auch in der Umsetzung des hier beschriebenen Projekts Threads eine Anwendung finden (siehe Abschnitt 3.2).

Ein Prozess bezeichnet den Ablauf eines Programms auf einer Recheneinheit, welchem innerhalb des Betriebssystems ein bestimmte Speicherbereiche und Ressourcen zugeordnet werden. Dieser Prozess kann beliebig viele Threads instantiieren und beinhalten. Ein Thread beschreibt einen eigenständigen Ablauf innerhalb eines Programms, welcher in multitaskingfähigen Umgebungen zeitgleich und unabhängig von anderen Threads ausgeführt werden kann. Da diese sich denselben Speicherbereich und damit einhergehende Ressourcen teilen, ist der Verwaltungsaufwand für Threads viel geringer und die Effizienz umso höher, weshalb diese in der Literatur häufig leichtgewichtige Prozesse genannt werden. In Korrelation dazu steht allerdings die Problematik, einen gleichzeitigen Zugriff mehrerer Threads auf eine Ressource zu unterbinden.

Auf Betriebssystemebene existiert ein so genannter Scheduler, welcher die Prozessorzeit auf einzelne Prozesse respektive Threads verteilt. Häufig handelt es sich hierbei um ein präemptives Scheduling,

was bedeutet, dass Threads während ihrer Ausführung unterbrochen werden können.[40] Durch die Anwesenheit von Lesern (Threads, welche Variablen lesen) und Schreibern (Threads, welche den Inhalt von Variable verändern) kann es zu Inkonsistenzen in der Verarbeitung kommen. Ein Schreiber kann während der Abarbeitung eines Segmentes unterbrochen werden, was dazu führt, dass mögliche Leser zum Teil geänderte und zum Teil ungeänderte Daten verwenden. Durch eine Synchronisierung dieser Routinen mit Hilfe so genannter Monitore, kann der gleichzeitige Zugang gesperrt und eine sequentielle Bearbeitung erzwungen werden. Das damit einhergehende Problem, dass ein Schreiber auch andere Leser aussperrt, ist in der Literatur als Leser-Schreiber-Problem bekannt [Harold, 2004; Jobst, 2011].

Java implementiert solche Monitore in Form von synchronized-Statements bzw. -Blöcken. Es wird ein exklusiver Zugang zu einem Objekt ermöglicht, welcher jedoch auch mit Einschränkungen verbunden ist. Handelt es sich bei dem Objekt beispielsweise um die Standardausgabe, so wird diese nicht im Allgemeinen auch allen unbeteiligten Threads verwehrt, sondern nur solchen, welche versuchen, dieses Objekt auch zu synchronisieren. Speziell im Hinblick auf Methoden bedeutet dies, dass sobald ein Thread eine Methode betritt, andere Threads so lange suspendiert werden, bis die Abarbeitung der Methode vollzogen wurde. In der Programmierung ist dies als Verbraucher-/Erzeuger-Problem bekannt [Jobst, 2011; Harold, 2004].

Der Einsatz solcher Monitore kann nach [Lea, 1999] wiederum zu weiteren Problemstellungen führen:

Zu häufiger Einsatz kann zu Performanzeinbußen führen, dadurch, dass zu viel sequentiell abgearbeitet wird, wodurch die ursprünglich geplante Parallelität ad absurdum geführt wird oder erhöhen die Gefahr auf einen Deadlock[41].

Unvorsichtiger Einsatz führt zum Beispiel dazu, dass komplette Methoden synchronisiert werden, obwohl lediglich eine Variable geschützt werden soll.

2.5. Android

Wie bereits in der Zielsetzung des Projektes[42] erwähnt wurde, sollen für die Interaktionen der Spieler Smartphones auf Android-Basis verwendet werden. Eine Begründung dieser Entscheidung findet sich im Abschnitt 3.1.7. Als Überblick und zum Verständnis der Funktionsweise Androids befindet sich in

[40] Es existieren darüber hinaus eine Vielzahl weiterer Möglichkeiten wieso Threads pausiert werden können, eine ausführliche Behandlung ist in [Harold, 2004] zu finden

[41] In [Harold, 2004, Kap. 5] werden diverse Präventionstechniken gegen den zu häufigen Einsatz von Monitoren beschrieben.

[42] Vgl. Abschnitt 1.2

diesem Kapitel eine kompakte Zusammenfassung dieses mobilen Betriebssystems und seiner Komponenten.

Android war ursprünglich als ein Digitalkamera-Betriebssystem geplant, womit so genannte „Smart-Kameras“[43] entwickelt werden sollten [Jay Alabaster, 2013]. Seitdem Google im Jahre 2005 sowohl das Unternehmen als auch die Software aufkaufte und seit 2007 die Entwicklung Androids mit der Open Handset Alliance[44] aufgenommen hat, ist Android als ein mobiles Betriebssystem fest im Markt etabliert. Obwohl es lediglich für Smartphones entwickelt wurde, ist es mittlerweile auf Tablets, Netbooks und auch in virtuellen Maschinen verschiedener Gastsysteme lauffähig.

Android wird als quelloffenes Betriebssystem vermarktet und steht allen Herstellern kostenlos zur Verfügung. Aus diesem Grund und da es im Allgemeinen als sehr flexibel in der Anpassung gilt, wird es mittlerweile[45] von 320 Anbietern in 160 Ländern erfolgreich eingesetzt [Eric Schmidt, 2013].

Im zweiten Quartal 2013 schrieb das Marktforschungsunternehmen *IDC* Android einen Marktanteil von fast 80 Prozent zu [IDC - Press Release, 2013]. Seitdem das erste Android-Gerät im Oktober 2008 auf den Markt kam, wachsen die Marktanteile und damit die Verbreitung und der Bekanntheitsgrad Androids stetig und sehr schnell, wie Abbildung 2.8 zeigt[46].

Ausschlaggebender als die insgesamt verkauften Android-Geräte sind hingegen tägliche Nutzungsstatistiken respektive die Nutzung von Apps. Hugo Barra, Vizepräsident für Android-Produkt-Management, gab noch im September 2012 bekannt, dass täglich rund 1,3 Millionen Android-Geräte aktiviert werden [Barra, 2012]. Mittlerweile sind es täglich bereits 1,5 Millionen aktivierte Geräte und insgesamt 900 Millionen [Czerulla, 2013]. Darüber hinaus wurden bereits 48 Milliarden Apps aus Google Play heruntergeladen.

Diese weite Verbreitung und Offenheit des Betriebssystems führt allerdings auch zu einem negativen Effekt, nämlich einer sehr starken Fragmentierung. *Staircase*, Entwickler einer Maps-ähnlichen App, untersuchte ein halbes Jahr lang, auf welchen Geräten ihre App installiert worden war. Als Ergebnis dieser Untersuchung wurden 3.997 Geräte von 599 Herstellern festgestellt [Andreas Floemer, 2012]. Damit werden besonders die Appentwickler vor eine große Herausforderung gestellt, da die App trotz unterschiedlicher Komponenten, wie beispielsweise der Displayauflösung, geräteübergreifend funktionieren soll. Der Testaufwand, welcher in gewöhnlichen Softwareprojekten bereits zwischen 30 und 40% der Gesamtkosten ausmacht[Pol u. a., 2002], wächst in der Appentwicklung damit noch einmal

[43] diese sollten sich unter anderem dadurch auszeichnen, dass sie besonders einfach mit Computern kommunizieren können.
[44] Die OHA ist ein Konsortium aus mittlerweile 84 Unternehmen, deren Ziel es ist offene Standards für Mobilgeräte zu schaffen. Neben Google sind unter anderem die Folgenden angehörig: T-Mobile, eBay, Intel, Samsung, nVidia, Texas Instruments und Qualcomm.
[45] zum Zeitpunkt dieser Ausarbeitung
[46] Im Anhang (B.3) befindet sich eine weitere Abbildung, welche die Marktverteilung der mobilen Betriebssysteme in den vergangenen sieben Jahren aufzeigt.

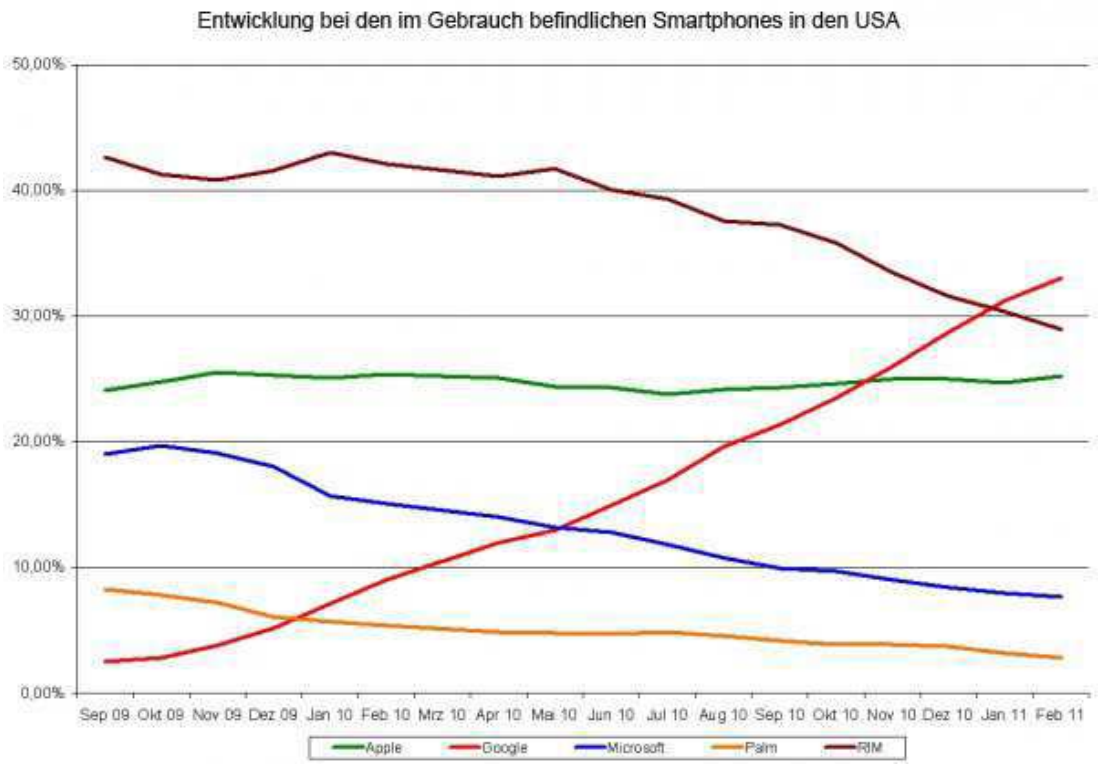


Abbildung 2.8.: Eine Visualisierung der comScore-Reports von September 2009 bis Februar 2011 zeigt die in Gebrauch befindlichen Geräte in den USA.

Aus [Lion13, 2011].

um einen großen Faktor an. Die Softwareschmiede *animoca* zeigt, wie aufwendig es mittlerweile geworden ist Apps zu testen [Kim-Mai Cutler, 2012]. In ihren Büroräumen haben sie über 600 unikale Android-Geräte ausmachen können.

Wie jedes andere Betriebssystem für Smartphones respektive Tablets, wird Android ausschließlich per Touchscreen bedient, neben einigen wenigen benötigten Software- oder Handwaretasten. Seitdem Android auch auf Netbooks lauffähig ist, werden auch andere Bedienelemente wie eine Maus oder Tastatur unterstützt. Damit auch erweiterte Funktionalitäten abgebildet werden können, setzt Android, wie auch andere mobile Betriebssysteme, auf bestimmte Gesten. Wegen der Vielfalt in diesem Bereich existieren bereits Touch-Gesten Referenz-Kärtchen zum Ausdrucken, welche die gängigen Gesten für Smartphones übersichtlich darstellen [Craig Villamor u. a., 2010]. Damit kann ein Smartphone für

diverse Arbeiten eingesetzt werden; vom einfachen Mausersatz bei Präsentationen bis hin zu einem komplexen Interaktionsgerät für virtuelle Realitäten [Craig Villamor u. a., 2010].

2.5.1. Architektur

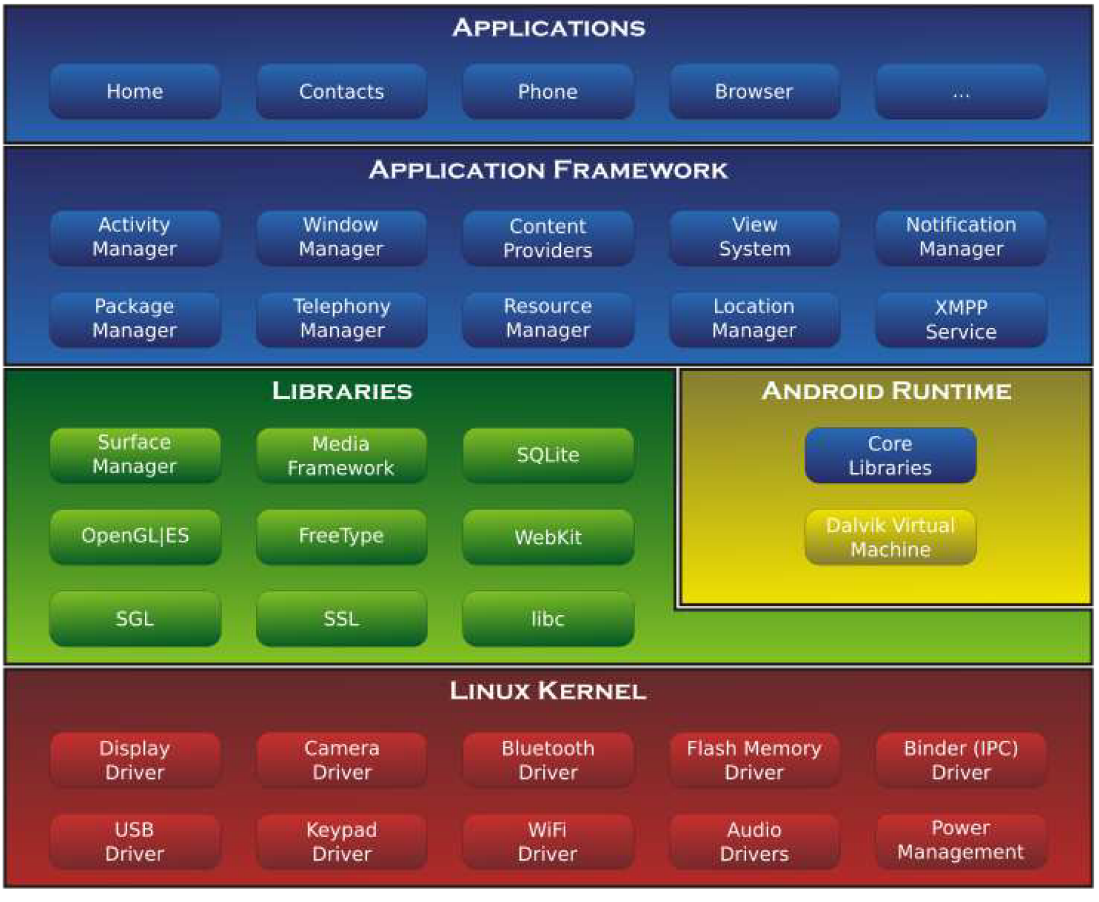


Abbildung 2.9.: Eine Übersicht über die Architektur von Android.

Aus [Dan Bornstein (Google), 2008].

Die Basis von Android bildet ein Linux Kernel, welcher zum einen für die Speicher- und Prozessverwaltung zuständig ist und zum anderen diverse Treiber zur Verfügung stellt. Ebenso wird eine Schnittstelle zur Netzwerkkommunikation zur Verfügung gestellt. Darüber befindet sich die Laufzeitumgebung,

welche auf der Dalvik Virtual Machine (DVM) basiert, sowie weitere zur Verfügung gestellte Bibliotheken. Für eine ausführlichere Beschreibung der OpenGL-, respektive Audio/Video-Nutzung, besonders im Zusammenhang der Spielentwicklung wird auf [Silva, 2012] verwiesen.

Die Dalvik Virtual Machine stellt eine angepasste Java Virtual Machine dar, welche auf der offenen JVM Apache Harmony basiert. Dadurch lassen sich Anwendungen auch in der Programmiersprache Java schreiben. Die Anpassungen hatten u.a. das Ziel, die speziellen Anforderungen mobiler Endgeräte zu erfüllen: CPU-effizient, speichereffizent, Betriebssystem ohne Swap-Space und Akkubetrieben [Dan Bornstein (Google), 2008]. Die DVM führt ebenfalls Byte-Code aus, basiert allerdings auf einer Registermaschine, anstatt wie andere virtuelle Maschinen auf einem Kellerautomaten. Die Vorteile sind u.a. das Verhindern unnötiger Speicherzugriffe, Ressourcenschonung und Effizienz respektive Schnelligkeit [Becker u. Pant, 2009, Kap. 2.2].

Eine Ebene darüber befindet sich das Anwendungsframework. In diesem befinden sich Komponenten, welche Systemdienste wie den *Telephony Manager* bereitstellen, über den beispielsweise ein Anruf initialisiert werden kann.

In der obersten Ebene befinden sich die Applikationen selber, sowohl Standardapplikation, als auch eigens entwickelte.

2.5.2. Komponentenmodell

Android definiert ein einfaches Komponentenmodell. Gruhn u. Thiel definieren ein solches wie folgt:

> *„Ein Komponentenmodell legt einen Rahmen für die Entwicklung und Ausführung von Komponenten fest, der strukturelle Anforderungen hinsichtlich Verknüpfungs- bzw. Kompositionsmöglichkeiten sowie verhaltensorientierte Anforderungen hinsichtlich Kollaborationsmöglichkeiten an die Komponenten stellt. Darüber hinaus wird durch ein Komponentenmodell eine Infrastruktur angeboten, die häufig benötigte Mechanismen wie Verteilung, Persistenz, Nachrichtenaustausch, Sicherheit und Versionierung implementieren kann.“* [Gruhn u. Thiel, 2000, S. 293]

Komponenten werden demnach in Form und Ausprägung spezifiziert, ebenso wie die Kommunikations- und Verbindungsschnittstelle. Weitere Beispiele für Komponentenmodelle sind Enterprise JavaBeans (EJB) oder Open Services Gateway initiative (OSGi).

In Android existieren u.a. die folgenden Komponenten: Activities, Services und Broadcast Receiver. Erstere sind vergleichbar mit Applets. Sie sind für die sichtbaren Teile der Anwendung zuständig und

dienen der Interaktion mit dem Nutzer. Zur weiteren Unterteilung existieren darüber hinaus *Subactivities*. Diese werden von anderen Activities aus gestartet, führen Berechnungen durch und geben das Ergebnis an die aufrufende Instanz zurück. Laut Becker finden diese oft in Master-Detail-Szenarien[47] Einsatz [Becker, 2012, Kap. 4]. Services besitzen keine eigene Benutzeroberfläche, da sie im Hintergrund arbeiten. Sie bieten dafür Schnittstellen für den Zugriff durch andere Komponenten. Insofern kann ein Service nicht nur als Alternative zu einem Thread angesehen werden, sondern viel mehr als ein echter Servicedienstleister, welcher im Hintergrund Berechnungen durchführt und jederzeit von Komponenten angesprochen werden kann. Broadcast Receiver dienen dazu, um auf Systemereignisse reagieren zu können. Wenn beispielsweise die App über eine abgebrochene Internetverbindung informiert werden soll, kann ein Broadcast Receiver verwendet werden, um auf dieses Ereignis zu reagieren. Außerdem dienen sie auch dem Empfang von *Intents* - einer weiteren Komponente Androids.

Um Activities sowie Services miteinander verbinden zu können, existieren verschiedenartige Intents. Eine lose Kopplung von Komponenten ist mittels impliziten Intents möglich. Die Empfängerkomponente ist beim Aufruf nicht bekannt, was dazu führt, dass eine beliebige Anzahl von Activities oder Services auf diesen reagieren kann. Welche letztlich den Vorzug erhält und aufgerufen wird, hängt von der Android-internen *Intent-Resolution* ab. Um eine konkrete Komponente aufzurufen existieren so genannte explizite Intents. Eine solcher kann mit weiteren Daten versehen werden. Sofern eine Activity beispielsweise Benutzereingaben entgegennimmt und diese einem Hintergrundservice zu Berechnungszwecken zur Verfügung stellen muss, kann ein so genanntes *Bundle*, welches die zusätzlichen Informationen beinhaltet, an einen Intent übergeben werden (siehe Abb. 2.10). Neben den genannten existieren ferner *Broadcast Intents*, welche hauptsächlich auf Betriebssystemebene verschickt werden. Hierunter fallen zum Beispiel Ereignisse über den Akkuzustand oder die Internetverbindung, welche sich einen solchen Broadcast Intent zu Nutze machen.

Als letzte, für das Ziel dieser Arbeit benötigte Komponente, ist die *View* zu nennen. Views sind letztlich Elemente für die Benutzerschnittstelle. Es ist die Basisklasse für alle visuellen Komponenten. Speziell hiervon abgeleitet sind Widgets, welche für die direkte Kommunikation mit dem Benutzer verwendet werden (Buttons, Eingabefelder, etc.).

2.5.3. Threads in Android

In der App-Entwicklung wird der parallelen Programmierung ein viel höherer Stellenwert zugeschrieben als gewöhnlichen Softwarelösungen. Parallelität in der Entwicklung dient häufig in erster Linie

[47] Master-Detail-Szenarien sind in der Softwareentwicklung im Zusammenhang mit Bedienoberflächen auch als Master-Detail-Beziehung bekannt. Die Master-Ansicht zeigt lediglich die Titel von Einträgen an. Bei einem Klick auf einen dieser Einträge, werden zusätzliche Informationen zu diesem Objekt an anderer Stelle angezeigt.

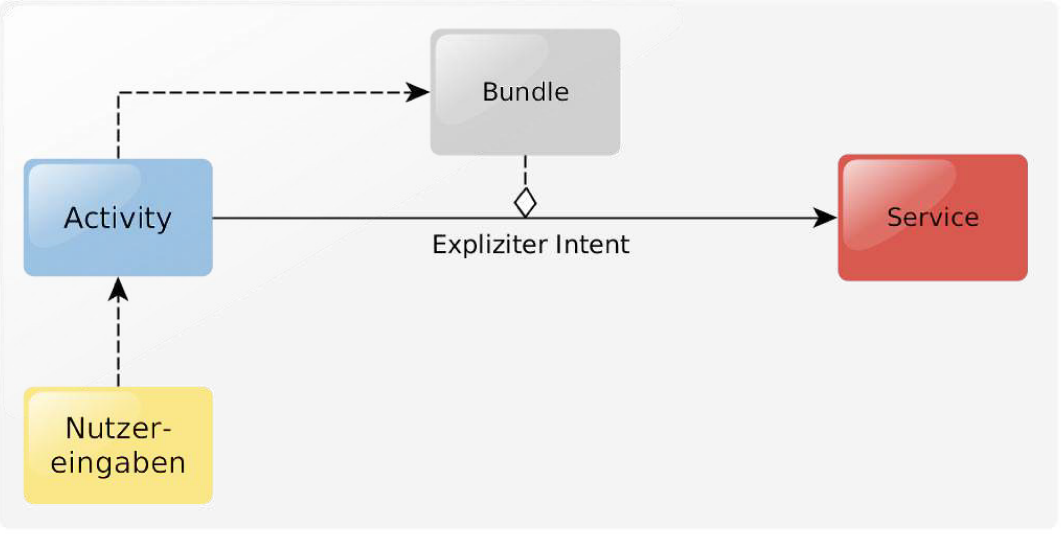


Abbildung 2.10.: Durch Explizite Intents können Daten an bestimmte Komponenten gesendet werden. Ferner kann durch diese Art von Intents auch eine andere Komponente (Activity) gestartet weden.

einer möglichen Effizienzsteigerung. Im Bereich der mobilen Applikationen sind zwangsläufig immer Benutzeroberflächen involviert. Lange Berechnungen führen dabei zum Einfrieren dieser und verschlechtern damit die Benutzbarkeit der Anwendung [Becker u. Pant, 2009; Young, 2009].

Android startet für jede Applikation einen neuen Prozess, welcher wiederum einen speziellen UI-Thread erzeugt. Unter dessen Aufgabenbereich fallen alle sichtbaren Teile einer Anwendung. Beim Ausführen von langanhaltenden Berechnungen in Activities reagiert Android mit einem Application-not-responding-Event. Dieses wird bereits ausgelöst, wenn Widgets nicht innerhalb von fünf Sekunden auf Berührungen reagieren [Rogers, 2009]. Daher müssen bei der Androidentwicklung vermehrt Threads eingesetzt werden.

Der große Vorteil, den Android dabei Entwicklern bietet, ist, dass der Thread den gleichen Speicherbereich verwendet und damit prinzipiell auf alle Objekte innerhalb der Anwendung zugreifen kann. Dadurch wird dem zusätzlichen Aufwand einer Interprozesskommunikation entgegengewirkt. Ausgenommen hiervon sind jedoch jegliche Benutzerschnittstellenelemente. Änderungen an diesen sind lediglich dem UI-Thread vorenthalten, welcher über so genannte *Handler-Objekte* von anderen Threads aus angesprochen werden kann.

KAPITEL 3

Realisierung

Nachdem die notwendigen technischen Grundlagen vorgestellt sowie die im Kontext dieser Arbeit verwendeten und substantiellen Fachtermini erklärt wurden, widmet sich dieses Kapitel der konkreten Umsetzung des Projektes. Zu diesem Zweck werden zunächst die Designentscheidungen näher beschrieben, welche die Umsetzung im weiteren Verlauf definieren. *Designentscheidungen* im Sinne dieser Arbeit sind solche Entscheidungen, die sowohl verwendete Werkzeuge zur Prototyp-Erstellung festlegen, als auch konkreten Aufschluss darüber geben, warum und an welcher Stelle Technologie-Entscheidungen notwendig waren.

Im zweiten Teil dieses Kapitels werden die konkreten Konzepte und ausgewählte Problemstellungen präsentiert, an dessen Beispielen die Entwicklung des Prototypen tiefergehend verdeutlicht werden kann.

3.1. Designentscheidungen

Um den Prototyp umzusetzen waren gewisse Auswahlentscheidungen nötig, die nun an dieser Stelle behandelt werden. Ferner werden die ausgewählten Technologien respektive Paradigmen hier eingeführt und auf die entsprechenden Detailabschnitte der Arbeit verwiesen.

Nicht alle Elemente eines Rollenspiels lassen sich sinnvoll für die digitale Welt adaptieren. Die im Abschnitt 2.1 identifizierten Punkte werden daher sequentiell betrachtet und in kumulative, übergeordnete Einheiten zusammengefasst. Zudem finden sich im Verlauf dieses Kapitels Entscheidungsbegründungen im Hinblick auf die Wahl einer Game-Engine, der damit kommunizierenden Programmiersprache(n), Design und Auswahl eines Netzwerkprotokolls, sowie die Vorstellung der gewählten Eingabegeräte und deren Betriebssystem.

3.1.1. Digitalisierung von Regelwerk und Spieldesign

Bei der Übertragung von Regelwerk und Spieldesign in die digitale Welt muss eine Unterscheidung getroffen werden.

Zunächst soll die Digitalisierung des *Regelwerks* besprochen werden. Traditionell wird dieses, mit samt seinen harten und weichen Regeln, in der Game-Engine verankert [Tychsen u. a., 2005, S.2]. Im Rahmen der in diesem Projekt angestrebten Digitalisierung wird jedoch eine Unterscheidung getroffen: Die harten Regeln werden durch die Game-Engine festgelegt (wie können sich die Spielenden bewegen, können sie fliegen, wie schnell sind die Spielenden …) und weiche Regeln, sowie die Hintergrundgeschichte, werden in der Decision-Engine verankert. Der Grund für diese Trennung muss durch einen Vorgriff genannt werden, dessen Details in Abschnitt 3.1.3 erläutert werden. Der Game Master soll eine vollständige Digitalisierung erfahren und keine menschliche Unterstützung mehr erhalten müssen. Zu diesem Zweck muss der digitale Game Master und die dahinter liegende Decision Engine Zugriff auf die „weichen" Regeln des Universums haben.

Bei der Digitalisierung des *Spieldesigns* gibt es wiederum mehrere Möglichkeiten. Neben einer grundsätzlichen Festlegung der Spielmechanik geht es beim Spieldesign auch darum, den Spielenden „ein Gefühl" der Welt samt den dazugehörigen Regeln zu vermitteln. Ein Spiel kann mitten in einem Handlungsstrang beginnen, wenn den Spielenden die nötigen Informationen zur Kontextuierung in anderer Form schon vermittelt wurden, beispielsweise durch erläuterndes Begleitmaterial in schriftlicher Form, einführende Videos, oder auch dadurch, dass das Spiel an eine bereits bekannte Welt[48] angelehnt wird [Sheldon, 2004, S.160f., S.183ff.]. Eine besonderes kontextsensitive Form sieht die Verbindung von Spielmechanik und handlungsrelevantem Design vor, bei der die Spielerinnen und Spieler nicht aus einer Darstellungsform in die andere wechseln müssen. Statt eines Videos zur Hintergrundgeschichte kann der Spieler die Geschichte beispielsweise im Dialog mit computergesteuerten Charakteren erfahren und sich selbst den Inhalt „erspielen". In diesem Fall muss jedoch sichergestellt werden, dass der Spieler diese Möglichkeiten erkennt und wahrnimmt [Sheldon, 2004, S.188f.]. Da das im Rahmen dieser Arbeit entstehende System nur einen Prototypen darstellen soll, ist vor allem die Wahl der Spielmechanik und Darstellungsform von Relevanz. Es ist an dieser Stelle jedoch sinnvoll zu erwähnen, dass durch das Spieldesign in vielerlei Hinsicht eine Einflussnahme möglich ist.

Als Darstellungsform von Spielmechanik und Spieldesign eignen sich sowohl rein textbasierte Lösungen, die zu Anfängen der Digitalisierung gang und gäbe waren[49] und mit Hilfe des Internets und

[48] Beispielsweise: „Das Spiel spielt in der Welt von Star Wars".
[49] Textbasierte Rollenspiele und Abenteuerspiele reichen etwa in die Gründungszeit des Internet zurück. Der erste bekannte Ableger war „Adventure" aus dem Jahre 1976 [Jerz, 2007]

insbesondere in Form von Foren-Rollenspielen immer noch aktiv genutzt werden, als auch visuell-interaktive Formen in zwei- oder dreidimensionaler Darstellung.

Für dieses Projekt wird eine vollständig dreidimensionale Digitalisierung gewählt, die sich vor allem auch aus dem Anspruch ergibt, *so viele Elemente* eines klassischen Pen-&-Paper-Rollenspiels *wie möglich* in die digitale Welt zu übertragen. Dazu gehört auch eine Visualisierung der erzählten und gespielten Welt. Rein textbasierte Lösungsansätze bieten wenig Mehrwert zur mündlichen Erzählform[50]. Insbesondere die Form des Foren-Online-Rollenspiels stellt sogar einen Rückschritt dar, da Spieler nicht mehr dazu angehalten werden, direkt und sozial miteinander zu interagieren. Entscheidungen können zeitversetzt getroffen werden und verhindern Interaktivität, was der für dieses Projekt gewählten Definition[51] eines Rollenspiels widerspricht [Pappe, 2011, S.29].

Visualisierte Computer-Rollenspiele in heutiger, bereits existenter Form haben viel mit Pen-&-Paper-Rollenspielen gemein, weisen allerdings auch einige Unterschiede auf, die Gegenstand aktueller Untersuchungen und Forschung sind [Tychsen u. a., 2005, S.5] [Tresca, 2011, S.134ff.]. Es gibt allerdings auch ein Problem, welches mit der vollständigen Visualisierung einhergeht: Dem Vorteil einer unmittelbaren, tatsächlichen Visualisierung der Spieler-Entscheidungen und Situationsveränderungen steht gegenüber, dass das visuell darstellbare auf eine Ansicht festgelegt wird, welche nicht der individuellen Interpretation einer korrespondierenden Situation im Pen-&-Paper-Bereich entsprechen muss. Auf diese Weise kann beispielsweise der persönliche Bezug des Spielers zum Spiel und des gespielten Charakters eingeschränkt sein [Tychsen u. a., 2005, S.4f.]. Bei der späteren Ausgestaltung der Spielwelt und den dazugehörigen Interaktionsformen muss dieser Punkt bedacht und, soweit möglich, kompensiert werden.

3.1.2. Auswahl der Game-Engine

Im Kapitel „Grundlagen“ im Abschnitt 2.2 wurde das Konzept der Game-Engine näher erläutert und zwei in Frage kommende, moderne Systeme miteinander verglichen. Es wurde gezeigt, dass die genannten Systeme, das „Cry Engine SDK“ und das „Unreal Development Kit“, in ihrem Funktionsumfang betreffend Grafikqualität und zur Verfügung gestellter Werkzeugefür die reine Spielerstellung nur marginale Unterschiede aufweisen. Auf rein technologischer Abilitätsebene lassen sich daher keine objektiven Ausschlusskriterien des einen oder anderen SDK identifizieren.

[50] Genau genommen verliert ein geschriebener Text im Gegensatz zu einer mündlichen Erzählung sogar an Information, wie beispielsweise Gestik und Mimik.
[51] Siehe Abschnitt 2

Die Auswahl der Game-Engine erfolgte demnach im Hinblick auf Praktikabilität sowohl im Arbeitseinsatz im *Team* als auch in Hinsicht auf die zur Verfügung stehenden Technologien, welche die geplante Umsetzung unterstützen können.

Wie bereits erwähnt wurde, stellt das *Unreal Development Kit* im Gegensatz zum *Cry Engine SDK* die Möglichkeit zur Verfügung, rechenintensive Arbeitsschritte über das Netzwerk verteilt auszulagern. Gerade in späteren Arbeitsstufen, in denen die visuellen Szenen durch ihre Größe immer komplexer werden, ist dies ein nicht zu unterschätzender Vorteil im Arbeitsprozess. Da die Berechnungszeiten auf diese Weise klein gehalten werden, kann mit Hilfe des UDK der Ansatz eines *Rapid Prototyping* verfolgt werden, den Chua u. a. wie folgt definiert:

> *„An approximation of a product (or system) or its components in some form for a definite purpose in its implementation“* [Chua u. a., 2010, S.2]

Insbesondere die Whiteboxing-Phase einer Szene wird dadurch beschleunigt. Innerhalb weniger Stunden lassen sich bereits simple Geometrien erzeugen, die einen Eindruck des finalen Produkts vermitteln [Hauteville, 2011].

Ein weiterer ausschlaggebender Punkt für die Entscheidung, bei der Entwicklung das UDK einzusetzen, ist die Möglichkeit, Level oder Szenen direkt kaskadisch ineinander zu streamen [Epic Games, Inc., 2013c]. Dies hat zwei direkte Vorteile, die näher beschrieben werden sollen.

Der erste Vorteil liegt darin begründet, dass ein Level-Streaming dem Projektziel zugute kommt, den nichtlinearen Aufbau eines Rollenspiels so exakt wie möglich in die digitale Welt zu übertragen. Abhängig von einer Entscheidungssituation können Inhalte im Spiel in Echtzeit gegen andere ausgetauscht und der Spielhergang vollständig modularisiert werden. Auf dieses Weise wird ein modulares (nichtlineares) Storytelling ermöglicht. Module sind zwar vorgefertigt; abhängig von den Entscheidungen der Spieler können diese jedoch in gänzlich anderer Art und Weise durchlaufen, oder auch ganz ausgelassen werden [Sheldon, 2004, S.311] (vgl. Abbildung 3.1). Derartige Module können ferner ineinander verschachtelt werden und erlauben so ein individuelles Detaillevel, welches sich der jeweiligen Situation adaptiv anzupassen vermag. Das Ziel ist hierbei nicht mehr, eine *Auswahl* von Möglichkeiten zu bieten, sondern die Spieler durch ihre Handlungen selbst bestimmen zu lassen, welche Möglichkeiten sie sich selbst erschaffen oder wahrnehmen möchten.

Der zweite Vorteil ist wiederum technisch begründet. Die Fähigkeit des Level-Streaming ermöglicht es, ein und dieselbe Spielwelt parallel auf mehreren Computer gleichzeitig zu bearbeiten. Bei Bedarf werden die einzelnen Teile zusammengefügt. Im Sinne einer Modularität ist auf diese Weise eine parallele, dezentrale Entwicklung möglich.

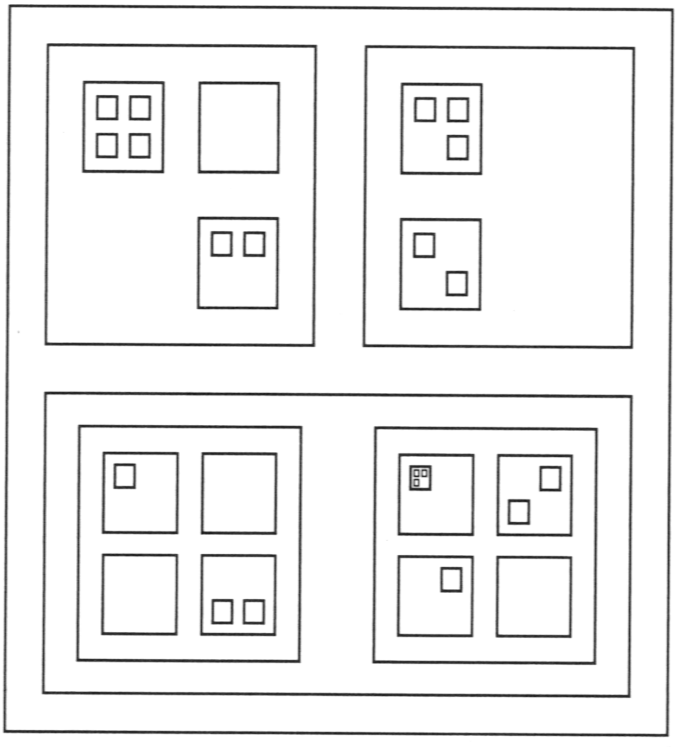


Abbildung 3.1.: Die Abbildung zeigt das Prinzip verschachtelte Module, in diesem Zusammenhang von Story-Modulen. Übertragen auf das Konzept von Level-Streaming können sich entsprechende Szenen-Abschnitte ebenfalls als Module vorgestellt werden, wodurch eine 1:1 Übertragung von Technik auf Story und vice versa möglich ist.

Aus [Sheldon, 2004, S.319].

3.1.3. Digitalisierung des Game Masters

Der Digitalisierung des Game Masters geht zunächst die Frage voraus, ob es sinnvoll ist, den Game Master *vollständig* oder lediglich *teilweise* zu digitalisieren. Der Unterschied zwischen einer Teildigitalisierung und einer vollständigen Digitalisierung soll an dieser Stelle kurz erläutert werden. Insbesondere Tychsen u. a. stellen die verschiedenen Möglichkeiten zur Digitalisierung eines Game Masters anschaulich dar und haben neben aktuellerer Literatur[52] nicht an Gültigkeit verloren.

Bei einer Teildigitalisierung wäre beispielsweise eine Spielmechanik denkbar, die es erlaubt, einem bestimmten Spieler Informationen an die Hand zu geben, woraufhin dieser auf Basis dieser Informa-

[52] Wie beispielsweise [Tresca, 2011, S.134ff.] .

tionen eine Entscheidung trifft und die Spielmechanik anweist, diese Entscheidung wirksam für alle Beteiligten umzusetzen. Es muss beachtet werden, dass bei jeder Form der Digitalisierung Abstriche gemacht werden müssen, die auf technische Limitierungen zurückzuführen sind. Ein menschlicher Game Master ist beispielsweise in der Lage, Informationen mit nonverbaler Kommunikation und Emotionen zu unterfüttern. Ein teildigitalisierter Game Master kann dies nicht leisten [Tychsen u. a., 2005, S.5]. Eine weitere Problematik beim Einsatz eines teildigitalisierten Game Masters ergibt sich dadurch, dass eine Digitalisierung (in diesem Fall die Digitalisierung eines Pen-&-Paper Rollenspiels) auch immer eine Quantifizierung darstellt. Eine vollständig explorierbare Welt ist *nicht umsetzbar*. Es könnte beispielsweise ein Spieler auf die Idee kommen, mit einem Charakter reden zu wollen, der für ein Gespräch nicht vorgesehen ist oder sich zu einem Ort am Horizont begeben wollen, der im digitalisiertem Spiel nicht erreichbar ist. Ein menschlicher Game Master im traditionellen Pen-&-Paper-Rollenspiel kann auf solche Spieler-Entscheidungen reagieren, ein teildigitalisierter Game Master jedoch nicht.

Es gibt einige Umsetzungen, in der eine Teildigitalisierung des Game Masters realisiert wurde. Das ebenfalls auf dem Dungeons & Dragons-Regelwerk basierende Spiel „Neverwinter Nights“[53] stellt hierzu beispielsweise das „AURORA Toolset“ zur Verfügung, mit dessen Hilfe eigene Welten, Dialoge und Geschichten durch einen menschlichen Game Master entwickelt werden können [Brown, 2007; MobyGames, 2013]. Die Ergebnisse sind anschließend statisch und können zur Laufzeit nicht verändert werden. Das heißt, Anpassungen an veränderte Situationen benötigen derart viel Zeit, dass es nicht praktisch ist, diese Anpassungen zur Laufzeit vorzunehmen. Der Spielleiter trifft Entscheidungen demnach weitestgehend im Vorhinein, setzt die Entscheidungen um und verhält sich danach *passiv* und nicht *reaktiv*. Eine Umsetzung in der Form, dass ein menschlicher Spieler unterstützend als Spielleiter in eine Digitalisierung eingreift, bringt demnach einige Restriktionen mit sich, die im Folgenden noch einmal zusammengefasst werden.

1. Wenn ein echter Spieler in einem echten Pen-Paper-Rollenspiel die Rolle des Game Masters übernimmt, findet eine Rollenverteilung auf Meta-Ebene statt. Dies bedeutet, dass der Game Master keinen Mitspieler mehr darstellt, sondern vollständige Kontrolle über die Spielwelt erhält. Dies ist ein Konzept, welches bei einer Teildigitalisierung nicht mehr funktioniert. Da die Spielmechanik in digitaler Form implementiert wird, erfolgt dadurch auch eine Limitierung. Diese findet in der Weise statt, dass die Welt und aktuelle Situation nicht mehr beliebig durch Beschreibungen modelliert und beeinflusst werden kann. Der teildigitalisierte Game Master wäre demnach nur ein verlängerter Arm der durch das Spiel vorgegebenen Grenzen, ohne jedoch über die bei einem traditionellen Pen-&-Paper-Spiel vorhandene Kreativitätsfreiheit zu verfügen. Der Game Master muss sich der grundlegenden Präsentation des Spiels ebenso unterordnen wie die restli-

[53] Veröffentlicht 2000 von Atari und Bioware [Tychsen u. a., 2005, S.1]

chen Spieler und kann nur in dem Maße eingreifen, welcher ohnehin von Vorneherein festgelegt wurde. [Tychsen u. a., 2005, S.5].

2. Normalerweise soll der Game Master eine Schnittstelle zwischen der Spielwelt und den Spielern darstellen und eine Schleife bilden: Aktionen der Spieler definieren die weitere Beschreibung der aktuellen und kommenden Situation [Tychsen u. a., 2005, S.3]. Im Fall der Teildigitalisierung des Game Masters müssten demnach prinzipiell zwei Schnittstellen geschaffen werden: Eine menschliche *und* eine digitale. Die digitale Schnittstelle müsste eine Logik kapseln, welche Entscheidungen darüber trifft, welche Informationen und Zuständigkeitsbereiche der menschliche Game Master erhalten soll. Der Game Master als menschliche Schnittstelle wiederum muss Entscheidungen darüber treffen, welche dieser Informationen er verwertet und ob er ggf. Informationen ins Spiel zurückfließen lässt. Hier ergibt sich ein Zuständigkeitsproblem. Pen-&-Paper-Rollenspiele leben von Interaktivität und der Spielfluss sollte nur unterbrochen werden, wenn es unbedingt nötig ist, beispielsweise wenn etwas ausgewürfelt werden muss [Bowman, 2010, S.27].

3. Durch die unter Punkt 1 genannte Restriktion, dass der menschliche Game Master nur innerhalb der bereits vorgegebenen Grenzen agieren kann, verlagert einen großen Teil der kreativen Arbeit zur Ausarbeitung einer Geschichte in die Vorarbeit zu einem Spiel. Der Game Master muss demnach einen Großteil der Geschichte im Vorfeld modellieren und ausarbeiten. Spontanität ist somit nicht mehr möglich, da der Game Master selbst weiß, was passieren wird [Tychsen u. a., 2005, S.5].

Aktuelle Forschungen versuchen einen Weg hin zum vollständig computergesteuerten Game Master zu finden [Tychsen u. a., 2005, S.8] [Tresca, 2011, S.134ff.]. Ein solcher digitaler Game Master ist, wenn von Forschungen im Bereich der echten künstlichen Intelligenz abgesehen wird, zwar determiniert, der Determinismus wird aber nicht mehr dadurch offenbar, dass er einen menschlichen Mitspieler in seiner Entscheidungsfreiheit einschränkt. Ein *vollständig* digitaler Game Master „weiss“ jederzeit alles über die aktuelle Situation[54] und gibt den Spielern definierte Schnittellen zur Eingabe. Dies begründet die Entscheidung, in diesem Projekt die Grundlage für einen *vollständig digitalen* Game Master zu realisieren: Mit Hilfe eines solchen Game Masters ist es durchaus möglich, die Spielwelt in Echtzeit zu verändern oder ganze Story-Elemente an die Aktionen der Spieler anzupassen, was mit einem nur teilweise digitalisiertem Game Master schwieriger zu realisieren wäre. Ein kontinuierliches Monitoring aller Spieler und das manuelle Update aller damit verbundenen Elemente und Verzweigungen wäre ein ansonsten zeitaufwendiges, manuelles Unterfangen. Durch vollständigen Determinismus werden Restriktionen auf technischer Seite nur noch durch die verwendete Game-Engine und die Fähigkeiten der-

[54] sämtliche Variablen und alle möglichen Story-Module sind gespeichert.

selben gesetzt [Tychsen u. a., 2005, S.8]. Das Konzept eines Game Masters in Computer-Rollenspielen ist in Abbildung 3.2 verdeutlicht.

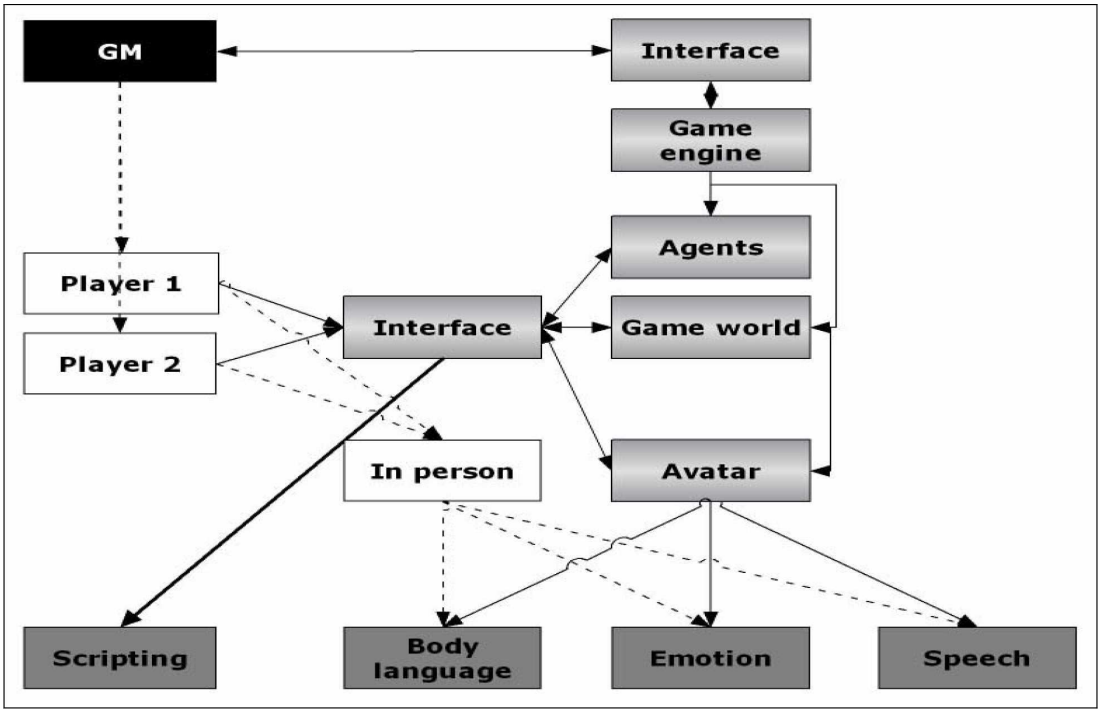


Abbildung 3.2.: Die Abbildung zeigt einen Flowchart eines Multiplayer-basierten Computer-Rollenspiels, welches einen Game Master (GM) enthält. Der Game Master hat die Möglichkeit, die Welt und alle darin befindlichen Elemente (in hellem Grau dargestellt). Die Spieler (Weiß dargestellt) können entweder direkt miteinander kommunizieren oder eine Kommunikation über die Avatare des Spiels durchführen lassen. Zur Eingabe der Spieler wird in jedem Fall ein Interface benötigt, welches die Aktionen der Teilnehmer an die zur Manipulation vorgesehenen Objekte weiterleitet.

Aus [Tychsen u. a., 2005, S.6].

3.1.4. Digitalisierung von Spielern und Hilfsmitteln zur Visualisierung

Dieser Abschnitt beschreibt die Möglichkeiten, wie Spieler miteinander und mit dem Game Master respektive dem digitalisierten Pen-&-Paper-Rollenspiel interagieren können. Im vorherigen Abschnitt wurde erläutert, dass in diesem Projekt die technische Grundlage dafür geschaffen werden soll, einen Game Master vollständig zu digitalisieren. Aus diesem Grund muss nun eine Schnittstelle konzipiert

werden, welche es den Spielern erlaubt, Eingaben an den virtuellen Game Master zu schicken und auch Ausgaben zu erhalten, beispielsweise Informationen zum eigenen Charakter, Inhalte des Inventars, Änderungen desselben etc.

Distanzbasierte Eingabemethoden würden in technischer Hinsicht vieles erleichtern, da prinzipiell pro Spieler ein gesamter PC mit samt aller verfügbaren Peripherie zur Ein- und Ausgabe bereit stehen würde. Im Abschnitt 2.1.4 jedoch wurde bereits beschrieben, dass es für das Projektziel von essentieller Wichtigkeit ist, die direkte, reale Kommunikation zwischen den Spielenden zu erhalten. Mit diesem Hintergedanken scheiden distanzbasierte Kommunikations- und Eingabemethoden[55] als Option für dieses Projekt aus.

In der Einleitung auf Seite 8 wurde bereits gezeigt, dass es für Spieler verschiedene Eingabemöglichkeiten gibt, wie beispielsweise Bewegungssteuerung über die „Microsoft Kinect“ oder auch die Eingabemethode eines Konsolen-Gamepads. Bei Pen-&-Paper-Rollenspielen können Spieler jedoch nicht nur Informationen verbreiten, sondern auch einfordern. Dazu gehört beispielsweise der schon erwähnte Charakterbogen, ein Inventar oder auch die Möglichkeit, Situationen für sich selbst auszuwürfeln. Üblicherweise hat ein Spieler in einem klassischen Pen-&-Paper-Rollenspiel darüber hinaus Zugriff auf weiterführende Informationen, die seinen Charakter betreffen. Das können beispielsweise Klassenbasierte Informationen sein: Wird ein Zauberer gespielt, so ist es sinnvoll, die verfügbaren Zaubersprüche nachschlagen zu können.

Auf der im August 2013 in Köln abgehaltenen Videospielmesse *GamesCom* wurden neue Bedienkonzepte für Konsolen vorgestellt, welche die hier vorgestellte Problematik ebenfalls adressieren. So erklärte der Chef des Entwicklungsstudios *Quantic Dream*[56], David Cage, dass oft die Beobachtung gemacht wurde, dass Personen, welche sich im gleichen Raum wie der Spieler aufhalten, oft am Spiel partizipieren und verbale Vorschläge zur weiteren Vorgehensweise einbringen [Stoecker, 2013]. Diese Gedanke wurde verfolgt bis hin zu einem Punkt, an dem weitere, im Raum befindliche Personen, tatsächlich ins Spielgeschehen eingreifen können und zwar mit Hilfe ihrer Tablet-Computer oder Smartphones.

Das Entwicklungsstudio *Ubisoft* geht noch einen Schritt weiter und ermöglicht für das Spiel *Watch Dogs* über die Utilisierung von Tablets oder Smartphones einen neuen Spielmodus, der sich nahtlos in das Hauptspiel einfügt (dieser Spielmodus wird in der Bildbeschreibung von Abbildung 3.3 näher erläutert). Die Entwickler greifen damit das von der *Wii U* erstmals vorgestellte Konzept auf und experimentieren mit Möglichkeiten der virtuellen Bildausweitung und Partizipationsmöglichkeiten mit touchfähigen Bildschirmen als Eingabegerät [Stoecker, 2013].

[55] Beispielsweise Videotelefonie.
[56] Bekannt für das Konsolenspiel *Heavy Rain*.

Abbildung 3.3.: Die Abbildung zeigt eine Spielszene des Titels *Watch Dogs*. Das große Bild zeigt die Ansicht eines Konsolenspielers, welcher den Avatar durch die Spielwelt steuert. Gleichzeitig kann ein weiterer Mitspieler (in diesem Fall in der Rolle eines Antagonisten) innerhalb einer stilisierten Karte der Spielwelt Manipulationen ausführen, die den ersten Spieler von seinem Ziel abbringen sollen.

Aus [Stoecker, 2013].

Smartphones eignen sich im Rahmen dieses Projektes als ideales Hilfsmittel. Die Gründe hierfür werden nachfolgend beschrieben.

Mit Hilfe von Smartphones ist nicht nur möglich Eingaben zu tätigen, sondern ebenso die Spieler mit Feedback zu versorgen. Normalerweise trennen Eingabegeräte den Nutzer eher von einer virtuellen Welt. Durch den Bruch wird das Aufrechterhalten einer Präsenz, wie sie in Rollenspielen besonders wichtig ist, zum kritischen Faktor.

Jedes Eingabegerät hat grundsätzlich verschiedene Vor- und Nachteile; die Reduktion der Nachteile resultiert meistens in einem höheren Preis. Smartphones bieten dabei eine Möglichkeit, gleichzeitig die Akzeptanz der Nutzer zu erhöhen und Brüche in der Präsenz zu vermeiden. Grundsätzlich sind

Smartphones als reine *Eingabegeräte* für den Nutzer eine eher teure Anschaffung. Dem gegenüber steht die Funktionsvielfalt (auch im Alltag), weshalb in Deutschland jeder Dritte (jeder Zweite der unter 30-Jährigen) ein Smartphone besitzt [BITKOM Pressemitteilung, 2012]. Moderne Smartphones beherrschen nicht nur die Darstellung von visuellen Informationen sondern auch Vibrationen und die Wiedergabe von Tönen. Durch diese Möglichkeiten kann ein breites Medienfeld zwischen dem darstellenden Spiel und dem Nutzer aufgespannt werden, was wiederum eine tiefere Immersion ermöglicht. Die Implementierung von Applikationen, die auf sämtliche Fähigkeiten der Hardware zugreifen können, erfordert dabei eine vergleichsweise niedrige Einarbeitungsschwelle [Hao Chuah u. Lok, 2012a, S.5].

Neben verschiedenen Möglichkeiten der Medienwiedergabe verfügen Smartphones über diverse Sensoren, mit dessen Hilfe die Spieler, neben der Toucheingabe, zusätzliche Informationen übermitteln können. Dazu gehört insbesondere die Lage im Raum oder auch die Beschleunigung.

Im Gegensatz zu normalen Controllern erlauben Smartphones direkte Manipulation und setzen kein Anlernen des Bedienkonzeptes voraus, da die Rückmeldung unmittelbar visuell erfolgt und sogar an individuelle Bedürfnisse angepasst werden kann [Hao Chuah u. Lok, 2012a, S.1]. Durch die Doppelfunktion eines Smartphones als Eingabe- sowie Ausgabegerät wird die Virtualität zum Nutzer hin verschoben und erlaubt damit die Realisierung einer Mixed Reality (MR)-Anwendung.

Eine Besonderheit bei der Projektkonzeption, die an dieser Stelle erwähnt werden soll, ist die Möglichkeit, mit Hilfe von Smartphones sogar erweiterte Bedienkonzepte einzubetten, bei denen die Spielgeschichte als MR-Schnittstelle zwischen Digitalisierung und Realität genutzt werden kann. Da die Spieler räumlich nicht voneinander getrennt sind respektive Smartphones als digitale Schnittstelle zum Host-System benutzen, können Aktionen, die vorher möglicherweise „geschauspielert“ wurden, nun tatsächlich physikalisch simuliert werden. Wenn beispielsweise der Fortgang der Geschichte an einer Stelle voraussetzt, dass die Spieler gemeinsam eine schwere Falltür öffnen, kann dies simuliert werden, indem die Spieler Ihre Smartphones symbolisch zusammenhalten und gemeinsam eine Aufwärtsbewegung durchführen.

Ein anderes Beispiel wäre, dass die Spieler unbemerkt durch einen bestimmten Abschnitt schleichen müssen und nicht gehört werden dürfen. Hier könnten die Mikrofone der Smartphones aktiviert werden, um bei Überschreitung einer bestimmten Lautstärke eine Aktion im Spiel auszulösen. Also müssten die Spielerinnen und Spieler tatsächlich, sowohl in der virtuellen als auch in der realen Welt, in diesem Fall „schleichen“. Im Hinblick auf das Projektziel eröffnet dies weitere Möglichkeiten der sozialen Interaktion.

3.1.5. Wahl des passenden Netzwerkprotokolls

Um einen möglichst hohen Immersionsgrad zu erreichen, muss das im vorherigen Abschnitt als Eingabemedium vorgestellte Smartphone drahtlos an die Game-Engine gekoppelt werden. Eine kabelgebundene Anbindung an einen Computer würde die zur Verfügung stehenden Freiheitsgrade einschränken. Eine drahtlose Verbindung wirft jedoch einige Probleme auf. Deren Hintergründe sollen an dieser Stelle erläutert werden.

Das *Engagement* des Spielers beschreibt das subjektive Gefühl der „Versunkenheit“ in die aktuelle Aufgabe respektive in das Spiel. Das Engagement ist einerseits abhängig vom persönlichen Spaß, andererseits, wenn auch nicht direkt proportional, zum Grad der Präsenz respektive dem Immersionsgrad [Heisenberg, 2012a, S.12f.]. Auf multisensorischer Basis bekommen die Nutzer des Systems einen Reiz auf visueller Ebene und teilweise auf taktiler Ebene, namentlich durch die Eingabemethode des Touchscreens. Haptisches Feedback wird dabei teilweise durch Vibrationen des Smartphones gewährleistet. Die Anzahl zur Verfügung stehender Stimuli für ein Gefühl der Immersion ist daher begrenzt, weshalb das Ziel sein muss, das Engagement so hoch wie möglich zu halten. Dies bedeutet auch, Latenzen so gering wie möglich zu halten und ein möglichst responsives Eingabesystem zu implementieren. In der Regel werden dabei Eingabe-Latenzen von 50ms und weniger angestrebt, was im Umkehrschluss heisst, dass das Netzwerkprotokoll mindestens 20 Updates pro Sekunde verschicken muss - für jeweils einen Spieler [Michael Abrash, 2012]. Ferner mischen sich Faktoren wie Genauigkeit, Drift und Asynchronität in die Problematik mit ein. Aus diesem Grund kommt der Netzwerkanbindung eine besondere Bedeutung zu, um Geschwindigkeits- oder Datenverlust bei der Übertragung unbedingt so weit wie möglich zu vermeiden.

Aus diesem Grund muss durch die Wahl des Netzwerkprotokolls eine Balance zwischen zur Verfügung stehenden Bandbreite, der zu sendenden Größe an Daten und vor allem der Architektur gefunden werden. Wie in Abschnitt 2.3.2 erwähnt, arbeitet der Code einer Netzwerkapplikation zu 90% in der Applikationsebene und muss lediglich mit der Transportebene kommunizieren, sodass die zur Verfügung stehenden Protokolle lediglich auf diese Ebene beschränkt werden und damit namentlich aus TCP und UDP bestehen.

In dem geplanten Projekt existieren verschiedenartige Daten, welche über das Netzwerk an einen der Teilnehmer gesendet werden müssen. Zum einen können dies beispielsweise Spielentscheidungen sein und zum anderen weniger wichtige Bewegungsdaten eines der Smartphone-Geräte. Erstere sind aus dem Grund vernachlässigbar, da Bewegungsänderungen kontinuierlich in kurzen Intervallen übertragen werden. Verlorengegangene Pakete werden umgehend durch aktuellere ersetzt. Insbesondere für erstere ist ein garantierter Empfang für den Erfolg der Anwendung unabdingbar. Sollten wichtige Spielentscheidungen der Spieler den Spielleiter nicht oder in falscher Reihenfolge erreichen, so kann

dieser die Story und Spielwelt nicht verändern und Ereignisse an die Spieler zurückführen. TCP ist prädestiniert für eine garantierte Lieferung sowie richtige Reihenfolge von Daten, welche über ein Netzwerk gesendet werden, wohingegen UDP keine dieser beiden Anforderungen erfüllt.

Zwar kann UDP auf Applikationsschicht um solche Funktionen erweitert werden, allerdings würde es sich dabei um eine softwareseitige Erweiterung handeln. TCP stellt diese Funktionalität bereits auf Hardwareebene und damit performanter zur Verfügung [Watte u. a., 2005]. Speziell angefertigte UDP-Protokolle implementieren im Wesentlichen die wichtigsten Funktionen von TCP, versuchen dabei jedoch die Einfachheit (Schnelligkeit) von UDP beizubehalten. Patrick Wyatt, welcher an der Entwicklung des Spiels *Guild Wars* beteiligt war, rät davon ab, für kleinere Anwendungen ein individuelles Netzwerkprotokoll zu implementieren. Die Entwickler von *Guild Wars* haben ein solches individuelles Netzwerkprotokoll implementiert, was darin resultierte, dass die Kernbibliotheken allein rund 50.000 Codezeilen umfassen. Zur vollkommenen Stabilität und Fehlerfreiheit wurden darüber hinaus einige Jahre Entwicklungszeit benötigt [Patrick Wyatt, 2012]. Es wird viel Zeit, eine Menge Wissen im Bereich von Netzwerken, sowie typischerweise ein großes Entwicklungsteam benötigt, um die Entwicklung eines eigenen Netzwerkprotokolls rechtfertigen zu können. Die mitwirkenden Mitarbeiter sollten bereits Netzwerkspiele implementiert haben und die Probleme sowie Stärken und Schwächen der einzelnen Protokolle bis auf Low-Level-Ebene nachvollziehen können.

TCP stellt die im Rahmen dieses Projektes benötigten Funktionalitäten zur Verfügung. Indem der Nagle-Alorithmus ausgeschaltet wird, können die damit für Spiele einhergehenden Probleme erfolgreich umgangen werden. Die Entwicklung eines automatischen Sendewiederholungssystems auf Basis von UDP ist damit nicht gerechtfertigt („*make games, not engines*“[57]) und würde zu viel Zeit in Anspruch nehmen, welche im Resultat kein besseres Ergebnis liefert.

Ferner ist ein gleichzeitiger Einsatz der beiden Protokolle nicht empfehlenswert, da die Qualität von Echtzeitanwendungen, welche UDP verwenden, immer von TCP und der Art, wie TCP das Netzwerk verwendet, beeinflusst werden [Hidenari Sawashima u. a., 1997]. Ein Ergebnis des Papers ist, dass der UDP-Paketverlust öfter auftritt, wenn gleichzeitig TCP-Verbindungen synchronisiert werden[58].

Auf den Einsatz von Bibliotheken wurde ebenfalls verzichtet, da zum einfachen Paketversand Java bereits eine ausgezeichnete API bietet und keine komprimierten, skalierbaren und verschlüsselten Nachrichten verschickt werden sollen [Patrick Wyatt, 2012]. Ein weiterer Grund zum Verzicht auf externe Bibliotheken ist die schlechte Code-Wartbarkeit. Änderungen erfordern das Einarbeiten in eine komplexe Struktur und ungewollte Nebeneffekte könnten zu spät identifiziert werden.

[57] bekanntes Zitat in der Spieleentwicklung von Josh Petrie, http://scientificninja.com/blog/write-games-not-engines.

[58] Dem interessierten Leser wird zur weiterführenden Lektüre [Hidenari Sawashima u. a., 1997] empfohlen.

Zu guter Letzt bietet die ausgewählte Game-Egine, das UDK, eine umfangreiche und konfigurierbare TCP-Schnittstelle nativ an, was die Entscheidung für TCP schließlich besiegelt [Epic Games, Inc., 2013d].

3.1.6. Wahl des Netzwerkspielmodells

Neben der Auswahl eines Netzwerkprotokolls musste sich für ein Netzwerkspielmodell (siehe Abschnitt 2.3.3) entschieden werden.

Ein Peer-2-Peer-Ansatz bietet sich schon aus dem Grunde nicht an, da alle Daten an jeweils alle verbundenen Teilnehmer gesendet werden. Die Smartphones dienen in diesem Prototypen der Interaktion mit der Spielwelt und nicht zur Anzeige selbiger, weshalb durchgeführte Aktionen eines Spielers alle anderen Spieler nicht interessiert respektive gar nichts mit diesen Daten anfangen können. Lediglich der Game-Master respektive die Spielwelt, welche durch die Game-Engine realisiert wird und diese darstellt, benötigen die Eingaben aller Spieler, um zum Beispiel Bewegungsänderungen der Spielfiguren oder ähnliches vornehmen zu können.

Die in Abschnitt 2.3.3 aufgelisteten Vorteile des Servers in einem Client-Server-Modell eignen sich hervorragend für das geplante Vorhaben:

- Eine zentrale Stelle für die Client- und Datenverarbeitung macht es einfacher Benutzer zu überwachen und diese zu administrieren.
  - Ein Verbindungsabbruch eines Spielers kann hier einfacher registriert und entsprechend behandelt werden. Eine solcher Vorfall ist in einem Peer-2-Peer-System schwieriger zu behandeln, da es keine zentrale Stelle gibt, an der sich ein Client anmelden und Daten beziehen kann.
- Hauptberechnungen auf einem leistungsstarken Server ausführen zu lassen, ermöglicht es weniger effiziente Clients, wie mobile Endgeräte, zu verwenden.
  - Da sich besonders in einem Rollenspiel eine Vielzahl an Berechnungen anhäufen und auf viele unterschiedliche Situationen reagiert werden muss, ist es sinnvoller, diese auf einen leistungsstarken Server auszulagern, als jedes Mobilgerät diese Berechnungen durchführen zu lassen.
  - Ferner ist es, zumindest in Rahmen dieses Projekts, nahezu unmöglich auf verschiedenartigen Geräten einen Determinismus zu implementieren.

- Der Server kann nicht ausschließlich von Spielclients verwendet werden, sondern zum Beispiel auch von einem Überwachungsclient, welcher aktuelle Statistiken und Geschehnisse anzeigt.
  - Der volldigitalisierte Game Master behält auf einem zentralen Server die Kontrolle über Spieler sowie Spielwelt und kann jederzeit gesendete Daten abfangen sowie für andere Zwecke, wie die der Entscheidungsfindung oder Storyanpassung, verwenden.
  - Neben den Smartphones kann auch die Game-Engine ausgelagert und als Client registriert werden, wodurch eine maximale Ausnutzung des Modularisierungskonzeptes erreicht wird. Damit ist nicht nur das Interaktionsgerät der Spieler auswechselbar, sondern auch die Game-Engine.

Von allen Punkten ist der letzte ein entscheidender. Dadurch, dass die Smartphones die Interaktionen der Spieler mit der Spielwelt abbilden und somit die Interaktionsdaten an die Game-Engine senden müssen, bilden Server und Game-Engine eine Einheit. Beim Wechsel der Engine müsste eine vollständige Neuimplementierung des Servers, sowie der gesamten Netzwerkkommunikation und Spiellogik stattfinden. Durch die Entkopplung dieser beiden Komponenten ist prinzipiell sogar das Netzwerkprotokoll austauschbar.

3.1.7. Wahl des mobilen Betriebssystems

Zur Verwendung von Smartphones für Interaktionen und Ausgaben muss im Rahmen dieses Projektes eine eigene Applikation entwickelt werden. Diese Implementierung kann auf verschiedene Weisen erfolgen. Zu nennen sind hier native und *hybride Applikationen* sowie reine *Webapps*.

Bei letzterer handelt es sich um eine normale Webseite, welche in der Form gestaltet wird, dass sie den besonderen Anforderungen von Smartphones (kleine Displays, fehlende Tastatur, …) gerecht wird. Der Nachteil von dieser Art Applikationen liegt in der fehlenden Möglichkeit Gerätefunktionen zu verwenden, was allein für die Kommunikation mit dem Server unabdingbar ist. Zwar wird durch die *Device APIs Working Group*[59] die Entwicklung von clientseitigen APIs vorangetrieben, diese befinden sich allerdings in einem noch nicht für den Einsatz ausreichenden Stadium.

Hybride Apps sind eine Erweiterung von Webapplikationen. Entwickelt wird auch hier mit bewährten Webtechnologien, allerdings erweitert um die Möglichkeit der Nutzung von APIs, welche gerätespezifische Funktionen kapseln und dem Entwickler für gewöhnlich in der Programmiersprache

[59] Es handelt sich hierbei um eine Gruppe des W3C, welche clientseitige APIs entwickelt, die auf Gerätefunktionen zugreifen können: http://www.w3.org/2009/dap/ .

Javascript zur Verfügung stellen. Die daraus resultierende Anwendung verliert allerdings ihre Lauffähigkeit im Browser und wird daher in ein gerätespezifisches Format konvertiert, welche insgesamt auch das „Look-&-Feel“ einer nativen Anwendung vermittelt. Der Umfang einer solchen API ist jedoch vergleichsweise gering und ermöglicht es auch nur fest definierte Gerätefunktionen aufzurufen, eine Individualisierung oder Modifizierung auf niedrigerem Level ist nicht möglich.

Eine letzte Alternative bilden native Anwendungen. Diese werden in einer für ein mobiles Betriebssystem spezifischen Programmiersprache entwickelt. Zur Zeit existieren eine Vielzahl solcher Betriebssysteme auf dem Mobilfunkmarkt. Zu den bekannteren gehören unter anderem: *Android*, *iOS*, *Blackberry* und *Windows Phone*. Hinzukommen jedoch eine Vielzahl weiterer, wie zum Beispiel: *Tizen*, *Firefox OS* oder *Ubuntu Touch OS*.

Da im Bereich der Netzwerkimplementierung grundlegende Modifikationen am Protokoll vorgenommen werden müssen, ist die Verwendung nativer Anwendungen unausweichlich. Weder Webapps noch hybride Apps bieten die Möglichkeit dahingehend Anpassungen vornehmen zu können. Darüber hinaus existieren in Pen-&-Paper-Rollenspielen eine Vielzahl möglicher Interaktionsformen und -möglichkeiten, welche in einer Digitalisierung ebenfalls möglich sein sollten. Eine Abbildung von Nutzerinteraktionen in technische Äquivalente erfordert eine ausreichend umfangreiche Programmiergrundlage, welche ausschließlich durch native Plattformen gewährleistet ist. Aufgrund der weiten Verbreitung Androids und der persönlichen Vorkenntnisse mit diesem Betriebssystem, ebenso wie der Programmiersprache Java, fiel die Wahl auf Googles Plattform.

3.1.8. Wahl der Programmiersprachen

Die Wahl der verwendeten Programmiersprachen leitet sich direkt aus den festgelegten Werkzeugen ab, die zur Umsetzung verwendet werden. Wie im vorheringen Abschnitt (3.1.7) erläutert wurde, wird Android als mobiles Betriebssystem festgelegt. Applikationen für dieses Betriebssystem werden in *Java* programmiert, womit die Benutzung dieser Sprache für dieses Projekt implizit ist.

Java bietet umfangreiche Konfigurations- und Steuerungsmöglichkeiten für verschiedene Netzwerkeigenschaften, was der weiteren Entwicklung entgegen kommt. Hinzu kommt die umfangreiche Erfahrung, die unter anderem im Rahmen des Studiums mit Java gesammelt werden konnte. Insbesondere letzter Punkt bestätigte die Wahl, die strukturellen Elemente des Prototypen, wie den Server oder auch die Digitalisierung des Spielleiters und die damit zusammenhängende Konzeption einer *Decision Engine*, in Java vorzunehmen.

Durch die Festlegung der Game-Egine auf das Unreal Devleopment Kit kommt ebenfalls die bereits vorgestellt Scripsprache *UnrealScript* zur Anwendung. Dem Projekt kommt dabei zugute, dass Unre-

alScript viele Einflüsse höherer Sprachen, wie auch Java, mitgenommen hat und deren Syntax weitgehend beibehält. Dabei wurden explizit Konzepte von Java als Designziel der Sprache übernommen [Epic Games, Inc., 2013e]:

- Automatische Speicher-Bereinigung („Garbage Collection“)
- Strenge Typisierung
- Clientseitige, sichere Ausführung des Codes in einer „Sandbox“
- Anlehnung der Syntax an C/C++ und Java

Die Einarbeitungszeit in UnrealScript konnte dementsprechend reduziert werden.

3.2. Konzept

Nach den theoretischen Grundlagen sowie der Betrachtung der Auswahlentscheidungen, folgt nun die Beschreibung des Konzeptes und der konkreten Implementierung. Im Verlauf des Abschnittes werden verschiedene Teile des Konzeptes und deren Realisierung einzeln behandelt. Darüber hinaus werden an einigen Stellen Probleme, die es während der Umsetzung zu bewältigen gab sowie deren Lösungen vorgestellt.

Zu Beginn dieses Abschnittes soll das Gesamtkonzept der Anwendung beschrieben werden. Zum besseren Verständnis wird daher die Visualisierung in Form einer schematischen Skizze (Abb. 3.4) verwendet.

Implementiert wurde eine multi-threaded Client-Server-Applikation. Es existieren zwei Arten von Clients, der Smartphone- (1) sowie der Game-Engine-Client (2), welche sich mit einem zentralisierten Server (3) verbinden. Die Verbindung findet durch unterschiedliche Übertragungsmedien und -methoden statt.

Ein Thread-basierter Ansatz stellte sich bei der Implementierung dieses Prototypen als die beste Alternative heraus, da vergleichbar agierende Ansätze (asynchrone Sockets) eine zu hohe Komplexitätsstufe mit sich bringen. Insgesamt handelt es sich hierbei bereits um eine vergleichsweise komplexe Anwendungsstruktur, sodass der zusätzliche Overhead umgangen und auf die bereits bekannte Threadprogrammierung zurückgegriffen wurde. Darüber hinaus wären zumindest für die Verwaltung mehrerer Verbindungen Threads notwendig gewesen und die Implementierung zweier verschiedener Paradigmen sollte in einem fachgerechten Programmierstil vermieden werden.

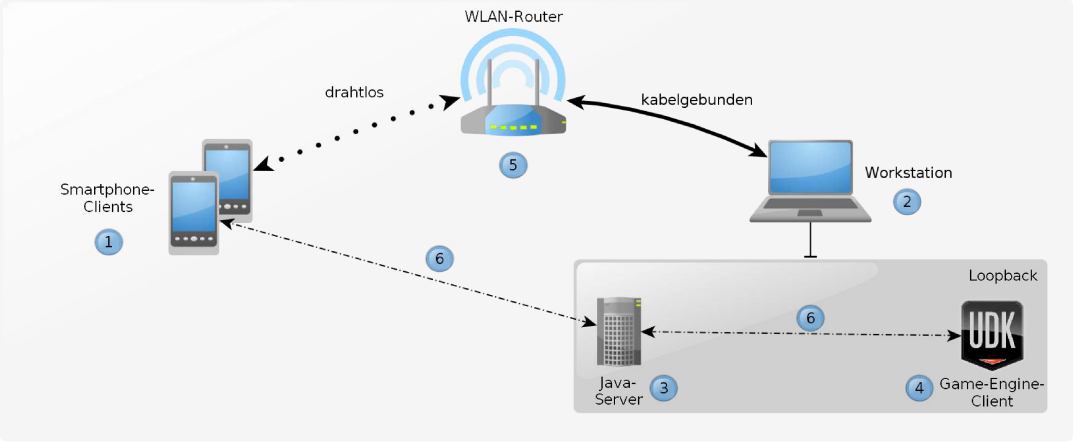


Abbildung 3.4.: Die Abbildung zeigt das Gesamtkonzept des umgesetzten Prototypen.

Der Smartphone-Client verfügt über zwei Threads, einerseits zum Lesen und andererseits zum Schreiben von Nachrichten. Dadurch ist es möglich, während dem Senden von Nachrichten wichtige Events zeitgemäß zu empfangen, obwohl Sockets standardmäßig blockierend implementiert sind. Das Interaktionsgerät als solches sendet nahezu ununterbrochen Bewegungsdaten, sodass ein Empfangen von speziellen Spielevents nicht mehr möglich gewesen wäre (mehr dazu in Abschnitt 3.2.1).

Der Game-Engine-Client ist für die Visualisierung des Spiels verantwortlich und hält die verschiedenen Module vor, durch welche sich die Spieler bewegen können. Die gesamten Auswirkungen der Anwendungslogik laufen letztlich hier zusammen und werden durch die Game-Engine beständig ausgewertet. Mit Hilfe des UDK wird demnach die Spielwelt dargestellt, welche ebenfalls die Visualisierung der Spieler (sogenannte Avatare) beinhaltet und die Digitalisierung der *erlebten Geschichte* gemäß des Projektziels verwirklicht.

Der Server verfügt ebenfalls über mehrere Threads, genauso wie über einen internen Speicher, indem alle verbundenen Clients, Verbindungsinformationen und Input- sowie Outputstreams hinterlegt werden (siehe Abschnitt 3.2.3).

Nachfolgend wird Abbildung 3.4 hierarchisch durchlaufen und erläutert.

3.2.1. Smartphone-Clients

Das Spiel- und Nutzungserlebnis der Spieler soll so natürlich wie möglich wahrgenommen werden, sodass eine Kabelverbindung ausgeschlossen wurde. Aus diesem Grund wird für die Datenübertragung von und zu den Smartphones ein kabelloses Netzwerk verwendet.

Die Android-Anwendung ist als Client realisiert worden und verbindet sich über ein eigens für das Projekt angelegtes Netzwerk zu einem WLAN-Router (5).

Bevor das Gerät zur Interaktion mit der Spielwelt verwendet werden kann, wird zunächst eine Verbindung zum Server benötigt. Die erste Aufgabe der Applikation stellt somit die Netzwerkinitialisierung dar, welche in Form einer Activity und entsprechenden Views sowie Widgets dem Benutzer angezeigt wird (siehe Abb. 3.5 und 3.6 (e)).

Abbildung 3.5.: Die Abbildung zeigt ein Screendesign der Start-Activity innerhalb der Android-App.

Beim Betätigen des Buttons durch den Nutzer wird ein Hintergrundservice erstellt (Abb. 3.6 (g)), welcher für die Laufzeit der Applikation die gesamte Netzwerkfunktionalitäten kapselt und den Activities in Form von Methoden zur Verfügung stellt. Innerhalb des Services werden weitere Threads erstellt, da der Service per se im Hauptthread läuft und dadurch die Bedienelemente bei langandauernden Operationen einfrieren würden. Der *ConnectThread* (Abb. 3.6 (k)) übernimmt dabei zunächst den Versuch eines Verbindungsaufbaus. Bei erfolgreichem Abschluss der Operation, wird zum einen die Information an den Service zurückgemeldet, welcher daraufhin die Activity wechseln kann und zum anderen werden zwei weitere Threads kreiert, welche für das Lesen und Schreiben im Netzwerk zuständig sind.

Die Applikation stellt, wie bereits in Abschnitt 3.1.4 erörtert, das Interaktionsgerät für die Spieler dar. Es existieren verschiedene Interaktionsformen und Ansichten, welche durch jeweils eine Activity dem Benutzer zur Verfügung gestellt werden (in Abb. 3.6 dargestellt als (h) und (f)).

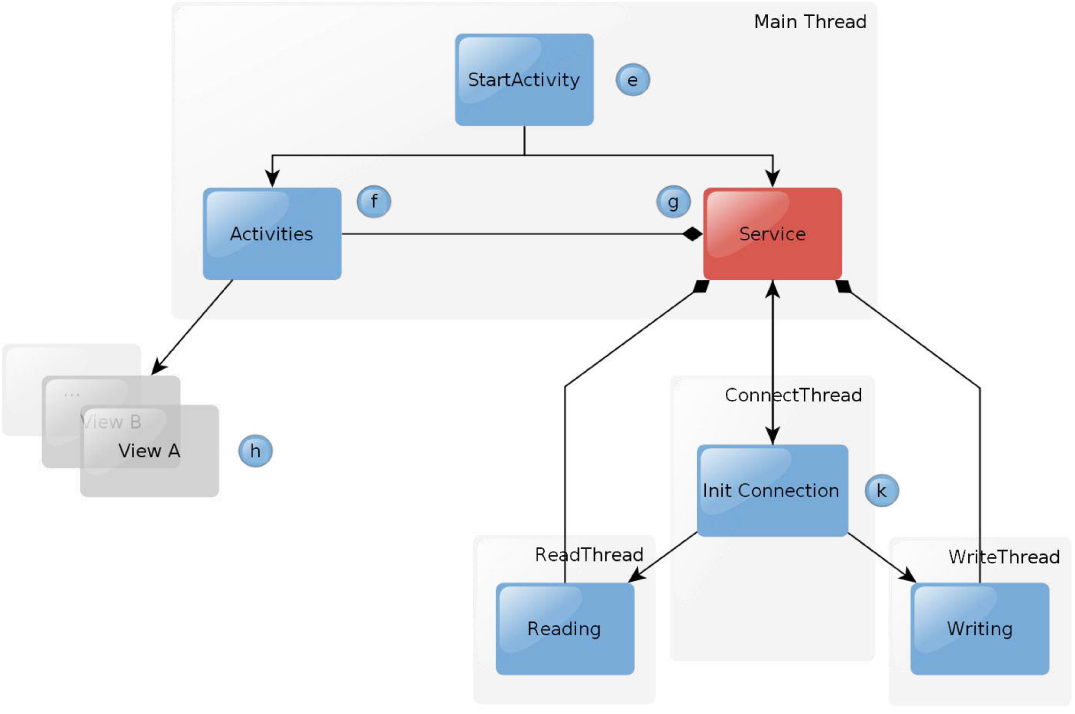


Abbildung 3.6.: Die Architektur folgt den Konventionen des Betriebssystemherstellers Google.

Die Hauptansicht ist eine Art *Touchpad*, durch welchen der Avatar im Spiel gesteuert werden kann. Die Steuerung geschieht nicht mittelbar, sondern indirekt über ein Fadenkreuz.

Durch die Bewegung eines Fingers auf der Touchpad-Adaption wird im Spiel das Fadenkreuz bewegt. Durch das Betätigen eines speziellen Interaktionsbuttons wird die Bewegung daraufhin initialisiert. Der Vorteil bei der Verwendung dieses Steuerungskonzeptes ist, dass das Fadenkreuz als verlängerter Arm in der Spielwelt dient. Zeigt dieser zum Beispiel auf ein Item, so kann der zusätzliche Button zum hin laufen und aufheben dieses Items verwendet werden. Es könnte jedoch auch ein weiteres Menü erscheinen und dem Spieler mehrere Auswahlmöglichkeiten anzeigen oder das aus dem Web bekannte Konzept des Hover-Effektes[60] könnte implementiert werden. Abbildung 3.7 zeigt, wie man im realisierten Prototypen durch überfahren mit der Maus zusätzliche Informationen anzeigen lassen kann.

[60] beim Bewegen der Maus über bestimmte Elemente, werden zusätzliche Informationen in Form eines Pop-up angezeigt.

Abbildung 3.7.: Die Abbildung zeigt, wie man im umgesetzten Prototypen einen Gegenstand (hier eine Statue) mit der Maus überfahren kann und dadurch weitere Informationen erhält. Oben links im Bild sieht man: „Actor selected: Static Mesh Actor“.

Die Augen der Spieler folgen dem Fadenkreuz auf dem Remotebildschirm, während sie ihren Daumen auf dem Smartphonedisplay bewegen. Der Delay zwischen dem Zeitpunkt wenn der Nutzer seinen Daumen bewegt und die Bewegung auf dem Display umgesetzt wird, ist hier entscheidend. Je kürzer der Delay ausfällt, umso natürlicher wird die Eingabe wahrgenommen und desto höher ist das Engagement. Um dieses Ziel zu erreichen, muss jede Bewegung übertragen werden, was durch das Ausschalten des Nagle-Algorithmus auch möglich ist. Dem vermehrten Netzwerkverkehr wird durch ein eigenes kabelloses Netzwerk entgegengewirkt (siehe Abschnitt 3.2.4).

Die Interaktionsdaten werden von den Clients an den Server gesendet, welcher diese an die Game-Engine weiterleitet. Eine direkte Kommunikation ist aufgrund des gewählten Netzwerkspielmodells nicht möglich, allerdings auch nicht gewünscht. Der Service bietet den Activities dafür diverse Methoden an, womit der Status der Netzwerkverbindung abgefragt oder Daten gesendet werden können.

In umgekehrter Kommunikationsrichtung kann das UDK über den Server den Interaktionsgeräten zum Beispiel Anweisungen zum Wechseln der Eingabemethode senden. Die Informationen hierzu werden im *ReadThread* erhalten, zum Service weitergereicht und entsprechend verarbeitet.

3.2.2. Workstation

Auf der Workstation befindet sich sowohl der Server (3), welcher in Java implementiert wurde, als auch der Game-Engine-Client (4). Um die Anzahl der Paketverluste und mögliche Problemstellungen, wie zum Beispiel einen zusätzlichen Delay zu vermeiden, ist die Workstation per Ethernetkabel mit dem WLAN-Router (5) verbunden worden. Dadurch, dass sich die Komponenten (3) und (4) auf derselben Workstation befinden, können die Daten zwischen diesen über das Loopback-Device latenzfrei übertragen werden.

Durch das Client-Server-Modell und die Entscheidung, die Game-Engine als eigenen Client zu realisieren, kann zwar jede Komponente beliebig ausgetauscht werden, allerdings erhöht sich dadurch auch die Menge an zu sendenden Daten und die Gefahr eines Paketverlustes, welcher die gesamte Kommunikation für mehrere Millisekunden blockieren könnte. Sind beispielsweise zwei Spieler zu dem Server verbunden, so senden diese ihre Bewegungsdaten an den Server, welcher die Daten an die Engine weiterleitet. Somit dupliziert sich die Anzahl der gesendeten Pakete. Mit der Entscheidung, die beiden Komponenten auf derselben Workstation laufen zu lassen, entfällt diese Problematik jedoch.

Ferner wurde dadurch die Möglichkeit geschaffen, den Game Master und die Decision Engine vollkommen unabhängig von der Game-Engine und der anderen Komponenten zu implementieren, um ein vollständig modulares System zu erhalten. Die Schnittstelle zwischen dem Game Master und der Spielwelt ist unabhängig von der Leistung sowie dem Umfang des Netzwerkprotokolls und beeinflusst somit die Immersionsfähigkeit des Spieles nicht.

3.2.3. Java-Server

Der Server (3) bildet in der Anwendung die Kommunikationsbasis. Jegliche Daten werden über den Server an Clients gesendet, da durch das Client-Server-Modell keine Direktadressierung zwischen Clients möglich ist. Darüber hinaus gehen alle Verbindungsanfragen hier ein (siehe Abb. 3.8 (a)) und werden entsprechend behandelt.

Da zwei verschiedene Clientarten existieren, wurde eine Art Lobbyfunktionalität umgesetzt. Verbundene Clients werden zunächst in diese verschoben, bis eine Kontrollnachricht den Server erreicht, welche sie identifiziert (siehe Abb. 3.8 (b)).

Sobald eine Klassifizierung möglich ist, wird ein neuer Thread erzeugt, welcher zum Lesen respektive zum Schreiben von Daten dieses Clients verwendet wird. Es wird eine Unterscheidung der neu generierten Threads vorgenommen, da diese unterschiedliche Funktionalitäten implementieren.

Der Ablauf eines PlayingThreads (Abb. 3.8 (d)) sieht wie folgt aus:

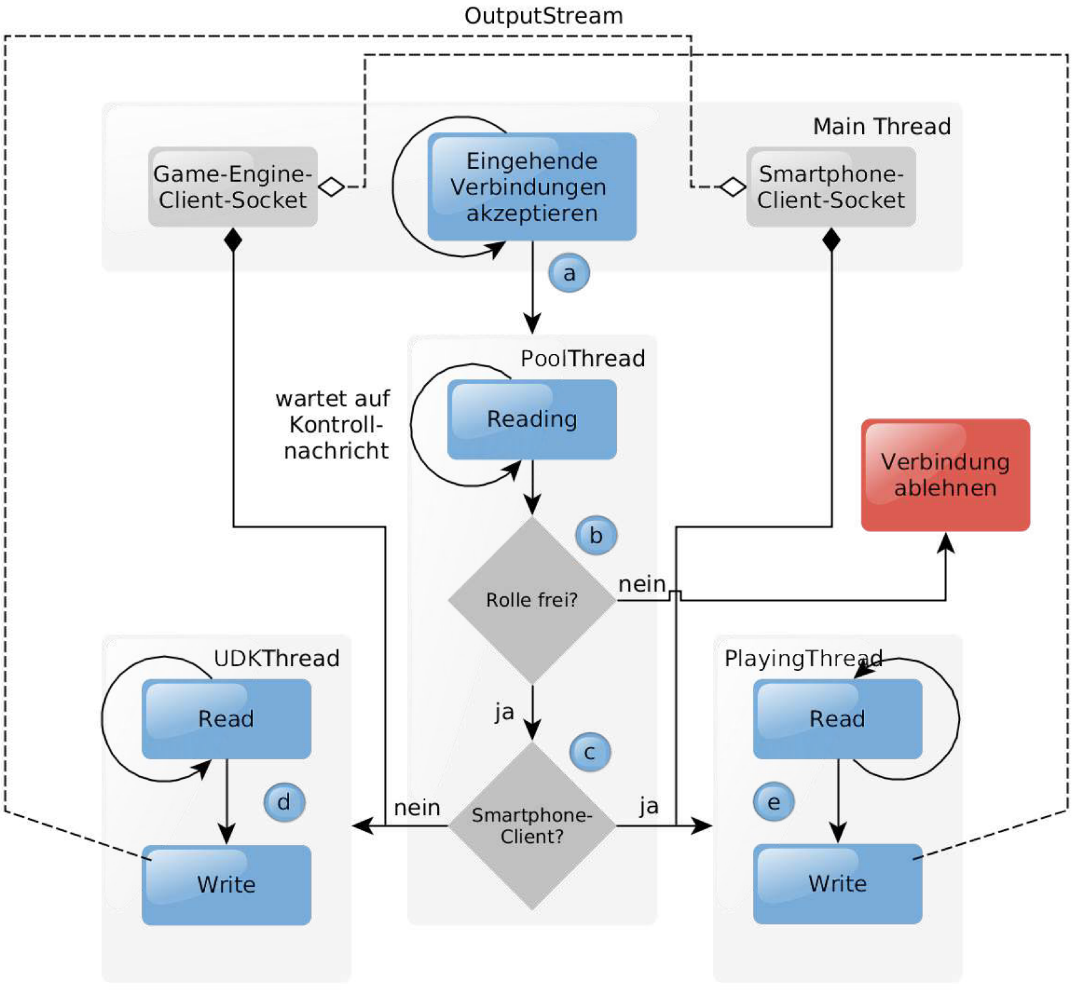


Abbildung 3.8.: Der Server speichert und verwaltet alle eingehenden Verbindungsanfragen sowie darüber hinaus auch alle Threads und Kommunikationsdaten.

- Hole Outputstream des verbundenen Game-Engine-Clients und speichere diesen zwischen.
- Versuche Daten zu lesen (blockierend).
- Bei erfolgreicher Leseoperation, erweitere Nachricht um Spieleridentifikator.
- Verwende gespeicherten Stream zum Weitersenden des modifizierten Datensatzes.

Hingegen ist der Workflow eines UDKThreads (Abb. 3.8 (c)) der folgende:

- Hole alle verbundenen Smartphone-Clients und speichere diese zwischen.
- Rufe zu jedem den Outputstream ab und speichere auch diesen.
- Versuche Daten zu lesen (blockierend).
- Prüfe ob Story-relevante Daten vorhanden sind.
  - Falls dem so ist, übergebe die Daten an den Game Master.
  - Handelt es sich lediglich um Interaktionsdaten, prüfe welcher Client der Empfänger ist und sende die Daten unter Verwendung der zwischengespeicherten Streams weiter.

Zum Weitersenden der Daten beziehen beide den Outputstream der jeweils anderen Clients. Obwohl es prinzipiell auch möglich ist Streams eines Sockets mehrfach anzufragen und zu speichern, wird von dieser Vorgehensweise abgeraten, da es sonst zu Inkonsistenzen und Interferenzen kommen kann. Aus diesem Grund speichert lediglich der Server alle relevanten Daten und gibt diese über öffentliche Methoden anderen Komponenten frei.

Der Server (3) fungiert in der Applikation nicht ausschließlich als Serviceanbieter sondern beherbergt darüber hinaus auch weitere Funktionalitäten, wie die des Game Masters (7) und der Decision Engine (8) (siehe Abb. 3.9).

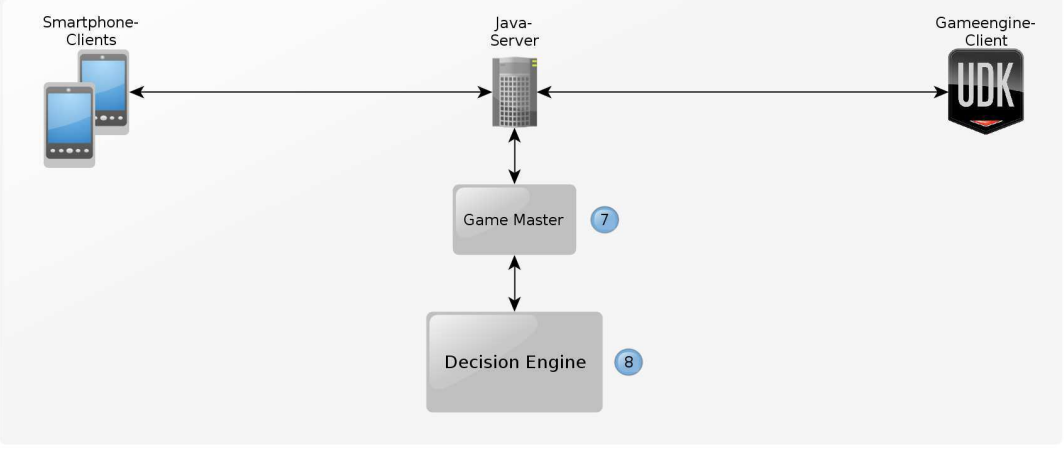


Abbildung 3.9.: Der umgesetzte Server nutzt die Kommunikationsdaten für weitere Aufgaben und stellt diese daher dem Game Master und der Decision Engine zur Verfügung.

Die Kommunikationsdaten werden in jedem Fall über den Server gesendet, sodass diese Stelle gerade zu prädestiniert dafür ist, um hier den Game Master anzusiedeln. Dieser überprüft, ob die gesendeten

Daten in irgendeiner Art und Weise für den weiteren Storyverlauf respektive Änderungen in der Welt genutzt werden können und macht gegebenenfalls von der Decision Engine Gebrauch.

3.2.3.1. Threadproblematik

Bei der Verwendung von Threads gilt es mehrere Probleme zu lösen respektive Vorkehrungen zu treffen. Prinzipiell dauert die Threaderstellung sowie -entfernung Zeit, welche die Performanz beeinträchtigen kann. Im vorliegenden Prototypen werden daher alle benötigten Threads beim Verbindungsaufbau bereits initialisiert und zwischengespeichert.

Des Weiteren hat die mit Threads einhergehende Asynchronität zwar den Vorteil den restlichen Programmablauf nicht zu behindern, genau dies verhindert jedoch auch den Zugriff auf Variablen oder Objekte außerhalb des Threads. Die hier ausgelagerten Operationen betreffen jeweils das Lesen respektive Schreiben von Daten auf Sockets, da diese Operationen blockierend umgesetzt sind. Besonders für die Leseoperationen ist es jedoch wichtig erhaltene Daten weitergeben zu können.

Der erste Lösungsansatz bei der Implementierung des Servers war eine lokale Variable im Thread zu erzeugen, in welcher das Ergebnis respektive die benötigten Daten abgelegt wurden, sodass der Threaderzeuger diese auslesen konnte. Bei der Verwendung dieses Ansatzes wurden jedoch Race Conditions festgestellt, welche durch die folgenden Möglichkeiten zu Stande kamen:

- Der Sheduler auf Betriebssystemebene kümmert sich um das Ausführen und Pausieren von Threads. Threads wurden häufiger nach erfolgreicher Leseoperation, jedoch vor dem Speichern der Daten in die Variable pausiert, was dazu führte, dass an anderer Stelle ein Zugriff auf eine nicht initialisierte Variable stattgefunden hat.
- Die Ausführungsgeschwindigkeit von Netzwerkoperationen ist unbekannt und veränderbar. Ein Lesezugriff erfolgte teilweise so schnell, dass die Abfrage des Variableninhaltes nachgelagert und somit erfolgreich ausgeführt wurde, eine längere Blockierung des Lesens und damit der Zugriff auf eine leere Variable ist allerdings ebenfalls aufgetreten.

Zur Lösung dieser Problematik wurde ein in Java gängiger und unter anderem auch von Harold empfohlener Ansatz verwendet. Bei der Erzeugung des Threads wird der Ersteller als Parameter mit übergeben und vorher mit öffentlichen Callback-Methoden versehen. Zur Übertragung der im Thread befindlichen Daten können somit diese Methoden verwendet werden. Sofern mehrere Empfänger von den Daten benachrichtigt werden sollen, ist dieses Konzept sehr einfach durch die Erweiterung um ein Observer-Pattern ebenfalls möglich.

Ferner bringt die Verwendung von Threads auf Serverseite das Risiko etwaiger Synchronisierungsprobleme mit sich.

```
1  MasterProjektSocket mps = new MasterProjektSocket([...]);
2  PrintStream out = new PrintStream([...]);
3  mps.setInputStream(input);
4  mps.setOutputStream(out);
5
6  Thread pt = new Thread(new PlayingThread(mps,server));
7
8  server.addClientSocket(mps);
9  mps.setThread(pt);
10 mps.setPlayerNumber(
11  server.getGroupOfClients("android").size()
12 );
```

Quelltext 3.1: Ohne Monitor kann insbesondere Zeile 10 zu falschen Werten führen.

Verbindungsanfragen können parallel eintreffen, sodass zur gleichen Zeit mehreren PoolThreads existieren, welche auf die Identifizierungsdaten des Clients warten. Sobald diese eingetroffen sind, werden Daten zu diesem Client im Server zwischengespeichert (siehe 3.1 Zeile 8). Nach der Erstellung eines speziellen Objektes (*MasterProjektSocket*) und der Zuweisung spezieller Eigenschaften, wird dieses in einer *ArrayList* des Servers zwischengespeichert. Insbesondere Zeile 10 greift unter Verwendung einer Methode auf selbiges Array zu, um dem zuvor erstellten Socket-Objekt noch eine Spielernummer mitzuteilen. Im folgenden soll die auftretende Inkonsistenz verdeutlicht werden:

1. Thread #1 bearbeitet den in 3.1 dargestellten Codeblock, bis Zeile einschließlich Zeile 9.
2. Der Scheduler suspendiert Thread #1 und übergibt die zur Verfügung stehende Rechenzeit an Thread #2.
    - Das Server-Array enthält zu diesem Zeitpunkt ein Socket-Objekt.
3. Thread #2 bearbeitet denselben Codeblock ebenfalls bis einschließlich Zeile 9.
4. Nun suspendiert der Scheduler Thread #2 und wechselt zu #1.
    - Das Server-Array enthält zu diesem Zeitpunkt zwei Socket-Objekte.
5. Thread #1 weist dem Socket-Objekt als Spielernummer eine zwei zu.
6. Die Abarbeitung dieses Threads ist beendet, sodass Thread #2 fortgesetzt wird.
7. Dieser weist seinem Socket-Objekt als Spielernummer ebenfalls eine zwei zu.

Durch die Verwendung eines Monitors kann einer solchen Inkonsistenz entgegengewirkt werden (siehe 3.2).

```
1  synchronized(this) {
2      server.addClientSocket(mps);
3      mps.setThread(pt);
4      mps.setPlayerNumber(
5          server.getGroupOfClients("android").size()
6      );
7  }
```

Quelltext 3.2: Durch den Einsatz eines Java-spezifischen Monitors (synchronized), wird der Codeblock für andere Threads blockiert.

Durch die Verwendung eines synchronized-Statements kann der Bereich für andere Threads blockiert werden, wodurch eine sequentielle Abarbeitung erzwungen wird. Im Bereich der Serverentwicklung musste speziell auf solche Problemsituationen geachtet und diese verhindert werden. Ferner muss jedoch ein zu häufiger Einsatz vermieden werden, da es sonst zu einer vollständig sequentiellen Abarbeitung kommen kann, was den Einsatz von Threads ad absurdum führt. Im Falle des hier implementierten Servers sind die Einsätze von Monitoren genauestens auf Sinnhaftigkeit evaluiert worden. Das Resultat war, dass die Codesegmente innerhalb solcher Blöcke so schnell abgearbeitet werden, dass eine Suspendierung währenddessen ausgeschlossen werden kann, wodurch etwaige Perfomanzeinbußen nicht auftreten können. Dennoch ist die Verwendung von synchronized-Statments mit Bedacht geschehen.

3.2.4. WLAN-Router

Um einen optimalen Datendurchsatz im Netzwerk zu erreichen wird ein eigener WLAN-Router verwendet. Durch das Ausschalten des Nagle-Algorithmus wird ein hoher Datendurchsatz erreicht, der mit anderen WLAN-Kanälen interferieren kann, wenn zu viele Pakete zur gleichen Zeit versendet werden. Das Aufspannen eines eigenen kabellosen Netzwerks ermöglicht es, diesen erhöhten Netzwerkverkehr zu kompensieren.

Sofern benötigt, können durch den Einsatz eines eigenen Routers auch weitere Optimierungen vorgenommen werden. Hierfür bieten einige Router beispielsweise die Möglichkeit bestimmte Einstellungen vorzunehmen, welche gewisse Latenzpriorisierungen ermöglichen („behandle Teilnehmer des Netzwerks mit erhöhtem Datendurchsatz bevorzugt“).

3.2.5. Game Master - Schnittstellenkonzeption

In Abschnitt 3.1.3 wurde bereits erläutert, dass der Game Master (in diesem Kapitel auch „Game Master (GM)“ genannt) eine vollständige Digitalisierung erfahren soll. Das Digitalisierungskonzept, welches in diesem Kapitel vorgestellt wird, ergibt sich dabei in logischer Schlussfolge aus den Anforderungen, welche ein digitaler Game Master für ein Rollenspiel erfüllen muss. Der Game Master kapselt eine Schnittstelle zwischen auslösenden Ereignissen und daraus resultierenden veränderten Spielsituationen. An dieser Stelle wird nun erläutert, wie diese Schnittstelle konkret konzeptioniert wurde und welche Gedanken zu diesem Konzept geführt haben.

Wie dabei bereits im Rahmen von Abschnitt 2.1.1 dargelegt wurde, verlaufen Rollenspiele hochgradig nichtlinear. Insbesondere die erzählte und erlebte Geschichte wird dabei in einem ständigen Kreislauf zwischen Game Master und Spielern vorangetrieben. Abgegrenzt wird hier vom linearen Gameplay respektive linearem Storytelling: Bei Pen-&-Paper Rollenspielen gibt es keinen „Goldenen Pfad“, sowohl im Bezug auf Story als auch auf Gameplay.

Im Rahmen der Realisierung des GM musste daher ein Konzept gefunden werden, welches den vollständig nichtlinearen Ansatz der Pen-&-Paper-Rollenspiele repliziert. Unterschieden werden muss hierbei zwischen nichtlinearem Gameplay und nichtlinearem Storytelling. Das nichtlineare Gameplay zeichnet durch vollständige Abwesenheit von Pfaden *innerhalb einer Szene* („Level“) aus, auf denen sich ein Spieler durch eine gegebene Spielwelt bewegt[61] [Sheldon, 2004, S.170].

nichtlineares Gameplay bedeutet jedoch nicht zwangsweise, dass die der Spielwelt zu Grunde liegende Story ebenfalls nichtlinear verläuft (siehe Abbildung 3.10). nichtlineares Gameplay kann die Illusion von Freiheit erzeugen, indem ein gewisser Grad ebendieser Freiheit innerhalb der Spielabschnitte eingeräumt wird. Dem gegenüber steht nichtlineares Storytelling, wie es in Pen-&-Paper-Rollenspielen praktiziert wird.

Die Antwort auf die Frage, wie nichtlineares Storytelling und nichtlineares Gameplay verbunden werden kann, heißt *modulares Storytelling*. Ein „Modul“ kann in diesem Zusammenhang auch abstrakt als „Objekt“ im Sinne von „objektorientiert“ betrachtet werden: Die Grundidee des modularen Storytelling sieht vor, dass Story und Gameplay einander komplettieren. Das Gameplay kann modulweise implementiert werden, wie es beispielsweise in Abbildung 3.10 verdeutlicht ist. Auf synonyme Art kann eine Geschichte ebenfalls in Modulen geschrieben werden, indem nichtlineares Storytelling und episodenbasiertes Storytelling miteinander kombiniert werden [Sheldon, 2004, S.275]. Modulares Storytelling ist nicht zwangsläufig an ein nichtlineares Medium gebunden:

[61] Abzugrenzen ist dies von der Tatsache, dass lwtztlich jeder Spieler im jeweiligen Entscheidungsmoment seinen eigenen, weiteren Pfad definiert.

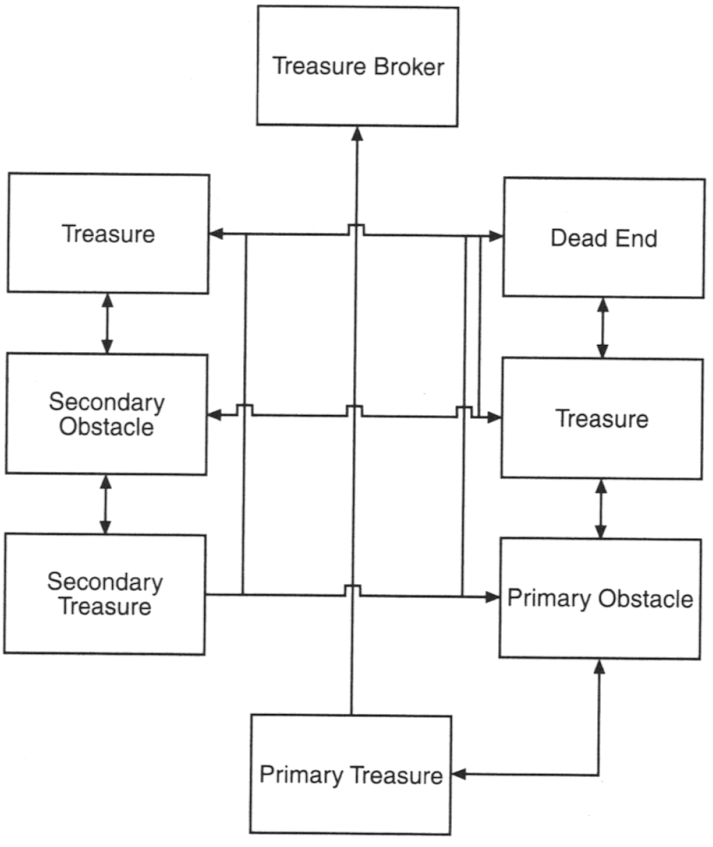


Abbildung 3.10.: Die Abbildung zeigt das Prinzip des nichtlinearen Gameplays anhand eines abstrakten Beispiels. Der Spieler hat mehre Möglichkeiten, zwischen verschiedenen Räumen zu wechseln und sich einen Weg durch die gegebenen Abschnitte zu suchen. Wichtig zu erwähnen ist hierbei, dass alle Abschnitte und deren Verbindungen untereinander bereits vorgegeben sind. nichtlinear meint in diesem Zusammenhang, dass es keine vorgegebene Reihenfolge gibt, in der die existierenden Pfade beschritten werden müssen. Die zu Grunde liegende Geschichte wird weiterhin linear erzählt, indem die Übergänge zwischen den verschiedenen Abschnitten immer vorgegeben sind.

Aus [Sheldon, 2004, S.169].

> *„If we decide that all linear media story structures are the same, just because pages follow each other in the same order in a book, [...] we are narrowing our focus far too much.“*

[Sheldon, 2004, S.296f.]

Ein Beispiel für modulares Storytelling innerhalb eines linearen Mediums ist der bekannte Film *Rashōmon*[62]. Die Geschichte um ein Verbrechen wird im Rahmen von Rückblenden aus der jeweiligen Sicht der Protagonistinnen und Protagonisten erzählt. Die Hintergrundgeschichte ist vage festgelegt, die tatsächliche Geschichte wird jedoch nichtlinear und modulweise durch die Charaktere aufgebaut. Letztlich bleiben die konkreten Ereignisse unklar und müssen durch den Zuschauer interpretiert werden. Der Film hat damit eine diffuse Hintergrundgeschichte (das Verbrechen), Rückblenden als stilistisches Mittel, um die Story aus verschiedenen Blickwinkeln (unabhängig voneinander) zu erzählen und letztendlich eine Erzählstruktur, die nicht linear von A über B nach C verläuft sondern ganz im Gegenteil diffus durch die Ausführungen der Charakter aufgespannt wird [Sheldon, 2004, S.296]. Insbesondere letztes ist für eine Nichtlinearität entscheidend: Eine Geschichte, deren Module in eine klare Reihenfolge arrangiert werden können, ist linear zu nennen[63].

Die einfachste Form, modulares (und damit nichtlineares) Storytelling zu ermöglichen, ist die sogenannte „Netzform“ auch *Web* genannt (siehe Abbildung 3.11) [Sheldon, 2004, S.307].

Diese Form, eine Geschichte zu erzählen weist zwar immer noch vorgegebene Pfade auf, jedoch sind die Übergänge zwischen den verschiedenen Story-Elementen für den Rezipienten nicht mehr ersichtlich. Die Rolle des Autors wird gelockert und er verliert die Hoheit über den Fortgang der Geschichte, da zwischen mehreren Story-Abzweigungen gewählt werden kann. Auf diese Weise bietet die Netzform einen Übergang linearer Erzählmethoden hin zum vollständig nichtlinearen Gamedesign [Sheldon, 2004, S.309]. Das modulare Storytelling ist letztlich die einheitliche Verbindung von Story und Gameplay (siehe Abbildung 3.12) [Sheldon, 2004, S.11].

Beim vollständig modularen Storytelling findet keine (auktoriale) Kontrolle mehr statt, in welche Richtung die Geschichte sich tatsächlich entwickelt. Stattdessen wird dem Spieler/der Spielerin freigestellt, den Verlauf der Geschichte nach seinen respektive ihren Wünschen frei zu gestalten [Sheldon, 2004, S.313]. Sheldon sagt hierzu:

> *„No other structure is better at integrating story and gameplay than modular.“*
>
> [Sheldon, 2004, S.313]

Während manche Elemente innerhalb von Modulen Story-relevant sind, sind andere es vielleicht nicht [Sheldon, 2004, S.313]. Mit Hilfe des modularen Storytelling kann sich der Spieler (respektive die Spielgruppe) nun völlig frei in der Spielwelt bewegen. Jede Aktion und jeder Entscheidungspunkt, den ein Spieler dabei trifft, wird festgehalten und persistiert. Diese gewonnenen Informationen können

[62] Rashōmon, Erscheinungsjahr 1950 [Kurosawa, 1950].

[63] Als Gegenbeispiel sei dem interessiertem Leser der Film *Memento* aus dem Jahre 2000 empfohlen, der im ersten Augenblick nichtlinear erscheint, dessen Module jedoch in eine lineare Reihenfolge arrangiert werden können. Dies ist sogar ein explizites Feature der DVD-Fassung [Nolan, 2000]. Bei *Rashōmon* ist ein solches Re-Arrangement nicht möglich.

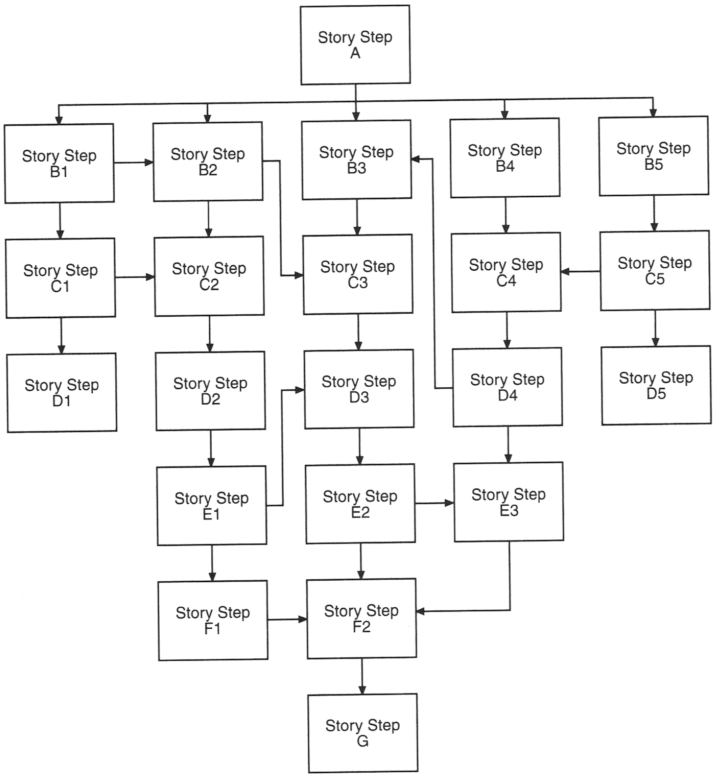


Abbildung 3.11.: Das Netzbasierte Modulkonzept bietet die simpelste Form des echten nichtlinearen Storytelling. Zwar verfügt das Netz noch immer über vorgegeben Pfade, der tatsächliche Verlauf der Geschichte ist jedoch nicht mehr vorhersehbar und wird durch den Rezipienten bestimmt. Somit bietet dieses Konzept einen Übergang zwischen linearem und modulbasiertem Storytelling, wie es auch in Pen-&-Paper-Rollenspielen verwendet wird.

Aus [Sheldon, 2004, S.308].

an sogenannten Entscheidungspunkten herangezogen werden, um das jeweils nächste Modul zu bestimmen. An dieser Stelle kann auch von einem Synergie-Effekt gesprochen werden: Für sich alleine betrachtet mögen manche Entscheidungen keinen großen Einfluss haben, insgesamt können Sie den weiteren Verlauf der Geschichte jedoch massiv beeinflussen [Sheldon, 2004, S.317]. Die Unreal Engine unterstützt die modulare Herangehensweise dabei in technischer Hinsicht durch das bereits erwähnte Level-Streaming (siehe Abschnitt 2.2.2). Abbildung 3.13 verdeutlicht dabei die verwirklichte Kommunikation im Rahmen dieses Projektes.

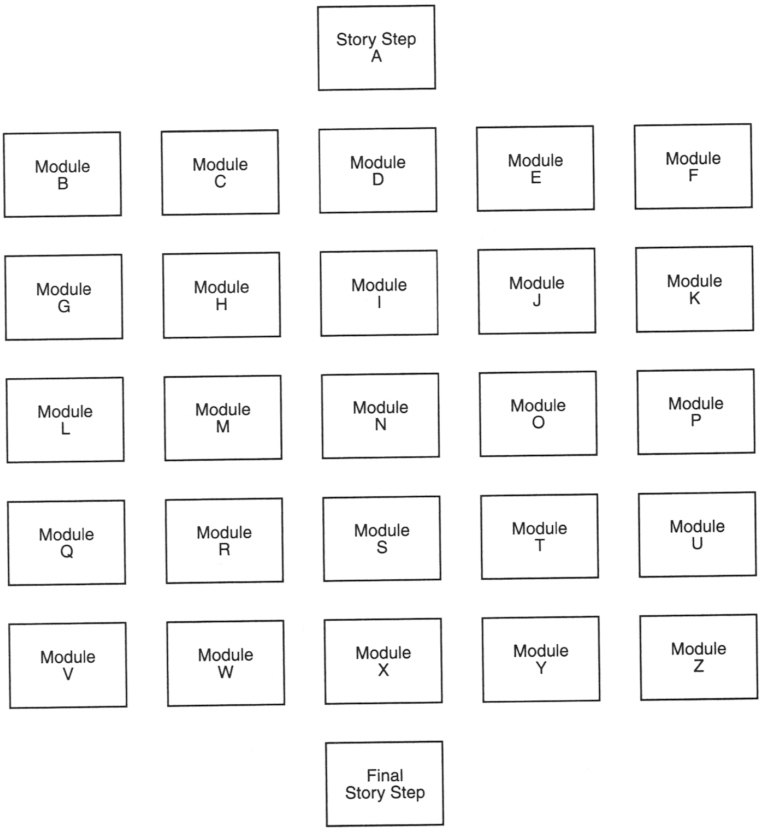


Abbildung 3.12.: Die Abbildung zeigt vorgegebene Module einer Geschichte. Beim modularen Storytelling herrschen keine vorgegebenen Verbindungen der Module untereinander; es kann beliebig von jedem zu jedem Modul gesprungen werden. Ausnahmen bilden in der Regel das erste („point of attack“) und das letzte („final resolution“) Modul, was sich allerdings auch weiter auffächern lässt, in dem mehrere Anfänge respektive alternative Enden vorgesehen werden [Sheldon, 2004, S.312].

Aus [Sheldon, 2004, S.311].

Die Spiel*geschichte* wird demnach durch Digitalisierung des Game Masters mit der Engine verbunden. An dieser Stelle soll nochmals abgegrenzt werden: Ziel des Projektes ist es nicht, ein Werkzeug zur Erstellung von dynamisch veränderbaren Spielinhalten zu erschaffen, sondern zu zeigen, wie Pen-&-Paper-Rollenspiele sinnvoll digitalisiert werden können. Die Konzeption des digitalen GM fußt dem-

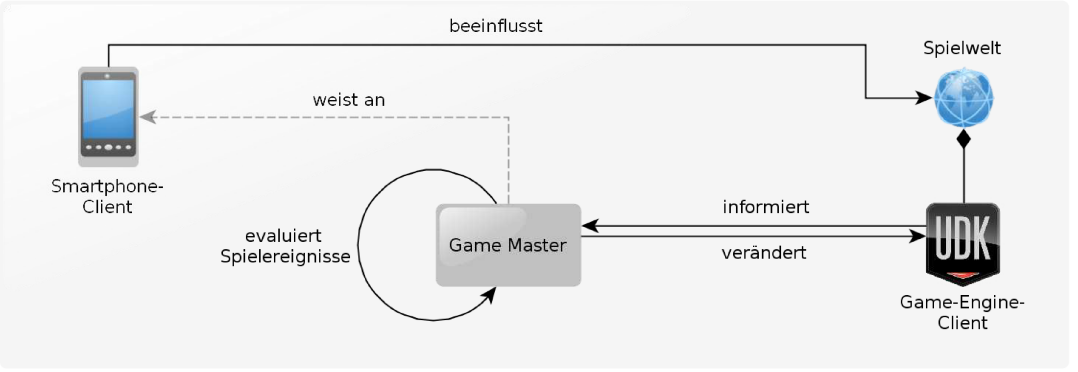


Abbildung 3.13.: Vereinfachtes Schema der verwirklichten Kommunikation mit der GM-Schnittstelle

nach auf der Prämisse, dass die Spielgeschichte in endlicher Anzahl an Modulen bereits vorliegt und nicht zur Laufzeit von Mensch oder Computer generiert wird[64].

Die Nutzer beeinflussen die von der Unreal Engine bereitgestellte Spielwelt indirekt durch die Benutzung ihrer Smartphones, beispielsweise durch schlichtes Wählen eines Pfades oder auch durch bestimmte Gespräche, gewisses Verhalten über einen längeren Zeitraum et cetera. Über ein eigens dafür erstelltes Kommunikationsprotokoll (siehe Abschnitt 3.2.7) informiert der Game-Engine-Client den Game-Master sowohl über alle Aktionen der Spieler als auch über alle veränderlichen Elemente der Spielwelt. Über eine sogenannte Decision Engine werden die Ereignisse ausgewertet und darauf folgend wahlweise der Game-Engine-Client oder auch die Smartphone-Clients angewiesen, die neu errechnete Situation herbeizuführen. Letzteres könnte in einer Anweisung bestehen, den Bildschirminhalt des Smartphones zu ändern oder ein Geräusch abzuspielen. Anweisungen an die Game-Egine könnten darin bestehen, ganze Level an bestimmten Stellen zu laden oder Elemente anzupassen.

Die Anweisungen erfolgen dabei über Remote Procedure Calls, was den modularen Aufbau der Anwendung zusätzlich unterstützt. Der GM brauch nur Kenntnis darüber, welche Methoden die Game-Engine und die Smartphones bereitstellen und kann diese entscheidungs-basiert zur Ausführung bringen. Somit wird gewährleistet, dass das System dezentral und unter Austauschbarkeit sämtlicher involvierter Elemente und Plattformen einsetzbar ist. Die Auswertung eines solchen RPC ist beispielhaft in Quelltext 3.3 dargestellt.

[64] Vgl. hierzu auch Abschnitt 3.1.3. Bei Pen-&-Paper-Rollenspielen liegt die erzählte Geschichte allenfalls in Stücken vor, der (reale) GM hat jedoch Freiheiten, diese nach Gutdünken auszugestalten; die Spielmodule sind in der Realität demnach praktisch unendlich in ihrer Anzahl. Im Fall einer Digitalisierung des GM muss die Anzahl der Module endlich gemacht werden.

```
1 DataJSon = class'JSonObject'.static.DecodeJson(toParse);
2 //[...]
3 functionName = DataJSon.GetStringValue("function");
4 functionParams = DataJSon.GetStringValue("params");
5 ConsoleCommand(functionName@functionParams,true);
```

Quelltext 3.3: Die Abbildung zeigt die Auswertung eines RPC im Game-Engine-Client. Nach Deserialisierung des übergebenem JSON-Objektes wird die Funktion samt Parametern an die Game-Engine übergeben und ohne weitere Prüfung direkt ausgeführt.

3.2.5.1. Anwendungsfall des Game Masters und der Decision Engine

Im Rahmen dieses Projektes wurde eine simple Decision-Engine auf Basis von Verzweigungen implementiert: „Wenn Ereignis *A* vorliegt, so löse die Aktion *B* aus“. Zu Demonstrationszwecken wurde daher eine simple Szenerie erstellt, die verschiedene Möglichkeiten offeriert, den digitalen Game Master und somit die Decision-Engine anzusprechen. Die erste Demonstration besteht darin, dass die Spielenden durch Wahl eines von zwei Wegen bereits eine Entscheidung treffen können. Gerade wenn die Spielenden Gang *A* durchqueren, wird Gang *B* gesperrt. Abbildung 3.14 zeigt den dabei verwendeten Aufbau.

Zeitgleich wird der Game Master informiert, dass ein bestimmter Entscheidungspunkt erreicht wurde und dass dieser für die Fortentwicklung der Story wichtig ist. Dieses Ereignis wird von der Decision-Egine ausgewertet und daraufhin ein RPC ausgelöst, welcher den Game-Engine-Client anweist, einen bestimmten Level-Abschnitt in der Welt zu laden. In diesem Anwendungsfall handelt es sich um einen Abschnitt hinter einer verschlossenen Tür (vgl. Abbildung 3.15).

Die zweite Demonstration besteht in der Simulation eines sozialen Ereignisses. Ein „soziales Ereignis“ im Sinne dieses Projektes wird definiert als:

> *Ein reale Aktion mit virtuellem Ergebnis, welche in kollaborativer Art und Weise durchgeführt werden muss.*

Kollaborativ heißt hierbei ausdrücklich nicht kooperativ: Bei kooperativen Aktionen kann immer noch jede/r Spielende/r sozial isoliert agieren. Wie in Abschnitt 3.1.4 bereits erwähnt wurde, beeinflusst die Wahl von Smartphones als Eingabegerät die Möglichkeiten, die im Rahmen einer Mixed Reality-Anwendung zur Verfügung stehen. Konkret wird die Möglichkeit eröffnet, Aktionen der virtuellen Welt in der realen Welt nachzustellen. Auf diese Weise kann das Erlebnis zum Nutzer hin verschoben werden, was dem Ziel der Diffusion von realer und virtueller Welt entgegenkommt. Ein soziales Spie-

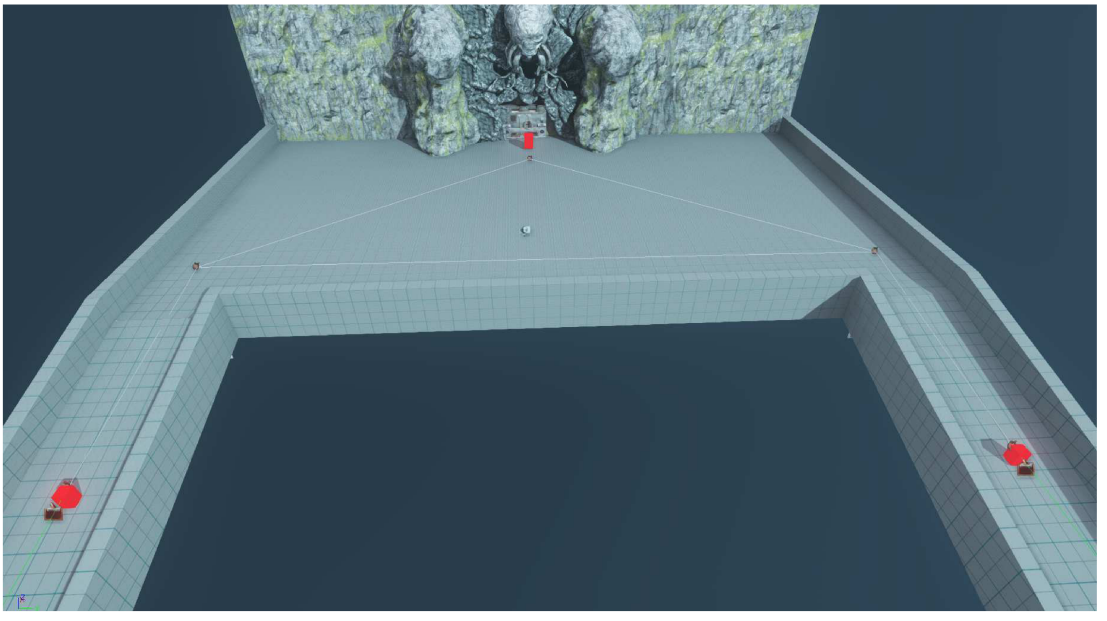


Abbildung 3.14.: Die Abbildung zeigt das Setting des Prototypen: In jedem Gang (rechts und links) sind Entscheidungspunkte versteckt (zu sehen als rote Kuben). Entscheiden sich die Spielenden für einen Gang, wird der jeweils andere Gang gesperrt. Schon durch die Wahl des Ganges beeinflussenden die Spielenden, welcher Abschnitt hinter der Tür (roter Kubus in der Bildmitte Oben) wartet.

len im Sinne eines Pen-&-Paper-Rollenspiels kann zudem zusätzlich gefördert werden, da bestimmte Aktionen sowohl virtuell als auch real stattfinden (müssen).

Nachdem durch die Wahl des Pfades der weitere Spielverlauf bereits beeinflusst wurde, müssen die Spielenden nun in gemeinsamer Aktion „eine Tür aufschieben“. Dazu begeben sich beiden Spielenden virtuell vor eine massive Stahltür. Der Game-Engine-Client informiert den Game Master über die entsprechende Positionierung der Spielenden und über die Möglichkeit, hier eine Aktion ausführen zu können. Die Smartphone-Clients werden darüber benachrichtigt und informieren die Spieler, dass eine Aktion durchgeführt werden kann. Sofern die Spielenden dieser Aufforderung nachkommen, bekommen sie auf dem Display ihrer Smartphones dargestellt, eine „Schiebe“- respektive Wischgeste auszuführen (siehe Abb. 3.16). Um die Gemeinsamkeit der Interaktion abzubilden, ist eine nahezu zeitgleiche Durchführung der Geste von beiden Spielern notwendig. Um diese zeitliche Korrelation festzustellen, wird der Server von der Game-Engine über solch ein anstehendes Ereignis informiert, woraufhin ein spezielles Objekt angelegt wird, welches das weitere Vorgehen übernimmt. Die Interaktionsdaten der einzelnen Spieler erreichen nach der Ausführung auf dem Smartphone den Server

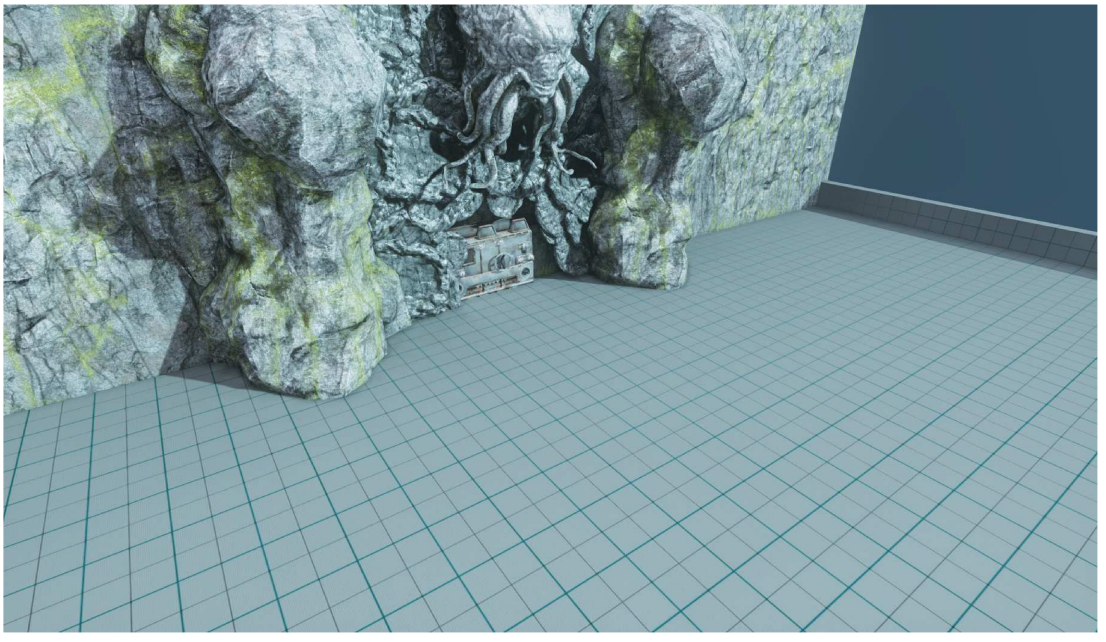


Abbildung 3.15.: Die Abbildung zeigt einen Aktionspunkt des Spiels: Eine verschlossene Tür, welche in kollaborativer Arbeit gemeinsam von den Spielenden geöffnet werden muss.

und werden unmittelbar an dieses Objekt, dem *CollaborationChecker*, weitergeleitet. Zur Feststellung der „gemeinsamen“ Durchführung werden Zeitstempel im Nanosekundenbereich verwendet. Wenn beide Spieler die Aktion innerhalb eines kurzen Zeitrahmens ausgeführt haben, gilt die Aktion als „gemeinsam“ durchgeführt. Da der Zeitrahmen sehr eng bemessen ist, müssen sich die Spielenden real absprechen, wann sie mit der Geste beginnen. Auf diese Weise wird zusätzlich eine reale Interaktion gefordert und forciert. Bei erfolgreicher Ausführung der Geste stößt der Game-Master per RPC direkt das Öffnen der Tür an. Hinter der Tür wartet daraufhin der Level-Abschnitt, welcher vorher durch die Wahl des Ganges festgelegt wurde. Ferner werden auch die Smartphone-Clients darüber benachrichtigt die View zu wechseln.

3.2.6. Game-Engine-Client

Der Game-Engine-Client besteht aus einer mit dem UDK realisierten Anwendung und übernimmt mehrere Kernaufgaben des Prototypen.

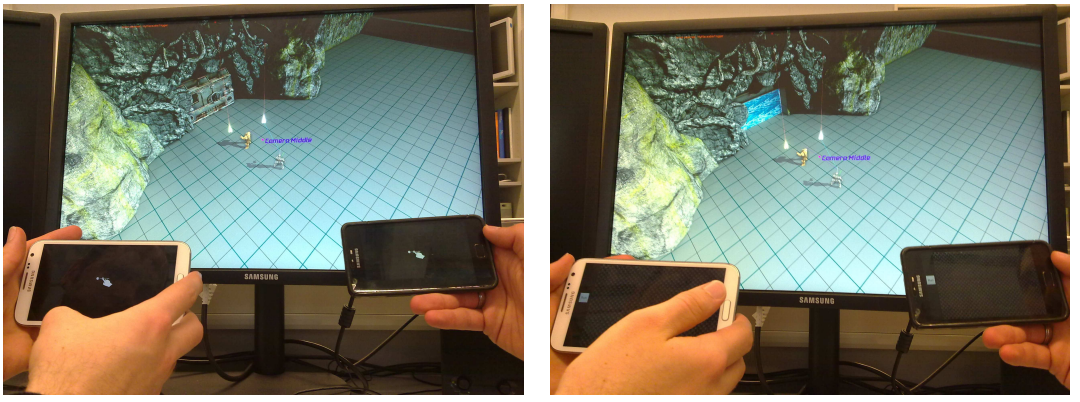


Abbildung 3.16.: Auf der linken Abbildung ist die Interaktionsview auf den Smartphone-Clients zu sehen. Nach einer gemeinsamen Ausführung der Wischgeste öffnet sich die Tür in der Spielwelt (rechte Abb.).

Der Game-Engine-Client nimmt Befehle des Game-Masters entgegen und setzt diese in Visualisierungen um, indem Updates der Spielwelt vorgenommen werden (siehe Abbildung 3.13). Innerhalb des Game-Engine-Clients werden sämtliche möglichen Ausprägungen des Spiels und der Story vorgehalten, deren konkrete Zusammensetzung jedoch zur Laufzeit per Remote Procedure Call vom Game Master bestimmt wird.

Im Rahmen dieses Prototypen wurden drei Module realisiert: Eine „Hauptwelt“, in der die Spieler starten, sowie zwei „Nebenmodule“ welche als Demonstration eines dynamisch-modularen Aufbaus dienen. Die Nebenmodule sind dabei lediglich als optische Überprüfung dafür gedacht, dass Spieler mit ihren Aktionen respektive Verhalten eine Auswirkung auf den weiteren Verlauf des Spiels und somit der Story nehmen können. Die Konzepte hinter diesen Modulen wurden im vorangegangenen Abschnitt näher erläutert.

Während der Entwicklung auf Seiten des Game-Engine-Clients mussten mehrere Probleme gelöst werden, von denen zwei insbesondere als exemplarisch für die Herangehensweise an eine Lösung näher betrachtet werden sollen.

3.2.6.1. Realisierung mehrerer Spieler

Da das Ziel des Projektes die Digitalisierung eines Pen-&-Paper-Rollenspiels ist, welches von mehreren Spielern gespielt werden kann, musste auch eine Möglichkeit geschaffen werden, die Partizipation mehrerer Spieler durch den Game-Engine-Client visualisieren zu lassen.

Das für die Umsetzung der Visualisierung gewählte Unreal Development Kit verfügt zwar über eine umfangreiche Unterstützung für Multiplayer-Spiele, jedoch nicht in einer Form, die auf den geplanten Prototypen anwendbar gewesen wäre. Durch die Abstammung der Unreal-Engine vom Gebiet der sogenannten Ego-Shooter[65] lässt sich an der Unreal-Egine ein Spieler-zentriertes Entwicklungskonzept erkennen, bei dem grundsätzlich von der Prämisse ausgegangen wird: *Ein* Mensch steuert *einen* Avatar mittels *einer* Eingabemethode durch *eine* Spielwelt.

Dies bedeutet, dass zwar vorgesehen ist, dass mehrere Spieler mit verschiedenen Endgeräten miteinander spielen können, die gleichzeitige Darstellung mehrerer vom Menschen kontrollierten Avatare auf einem gemeinsamen Bildschirm jedoch ein Problem darstellt.

Es ist möglich, mehrere Instanzen der Engine auf einem Endgerät zu starten, um beispielsweise ein Splitscreen-Verfahren zu ermöglichen, allerdings stellt dies nur eine andere Form der Multiplayer-Unterstützung dar. Hier werden zwei Instanzen der Engine logisch getrennt und über zwei getrennte Bildschirmabschnitte ausgegeben.

Um mehrere Spieler auf einem gemeinsamen Bildschirm zu simulieren, wird daher auf eine Lösung zurückgegriffen, die das vom UDK vorgesehen Konzept nicht unterwandert und gleichzeitig für den Zweck des Prototypen nutzbar macht. Dabei gibt es einen „echten“ Spieler, mit dessen Instanz die Logik des gesamten Spiels verbunden ist. Dies ist automatisch derjenige Spieler, welcher sich als erstes zum Server verbindet.

Jeder weitere Spieler erzeugt einen Non Player Character (NPC), in diesem Fall einen sogenannten *Bot*. Bots sind in ihrer Natur vom Computer gesteuerte Mitstreiter oder Gegner, welche menschliche Mitspieler simulieren oder ersetzen sollen. Im Fall der Realisierung des Prototypen erlaubt das Erzeugen von Bots die Darstellung mehrerer Avatare auf einem gemeinsamen Bildschirm. Bots können Befehle oder Verhaltensmuster auferlegt werden, beispielsweise einen Spieler anzugreifen oder sich fortzubewegen. Diese werden normalerweise intern durch die Game-Engine verwendet. Der verfolgte Lösungsansatz an dieser Stelle ist, echten Spielenden die Hoheit über genau diese Befehle zu geben. Das heisst: Normalerweise steuert ein Computer die Bots. Im realisierten Prototypen steuern echte Menschen die Bots. Ein Unterschied in der Bedienung ist für den Endbenutzer im Ergebnis nicht erkennbar. Bis auf den „echten“ Avatar handelt es sich demnach lediglich um computergesteuerte Avatare, deren Steuerung jedoch retrospektiv an die echten, menschlichen Spieler übergeben wurde.

Abbildung 3.17.: Die Abbildung zeigt die Berechnung der Kamera-Mitte in Aktion: Obwohl die Spielenden an verschiedenen Rändern des Bildinhaltes stehen, zentriert die Kamera auf die Bildmitte.

3.2.6.2. Kalkulation der Kamera

Das UDK wurde vornehmlich für Spieler in der Ego- oder auch einer (limitierten) Third-Person-Perspektive entwickelt, was auch darin resultiert, dass das Kamera-System hierarchisch tief in der Engine verankert ist [Thorn, 2012, S.629f.]. Um ein gemeinsames Spielerlebnis zu ermöglichen, musste jedoch eine Lösung gefunden werden, alle Spielenden zu jedem Zeitpunkt gemeinsam durch eine Kameraperspektive einzufangen. Nach Vorbild diverser Rollenspiel-Umsetzungen wie zuletzt *Diablo 3* für die Playstation 3 wurde hierfür eine sogenannte isometrische Perspektive gewählt. In Abbildung 3.2.6.2 ist sowohl die isometrische Perspektive als auch die Kamerakalkulation zu sehen.

Eine solche Perspektive erlaubt es allen Spielenden, das Spielgeschehen aus einer übergeordneten Sicht zu verfolgen. Ziel der Kamera-Modellierung war, eine neutrale Kamera zu erschaffen, welche keinen Avatar bevorteilt oder benachteiligt und sich adaptiv an das Spielgeschehen anpasst. Hier ergab sich jedoch die Problematik, dass sowohl das gesamte Steuerungssystem als auch die Kamera an den einzi-

[65] Insbesondere die *Unreal Tournament* - Reihe des Engine-Herstellers dient regelmäßig als Grafikdemonstration der jeweils neusten Generation der Unreal-Engine: *„Unreal introduced players to the Unreal Universe [...]. It also put the Unreal Engine technology on the map as a serious contender in the game engine middleware business [...]“* [Epic Games, Inc., 2013j]

gen „echten“ Spieler gebunden sind, wie es bereits im vorherigen Abschnitt beschrieben wurde. Nach Justierung der Kamera-Parameter war es demnach zwar möglich, den (Haupt-)Spieler isometrisch im Bild einzufangen, jedoch folgte die Kamera ab diesem Zeitpunkt starr den Bewegungen des korrespondierenden Spielers und hielt diesen im Mittelpunkt, da dieser immer noch das Kamera-Ziel darstellte. Das Resultat war eine isometrische Kamera, die nur dann alle Spielenden im Bild einfing, wenn diese sich in relativer Nähe zum Haupt-Spieler aufhielten.

Um dieses letzte Problem zu lösen, wird zwischen allen Avatar-Positionen zur Laufzeit des Spiels regelmäßig die geometrische Mitte errechnet und dieser als neuer Bezugspunkt der Kamera gesetzt. Damit ist es möglich, alle Spieler gleichzeitig im Bild einzufangen und den Bewegungen dieser sanft zu folgen. Um insbesondere eine sanfte Bewegung zu ermöglichen, wird vom Konzept der in Abschnitt 2.2.2.1 bereits erwähnten *Ticks* Gebrauch gemacht. In Quelltext 3.4 ist dargestellt, wie die jeweils neue Kamera-Position errechnet wird. Zunächst wird geprüft, ob sich die Distanz der Kamera zum Bezugspunkt im Vergleich zum vorherigen Distanzwert im letzten *Tick* verändert hat. Ist dies der Fall, wird die neue Kamera-Distanz sanft zum neuen Bezugspunkt interpoliert. Dem Gegenüber wird davon ausgegangen, dass die Kamera-Position sich in grundsätzlich in jedem Tick ändert. Deshalb wird die neue Position kontinuierlich zwischen der letzten bekannten und der jeweils neuen interpoliert.

```
1 if(Dist != FreeCamDistance) {
2    Dist = Lerp(Dist, FreeCamDistance, 0.15f);
3 }
4 Pos = Loc - Vector(Rot) * Dist;
5 OutVT.POV.Location = VLerp(Pos, OrigPOV.Location, 0.9f);
```

Quelltext 3.4: Die Abbildung zeigt die Kalkulation der Kamera-Distanz und der Kamera-Position, wobei die neuen Werte in Abhängigkeit der Werte im vorherigen *Tick* interpoliert werden.

3.2.7. Kommunikationsprotokoll

Für die Kommunikation zwischen den Teilnehmern wurde ein kompaktes Datenformat benötigt, welches nicht zu viel Overhead mit sich bringt, für Menschen les- und wartbar ist und sich für den Datenaustausch zwischen Anwendungen, welche auch in unterschiedlichen Programmiersprachen geschrieben sein können, eignet. Aufgrund der gemeinsamen Codebasis der Smartphone-Clients und dem Server, hätte sich ein Direktaustausch von Javaobjekten zwar angeboten, was aufgrund der von Java verschiedenen Codebasis des Game-Engine-Clients jedoch nicht möglich war.

3.2.7.1. Byte-basierter Ansatz

Als erster Ansatz wurde das Übertragen einzelner Bytes respektive Bytestreams implementiert. Auf der Java-Seite ergaben sich beim Lesen solcher Daten allerdings Probleme. Sockets bieten zum Überprüfen des Vorhandenseins von Daten Methoden an. In einer Endlosschleife wird eine solche Methode aufgerufen, welche die Anzahl der zum Lesen zur Verfügung stehenden Bytes anzeigt. Im nächsten Schritt kann diese Anzahl verwendet werden um die Leseoperation auszuführen. Die zurückgegebene Anzahl entspricht allerdings lediglich einem geschätzten Wert [Krishnamoorthy Dharmalingam, 2009; Oracle, 2011; Harold, 2004], was dazu führen kann, dass in einer Leseoperation nicht der vollständige Datensatz erfasst wird. Bei dieser Vorgehensweise müsste somit ein eigener Caching-Algorithmus implementiert und die Datensätze mit einem eindeutigen Trennsymbol versehen werden.

Als Problemlösung wurden die Inputstreams daher mit einem weiteren Filter Stream versehen, welcher das so genannte *Buffering* bereits implementiert. Die Datensätze mussten daraufhin noch mit einem eindeutigen Trennsymbol versehen werden. Gewählt wurde hierbei die Übertragung als einzelne Zeile, welche sich dadurch auszeichnet, dass sie mit einem Zeilenumbruchzeichen endet. Durch die Verwendung eines entsprechenden Schreibfilters, konnten Datensätze ohne weiteren Overhead als Zeile gesendet und empfangen werden.

3.2.7.2. JSON-basierter Ansatz

Das Senden von Bytestreams respektive einzelnen Bytes, je nach Einsatzzweck erfordert einerseits einen zusätzlichen Logikteil auf allen involvierten Teilnehmern sowie andererseits entsprechende *Parser* und *Composer*.

Die zusätzliche Logik muss beispielsweise beim Empfang einer Nachricht die Länge und das Vorhandensein gewisser Identifikatoren überprüfen um die Nachricht entsprechend verarbeiten zu können. Der Parser muss längere Nachrichten in einzelne Bestandteile zerlegen und für die Clients zur weiteren Verwendung aufbereiten, zum Beispiel durch eine Typumwandlung. Entsprechend wird für die Umkehrrichtung ein Composer benötigt, welcher den Datensatz richtig „zusammenbaut“. Diese weiteren, benötigten Funktionalitäten müssen jeweils implementiert und getestet werden, sowie für spätere Einsatzzwecke erweiterbar sein. Da bereits ein offener Standard existiert, welcher alle Anforderungen erfüllt und in die verwendeten Entwicklungsumgebungen integriert werden kann, wurde sich für die Verwendung von der JavaScript Object Notation entschieden.

Ein konkretes Beispiel zur Verwendung von JSON im Rahmen des Projektes ist mit Quelltext 3.5 gegeben. Konkret handelt es sich hierbei um den Zusammenbau und Übersendung eines Datensatzes vom

```
if(Owner.StoryRelevant){
    MainJSon = new () class'JSonObject';
    ActionJSon = new () class'JSonObject';
    DataJSon = class'JSonObject'.
                static.DecodeJson(Owner.JSonData);

        ActionJSon.SetObject("data",DataJSon);
        ActionJSon.SetIntValue("type",Owner.TriggerType);
        MainJSon.SetObject("action",ActionJSon);
        MainJSon.SetIntValue("player",Owner.ToucherID);
        MainJSon.SetBoolValue("story",true);

    // Send the encoded JSonObject
    serializedJSonToSend =
    class'JSonObject'.static.EncodeJson(MainJSon);

    SendText(serializedJSonToSend);
}
```

Quelltext 3.5: Der Quelltext zeigt den Zusammenbau eines Story-Relevanten JSON-Objektes auf Seiten des Game-Engine-Clients.

Game-Engine-Client an den Server, welcher ein Story-entscheidendes Kriterium beinhaltet. In diesem Fall handelt es sich um ein Ereignis, bei dem ein sogenannter Story-entscheidener Trigger ausgelöst wurde, was bedeutet, dass die dem Game-Master hintergeschaltete Decision-Engine das Ereignis als Story-relevant bearbeiten muss. Eine Auswertung des Ereignisses kann daraufhin als Remote Procedure Call erfolgen, wie in Abschnitt 3.2.5 beschrieben.

```
{
        action: {
                type: (int)
                data: {
                        x: (int)
                        y: (int)
                        act: boolean
                }
        }
}
```

Quelltext 3.6: Im Beispiel ist das JSON-Objekt, welches die Smartphone-Clients zum Senden eines Bewegungsupdates an den Server verwenden, dargestellt.

Ein generisches Beispiel eines verwendeten JSON-Objektes ist in Quelltext 3.6 zu sehen. Hierbei handelt es sich um eine Datensatzstruktur, wie sie von einem Smartphone-Client zur Übermittlung von Bewegungsupdates verwendet wird. Der Schlüssel *type* stellt einen Identifikator für die Art der Nachricht dar, so dass eine Zuordnung ohne eine Komplettanalyse (d.h ohne das komplette durchparsen aller Unterobjekte) möglich ist. Dies beschleunigt die Abarbeitung einer Nachricht um ein Vielfaches.

KAPITEL 4

Zusammenfassung und Fazit

Das Ziel dieser Arbeit bestand darin, das Konzept von Pen-&-Paper-Rollenspielen auf die digitale Welt zu übertragen. Die Digitalisierung dieses in seinen natürlichen Ursprüngen analogen Spielprinzipes fand unter Verwendung von modernen Ein- und Ausgabemethoden statt. Dabei war es ein explizites Projektziel, dass die Digitalisierung einerseits das Spielerlebnis bereichern soll und andererseits insbesondere die Verwendung von zusätzlicher Technologie nicht als Barriere im Sinne einer Zugänglichkeit oder Reduzierung des Engagements empfunden wird.

Um dieses Ziel zu erreichen, wurde ein Prototyp konzeptioniert und implementiert, der technische Lösungen und Konzepte kapselt, welche in kommerziellen Produkten und ähnlichen Ansätzen aus Wissenschaft und Forschung bisher in dieser Art noch nicht realisiert wurden. Dazu gehört unter anderem die Entscheidung, Smartphones als Ein- und gleichzeitig als Ausgabegerät zu verwenden. Auf diese Weise wurde eine Mixed Reality-Anwendung „erschaffen“, welche das Spielerlebnis vom zentralen Ausgabemedium weg und hin zum Nutzer verschiebt. Diese Verschiebung resultiert in einer Diffusion von virtueller und realer Welt ineinander. Ausdrückliches Projektziel hierbei war ebenfalls, die soziale Interaktion zu erhalten und zu fördern, sowohl virtuell, als auch real. Dieses Ziel wurde erreicht, indem virtuell ein gemeinsamer sozialer Raum erschaffen wurde, indem die Spielenden die Repräsentation ihrer Charaktere immer gemeinsam auf einem Bildschirm sehen. Dieses Erlebnis wurde mit der Realität verschmolzen, indem die Spielenden virtuelle Aktionen tatsächlich in der Realität durchzuführen haben. Eine solche Aktion wurde exemplarisch in der Form implementiert, dass die Spielenden eine virtuell vorhandene „schwere“ Stahltür „aufschieben“ müssen und diese Aktion in der Realität durch eine gemeinsame Wischgeste durchgeführt wird.

Weiterhin wurde ein vollständig digitaler Spielleiter implementiert, welcher die Aktionen und Entscheidungen der Spieler modular unabhängig auswertet und in neue Situationen überführt. Modular unabhängig in diesem Kontext bedeutet zweierlei: Einerseits ist der digitale Spielleiter als eigenständiges Modul erschaffen worden, welches unabhängig von gewählten Ein- und Ausgabegeräten ein-

gesetzt werden kann und ausschließlich über etablierte Netzwerkstandards kommuniziert. Andererseits bedeutet die modulare Unabhängigkeit, dass Entscheidungen über einen Spiel- respektive Storyverlauf in Echtzeit einzig auf Basis der zur Verfügung gestellten Informationen erfolgen und nicht im Vorhinein prädestiniert festgelegt wurden. Das Konzept eines *vollständig digitalen*[66] Spielleiters wurde gewählt, um Brüche in der Befehlsweitergabe zu vermeiden und um ein responsives Echtzeit-Spielerlebnis zu ermöglichen.

Um die Kommunikation der Eingabegeräte mit der darstellenden Visualisierung zu gewährleisten, wurde ein Server-Client-Modell gewählt. Wie bereits erwähnt unterstützt dies wiederum den Vorteil der Modularität, da hier sämtliche Komponenten des Systems untereinander austauschbar sind. Um einen Bruch in der Eingabekette zu vermeiden und das *Engagement* der Spielenden aufrecht zu erhalten, wurde zudem auf eine drahtlose Kommunikation gesetzt. Um die Schwierigkeiten, die bei einer solchen Kommunikationsform auftreten können, zu kompensieren, wurde eigens für dieses Projekt ein auf TCP basierendes Netzwerkprotokoll entwickelt. Durch den Aufbau der Anwendung als Client-Server-Modell konnte hierbei an den kritischen Stellen durchweg auf die Programmiersprache *Java* zurückgegriffen werden, welche eine tiefergehende Konfiguration speziell des Paketmanagements erlaubt. Wichtig war hierbei insbesondere die Möglichkeit, den so genannten Nagle-Algorithmus der TCP-Schicht auszuschalten, um den Datenaustausch zwischen den Endgeräten signifikant steigern zu können.

Das Gesamtkonzept der Anwendung besteht schließlich darin, dass die menschlichen Spieler ihre Rollen in digitaler Form wahrnehmen können, in dem sie Smartphones als Ein- und Ausgabegeräte benutzen. Diese Ausgabegeräte stellen Clients dar, welche mit einem den Datenstrom regulierenden Server verbunden werden. Gleichzeitig ist der Server mit einem weiteren Client verbunden, welcher die Ergebnisse der Aktionen der Spieler visualisiert. Hierfür wurde das Unreal Development Kit herangezogen. Der Action-Response-Circle findet somit immer zwischen den einzelnen Modulen statt: Spieler verändern die Spielwelt über ihre Eingabegeräte, die Spiel-Engine informiert den digitalen Spielleiter über diese Veränderungen und auf Basis einer mit dem Spielleiter verbundenen Decision Engine wird die visualisierende Engine per Remote Procedure Call angewiesen, diese Veränderungen umzusetzen.

Im Rahmen dieses Projekts wurde damit eine technologische Grundlage geschaffen, die es erlaubt, Pen-&-Paper-Rollenspiele vollständig zu digitalisieren. Auf dieser Basis sind vielerlei nachfolgende Arbeiten vorstellbar. So kann nun beispielsweise erforscht werden, wie bestimmte soziale Interaktionen das Spielerlebnis weiter intensivieren können. Einige Arten kollaborativer Aktionen werden hierbei eher auf Akzeptanz stoßen als andere, was im Rahmen einer vergleichenden Evaluation eruiert werden kann. Weiterhin ist es im Hinblick auf einen gemeinsamen, sozialen Spielfluss interessant, eine

[66] im Gegensatz zu einem nur teilweise digitalisiertem Spielleiter

Unterscheidung zwischen den Konzepten *Kooperation* und *Kollaboration* zu erarbeiten. Gegebenenfalls verschiebt sich mit Anzahl der Spielenden, welche in Rahmen dieses Prototypen noch auf zwei Menschen begrenzt ist, der Schwerpunkt mehr zu einer Kooperation, da kollaborativer Tätigkeiten ab einer bestimmten Anzahl der Involvierten schwer(er) zu koordinieren sind.

Ein weiterer Forschungszweig könnte sein, auf welche Art und Weise Aufbau und Verlauf respektive Verschachtlung von Story-Modulen Einfluss auf das Engagement und die Immersion der Spielenden nehmen. Jeder Mensch nimmt eine Geschichte individuell wahr und Gewichtet ihre Bestandteile unterschiedlich in ihrer Relevanz und Qualität[67]. Mit steigernder Anzahl der Teilnehmer entstehen Spannungsfelder und Diskrepanzen, wenn ein Konzens erarbeitet werden muss, in welche Richtung eine Geschichte weiter verlaufen soll.

Auf technologischer Basis kann erforscht werden, wie Spiele respektive Spiel-Module und Geschichts-Module in Einklang gebracht werden können. Die Aussage von Sheldon

> *„No other structure is better at integrating story and gameplay than modular.“*

[Sheldon, 2004, S.313]

kann dahingehend untersucht werden, wie genau Story und Gameplay ineinander verschachtelt werden können. Hier spielen auch Ereignisse und deren Visualisierung eine entscheidende Rolle. Eine besondere Rolle nimmt hier die Schnittstelle ein, die neue Story-Abschnitte anhand der gegebene Informationen generiert. Hier besteht viel Raum zur Entwicklung eines Konzeptes, beispielsweise in Form eines endlichen Entscheidungsautomaten, welcher die Komplexitätsanforderungen einer solchen Decision Engine beherrschbar macht.

[67] Dies ist alleine schon dadurch ersichtlich, dass ein und der selbe Film oder dasselbe Buch grundsätzlich die gesamte Bandbreite der Kritik, von gut bis schlecht, zu erwarten hat. Was allgemein als „gute“ oder als „schlechte“ Geschichte angesehen wird, ist nicht Bestandteil dieser Ausarbeitung, ist dennoch sicherlich ein weiterer, interessanter Diskussionspunkt.

Digitalisierung eines Pen-&-Paper-Rollenspiels mit Übertragung von Interaktionen in die reale Welt

MASTERARBEIT

„Gälte es stets alles von der Wirklichkeit aus zu beurteilen, wären Spiele sinnlos; aber wenn Spiele sinnlos wären, was sonst gäbe es noch zu tun?“

(Leo Tolstoi; von den Autoren übersetzt aus [Tolstoy, 2009, S. 48]*)*

KAPITEL 5

Einleitung

Gleich zu Beginn wird durch das einleitende Zitat dasjenige Medium in Frage gestellt, um welches es im Kern dieser Ausarbeitung geht: Spiele. Während Tolstois Novellenfigur mit seiner Aussage das Spielen an sich zwar rechtfertigt, wirft dieses Zitat dennoch gleich mehrere Aspekte auf: Sind Spiele von der Realität abzugrenzen? Sollten Spiele als „sinnlos“ betitelt werden respektive sind sie gar als Antagonist der „wichtigen“ Dinge im Leben zu betrachten?[68] Viele wissenschaftliche Ausarbeitungen beschäftigen sich wiederkehrend mit diesen und weiteren Fragen, wobei im Kern meist darüber geforscht wird, *warum* Menschen spielen [Fine, 2002, S. 5]. Diese Arbeit beschäftigt sich jedoch vor allem damit, *wie* sie das tun - und welche Rolle die fortschreitende Digitalisierung sinnvoll darin einnehmen kann. Von dieser ersten Differenzierung abgesehen, ist eine allgemeingültige Definition des Begriffs „Spiel“ grundsätzlich hilfreich, wobei inzwischen auf eine Vielzahl von bereits erarbeiteten Vorschlägen zurückgegriffen werden kann. Die von den Autoren Avedon u. Sutton-Smith erarbeitete Formulierung hat sich hierbei als Standard[69] etabliert:

> *„Games are an exercise of voluntary control systems in which there is an opposition between forces, confined by a procedure and rules in order to produce a disequilibrial outcome.“*
>
> [Avedon u. Sutton-Smith, 1971, S. 405]

Insbesondere die Aspekte des *Vorhandenseins von Regeln*, die *Orientierung auf ein Ziel hin* und der *vom vorherigen Zustand verschiedene Ausgang eines Spiels* lassen sich in einer Vielzahl[70] von unterschiedlichen Definitionsansätzen wiederfinden[71]. Eine definitiv unumstrittene Eigenschaft haben viele Spiele

[68] wie Fine resümiert [Fine, 2002, S. 5]

[69] vgl. hierzu auch [Fine, 2002, S. 7]

[70] Für eine nähere Betrachtung kann hierfür auch die Tabelle B.1 im Anhang auf Seite xxi herangezogen werden.

[71] In einer vergleichenden Darstellung erarbeiteten Salen u. Zimmerman auch folgende, neuere Definition, welche als Resultat einer vergleichenden Analyse von Revisionsvorschlägen versucht, eine Mehrzahl an Kennzeichen zu vereinen:

> *„A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome.“*
>
> [Salen u. Zimmerman, 2003, S. 80]

ebenfalls gemein: Sie können (und werden oftmals) als soziale Tätigkeit von mehreren Personen ausgeführt[72]. Zwar stellen Spiele, in einem größeren Kontext betrachtet, nur eine von vielfältigen Möglichkeiten dar, dem inneren Bedürfnis der Menschen nach Sozialisierung eine Plattform zu bieten[73]. Doch gerade hier spiegelt sich der Prozess der Digitalisierung und Globalisierung in besonders deutlicher Form wider und veranschaulicht die Bedeutung des technischen Wandels für den Umgang der Menschen mit- und untereinander.

Das Bedürfnis der Menschen, unter Verwendung von Spielen sozial miteinander zu interagieren, ist bis in die Antike hinein nachvollziehbar und in allen vergangenen und existierenden Kulturen wiederzufinden. Bereits vor mehr als 5000 Jahren erfanden die Ägypter das Brett- und Gesellschaftsspiel *Senet*, das als gut erhaltene und spielbare Grabbeigabe des berühmten Tutanchamun zu finden war [Hageman, 2005, S. 5]. Die mesoamerikanischen Völker waren für ihr Ballspiel bekannt und die teils weitläufig dafür angelegten Anlagen sind auch heute noch frei begehbar [Foster u. Mathews, 2005, S. 195ff.]. Im heutigen Zeitalter dominieren[74] Computer- und Videospiele den weltweiten Spielemarkt und sind als solche in der Mitte der Gesellschaft angekommen[75]. Eine im Jahre 2013 veröffentlichte Studie des Bundesverband Interaktive Unterhaltungssoftware e.V. (BIU) hat ergeben, dass mehr als jeder dritte Bundesbürger am Computer spielt - in der Altersgruppe bis zu dreißig Jahren sind es sogar acht von zehn Menschen [Brehm, 2013]. Wie in Abbildung 5.1 deutlich wird, sind derzeit[76] in Spitzenzeiten knapp 6 Millionen Spieler und Spielerinnen weltweit gleichzeitig auf der größten[77] Vertriebsplattform für Computerspiele, *Steam*, online.

Da diese Definition (zum Zeitpunkt dieser Arbeit) recht neu ist, hat sie noch keinen signifikanten Einzug in die neuere Literatur und Forschung erhalten, bietet jedoch eine fundierte Neubewertung. Im späteren Verlauf dieser Arbeit wird gezeigt werden, dass insbesondere der Aspekt eines quantifizierbaren Ziels, welches als geforderte Eigenschaft in vielen Definitionen enthalten ist, für die Betrachtung des Kernthemas dieser Arbeit von Bedeutung ist, da die im Titel genannten Pen-&-Paper-Rollenspiel hier eine Sonderrolle einnehmen und sich den gängigen Definitionen entziehen (vgl. auch Abschnitt 8.0.1 auf Seite 156).

[72] Johann Huizinga übernahm diesen Aspekt sogar in seine 1938 verfasste Definition des Begriffes „Spielen“ und stellt das Element der Bildung sozialer Gruppen explizit heraus:

> [Play is] *„ a free activity standing quite consciously outside 'ordinary' life as being 'not serious' but at the same time absorbing the player intensely and utterly. It is an activity connected with no material interest, and no profit can be gained by it. It proceeds within its own proper boundaries of time and space according to fixed rules and in an orderly manner. It promotes the formation of social groupings that tend to surround themselves with secrecy and to stress the difference from the common world by disguise or other means.“*

[Huizinga, 1955, S. 13] [Salen u. Zimmerman, 2003, S. 75].

[73] Dennoch ist diese Möglichkeit groß genug, um ganze Bücher darüber zu verfassen, wie Spiele (in diesem konkreten Fall: *Rollenspiele*) als soziale Welten aufgefasst werden können (siehe Fine - *Shared Fantasy: Role Playing Games as Social Worlds* [Fine, 2002]).

[74] Die Entwicklung des Computerspiel-Titels *GTA V* beispielsweise kostete umgerechnet etwa 200 Millionen Euro und erwirtschaftete am ersten Verkaufstag einen Umsatz von 800 Millionen Euro [Kuri, 2013b].

[75] Eine vergleichende Schaugrafik zur historischen Entwicklung der Spiele befindet sich im Anhang unter B.1. Eine Veranschaulichung der Statistiken des weltweiten Videospielmarkts ist ebenfalls unter [Wikia.com, 2013] zu finden.

[76] zur Entstehungszeit dieser Arbeit

[77] Nach Schätzungen von *Forbes* besitzt *Steam* 50%-70% Marktanteil im Jahr 2011 [Chiang, 2011].

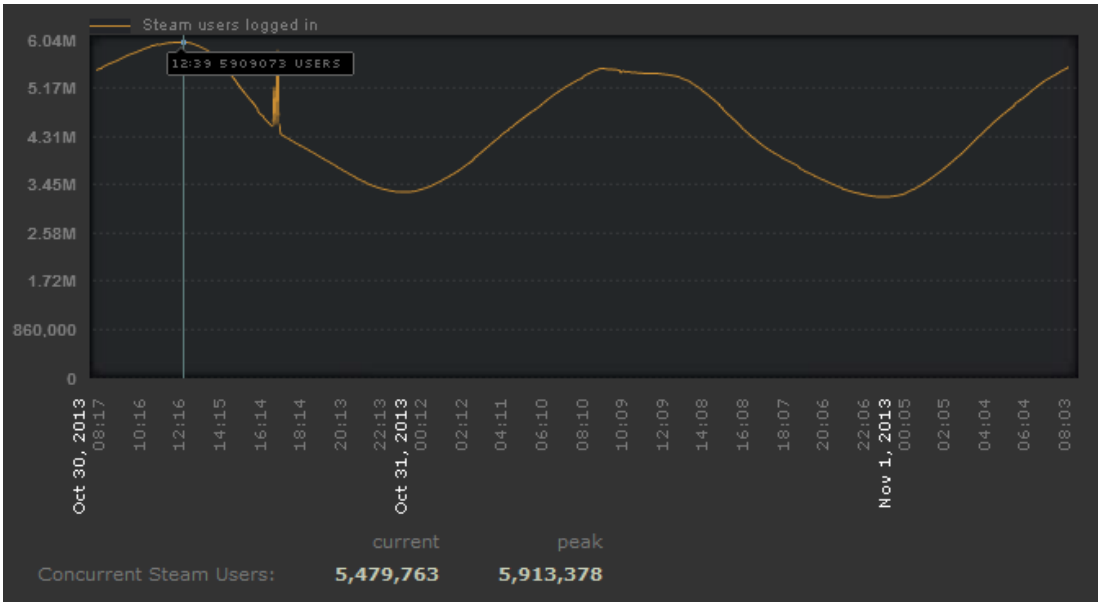


Abbildung 5.1.: Anzahl der weltweit aktiven Spieler bei Steam am 01. November 2013.

aus [Valve Corporation, 2013c]

Insbesondere an diesem Beispiel zeigt sich, dass der Zugang zu tausenden von Spielen[78] mit fortschreitender Technik immer stärker vereinfacht wird. So ist es jetzt schon Standard, dass Millionen von Menschen rund um den Globus[79] miteinander unbekannte Welten erforschen oder ein Kräftemessen in kompetitiver Manier veranstalten. Die räumliche Distanz zwischen einer beliebig großen Anzahl an Teilnehmern wird dadurch schließlich immer weniger greifbar und verliert an Relevanz.

Letztlich entsteht durch diese weltweite Vernetzung aufgrund gemeinsamer Interessen, in diesem Fall beispielsweise das Interesse an einem bestimmten Spiel, ein sozialer Raum oder eine *Sphäre*. Begrifflich ist hierunter kein tatsächlich *physikalisch* vorhandener Ort zu verstehen, sondern ein temporärer Interessenverband, welcher durch die Kommunikation und soziale Interaktion der Gruppenmitglieder in seiner Gestalt erst definiert wird [Dringenberg u. a., 2002, S. 33]. Ähnlich wie in der realen Welt entstehen in solchen sozialen Räume Beziehungsgeflechte der Menschen untereinander. Dies geschieht oftmals sogar schneller als im echten Leben, da die Kommunikation leichter fällt und Barrieren wegfallen oder nicht zur Geltung kommen [Dringenberg u. a., 2002, S. 37].

[78] Alleine *Steam* bietet über 2000 verschiedene Spiele für unterschiedliche Plattformen an [Valve Corporation, 2013d].
[79] Eine Schaugrafik zur Visualisierung befindet sich im Anhang unter Abbildung B.2 auf Seite xix.

Diesen positiven Aspekten steht gegenüber, dass die rein digital existierenden Interessenräume schnell verfallen können. Insbesondere soziale Beziehungen oder Bekanntschaften lassen sich durch einen Klick herstellen oder auch beenden: Menschen werden dadurch füreinander parzellierbar[80]. Genau an diesem Kritikpunkt setzt die vorliegende Arbeit an: Die Vorteile und das soziale Gefüge eines von Angesicht zu Angesicht gespielten Spieles sollen beibehalten und mit modernen, digitalen Elementen bereichert und erweitert werden.

5.1. Zielsetzung und Motivation der Arbeit

Wie kann eine Möglichkeit geschaffen werden, Menschen digital miteinander spielen zu lassen, ohne ihr soziales Gefüge damit zu stören? Dieser Kerngedanke soll innerhalb dieser Arbeit und der damit einhergehenden Implementierung beantwortet werden. Insbesondere die ständig wachsende Bedeutung sozialer Netzwerke und die steigende Verbreitung von Smartphones zeigen, dass die Digitalisierung von sozialer Interaktion einen kontinuierlich wachsenden Stellenwert im Leben vieler Menschen einnimmt[81] [Beyer, 2010, S. 56ff]. Zunehmend findet hierbei nicht nur eine Unterstützung realer Interaktion durch Technik, sondern teilweise eine vollständige Substitution statt. Ziel der Arbeit ist es jedoch nicht, diese Entwicklung aufzuhalten oder umzukehren, sondern genau an diesem Punkt aufzugreifen und weiterzuführen: Technik soll keine Barriere darstellen, welche eine soziale Interaktion verhindert, sondern diese unterstützen und fördern.

Das Ziel dieser Arbeit wird daher definiert als: Die Konzeption und Umsetzung einer MR-Anwendung, bei der die reale Interaktion zwischen den Menschen erhalten bleibt. Als inhaltliche Thematik der Anwendung wird hierbei aus mehreren Gründen das Konzept eines *Pen-&-Paper-Rollenspiels* gewählt, welches im Folgenden im Rahmen der Motivation näher beschrieben wird.

5.1.1. Motivation der Themenwahl

Wenn Spiele als Sonderform der sozialen Interaktion betrachtet werden, nehmen Pen-&-Paper-Rollenspiele innerhalb dieser Sonderform eine weitere Unterrolle ein, in der Hinsicht, dass hierbei soziale Interaktion sowohl zum Instrument als auch zum Inhalt des Spiels wird. Mit Bezug auf den vorhergehenden Abschnitt[82] wurde bereits erläutert, dass Menschen durch ein gemeinsames Ziel, eine gemeinsame Tätigkeit oder ein gemeinsames Interesse einen sozialen Raum bilden. Das Konzept eines

[80] Die Einteilung von Menschen in Gruppen wird beim sozialen Netzwerk *Google+* explizit als Feature genannt und umworben. Zu einem späteren Zeitpunkt wurde dieses Konzept ebenfalls durch das soziale Netzwerk *Facebook* adaptiert [Lippert, 2011].
[81] Dies geht sogar bis hin zum Zwang, ständig erreichbar zu sein [Beyer, 2010, S. 56ff].
[82] vgl. hierzu Seite 103

Pen-&-Paper-Rollenspiels sieht hierbei eine Forcierung eines solchen Raumes vor, bei dem die Teilnehmenden bewusst miteinander in Interaktion treten, um eine gemeinsame soziale Welt zu erschaffen. Diese Welt wird innerhalb eines Pen-&-Paper-Rollenspiels lediglich durch ein so genanntes *Regelwerk* eingegrenzt. Im Gegensatz zu klassischen Gesellschafts- respektive Brettspielen[83] weist diese Spielart keinen kompetitiven Charakter auf. Die Spielenden definieren den Sinn und Zweck des Spiels dabei selbst und entwickeln eine Geschichte durch gemeinsame Entscheidungen und Aktionen in einem kontinuierlichem Kommunikationskreislauf mit einem so genannten Spielleiter/einer Spielleiterin[84]. Folgesituationen werden hierbei immer anhand der Situationen definiert, welche vorher durch die Entscheidungen der Spielenden herbeigeführt wurden. In Tolstois Novelle werden Spiele von der Realität abgegrenzt[85]. Rollenspiele hingegen zeigen eine Durchmischung von Realität und Spiel und entziehen sich einer festen Definition. Innerhalb ihrer Grenzen verfügen Pen-&-Paper-Rollenspiel jedoch über eine feste Struktur mit definierten Regeln, bei gleichzeitig maximaler Freiheit in der tatsächlichen Umsetzung. Durch diese Rahmenbedingungen und Eigenschaften ist die Gattung der Pen-&-Paper-Rollenspiele prädestiniert zur Entwicklung eines Konzepts, welches die Vereinigung eines realen und eines virtuellen Raums anstrebt.

5.1.1.1. Exkurs: Das Pen-&-Paper-Rollenspiel

Pen-&-Paper-Rollenspiele unterschieden sich in vielerlei Hinsicht von anderen Spielarten und sollen zum besseren Verständnis an dieser Stelle kurz erläutert werden. Wie im späteren Verlauf dieser Arbeit deutlich wird, entziehen sich Rollenspiele im Allgemeinen der in diesem Kapitel vorgestellten Definition respektive werden nicht vollständig von ihr abgedeckt, insbesondere in der Hinsicht, dass beim Rollenspiel nicht auf ein bestimmtes Ziel hingearbeitet wird. Im Rahmen dieser Arbeit soll daher eine neue Definition gelten, wobei auf den Vorschlag von Schick zurückgegriffen wird:

> *„A role-playing game must consist of quantified, interactive storytelling."*[86]

[Schick, 1991, S. 10].

Auffallend ist hier die explizite Nennung des „interactive storytelling", welches auch den eigentlichen Inhalt des Spieles ausmacht. Die Spielenden führen durch ihre Handlungen Entscheidungen herbei („interactive") und definieren auf diese Weise eine individuelle Folgesituation und damit zusammenhängend eine Geschichte („storytelling"). Die Spiele laufen dabei in der Phantasie der Teilnehmenden

[83] Wie beispielsweise „Monopoly" oder „Mensch ärgere Dich nicht" im deutschsprachigem Raum.
[84] Im weiteren Verlauf dieser Arbeit wird der englische und geläufige Begriff engl. GM verwendet [Schick, 1991, S. 10].
[85] vgl. Seite 101
[86] Diese Attribuierung wird von Schick selbst als „working definition" bezeichnet [Schick, 1991, S. 10]. Eine differenziertere Betrachtung dieser Definition, auch im Zusammenhang mit der generalisierten Definition des Begriffes „Spiel", ist im Abschnitt 8.0.1 auf Seite 156 zu finden.

ab, wobei eine Ausgangssituation im Vorhinein durch den bereits erwähnten Game Master festgelegt wird. Der Game Master dient als Schnittstelle zwischen den Spielenden und der Welt, mit welcher interagiert wird, wobei er hierbei an ein sogenanntes Regelwerk gebunden ist. In diesem Regelwerk wird festgelegt, was im Spiel tatsächlich möglich ist („quantified“) und dient sowohl als Nachschlagewerk als auch Richtlinie für den Game Master, wohingegen die Spielenden das Regelwerk nicht kennen müssen.

Plastisch konstruiert sähe eine konkrete Situation beispielsweise folgendermaßen aus: Die Spielenden sitzen gemeinsam an einem Tisch, wobei dieser meist mit diversen unterstützenden Spielmaterialien gefüllt wird, unter anderem Würfel und/oder Karten der Spielwelt. Nun beschreibt der Game Master eine Situation:

> *„Ihre Gruppe steht vor einem großen, dunklen Wald. Ihr Ziel liegt hinter diesem Wald. Der Wald ist bekannt für seine Waldtrolle[87], welche in ihrer Heimat nicht gestört werden wollen. Die einzige andere Möglichkeit hindurch wären die Höhlen, die unter diesem Wald hindurch führen, jedoch werden diese von riesigen Höhlenspinnen bevölkert. Was gedenken Sie zu tun?“*

Die „Gruppe“ bezeichnet in diesem Fall den Zusammenschluss der Spielenden innerhalb der Phantasiewelt. Sie besteht dabei aus den Alter Egos der Teilnehmenden, beispielsweise könnte jemand in die Rolle eines Zwergs schlüpfen, während eine weitere Teilnehmerin eine heldenhafte Amazone spielt. Im Folgenden muss sich die Gruppe beraten, wie weiter zu verfahren ist. Der Spieler, der den Zwerg spielt, könnte seinen Charakter beispielsweise vorschlagen lassen, den Weg durch die Höhlen zu nehmen, da sich Zwerge unter Tage gut zurecht finden. Die Spielerin der Amazone könnte hingegen die Ansicht vertreten, dass der Weg durch den Wald die bessere Alternative darstellt, weil sich hier notfalls mehr Fluchtmöglichkeiten bieten. Vielleicht führt die Gruppe auch einen Zauberer mit, welcher noch einen ganz anderen Vorschlag mit einbringt. In jedem Fall wird in gemeinsamer Absprache eine Entscheidung getroffen, welche dem Game Master im Anschluss mitgeteilt wird. Dieser wertet die Entscheidung der Spielenden aus und konstruiert daraus eine Folgesituation. Meistens erhalten die Spielenden nach jeder Entscheidung oder Interaktion Erfahrungspunkte, mit deren Hilfe die Charaktere gewissermaßen evolviert werden können. Auf diese Weise entsteht ein zusätzlicher Anreiz, die Geschichte weiter zu erspielen, da sich die Spielenden kontinuierlich verbessern können.

[87] Dieses Beispiel ist an ein Fantasy-Universum angelegt, in dem Pen-&-Paper-Rollenspiele auch ihren Ursprung haben. Spieler und Spielerinnen können darin die Rollen von Zwergen, Zauberern und dergleichen annehmen, während auch alle Wesen und Gegebenheiten existieren können, die durch das Regelwerk zulässig sind. Das erste Beispiel einer derartigen klassischen Fantasy-Welt ist wäre *Dungeons and Dragons*, welches 1974 erschien und noch bis heute weiterentwickelt wird [Wizards of the Coast, Inc., 2003].

5.2. Stand der Wissenschaft und Forschung

Die vorliegende Arbeit setzt auf einem von den Autoren durchgeführten Projekt auf, welches im Rahmen des Masterstudiums zweckgebunden als Vorarbeit konzipiert wurde. Dieses Projekt trägt den gleichnamigen Titel dieser Ausarbeitung, konzentriert sich jedoch ausschließlich auf die rein technischen Aspekte, um eine Grundlage für die höhere Anwendungsentwicklung zu erschaffen, welche erst in dieser Arbeit durchgeführt wird. Die bereits implementierte Anwendung wird im Folgenden beschrieben, um daraufhin die Unterschiede und Ziele herauszuarbeiten, welche bisher nicht realisiert wurden und in dieser Arbeit eine Umsetzung erfahren sollen. Dem schließt sich eine kurze Übersicht vergleichbarer Arbeiten und Konzepte an.

5.2.1. Hintergrund und zur Verfügung stehende Basis

Im Rahmen des vorangegangenen Projektes wurde ein Prototyp konzeptioniert und implementiert, bei dem im Sinne der Zugänglichkeit die eingesetzten Technologien nicht als zusätzliche Barriere, sondern als Unterstützung im Sinne einer Mixed Reality-Anwendung wahrgenommen werden. Die Grundidee war hierbei, das System als Client-Server-Struktur zu realisieren, bei der die leichtgewichtigen Clients vornehmlich für die Eingabe und die Ergebnisrückführung zuständig sind, während ein leistungsstarker Server für die ressourcenintensive Koordination des Systems sorgt. Zudem verfügt die Anwendung über eine moderne grafische Ausgabe[88], welche ebenfalls als zusätzlicher Client auf der Server-Maschine realisiert wurde.

Das erarbeitete Resultat kapselt hierbei technische Lösungen und Konzepte, welche in dieser Art noch nicht in kommerziellen Produkten und ähnlichen Ansätzen aus Wissenschaft und Forschung realisiert wurden. Zu diesen Lösungen gehört, dass die Eingabegeräte gleichzeitig als Ausgabegeräte konzipiert wurden, um das Spielerlebnis und somit das Aufmerksamkeitsfeld des Nutzers vom zentralen Ausgabemedium weg und hin zu den Spielenden verschieben zu können. Durch ein solches Aufspannen eines Ereignishorizontes zwischen Spielausgabe und Nutzern konnte eine Diffusion zwischen Realität und virtueller Welt auf technischer Seite vorbereitet werden, wobei die generelle Funktionsfähigkeit dieser technischen Lösung anhand einer prototypischen Beispielsituation gezeigt wurde. Der schematische Aufbau des realisierten Systems ist dabei in Abbildung 5.2 festgehalten.[89]

Das zentrale Spielgeschehen wird hierbei auf einem gemeinsamen Ausgabemedium visualisiert, während die Spielenden ihre Eingaben unter Zuhilfenahme von Smartphones tätigen können. Die Wahl auf Smartphones fiel vor allem aus dem Grund, weil die Teilnehmenden damit nicht nur in der Lage

[88] Für die Darstellung der Spielwelt wurde die Unreal-Engine beziehungsweise das UDK gewählt.
[89] siehe Abschnitt 1.3

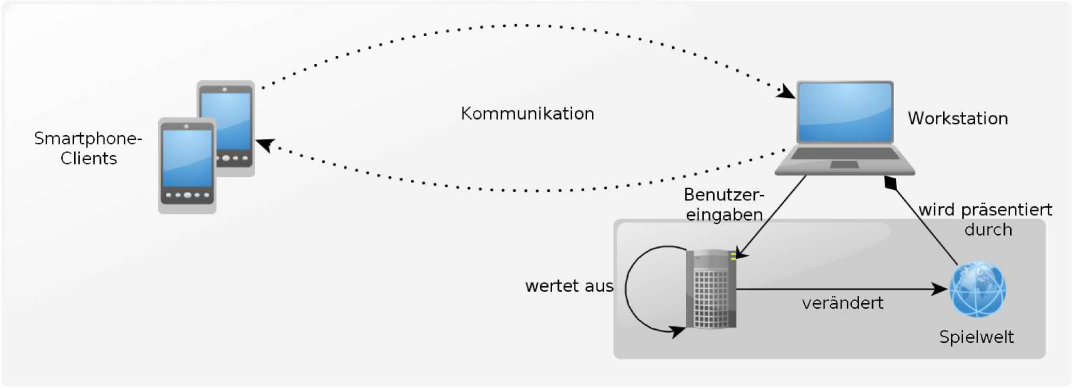


Abbildung 5.2.: Die Abbildung zeigt eine grobe Skizze des im Masterprojekt realisierten Systems. Die Spielenden sind mit Hilfe von Eingabegeräten in Form von Smartphones drahtlos mit einer Workstation verbunden. Diese Workstation beinhaltet die Logik zur Koordination des Netzwerkverkehrs, zur Visualisierung und Auswertung der Spielsituation.

sind, auf mehrere Weisen Eingaben zu tätigen, sondern weil es ebenso möglich ist, Ausgaben in einem zirkulärem Prozess direkt zu den Nutzern zurück zu führen[90]. Ferner lässt sich die berührungsempfindliche Oberfläche für *direkte* Interaktionen mit Objekten nutzen, da diese auf dem Display unmittelbar berühr- und manipulierbar sind, im Gegensatz zu anderen Eingabegeräten, welche hierfür zusätzliche Mechaniken benötigen und dadurch lediglich eine indirekte Interaktion zulassen [Hao Chuah u. Lok, 2012b; Forlines u. a., 2007].

Wie in der Abbildung 5.2 ebenfalls zu erkennen ist, besteht die Anwendung dabei aus einem geschlossenen *Action-Response-Circle*, bei dem die Eingaben der Smartphones von einem Server entgegengenommen werden. Dieser wertet die Benutzereingaben aus und gibt die Ergebnisse an die Spielwelt weiter, welche wiederum von den Spielenden rezipiert wird. Auf der anderen Seite informiert die Spielwelt respektive die zugrundeliegende Game-Engine einen als Server-Modul realisierten digitalen Game Master über Veränderungen, wie beispielsweise das Auslösen eines Ereignisses[91].

[90] siehe Kapitel 4

[91] Zum besseren Verständnis sei eine Beispielsituation gegeben: Die Spielenden steuern die digitalen Repräsentationen ihrer Charaktere mit Hilfe von Eingaben, welche über die Smartphones getätigt werden. Diese Steuerbefehle werden kabellos an einen Server übertragen. Anschließend werden diese Signale aufbereitet und an den darstellenden Client weiterleitet, welcher ebenfalls als Anwendung auf der Servermaschine implementiert ist. Dieser visualisierende Client setzt die Steuerbefehle dann in grafische Ausgaben um, beispielsweise in Positionsveränderungen der Avatare. Auf der anderen Seite kann die Game-Engine als Ereignis wahrnehmen, wenn ein Avatar in einen Trigger läuft. Dieses Ereignis wird dem Server mitgeteilt, welcher hierdurch einen Entscheidungsprozess anstößt, wie mit dieser Information umgegangen werden soll und ob dadurch weitere Ereignisse angestoßen werden.

Dieser Game Master beinhaltet als Kernmodul die technischen Vorbereitungen für eine so genannte „DE“. Im Rahmen des Masterprojekts konnte die generelle Funktionsfähigkeit einer Decision Engine in einem Anwendungsfall mit simplifizierten Verzweigungen gezeigt werden. Es wurde ferner vorgesehen, dass der Game Master unter Zuhilfenahme eines Remote Procedure Call (RPC) in der Lage ist, errechnete Veränderungen direkt per Push-Nachricht an die Game-Engine zu senden und somit die Spielsituation zu verändern und anzupassen.[92]

(a) Gameplay vor der Interaktion

(b) Gameplay nach der Interaktion

Abbildung 5.3.: In der linken Abbildung (5.3(a)) ist die Interaktions-Ansicht auf den Smartphone-Clients zu sehen. Auf dem Monitor dahinter sind die Avatare der Spielenden dargestellt, welche vor einer Tür stehen. Nach der Ausführung einer Wischgeste öffnet sich diese Tür in der Spielwelt und gibt in diesem Fall die dahinter liegende Wasserwelt frei (zu sehen in Abbildung 5.3(b)).

Schon beim Masterprojekt war es ausdrückliches Ziel, das System technisch so auszulegen, dass die soziale, reale Interaktion der Partizipierenden erhalten bleibt und zusätzlich gefördert wird. Ein Ansatz hierfür wurde realisiert, indem eine virtuelle Aktion durch eine gemeinsame Handlung in der Realität angestoßen werden kann. Eine Erläuterung hierzu ist in Abbildung 5.3 zu finden.[93]

Bei diesem Beispiel handelt es sich um eine virtuelle „schwere Stahltür“, welche aufgeschoben werden muss, um den nächsten Spielabschnitt betreten zu können. In der Realität müssen die Spielenden dazu einen Zeitpunkt absprechen, an dem die Aktion beginnen soll, um das Aufschieben anschließend durch eine Wischgeste auf ihren Smartphones simulieren zu können. Auf diese Weise wird einerseits eine reale Kommunikation gefördert und andererseits haben Aktionen der realen Welt Auswirkungen auf ihr digitales Pendant.[94]

[92] siehe Kapitel 4
[93] siehe Kapitel 4
[94] siehe Abschnitte 3.2.6 und 4

5.2.2. Ausgangspunkt und Ziele der Masterarbeit

Wie im vorherigen Abschnitt erläutert wurde, ist im Rahmen des Masterprojekts eine technologische Grundlage geschaffen worden, welche es erlaubt, Pen-&-Paper-Rollenspiele vollständig zu digitalisieren. An dieser Stelle setzt die vorliegende Arbeit an, wobei der Fokus auf die im Folgenden näher erläuterten Aspekte gelegt wird.

5.2.2.1. Umsetzung einer Mixed Reality-Anwendung

Die Implementierung im Masterprojekt war hart kodiert auf zwei Spielende begrenzt. In einem ersten Schritt wird diese Beschränkung auf zwei Teilnehmende aufgehoben und modular erweitert, sodass eine flexible Anzahl an Teilnehmerinnen und Teilnehmern ermöglicht wird. Das Augenmerk liegt hierbei auf dem Anspruch einen Interaktionsraum zwischen der digitalen Ausgabe und den Partizipierenden aufzuspannen, um auf diese Weise sowohl eine Intensivierung der Interaktion zwischen den Gruppenmitgliedern, als auch eine erhöhte Diffusion von realer und virtueller Welt zu erreichen. Eine Möglichkeit für solch eine Intensivierung wäre beispielsweise die Auswertung realer Geräusche, welche durch die Mikrofone der Smartphones eingefangen werden und Auswirkungen in der virtuellen Welt nach sich ziehen. Auf der anderen Seite wäre es denkbar, dass in der digitalen Welt befindliche Geräusche auf die Smartphones übertragen werden, um so ein räumliches Klangbild zu erzeugen.[95]

Die theoretisch herausgearbeiteten Ergebnisse der Interaktionsforschung werden in einem weiteren Schritt prototypisch implementiert und im Rahmen einer Evaluation eruiert. Hierbei ist von besonderem Interesse, wie sich die Einbringungen von Mixed Reality-Aspekten auf das soziale Gefüge der Partizipierenden auswirkt. Mit Hilfe einer vergleichenden Evaluation sollen verschiedene Aspekte überprüft werden, wie beispielsweise der empfundene Spaßfaktor und das Kommunikationsverhalten. Dazu werden zum einen Fragebögen erstellt, die das subjektive Spielerlebnis ermitteln sollen und zum anderen wird ein objektiver Beobachter eingesetzt, um die Ergebnisse der Fragebögen zu stützen, zu widerlegen oder zu ergänzen.

5.2.2.2. Modulares Storytelling

Modulares Storytelling ist eine mögliche Ausprägung von Erzählformen, bei der in sich abgeschlossene Module in unabhängiger Reihenfolge in Korrelation gesetzt werden können und auf diese Weise einen größeren Sinnzusammenhang bilden um einen nichtlinearen Narrationsaufbau zu ermöglichen[96] [Sheldon, 2004, S. 311]. Derartige Module bestehen aus vorgefertigten Abschnitten, welche,

[95] vgl. Kapitel 1
[96] Wie es insbesondere von Pen-&-Paper-Rollenspielen vorgesehen wird.

abhängig von den Entscheidungen der Rezipienten, individuell ineinander verzahnt werden können. Auf diese Weise können beispielsweise die Spielenden in Echtzeit Einfluss auf den weiteren Verlauf und Aufbau des Geschehens in einem Computerspiel nehmen.

Eine Variante des nichtlinearen Storytellings ist beispielsweise die *Hypermedia Novel*[97], bei der eine zu präsentierende Geschichte ebenfalls in mehrere narrative Module unterteilt wird (vgl. [Heiden u. a., 2001, S. 346]).

Innerhalb des Masterprojekts wurde mit Auswahl des UDK eine technische Grundlage ausgewählt, die ein modulares Storytelling ermöglicht. Die ausgewählte Game-Engine erlaubt es, durch sogenanntes „Level-Streaming“ Szenen oder Abschnitte kaskadisch ineinander zu verschachteln, dynamisch zu erzeugen oder auch zu entfernen. Auf diese Weise wird ein skalierender Aufbau ermöglicht, welcher sich der jeweiligen Spiel- und Entscheidungssituation individuell anzupassen vermag.

Im Rahmen der vorliegenden Arbeit soll dieses Konzept aufgegriffen und weiterverfolgt werden, indem eine kurze Geschichte tatsächlich implementiert wird, um die Frage zu beantworten, auf welche Art und Weise sowohl Aufbau der Geschichte als auch die Verschachtlung von Story-Modulen in der Praxis umsetzbar sind. Konkretisiert geht es dabei um den Aspekt, in welcher Art und Weise schon durch Wahl und Aufbau einer Geschichte Grundlagen für ein kollaboratives Verhalten im Sinne einer MR-Anwendung gelegt werden können. Insbesondere die Aussage von Sheldon:

> *„No other structure is better at integrating story and gameplay than modular“*

[Sheldon, 2004, S. 313]

dient dieser Betrachtung als Leitfrage, welche es zu beantworten gilt.

5.2.2.3. Verfeinerung des Netzwerkmodells

Durch den Umbau und die Erweiterung der bestehenden Anwendung auf mehr als zwei gleichzeitig spielende Menschen erhöhen sich die Anforderungen an eine effiziente Netzwerkkommunikation. Diese Anforderungen ergeben sich aus einem zu erwartenden erhöhten Netzwerkdurchsatz, verbunden mit einem verstärkten Fokus auf Interaktionen, welche zusätzliche Bandbreite und Rechenleistung beanspruchen. Ebenfalls wurde das derzeit verwendete Netzwerkprotokoll zwar erfolgreich zur Anwendung gebracht, jedoch bislang nicht ausführlich evaluiert und einem Stresstest im Bezug auf Stabilität und Latenzfreiheit unterzogen. Diesen Punkt gilt es daher zusätzlich zu berücksichtigen.

[97] Eine genauere Betrachtung der Hypermedia Novel im Kontext weiterer Erzählformen wird in Abschnitt 8.2.3.1 auf Seite 166 vorgenommen.

Das bestehende Netzwerkmodell muss weiterhin in der Art erweitert werden, dass die als Server-Modul realisierte Entscheidungsmaschine flexibel auf komplexe Interaktionsformen reagieren kann. Die hierfür notwendigen Daten müssen in einer sparsamen und effizienten Weise übertragen werden, wozu sowohl ein kompaktes sowie flexibles Datenformat als auch Kommunikationsmodell herausgearbeitet werden muss.

5.2.2.4. Entscheidungsmaschine

Eine Entscheidungsmaschine, auch „Decision Engine“ genannt, ist Grundlage für die Umsetzung einer modular verschachtelten Geschichte. Durch diese Entscheidungsmaschine werden alle für eine Entscheidungsfindung relevanten Variablen miteinander in Korrelation gesetzt, um das jeweils nächste Story-Modul respektive den damit einhergehenden Spiel-Abschnitt zu generieren. Bei Pen-&-Paper-Rollenspielen wird diese Rolle von einem realen Game Master übernommen, welcher die Entscheidungen und Absprachen der Spielenden auswertet und in neue Spielsituationen überführt. Bei einer Digitalisierung müssen zur Verfügung stehende Informationen jedoch sinnvoll quantisiert werden, um zur Entscheidungsfindung nicht auf (zusätzlichen) menschlichen Input angewiesen zu sein.

Aus diesem Grund werden verschiedene Lösungsansätze gegenübergestellt, welche die Komplexität einer solchen Entscheidungsmaschine beherrschbar machen. Dafür werden verschiedene Ansätze miteinander verglichen, beispielsweise das Konzept eines Entscheidungsbaums und das eines endlichen Automaten. Die Implementierung erfolgt hierbei als Teil des Game Masters, wobei die Entscheidungsmaschine nur zur Entscheidungsfindung zuständig ist.

5.2.3. Vergleichbare Arbeiten

Wie Menschen miteinander spielen, ohne dabei eine Separation der Teilnehmenden in räumlicher oder sozialer Hinsicht zu protegieren, ist Gegenstand vergangener und aktueller Forschungen und Projekte. An dieser Stelle soll lediglich eine grobe Übersicht gegeben werden, wohingegen das Kapitel „Related Work“[98] eine eingehendere Auseinandersetzung mit den hier angerissenen Themen beinhaltet. Zudem wird an geeigneter Stelle stärker ausdifferenziert, welche Konzepte in Hinsicht auf Interaktionen in realer und digitaler Form bestehen und welche Ansätze im Rahmen von MR-Anwendungen erforscht wurden[99]. Ein Spezialfall bietet sicherlich die Forschung und bestehende oder laufende Konzepte im Bezug auf die Digitalisierung von Pen-&-Paper-Rollenspielen.

[98] siehe Seite 215ff
[99] vgl. hierzu Abschnitt 7.2ff.

Ein generalisierender Einstieg in die Thematik kann mit der Funktionsweise und verschiedenen Anwendungsformen der Videospielkonsolen gefunden werden[100]. Videospielkonsolen, insbesondere in ihrer rezenten Form, bieten ein Konzept, welches soziales Spielen fördert oder technisch ermöglicht. Im Gegensatz zum PC oder auch zu Smartphones und Tablets, welche ebenfalls zum Spielen verwendet werden können, ist das Spielen auf Konsolen das prägende Charakteristikum für die Existenz derselben[101]. Dies schlägt sich unter anderem dadurch nieder, dass Konsolen einen gleichzeitigen Anschluss mehrerer Eingabegeräte („Controller“) erlauben, um ein gemeinsames Spielen vor einem zentralen Ausgabemedium zu ermöglichen. Durch weitere Eingabeformen, wie Bewegungssensoren oder die Anbindung und Erweiterung auf mehrere Aus- und Eingabequellen, wird ein Einsatz für mehrere Spielende bewusst gefördert und forciert[102].

Mit Erscheinen der Videospielkonsole *Wii* des Herstellers *Nintendo* wurden erstmals zwei neue Konzepte und somit Nutzungsszenarien für Videospielkonsolen präsentiert: Zum einen ist damit das Spielen als soziales Ereignis für die gesamte Familie möglich und zum anderen sind die Spielerinnen und Spieler nicht mehr statisch an ihre Controller gebunden, sondern können Interaktionen per Bewegungssteuerung im realen Raum ausführen [Sheldon, 2013, S. iv]. Der Anspruch an Interaktivität und die Bedeutung derselben, nicht nur auf Software-Ebene, wurde hierbei explizit hervorgehoben [Casey, 2006]. Mit dem Nachfolger dieser Konsole, der *Wii U*, wird versucht, dieses Konzept weiterzuführen. Hierfür tritt unter anderem ein mit Touchscreen ausgestatteter Controller als neue Eingabevariante in Erscheinung. Auf diese Weise kommt das Bedienkonzept der *Wii U* dem Ansatz dieser Ausarbeitung recht nahe[103]. Im Gegensatz zu einer Verwendung der technischen Möglichkeiten zur Erweiterung des Realitätsraums zwischen Medium und den Spielenden liegt der Fokus hierbei jedoch vornehmlich auf der Vereinfachung des sozialen Kontakts über weite Distanzen, beispielsweise durch Videotelefonie-Funktionen [Humphries, 2012].

Durch den Systemaufbau von Videospielkonsolen und den damit einhergehenden Aufbau kann ebenfalls eine Analogie zum Pen-&-Paper-Rollenspiel gezogen werden[104]. Eine erläuternde Veranschaulichung hierzu ist in Abbildung 5.4 schematisch festgehalten.

Mit Bezug auf das erläuterte Konzept des Pen-&-Paper-Rollenspiels[105] kann folgende Analogie gezogen werden: Der Game Master, mitsamt der damit zusammenhängenden Kenntnis des Regelwerks, wird durch die dem Spiel innewohnende Logik substituiert, welche durch die Konsole berechnet wird. Die Eingaben und Entscheidungen der Spielenden werden mit Hilfe von Controllern getätigt, während das

[100] vgl. auch Abschnitt 1.3
[101] Obgleich moderne Konsolen inzwischen für Einsatzszenarien verwendet werden können, welche über das reine Spielen hinaus gehen, wie beispielsweise die Verwendung als Browser oder Medienzentrale [Greif, 2013].
[102] vgl. Abschnitt 6.3
[103] vgl. hierzu Abschnitt 1.3.
[104] siehe Abschnitt 1.3.1
[105] vgl. Seite 105

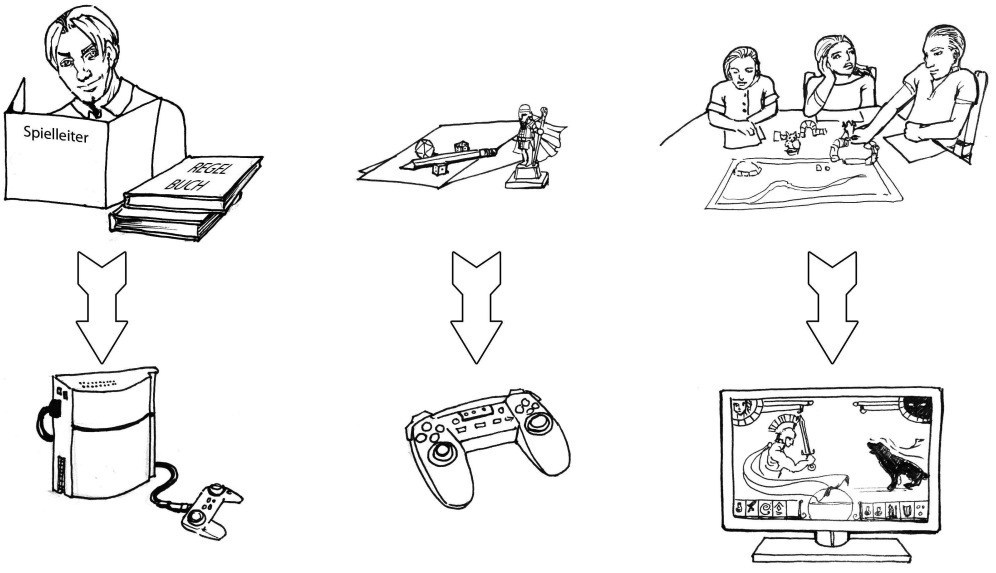


Abbildung 5.4.: Durch die Abbildung werden die Parallelen zwischen klassischem Pen-&-Paper-Rollenspiel und modernen Videospielkonsolen veranschaulicht, ein Konzept, welches dem in dieser Arbeit angestrebten Aufbau in Teilen ähnelt. Die Funktion des Game Masters (linker Teil der Abbildung) und des Regelwerks werden durch die Konsole und die ihr innewohnende Logik substituiert; die Software des jeweiligen Spiels enthält somit u.a. ein digitalisiertes Regelwerk und eine komplexe Übertragungsfunktion (KI). Eingabe- und Hilfsmittel wie Stifte, Würfel und Papiere fallen weg, da sämtliche Eingaben durch die angeschlossenen elektronischen Eingabegeräte, in diesem Fall Controller, getätigt werden (mittlerer Teil der Abbildung). Die eigentliche Spielsituation, bei der die Spielenden den Fortgang der Geschichte meistens in der Phantasie ausgestalten (allenfalls kommen auch hier Hilfsmittel wie Karten oder Spielfiguren zum Einsatz) wird direkt visualisiert, wobei die Visualisierung auf einem zentralen Ausgabemedium erfolgt und ebenfalls durch die Konsole berechnet wird (rechter Teil der Abbildung).

Konzept und Design der Autoren, Ausarbeitung i.A. Ulrike Kleinert [Kleinert, 2014].

Ergebnis der Entscheidungen unmittelbar auf einem zentralen Ausgabemedium visualisiert wird. Eine aktuelle Konsolen-Umsetzung zum Zeitpunkt dieser Arbeit, in welcher auch Bezüge auf das Rollenspiel

genommen werden, ist der Spieltitel *Diablo III*[106]. Hier wird eine Separation der Spielenden auch in der digitalen Welt verhindert, in dem nicht wie sonst üblich auf ein Splitscreen-Verfahren zurückgegriffen wird, sondern die Avatare der Spielenden gemeinsam auf einem Bereich visualisiert werden[107].

Das Entwicklungsstudio *Harebrained Schemes LLC* führte im Jahr 2013 eine Crowdfunding-Kampagne für ein *Golem Arcana* genantes Konzeptspiel durch, welche im Oktober desselben Jahres erfolgreich abgeschlossen wurde [Harebrained Schemes LLC, 2013c]. Bei diesem Spiel handelt es sich nach eigener Aussage um ein digital erweitertes Tabletop-Rollenspiel, bei dem die Spielmechanik mit Hilfe von digitalen Elementen erweitert wird [Wasserbäch u. Holland, 2013; Harebrained Schemes LLC, 2013b]. Die Spielenden können dabei reale Figuren auf einem Spielbrett aufstellen und ihre Eingaben mit Hilfe eines elektronischen Stifts tätigen, durch den Markierungen auf der Spielkarte und der Spielfiguren eingelesen werden. Dieser Stift kommuniziert die eingelesenen Informationen drahtlos an ein Smartphone oder Tablet, durch welche das Spiel sowohl grafisch unterstützt als auch inhaltlich koordiniert wird. Ziel des Konzeptes ist ein erleichterter Zugang, bei dem nicht erst umfangreiche Regelbücher gelesen werden müssen, um ein Spiel zu beginnen [Harebrained Schemes LLC, 2013b]. Eine Spielszene ist schematisch in Abbildung 5.5 festgehalten.

Abzugrenzen ist dieses Konzept vom klassischen Pen-&-Paper-Rollenspiel, da hier das Hauptaugenmerk nicht auf der dynamischen Entwicklung einer Geschichte liegt, sondern auf einem Kampfsystem mit gegebener Hintergrundgeschichte. Die Spielenden können jedoch stellenweise wählen, wie weit die digitale Unterstützung gehen soll, beispielsweise wenn es um die Notwendigkeit von Zufall geht: Ein Würfel kann in der realen Welt geworfen werden, um das Ergebnis in das System einzugeben, der Würfel kann aber auch rein virtuell gerollt werden [Harebrained Schemes LLC, 2013b]. Der interessante Ansatz dieser konkreten Umsetzung wird im Rahmen eines größeren Vergleichs in Abschnitt 12.3.1 nochmals aufgegriffen.

5.3. Aufbau der Arbeit

Im Sinne einer kurzen thematischen Zusammenfassung soll an dieser Stelle ein Überblick des generellen Aufbaus und der verwendeten Argumentationsstruktur dieser Ausarbeitung gegeben werden. Diese Arbeit gliedert sich in vier Teile, namentlich „Grundlagen“, „Verwandte Arbeiten und Konzept“, „Realisierung“ und „Evaluation und Konklusion“, deren Inhalte im Folgenden zusammengefasst werden.

[106] 2012, *Blizzard Entertainment* [MobyGames.com, 2014a]
[107] vgl. hierzu ebenfalls Abbildung 16.1 auf Seite 263

Abbildung 5.5.: *Golem Arcana*-Spielsituation

Screenshot aus einem Video des Entwicklers [Harebrained Schemes LLC, 2013d].

Im ersten Teil der Arbeit werden die benötigten wissenschaftlichen Hintergründe der involvierten Teilgebiete vorgestellt und näher erläutert. Begonnen wird dabei in Kapitel 6 mit Interaktionen im Allgemeinen sowie dem Begriff der Metapherhaftigkeit und der Human-Computer Interaction (HCI) im Speziellen. Besondere Aufmerksamkeit wird an dieser Stelle der Nutzung von Smartphones als Interaktionsgerät zuteil. Ferner existieren verschiedene Arten gemeinschaftlicher Interaktionen, wobei Kollaborationen und Kooperationen im Rahmen dieser Arbeit von besonderer Bedeutung sind. Im darauffolgenden Kapitel wird der Begriff der Mixed Reality genauer erläutert und die dazugehörige Taxonomie vorgestellt. Dem schließt sich eine Betrachtung an, wie Aufgabenbereiche sinnvoll an die Realität und Virtualität delegiert werden können, um Stärken und Schwächen der beiden Welten herauszuarbeiten und Vorschläge zu unterbreiten, wie die jeweiligen Stärken genutzt werden können. Hierfür wird der neue Begriff der Physical Is Good At - Digital Is Good At (PIGA-DIGA)-Listen eingeführt. Dem Kapitel über Mixed Reality-Grundlagen folgt in Kapitel 8 eine Beschreibung von Grundlagen und Konzepten des Storytellings mit besonderem Fokus auf Computerspiele. Damit zusammenhängend werden verschiedene Erzählstrukturen vorgestellt, wobei ein besonderes Augenmerk auf den Ansatz des sogenannten *modularen Storytellings*[108] gelegt wird. Ebenfalls wird das Konzept von Pen-&-

[108] nach Sheldon

Paper-Rollenspielen genauer erörtert, da ein Verständnis dieser Thematik im Rahmen der Arbeit von grundlegender Bedeutung ist. Im Anschluss daran findet eine Detailbetrachtung von verschiedenen Entscheidungsfindungsansätzen und generell dem Prinzip der Entscheidungsfindung in Zusammenhang mit einer künstlichen Intelligenz statt, dessen Begriffe kurz vorgestellt und für die Zwecke der vorliegenden Arbeit definiert werden. Ebenfalls in den Aufgabenbereich der künstlichen Intelligenz fällt die Funktionsweise einer Spiel-Kamera, sodass in Kapitel 10 verschiedene Systeme und deren prädestinierte Spielgenres vorgestellt werden, um auch hier verschiedene Vor- und Nachteile herauszuarbeiten. Als letztes werden im Grundlagenteil Netzwerke in Computerspielen und damit zusammenhängend Netzwerkprotokolle, speziell das eingesetzte TCP und dessen Eigenschaften behandelt. Insbesondere die Vor- und Nachteile in Computerspielen sowie durchführbare Optimierungsmöglichkeiten liegen dabei im Fokus.

Der zweite Teil der Arbeit beinhaltet einen Überblick über konkrete Arbeiten und Konzepte, die sich mit einem oder mehreren Teilbereichen dieser Arbeit decken. Vollkommen identische Ansätze wurden bis zur Fertigstellung dieser Ausarbeitung nicht gefunden, sodass das Gesamtkonzept dieser Arbeit als neuer Ansatz betrachtet werden kann. An den Überblick verwandter Arbeiten schließt sich der Teil der Arbeit an, in dem konkrete Überlegungen der wissenschaftlichen Grundlagen in Konzepte und Designentscheidungen überführt werden. Die Aufteilung folgt dabei dem Schema des ersten Grundlagenteils, sodass für jedes Grundlagenkapitel ein korrespondierendes Konzeptkapitel existiert. Für das hier entwickelte Gesamtkonzept war es von essentieller Bedeutung, sowohl den Begriff der Mixed Reality als auch den der Interaktion, besonders im Hinblick auf Kollaboration und Kooperation, zu definieren. Diese Definitionen befinden sich in den Kapiteln 13 respektive 15. Der Konzeptteil beinhaltet darüber hinaus weitreichende Überlegungen zum realisierten Storytelling und gibt sowohl einen feinals auch grobgranularen Überblick über den implementierten Level- und Interaktionsaufbau.

Dem Konzeptteil schließt sich im dritten Teil dieser Arbeit die Verschriftlichung der Realisierung der entwickelten Anwendungsarchitektur und ausgewählter, interessanter Komponenten an. Unter anderem wird ein Überblick des entwickelten Kommunikationsprotokolls und der damit zusammenhängenden sogenannten *Nachrichtenagenten* gegeben, welche in Abschnitt 18.1 und Kapitel 19 zu finden sind. Besondere Aufmerksamkeit kommt hierbei unter anderem dem *Interaction-Agent* zugute, welcher sich auf der Hauptkomponente, dem Server, befindet und weite Teile der Interaktionslogik der Anwendung implementiert. Ebenfalls betrachtet werden die Rollen der Smartphone-Clients (in Kapitel 21) sowie der für die Visualisierung der Anwendung zuständige Game-Engine-Client in Kapitel 22. Nach einem tieferen Einstieg in die konkreten Probleme und Lösungen der Kamera-Intelligenz endet das Kapitel mit der Verdeutlichung des Zusammenhangs zwischen Interaktionen, Story und Gameplay.

Im letzten Teil der Arbeit, ab Kapitel 24, ist die Niederschrift der durchgeführten Evaluationen zu finden. Dieses Kapitel ist in zwei Teile unterteilt, da sowohl die konstruierte Netzwerktechnologie als auch das aufgestellte Postulat und damit einhergehend das anvisierte Ziel der Arbeit untersucht wurden. Eine Zusammenfassung der Arbeitsergebnisse wurde vorgenommen, indem zwei Anwendungsfälle des Gesamtsystems aufgegriffen und näher erläutert werden. Nachfolgend und zugleich als letztes folgt ein Ausblick auf mögliche Verbesserungen und Anregungen für zukünftige Forschungsarbeiten.

Teil I.

Grundlagen

„[Interaktion, die]: aufeinander bezogenes Handeln zweier oder mehrerer Personen; Wechselbeziehung zwischen Handlungspartnern“

(Duden [Bibliographisches Institut GmbH, 2013]*)*

KAPITEL 6

Interaktionen

Wenn der Begriff „Interaktion“ gelesen wird, so ist oftmals der Gedanke einer Handlung zwischen zwei oder mehr Personen präsent, was sich auch in der dieses Kapitel einleitenden Definition des *Duden* niederschlägt.

Dabei kann dieser Begriff, gerade im heutigen, modernen Zeitalter des Computers, für mehr stehen als nur für einen Vorgang, welcher sich zwischen zwei oder mehr involvierten Menschen abspielt. Bei exakter Betrachtung müssen für eine Interaktion noch nicht einmal Handelnde respektive Handlungspartner in Form von Personen vorhanden sein; es ist durchaus ebenso möglich, dass beispielsweise ein Computersystem mit einem weiteren System interagiert[109]. Ebenso ist es vorstellbar, dass eine Katze mit einem Spielzeug spielt, denn auch dies fällt unter den Begriff der Interaktion. Gerade bei diesem zweiten Beispiel wird deutlich, dass eine Interaktion nicht zwangsläufig bidirektional intoniert erfolgen muss, denn die Spielzeugmaus der Katze hat keine Absicht; sie ist ein lebloser Gegenstand. Insbesondere in der Informatik ist es häufig anzutreffen, dass ein System oder allgemein ein (digitales) Objekt auf eine einseitige, durch einen Nutzer initiierte Handlung in irgendeiner Form reagiert. In diesem Zusammenhang kann ebenfalls von HCI gesprochen werden, für welche Hewett u. a. folgenden Definitionsvorschlag unterbreitet haben:

> *„Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.“*

[Hewett u. a., 1992, S. 5]

Es wird deutlich, dass das Forschungsfeld der HCI das Feld der Informatik ausweitet: Auf der einen Seite stehen Maschinen und die damit zusammenhängenden Phänomene, auf der anderen Seite ste-

[109] Ein interessantes Ergebnis zur Interaktion zweier künstlicher Intelligenzen miteinander ist im Rahmen des Projektes *AI vs. AI* am *Cornell Creative Machines Lab* entstanden. Hierbei unterhalten sich zwei so genannte *Chatbots* miteinander und visualisieren die Ausgaben über digitale Avatare [Labutov u. a., 2013].

hen Menschen und damit einhergehend weitere, große Forschungsfelder. Mit hierein spielen unter anderem Kommunikationstheorie, soziale Zusammenhänge, Ethnologie, Linguistik und vielerlei mehr (vgl. [Hewett u. a., 1992, S. 5]). Die Autorin Sick beispielsweise setzt Interaktionen in ein Gleichnis mit Übersetzungen und stellt heraus, dass eine Trennung von perzeptiver und kognitiver Sphäre nicht als möglich erscheint [Sick, 2012, S. 102]. Der Begriff *Übersetzung* wird dabei in diesem Zusammenhang nicht als rein linguistisches Wort betrachtet, sondern als notwendige Schnittstelle zur Interaktion und Interaktivität, die eine verwirklichende Rolle einnimmt - anstatt einer repräsentativen Rolle, wie es beispielsweise bei einem übersetzten Text der Fall wäre.

Ein Zusammenhang zwischen Interaktionen, Übersetzung und auch dem Kunstbegriff lässt sich ebenfalls in Richtung der Computerspiele und damit zu der im Rahmen dieser Arbeit entwickelten Implementierung herstellen. Die Frage, ob Computerspiele tatsächlich als Kunst respektive Kunst*form* anzusehen sind, sprengt an dieser Stelle sicherlich den Rahmen[110]. Wenn Computerspielen das Prädikat der Kunstform jedoch zugesprochen wird, so darf die Frage gestellt werden, ob Computerspiele ebenso wie Kunstwerke rezipiert werden können, sollten oder sogar *müssten*. Dies ist im Hiblick auf Interaktionen eine interessante Fragestellung, denn wie Sick heraus stellt, erfordert die Rezeption eines Kunstwerks die Partizipation des Betrachters [Sick, 2012, S. 103]. Dieser Beobachtung schließt sich die Frage an, wo die Grenze zur Interaktion verläuft und ob sich eine solche Grenze generell ziehen lässt. Anders ausgedrückt: Findet bereits eine Interaktion statt, wenn ein Objekt bewusst wahrgenommen wird?

Geelhaar sieht ein Zusammenwirken individueller, sozialer und kontextbezogener Faktoren, die zusammen erst eine Interaktion entstehen lassen [Geelhaar, 2012, S. 206]. Dabei wird zunächst davon ausgegangen, dass Nutzer und Nutzerinnen das (in diesem Fall) digitale Angebot bewusst wahrnehmen und dass aus dieser Wahrnehmung heraus eine individuelle, auf den Kontext bezogene (Interaktions-) Absicht entsteht. Darauf folgt, dass das digitale Angebot identifiziert und verstanden wird, woraufhin Interaktionsmöglichkeiten auch als solche erkannt werden. Einem Prozess der Navigation und Ansteuerung bestimmter Interaktionsmöglichkeiten in der digitalen Welt folgt anschließend die Auslösung gewünschter Ereignisse, um letztlich die Ausgabe respektive die Rückgabewerte der Ereignisse zirkulär in einen realen Kontext einzubetten (und daraus gegebenenfalls einen neuen Interaktionslauf zu starten). (Vgl. [Geelhaar, 2012, S. 206-207], [Geelhaar, 2006])

Für das Zustandekommen einer Interaktion zwischen Mensch und Maschine müssen demnach die menschlichen Sinne angesprochen werden – doch genau das passiert zunächst einmal nicht. Gegenstände können mehrere inhärente Eigenschaften besitzen, mit denen die Aufmerksamkeit des Men-

[110] Interessante Standpunkte hierzu sind sowohl in Audio-, Video- als auch Zeitungsbeiträgen zu finden; ebenso bietet das Computerspielmuseums in Berlin Führungen an, bei denen der Aspekt, ob Computerspiele eine Kunstform darstellen, mit erörtert wird [Gameshouse gGmbH, 2013; Cordes-Strehle, 2013; Biermann, 2013; Bulban u. Trotier, 2012].

schen erregt werden kann, wie beispielsweise visuelle oder akustische Merkmale[111]. Mit der Wahrnehmung eines interaktionsfähigen Systems alleine ist jedoch noch keine Interaktion entstanden, denn Interaktion ist die Verflechtung zwischen Wahrnehmung, Absicht, Ergebnisrückführung und ebenfalls der medialen Verkörperung des Perzeptions- respektive Interaktionsgegenstandes. Wird diese informelle Definition um den Begriff der *Vollständigkeit* ergänzt und auf die HCI angewandt, so wird auf diesem Weg das Konzept der *vollständigen*, immersiven Interaktion respektive der Virtual Reality[112] im Wortsinn erreicht, in der das Gehirn nicht mehr zwischen Realität und Virtualität zu unterscheiden vermag. [Robben, 2012, S. 19-25]

In der Regel starten Interaktionen jedoch in der realen Welt und involvieren hierbei den gesamten menschlichen Körper oder zumindest Teile davon. Wenn ein Buch gelesen wird, bezieht dies beispielsweise die Augen mit ein (indem Buchstaben erfasst werden), den Geist (Buchstaben verstehen, Sinnzusammenhänge bilden), den Vorgang des Lesens an sich (Seiten blättern) und auch das Buch als Trägermedium (Gewicht, Haptik, Olfaktorik[113]). Die Interaktion selbst kann sich dabei mit dem Trägermedium ändern: Es macht einen Unterschied, ob der gleiche Text in einem Buch gelesen oder beispielsweise in einen virtuellen Raum hineinprojiziert und unter Zuhilfenahme der Oculus Rift[114] betrachtet wird. [Robben, 2012, S. 19-25]

Letzteres Beispiel zeigt ebenfalls, dass Interaktionen einem Prozess der kontinuierlichen Re-Medialisierung unterworfen sind, in der Form, dass Interaktionen bei der Einführung neuer Medien und Technologien eine Metamorphose durchleben und in verändertem Gewand einen erneuten Aktualitätszyklus durchlaufen[115]. Eine besondere Rolle bei der vom Mensch ausgehenden Interaktion spielt dabei die Hand, welche gleichzeitig als Sinnes- wie auch Handlungsorgan dient. Ein plastischer Vergleich ist die Arbeit einer Malerin, welche einem zirkulären Prozess aus Perzeption und Gestaltung unterworfen ist: Ständig wird das gerade zu erschaffende Werk genau an der Stelle verändert, die zuletzt auch aktiv wahrgenommen wurde. [Robben, 2012, S. 25-28]

[111] Wobei dies natürlich keine intonierte Aufmerksamkeit darstellt; allerhöchstens die intonierte Aufmerksamkeit durch den Konstrukteur oder die Konstrukteurin des fraglichen Systems als proaktiver Gedankengang, sofern es sich um ein künstlich erzeugtes Objekt handelt.

[112] Dieser Begriff wird in Kapitel 7 auf Seite 139 genauer beschrieben.

[113] In einer persönlichen Anmerkung sei gestattet, dass immer noch weit unterschätzt wird, wie überaus gut ein frisch aufgeschlagenes Buch riechen kann und wie viel Einfluss das auf die Wahrnehmung und den Genuss der Lektüre nimmt.

[114] Bei der *Oculus Rift* handelt es sich um ein Head-Mounted-Display, welches über zwei eingebaute Bildschirme dreidimensionale Inhalte direkt zu den Augen überträgt. Die Brille kann dabei durch eingebaute Sensoren ebenfalls den Bewegungen des Kopfes folgen und erlaubt somit die immersive Perzeption einer virtuellen Welt [Oculus VR, Inc., 2013].

[115] Ein Beispiel mit kontradiktorischer Anekdote ist die Neuveröffentlichung des Romans *1984* von George Orwell als so genanntes E-Book. Das alte Medium des Buches wurde hier in eine neue, moderne Form überführt. Interessanterweise unterlag genau dieser Titel mit dem Inhalt einer totalitären und überwachten Gesellschaft einem Lizenzstreit, sodass das digitale Buch über einen Steuerungsbefehl von den Geräten der Kunden gelöscht wurde, sobald diese ihr Gerät mit dem Internet verbunden haben. Dies wirft auch Fragen darüber auf, ob eine jeweilige Re-Medialisierung nicht ebenfalls unter sozialkritischen und/oder besitzrechtlichen Aspekten betrachtet werden kann oder muss. [Braun, 2009]

6.1. Metapherhaftigkeit von Interaktionen

Eine wichtige Eigenschaft heutiger HCI ist die Metapherhaftigkeit, da Interaktionen der realen Welt im Regelfall nicht auf die virtuelle Welt übertragen werden können und umgekehrt. Um das Beispiel der Malerin wieder aufzugreifen, wäre ein digitaler Grafiker nicht in der Lage, seinen Bildschirm direkt mit Farbe anzumalen, um das Bild anschließend digital zu verfeinern. Stattdessen wird hier in der Regel[116] eine Metapher eingesetzt, bei welcher durch die Bewegung der Maus ein virtueller Pinsel über das ebenso virtuelle Bild bewegt wird. Im größeren Kontext bedeutet dies, dass alltägliche Interaktionen in einen Sinnzusammenhang gesetzt werden, welcher durch den Versuch einer konstruierten Selbstähnlichkeit erlerntes Verhalten der realen Welt in die virtuelle Welt überträgt. Durch den alltäglichen Gebrauch können Metaphern allerdings auch an Sinnhaftigkeit verlieren. Besonders häufig ist dies der Fall, wenn die Bildhaftigkeit so sehr in den Alltag übergangen ist, dass der reale Bezug verloren geht. Ein Beispiel dafür ist das Anlegen virtueller Ordner, bei denen kaum noch ein Mensch an echte physikalische Ordner denkt, die möglicherweise im Regal hinter einem stehen. [Robben, 2012, S. 28-30]

Die Durchdringung des Alltags mit Technologie kann gerade im Bereich der Computerspiele auch kontraproduktive Elemente nach sich ziehen. Björk u. a. stellen heraus, dass soziale Aspekte und Interaktionen in heutigen Computerspielen oftmals verloren gehen:

> *„With computer games, we no longer need to physically get together with friends and family to play games that involve other people than ourselves. In some sense, computers have turned game play into individual and isolated activities.“*

[Björk u. a., 2001, S. 1]

Um das menschliche Bedürfnis nach Sozialisierung zu reintegrieren, bietet sich eine Betrachtung der von Weiser entwickelten und 1991 erstmalig ausformulierten Vision des *Ubiquitous computing* (dt. „allgegenwärtiger Computer“[117]) an.

6.2. Der allgegenwärtige Computer

Das Konzept des allgegenwärtigen Computers beschreibt die vollständige Durchdringung und Integration der Technologie in den Alltag [Weiser, 1991]. Dabei werden die Nutzer in den Vordergrund

[116] Der Vollständigkeit halber sei erwähnt, dass durchaus Grafiktabletts und ähnliche Eingabegeräte existieren, welche zumindest teilweise reales Malen am Computer erlauben.
[117] übersetzt v.d. Autoren aus dem Englischen

gestellt, wodurch ein Bewusstsein für die Unterstützung menschlicher Bedürfnisse mit Hilfe von Technik geschaffen werden soll. Diese Durchdringung darf mittlerweile als Normalität betrachtet werden: Smartphones, Tablets, Navigationssysteme in Autos und nicht zuletzt die Entwicklung moderner Ansätze wie *Google Glass* oder die *Oculus Rift*, bei letzterer wiederum sogar zuletzt die Verbindung von virtueller Realität und Smartphones, dienen hierbei nur als Stichwörter [Kuri, 2013a; Janssen, 2013b]. Nicht zuletzt durch das Unternehmen *Apple* wurde deutlich, dass moderne Alltagstechnik[118] nicht nur eine unterstützende Funktion erfüllen, sondern auch Ansprüche an Identifizierungsmerkmale im Sinne eines Lifestyle-Produktes gestellt werden [Brinkbäumer u. Schulz, 2010]. [Robben u. Schelhowe, 2012, S. 7-9] Schmitt schreibt hierzu:

> *„Customers want products that dazzle their senses, touch their hearts and stimulate their minds."*

[Schmitt, 1999, S. 22]

Ein Beispiel für ein solches Interaktionsgerät, welches gleichzeitig als „Lifestyle-Gadget" angesehen wird, ist das *iPad*. Weshalb ist dieses Tablet beispielsweise so erfolgreich, obwohl es schon lange vorher Tablet-Computer gab[119]? Eine mögliche Erklärung findet Cusumano in der Verbindung zum „Apple-Universum", der leichten Zugänglichkeit und dem damit verbundenen *Appstore* [Cusumano, 2010, S. 193] (vgl. auch [Brinkbäumer u. Schulz, 2010]). Was ein erfolgreiches und „intuitives" Interaktionsgerät letztlich ausmacht, kann detaillierter untersucht werden, ebenso wie die verschiedenen Formen, in denen interaktive Systeme existieren. Das Konzept vom „allgegenwärtigen Computer" kann hierbei als Ausgangspunkt betrachtet werden, bei dem Systeme und Interaktionsgeräte auf ihre Eignung hin untersucht werden, den Nutzer in seinem natürlichen Verhalten zu unterstützen und nicht in neue Verhaltensweisen zu drängen (vgl. [Björk u. a., 2001, S. 1]).

[118] so genannte *Gadgets*

[119] Interessanterweise entstand sogar die Diskussion, ob nicht vielleicht Arthur Charles Clarke den Tablet-Computer „erfand", da er in seinem Roman *2001: A Space Odyssey* sehr detailliert beschreibt, wie heutige Tablet-Computer funktionieren:

"When (Floyd) had tired of official reports, memoranda and minutes, he would plug his foolscap-sized newspad into the ship's information circuit and scan the latest reports from Earth. One by one, he would conjure up the world's major electronic papers… Switching to the display's short-term memory, he would hold the front page while he searched the headlines and noted the items that interested him. Each had its own two-digit reference. When he punched that, a postage-sized rectangle would expand till it neatly filled the screen and he could read it with comfort. When he finished he could flash back to the complete page and select a new subject for detailed examination. Floyd sometimes wondered if the newspad and the fantastic technology behind it was the last word in man's quest for perfect communications."

[Arthur C. Clarke, 1968, Kapitel 9], vgl. auch [Puls, 2012].

6.2.1. Formen interaktiver Systeme

Vereinfacht ausgedrückt bietet ein interaktives System im Sinne der HCI eine Möglichkeit, von Benutzern durchgeführte Systemeingaben mit geringer Latenz[120] in eine Ausgabe zu überführen [Heisenberg, 2012b, S. 11]. Eine allgemeinere Definition wird von Stary angestrebt und trägt dem Umstand Rechnung, dass dem Menschen innerhalb eines Interaktionszyklus eine besondere Rolle zukommt:

> *„Ein interaktives System ist ein **sozio-technisches System**, da es nicht nur aus den technischen Komponenten Hard- und Software (beide ergeben das technische System **Computer**) besteht, sondern auch die soziale Komponente **Mensch** besitzt.“*

[Stary, 1996, S. 14]

In diese Definition spielt mit ein, dass zur Entwicklung intuitiver, verständlicher und ergonomischer Schnittstellen nicht nur das Studium der damit verbundenen Technologien, sondern in ebenso starker Art und Weise der Mensch mit seinen Fähigkeiten und sozialen Gefügen betrachtet werden muss. Die Erforschung des Menschen und seiner Fähigkeiten treibt damit die Entwicklung neuer Interaktionsansätze im Bezug auf bekannte Technologien voran, während die Erschaffung neuer technologischer Ansätze[121] im Gegenzug eine Weiterentwicklung darauf anwendbarer Interaktionsgeräte forciert [Stary, 1996, S. 13].

In jedem Fall findet in einem interaktiven System ein alternierender Austausch zwischen Mensch und Maschine statt. Eine besondere Rolle wird dabei den Schnittstellen zuteil, die diesen Austausch ermöglichen, wobei beachtet werden sollte, dass der Mensch über mehrere Möglichkeiten verfügt, seine Umwelt wahrzunehmen[122] [Stary, 1996, S. 13] [Heisenberg, 2012b, S. 15]. Heutige HCI zielt darauf ab, die traditionellen Windows, Icons, Menus und Pointer (WIMP)-Schnittstellen zu erweitern oder zu ersetzen und auf diese Weise, insbesondere im Bereich der dreidimensionalen Navigation, einen effizienteren und intuitiven Zugang zu modernen, interaktiven Systemen zu ermöglichen. Beispiele für Anwendungen und den Bedarf an solchen Interaktionsformen lassen sich nicht nur im Bereich der Computerspiele, sondern ebenso bei der Datenvisualisierung, der Medizin, dem Militär, der Architektur und vielen weiteren Disziplinen finden (vgl. Abbildung 6.1). [Heisenberg, 2012b, S. 16-26]

Ein generischer Aufbau eines Interaktionssystems ist in Abbildung 6.2 dargestellt. Insbesondere die Schnittstellen zwischen Mensch und System lassen sich in Gruppen der Ein- und Ausgabegeräte einteilen, wobei hybride Formen möglich sind, welche beide Gruppeneigenschaften in einem Gerät vereinen. Die Formen der Einteilung gliedern sich wie folgt:

[120] Eine Latenz ist dann als gering zu bezeichnen, wenn sie dem Anwendungszweck und der Intonation des Nutzers genügt.
[121] beispielsweise die Entwicklung von 3D-Engines
[122] Zum Begriff der Wahrnehmung und Affordanz siehe auch Abschnitt 7.3 auf Seite 147.

Abbildung 6.1.: Die Abbildung zeigt drei mögliche Anwendungsfelder von interaktiven Systemen. Im linken Teil der Abbildung ist hierbei eine Apparatur des Feldes der Medizin zu sehen, mit der virtuelle Augenoperationen durchgeführt werden können. Auf diese Weise können Chirurgen und Chirurginnen ihr Wissen erweitern oder festigen, bevor sie Operationen am lebenden Menschen durchführen. Im mittleren Teil der Abbildung ist zu sehen, wie Phobien mit Hilfe virtueller Umgebungen adressiert werden können, um auf diese Weise beispielsweise Flugängste zu lindern oder posttraumatische Belastungsstörungen durch Konfrontationstherapie zu behandeln. Der rechte Teil der Abbildung zeigt das Flugtraining von Helikoptern mit Hilfe einer interaktiven Anwendung; wobei in diesem speziellen Fall der Anflug auf eine Bohrinsel geübt wird.

aus [Virtually Better, Inc., 2014; VRmagic Holding AG, 2014; ADAC HEMS Academy GmbH, 2014]

Nach Wertedarstellung: Bei einer Klassifizierung nach Wertedarstellung kann nach Eingabegeräten mit *absoluter* und *relativer* Darstellung unterschieden werden. Ein Eingebergerät mit absoluter Wertedarstellung besitzt ein eigenes Koordinatensystem und verrechnet Parameter innerhalb desselben.

Relative Eingabegeräte hingegen besitzen kein eigenes Koordinatensystem, sondern integrieren veränderliche Werte in Korrelation zur vergangenen Zeit. Auf diese Weise lassen sich beispielsweise skalare Transformationsmetaphern erzeugen, bei denen eine bestimme Anzahl an Einheiten im Interaktionsgerät einer ebenso hohen Anzahl an Einheiten im System entspricht, welche zusätzlich mit einem Faktor verrechnet (beispielsweise multipliziert) werden.

Nach Freiheitsgraden: Eine Einteilung nach Freiheitsgraden wird anhand der möglichen, gleichzeitigen Bewegungsrichtungen vorgenommen, welche durch das Gerät unterstützt werden. Eine

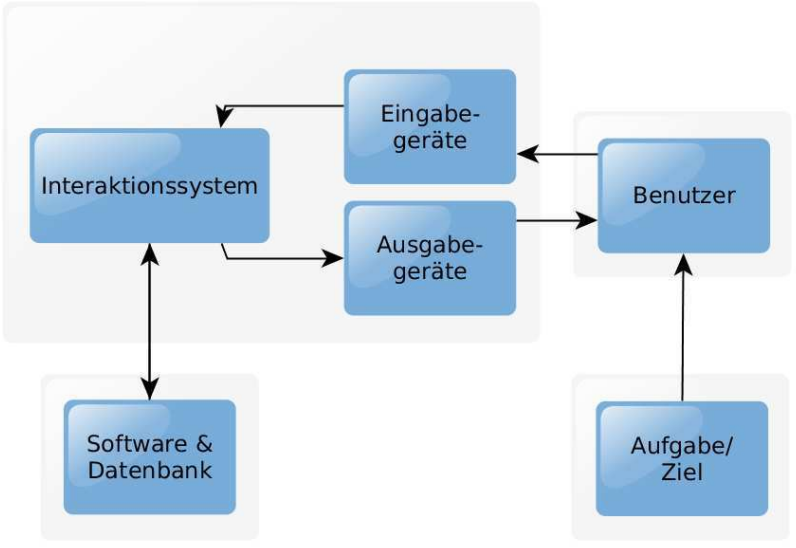


Abbildung 6.2.: Die Abildung zeigt den generischen Aufbau eines Interaktionssystems. Der Benutzer wird von einem Ziel angeleitet, mit dem System zu interagieren. Dazu stehen verschiedene Ein- und Ausgabegeräte bereit. Das System selbst wird mit einer Software betrieben und hält bestimme Daten vor, die auch verändert werden können.

nach [Heisenberg, 2012b, S. 34]

Computermaus besitzt beispielsweise zwei Freiheitsgrade, während eine Schieberegler auf einem Mischpult nur einen Freiheitsgrad unterstützt. Diese Art der Einteilung ist nicht immer ohne weiteres anwendbar, beispielsweise bei Bewegungs- oder Sprachsteuerungen.

Nach Performance-Parametern: Performance-Parameter erlauben weitere Unterteilungen. Sie sind generell unabhängig von den Freiheitsgraden und der Wertedarstellung des Eingabegeräts. Die Unterteilung gliedert sich auf in Genauigkeit (*accuracy*), Schwankungsbreite (*jitter*), Latenzzeit (*latency*) und Fehlerfortpflanzung (*drift*) [Heisenberg, 2012c, S. 14].

Die Genauigkeit bezeichnet hierbei die Differenz zwischen tatsächlicher und gemessener Objektposition und ist nicht mit dem Auflösungsvermögen (d.h. dem Quantisierungsgrad) des Eingabegerätes gleichzusetzen. Dem steht die Schwankungsbreite gegenüber, welche Aufschluss darüber gibt, ob der Messwert für ein stillstehendes Objekt konstant bleibt oder „springt“ beziehungsweise „zittert“.

Einer der wichtigsten Performance-Parameter ist die Latenz. Sie steht für die zwischen Aktion und Reaktion vergangene Zeit und muss möglichst niedrig gehalten werden, da die Interaktion

sonst erheblich erschwert wird; ebenso können Nebenwirkungen wie die *Simulator Sickness*[123] in immersiven Anwendungen durch die Latenz begünstigt werden.

Die Fehlerfortpflanzung letztlich bezeichnet die Aufsummierung einer Messungenauigkeit über die Messzeit.

[Heisenberg, 2012c, S. 11-22]

Neben den Eingabegeräten und ihren Klassifikationen gibt es eine große Variation von Ausgabegeräten, allen voran und am häufigsten verwendet, das Display. Der visuelle Sinn des Menschen kann hierbei sowohl monoskopisch als auch stereoskopisch stimuliert werden, um beispielsweise durch Ausnutzung der retinalen Querdisperation einen künstlichen Raumeindruck zu erzeugen [Zimbardo u. Gerrig, 2008, S. 146]. Eine mögliche Form eines hyriden Gerätes, welches sowohl Eingabe als auch Ausgabe vereint, ist der *Touchscreen* oder, in einem Gerät verbaut, das *Smartphone*. Da diesem im Zuge der vorliegenden Arbeit besondere Bedeutung zukommt, wird im nächsten Abschnitt näher beschrieben, wie sich Interaktionen mit diesen Geräten umsetzen lassen.

6.3. Vom Touchscreen zum Smartphone als Interaktionsgerät

Auf Seite 124 wurde die Vision des ubiquitären Computer von Weiser bereits vorgestellt. Konkret wird formuliert:

> *„There is more information available at our fingertips during a walk in the woods than in any computer system, yet people find a walk among trees relaxing and computers frustrating. Machines that fit the human environment, instead of forcing humans to enter theirs, will make using a computer as refreshing as taking a walk in the woods.“*

[Weiser, 1991]

Weiser bringt damit zum Ausdruck, dass beim Design und der Entwicklung von Computersystemen und insbesondere von Human-Computer-Interfaces immer vom Menschen, seinen Sinnen sowie von seinen Fähigkeiten ausgegangen werden soll (vgl. [Döring u. a., 2012, S. 115]). Demnach werden Schnittstellen als unsichtbar, angenehm und „natürlich“ empfunden, wenn es gelingt, diese in das natürliche Verhaltensmuster des Menschen einzupassen. In direkter Nachbarschaft des Begriffes der „Natürlichkeit“ ist oftmals der Begriff „intuitiv“ anzutreffen. *Intuitiv* bedeutet im allgemeinen Verständnis, dass

[123] Mit dem Begriff der *Simulator Sickness* werden im Allgemeinen verschiedene Faktoren zusammengefasst, bei denen sich widersprechende Signale des Körpers über Beschleunigung und Lage im Raum ein Unwohlsein hervorrufen. Eine empfehlenswerte Lektüre hierzu ist mit [Kolasinski, 1995] gegeben.

sich unbekannte Systeme, Anwendungen oder Aktionen ohne - oder allenfalls mit sehr geringem - kognitivem Aufwand verwenden lassen [Herrlich u. a., 2012, S. 138].

Das bedeutet jedoch auch, dass ein System, welches von einem Menschen als „intuitiv“ bezeichnet wird, nicht für einen anderen Menschen als ebenso intuitiv empfunden werden muss. Hierein spielt ein hohes Maß an Vorerfahrung mit gleichen oder vergleichbaren Systemen – bei letzteren auch das Abstraktionsvermögen, um bereits Erlerntes zu adaptieren und auf neue Sachverhältnisse anzuwenden. Jacob u. a. argumentieren, dass das individuelle Vorwissen der Menschen beim Design von Schnittstellen zwischen Mensch und Maschine genutzt werden solle:

> *„Basing interaction on pre-existing real world knowledge and skills may reduce the mental effort required to operate a system because users already possess the skills needed“*

[Jacob u. a., 2008, S. 4]

Wobei hierbei ebenfalls beachtet werden muss, dass die reale Welt nicht unreflektiert imitiert wird, sondern dass ebenfalls darüber nachgedacht wird, welche Aufgaben ein Computer durch Digitalisierung übernehmen respektive besser machen kann als reale Objekte[124] (vgl. [Döring u. a., 2012, S. 118]).

Im Vergleich zu den reinen Desktop-Systemen der 1990er Jahre ist das Modellieren von begreifbaren Schnittstellen eher komplexer und nicht einfacher geworden[125], weil mittlerweile viel mehr Möglichkeiten und Einsatzszenarien digitaler Systeme bereitstehen. Das bedeutet ebenfalls, dass ubiquitäre Systeme zunehmend in Umgebungen und Abläufe eingebunden sind, welche einer ständigen Veränderung unterworfen sind [Döring u. a., 2012, S. 115-116]. Einige Interaktionsschnittstellen verbinden dabei mehrere Elemente miteinander, wie es am Beispiel der Smartphones derzeit zu beobachten ist: Smartphones kombinieren Gesten- und Lageinformationen, werten Berührungen aus und besitzen gleichzeitig mechanische Schalter oder Knöpfe (vgl. [Döring u. a., 2012, S. 116]).

Durch solche Berührungen sind interaktive Oberflächen in der Lage, eine Verbindung zwischen realer und virtueller Welt zu erschaffen und auf diese Weise ebenfalls eine Aufweichung von Raumgrenzen zu erzeugen [Herrlich u. a., 2012, S. 135]. In diesem Zusammenhang steht entgegengesetzt zu einem Raum im Begriff eines *Containers*, der als *absoluter Raum* bezeichnet werden kann, der Begriff des *topischen Raums*. Der topische Raum unterschiedet sich vom absoluten Raum durch durchlässige Grenzen und den Einbezug des innerhalb des Raums *Erlebtem*. Der soziale Raum und der topische Raum stehen in engem Bezug zueinander und für eine Interaktion über die verschiedenen Raumgrenzen hinweg müssen Technologien benutzt werden, die Virtualität, Realität und Sozialität miteinander vereinen -

[124] Mehr zu diesem für Mixed Reality-Anwendungen wichtigen Aspekt ist im Abschnitt 7.2 auf Seite 145 zu finden.
[125] Mit dem technologischen Fortschritt geht oftmals die Erwartung einher, dass Schnittstellen eher zugänglicher und intuitiver gestaltet werden. Dies muss jedoch nicht unbedingt der Fall sein.

und genau an dieser Stelle kommen Smartphones als verbindendes Element ins Spiel. [Grüter, 2012, S. 323 - S. 324], [Grüter, 2012, S. 336], [Herrlich u. a., 2012, S. 135]

Das zentrale Element der Smartphones ist der so genannte *Touchscreen*. Ein Touchscreen verfügt über eine interaktive, berührungsempfindliche Oberfläche, die in besonderer Art und Weise ein ambivalentes Verhalten ermöglicht. Auf der einen Seite sind Menschen mit Hilfe von Smartphones in der Lage, reale Handlung auszuführen und damit digitale Prozesse anzustoßen. Auf der anderen Seite existiert bei der Verwendung von Smartphones ein geschlossener Feedbackkreislauf, den ein Touchscreen (und damit auch ein Smartphone) in jedem Fall erzeugt: Menschen können durch Berührung nicht nur Information vermitteln, sondern bekommen auch kontinuierlich visuelle Rückmeldung darüber, was die soeben getätigten Eingaben für einen Effekt hervorrufen. Auf diese Weise wird das analoge Signal der Berührung kontinuierlich quantisiert und ausgewertet, wodurch sich, je nach Quantisierungsgrad respektive je nach zeitlichem Auflösungsvermögen des Systems, eine Vielzahl von Anwendungsszenarien ergeben. Eine besonders schnelle Auswertung ist beispielsweise überall dort sinnvoll, wo eine analoge Steuerung erfolgt, wie bei der Navigation in virtuellen Welten [Israel u. a., 2009]. In diesem Sinne können Smartphones ebenfalls zur Realitätserweiterung oder als ergänzendes Ein- und Ausgabegerät verwendet werden. Eine Überlegung beim Einsatz eines Smartphones als Ergänzung beispielsweise zu einer Konsole ist der, dass sich der soziale Fokus vieler Konsolenspiele selbst dann niederschlägt, wenn es sich gar nicht um einen Mehrspielertitel handelt. Dieser inhärent soziale Faktor lässt sich analysieren und nutzen.

Der Chef des Entwicklungsstudios *Quantic Dream*[126], David Cage, erklärte in einem Interview beispielsweise, dass des öfteren die Entdeckung gemacht worden wäre, dass weitere im Raum befindliche Personen einem gerade aktiven Spieler oder einer Spielerin Hinweise gäben oder das Geschehen kommentieren würden [Stoecker, 2013]. Dieser Gedanke wurde vom Entwicklungsstudio *Ubisoft* zu einer Implementierungsidee weiterentwickelt, bei der weitere Personen mit Hilfe ihrer Smartphones oder Tablets direkt in das gerade aktive Spielgeschehen eingreifen können (vgl. Abschn. 3.1.4). Dieser Spielmodus wird in Abbildung 6.3 näher erläutert. Das Beispiel macht deutlich, dass das Bedürfnis der Menschen nach sozialer Teilhabe und Interaktion auch in den Fällen vorhanden ist, in denen technisch keine Möglichkeiten zu sozialer Partizipation vorgesehen wurden. Smartphones und Tablets können diesem Bedürfnis im Hinblick auf digitale Spiele entgegenkommen.

Bei der Interaktion mit Touchscreens (jedoch nicht ausschließlich dort) kann zwischen so genannten *Single-Point* und *Multi-Point*-Eingaben unterschieden werden [Herrlich u. a., 2012, S. 136]. *Single-Point* steht dabei für die zweidimensionale Steuerung eines Zeigers, am besten bekannt durch die Metapher des Mauszeigers. *Multi-Point*-Eingaben hingegen meinen nicht zwangsläufig Multi-Touch-Eingaben,

[126] bekannt für das Konsolenspiel *Heavy Rain* [Stöcker, 2010]

Abbildung 6.3.: In der Abbildung ist eine Szene des für das Jahr 2014 angekündigten Spiele-Titels *Watch Dogs* zu sehen. Im oberen, großen Bildbereich ist die Ansicht des „Haupt"-Spiels zu sehen, in dem der Avatar durch die virtuelle Welt gesteuert wird. Gleichzeitig kann ein weiterer Mitspieler oder eine Mitspielerin (in diesem Fall in der Rolle eines Antagonisten) innerhalb einer stilisierten Karte der Spielwelt Manipulationen ausführen, welche wiederum das Hauptspiel beeinflussen.

aus [Stoecker, 2013]

sondern generell die parallele Verwendung von mehr als zwei Freiheitsgraden zur Eingabe. Daneben existieren ebenfalls hybride Ansätze, beispielsweise die Eingabe mit mehreren Fingern auf einem gemeinsamen Gerät, wie es auf Touchscreens möglich ist. Forlines u. a. haben in einem Experiment mit einer großen Touchscreen-Wand nachweisen können, dass den Partizipanten das Auswählen und Interagieren leichter fiel, wenn statt zwei separaten Eingabegeräten ein direkter Kontakt zur Interaktionsfläche mit mehreren Händen möglich war [Forlines u. a., 2007, S. 654f.] (vgl. auch [Herrlich u. a., 2012, S. 136ff.]). Hierbei ist allerdings zu beachten, dass an dieser Stelle nicht fälschlicherweise die Annahme getroffen werden darf, dass ein Angebot zur beidhändigen Interaktion immer intuitiv

verständlich und für alle Menschen leicht zu adaptieren ist, sondern nur im Vergleich zur parallelen Verwendung mehrerer, individueller Eingabegeräte.

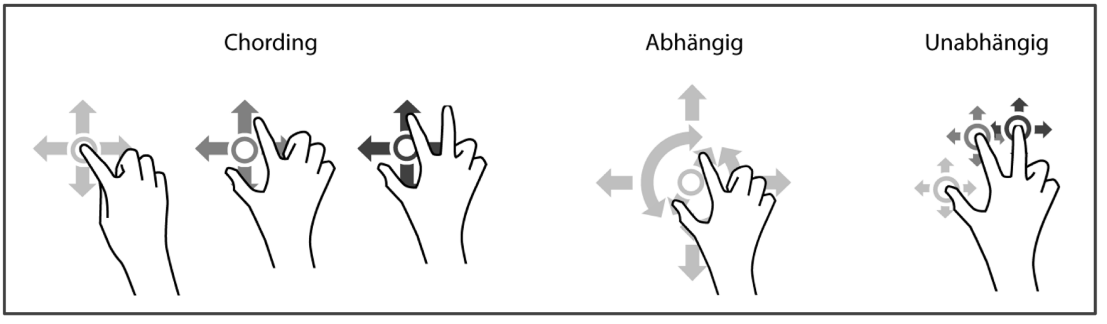


Abbildung 6.4.: Die Abbildung zeigt verschiedene Möglichkeiten zur Ausnutzung mehrere Freiheitsgrade bei Eingaben auf Touchscreens. Im linken Teil der Abbildung sind Mehrfingergesten respektive das so genannte *Chording* zu sehen. In der Mitte sind Möglichkeiten zur Ausnutzung der relativen Position der Finger zueinander dargestellt und im rechten Teil der Abbildung ist ein Beispiel für den unabhängigen Einsatz mehrerer Finger gegeben [Herrlich u. a., 2012, S. 143].

aus [Herrlich u. a., 2012, S. 143]

Ebenso scheint zu gelten: Eine Annäherung an die Wirklichkeit oder den gewohnten, physischen Raum ist nicht immer zielführend; beizeiten werden gerade „einfache“ Metaphern zum Ausführen von komplexen Aufgaben bevorzugt. Mit Rückbezug auf eine Szene des Films *Minority Report*[127], in welcher der Hauptdarsteller beidhändig im freien Raum mit einem Computer agiert und dabei dreidimensionale Aufgaben ausführt, konnten Bérard u. a. zeigen, dass die klassische Computermaus spezialisierte Eingabegeräte mit mehr Freiheitsgraden in Fragen der Effizienz übertrifft [Bérard u. a., 2009, S. 412f.]. Smartphones respektive Touchscreens scheinen damit einen gute hybride Mischung zwischen Eingabeformen mit vielen Freiheitsgraden und herkömmlicher, rein zweidimensionaler Eingabe darzustellen (vgl. [Herrlich u. a., 2012, S. 141]), da diese sowohl eine zweidimensionale, als auch mehrdimensionale Eingabemöglichkeit bieten können[128]. Die Dimensionalität wird dabei durch den Einsatz mehrerer Finger erhöht, was Matejka u. a. als „Chording“ bezeichnen [Matejka u. a., 2009, S. 1074] (vgl. auch Abbildung 6.4).

Zu den häufigsten virtuellen Manipulationen, die demnach über Touchscreens respektive Smartphones durchgeführt werden, zählen die Translation und Rotation. Schon das Wischen des Fingers auf

[127] siehe [Spielberg, 2002]
[128] Neben dem Touchscreen als Eingabemedium stellt das Smartphone bspw. durch den Beschleunigungssensor einen weiteren Eingabemechanismus zur Verfügung, welcher seinerseits über mehrere Freiheitsgrade verfügt.

dem Display, um beispielsweise eine Webseite nach unten zu scrollen, stellt eine Translation[129] der Webseiteninhalte dar. Obgleich das Hauptmerkmal von interaktiven Bildschirmen der direkte Kontakt des Fingers respektive der Hand zur Oberfläche darstellt, ist eine indirekte Interaktion häufig ergonomischer [Herrlich u. a., 2012, S. 135-S. 136].

6.4. Kooperative und kollaborative Interaktionen

Kooperationen sowie Kollaborationen werden eine besondere Aufmerksamkeit zuteil: Laut einer Studie, durchgeführt von Seif El-Nasr u. a., bevorzugen 55% der Spieler und Spielerinnen von Computerspielen im Alter zwischen 6 und 16 Jahren eine kooperative Art des Spielens gegenüber anderen Spielarten. 77% der Befragten beschäftigen sich gern mit Spielen die sowohl Kooperation als auch Konfrontation beinhalten [Seif El-Nasr u. a., 2010]. Zudem ist ein stetig steigendes Angebot an Kooperations- und Kollaborationsmodi in Spielen zu erkennen [Voida u. a., 2010]. Der Trend zum erhöhten Fokus auf kooperative Interaktion wird ebenfalls in [Rocha u. a., 2008; Islas Sedano u. a., 2013] erwähnt. Dieses Wiederauferstehen respektive Wiederentdecken vom kooperativen Spielen begründen die Autoren darin, dass Studien gezeigt haben, dass eine große Gruppe von potentiellen Spielern und Spielerinnen derzeit nicht aktiv und regelmäßig spielt, weil sie sagen, dass Spiele „nicht für sie gemacht sind“. Diese Gruppe favorisiert kooperative Erfahrungen und Spielerfahrungen, welche sie mit anderen teilen können - und dies im selben Raum („same physical space“) [Rocha u. a., 2008, S. 73].

Für den weiteren Verlauf ist es wichtig, zunächst eine Begriffsdefinition vorzunehmen:

Kollaboration: *„When collaborating, people work together (co-labor) on a single shared goal.”* [Sellier, 2014]. Im Kontext von Spielen betrachtet definieren Zagal u. a. den Begriff wie folgt: *„In collaborative games, players share common goals and outcomes. All gamers either win or lose together.“* [Voida u. a., 2010, S. 371].

Kooperation: *„When cooperating, people perform together (co-operate) while working on selfish yet common goals.”* [Sellier, 2014]. Auch hier existiert im Bereich der Spielentwicklung eine ähnliche Definition: *„In a cooperative game, players' goals are not necessarily in direct opposition, but neither are their goals completely aligned.“* [Voida u. a., 2010, S. 371].

[129] Allgemein formuliert handelt es sich bei einer Translation, Skalierung oder auch Rotation um eine *affine Abbildung*. Wenn zwei affine Räume (A, V_A) und (B, V_B) existieren, dann ist eine Abbildung $f\colon A \to B$ zwischen Räumen affin zu nennen, wenn eine Abbildung $\varphi\colon V_A \to V_B$ existiert, sodass für alle Punkte $P, Q \in A$ gilt (vgl. [Grassmann, 2007, S. 62ff.] [Mohr, 2013]):

$$\overrightarrow{f(P)f(Q)} = \varphi(\overrightarrow{PQ})$$

Insofern steht Kollaboration als Begriff für das Hinarbeiten auf ein gemeinsames Ziel. Es können dabei verschiedene Rollen eingenommen werden und unterschiedlich ausgeprägte Beiträge zum Erreichen des Ziels beigetragen werden. Bei der Kooperation hingegen wird an unterschiedlichen Zielen gearbeitet, welche jedoch beispielsweise einen gemeinsamen Austausch in Form von Erfahrungen oder Wissen ermöglichen. Es handelt sich somit um eine weniger starre Abhängigkeit als bei der Kollaboration. [Kaul, 2012]

Auch ein konkurrierendes Verhalten ist insbesondere in Spielumgebungen möglich und findet häufig Anwendung. Spieler entwickeln Strategien, welche denen anderer Spieler direkt entgegenstehen [Voida u. a., 2010, S. 371]. Die Grenzen zwischen den verschiedenen Modi sind jedoch nicht unbedingt als „harte" Grenzen zu betrachten, in der Form, dass auf der einen Seite konkurrierende und auf der anderen Seite kollaborierende (dazwischen kooperative) Aspekte stehen, sondern dass durchaus Synergieeffekte beobachtet werden können, wie Nardi u. Harris herausstellen:

> *„Even when players are competing, the object of having fun by playing the game is a form of collaboration; players could not compete if there were no one to play with. As in any game or sport, the larger field of collaboration is constituted by engaging in the game"*

Generell kann laut Lecky-Thompson festgehalten werden, dass Multiplayerspiele Spielern und Spielerinnen mehr Spaß bieten als Einzelspieler-Spiele [Lecky-Thompson, 2008, S. 56].

Seif El-Nasr u. a. explorierten in ihrer Studie rund 215 PC- und Videospiele mit Mehrspielermodus, um deren kooperative sowie kollaborative Natur zu analysieren und Erfolgs- und Risikofaktoren von kooperativen Spielen zu untersuchen. Wichtig ist hierbei zu erwähnen, dass Spiele im Allgemeinen eher als Co-op bezeichnet werden, auch wenn diese hauptsächlich kollaborative Elemente beinhalten[130]. Steam, als eine der größten Online-Plattformen, führt beispielsweise kein Spiel auf, welches mit dem Zusatz *Kollaboration* ausgezeichnet ist[131]. Dabei wurde zunächst analysiert, welche Kernfaktoren für ein kooperatives Spiel herangezogen werden können respektive müssen. Zunächst ist eine Kamera-Einstellung von Nöten, die kooperatives Spielen erlaubt. Weiterhin müssen gemeinsame Puzzles oder Interaktionen mit dem selben Objekt möglich sein und/oder es müssen limitierte Ressourcen existieren, welche eine Absprache unter den Spielenden fördern. [Seif El-Nasr u. a., 2010]

Ferner werden u.a. die folgenden Entwurfsmuster für Kooperationsspiele genannt [Seif El-Nasr u. a., 2010; Rocha u. a., 2008]:

[130] Anm. der Autoren: Dies scheint daran zu liegen, dass die Spieler in den Spielen über eigene Avatare verfügen und prinzipiell auch gegen den Kollaborationsgedanken verstoßen können. Ein rein kollaboratives Spiel könnte das Steuern eines einzelnen Avatars durch mehrere Spieler sein, in der Form, dass einer z.B. die Fortbewegung und ein anderer das Umsehen steuert.

[131] Islas Sedano u. a. haben bei ihren Recherchen nach Spielen aus diesem Bereich die beiden Stichworte *cooperative* und *collaborative* verwendet [Islas Sedano u. a., 2013]. Ersteres führte zu 342 Ergebnissen, letzteres hingegen lediglich zu 63. Nach genauerer Untersuchung stellte sich heraus, dass von diesen 63 lediglich 18 ausschließlich aus kollaborativen Elementen bestanden, 24 weitere enthielten zumindest einige Möglichkeiten.

Ergänzungsprinzip: Dieses ist eines der am häufigsten verwendeten Muster in kooperativen Spielen. Es existieren verschiedene Rollen, welche sich gegenseitig ergänzen.

Synergien zwischen Fähigkeiten: Eine Rolle kann eine andere unterstützen, indem deren Fähigkeiten verbessert respektive verschlechtert werden[132].

Synergien zwischen Spielzielen: Als Beispiel nennen Seif El-Nasr u. a. hier das Spiel *Team Fortress 2*, in welchem ein so genannter *Pyro* das Ziel hat drei Gegner zu besiegen, während dieser von einem *Sanitäter* verstärkt[133] worden ist. Umgekehrt hat der *Sanitäter* das Ziel einen *Pyro* zu verstärken, welcher eine bestimmte Anzahl an Gegnern verbrennt. Damit kooperieren Spieler, um ihre Ziele zu erreichen.

Spezielle Regeln: Diese bezeichnen Regeln, welche verwendet werden um Kooperation in Teams zu forcieren. Der bekannteste Vertreter ist hier in Ego-Shootern die Abschaltung der Möglichkeit, dem eigenen Team zu schaden[134].

Es wurde beobachtet, dass die Spielenden am ehesten bei gemeinsamen Puzzeln und Zielen zusammengearbeitet und dabei Strategien entwickelt haben. Eine weitere, bemerkenswerte Beobachtung war, dass sich die Spielenden am häufigsten im Hinblick auf die Kamera gegenseitig behindert haben, wenn beispielsweise alle in entgegengesetzte Richtungen liefen. Zusammenfassend kann festgehalten werden, dass gemeinsame Ziele und Aufgaben den größten Einfluss haben, was unter dem Begriff *Ergänzungsprinzip* zusammenläuft. Besonders vorsichtig müssen Spieldesigner bei der Entwicklung der Kamera sein, da es hier am häufigsten zu Konfliktsituationen kommt respektive sich die Spielenden hier gegenseitig behindern können[135]. [Seif El-Nasr u. a., 2010]

Nach Analysen der Kommunikation zwischen Spielenden von Multiplayerspielen durch Friedl; Koivisto; Mulligan u. Patrovsky; Seay u. a.; Taylor; Cheung u. a. kann als eines der Hauptergebnisse festgehalten werden, dass es unwahrscheinlich ist, dass solch ein Spiel erfolgreich wird, wenn es keine Möglichkeit sowie keinen Anlass zur Kommunikation gibt [Korhonen u. Koivisto, 2007] (vgl. [Friedl, 2002; Koivisto, 2003; Mulligan u. Patrovsky, 2003; Seay u. a., 2004; Taylor, 2009]). Die Reichhaltigkeit der menschlichen Kommunikationsvielfalt, wie Augenkontakt, Mimik und Gesten ist in virtuellen Welten nicht möglich respektive erfährt dort keine technische Möglichkeit zur Entfaltung, sodass soziale Aspekte vernachlässigt werden [Magerkurth u. a., 2004]. Kollaboration ist dabei eng verknüpft mit Kommunikation, da es häufig Gründe bzw. Material zum kommunizieren bietet [Korhonen u. Koivi-

[132] Wird häufig in MMORPGs eingesetzt. Die Effekte werden hierbei als *buffen* bzw. *debuffen* bezeichnet.
[133] im Computerspieler-Jargon: *gebufft*
[134] auch bekannt als „friendly fire“
[135] Dieser Aspekt ist insbesondere in Verbindung mit Kapitel 10 und den Abschnitten 16.2 sowie 23 von Bedeutung.

sto, 2007]. Damit wird insbesondere der direkten Face-2-Face-Kommunikation eine wichtige Rolle zuteil[136]:

> *„[…] face-to-face is the richest form of information processing because it provides immediate feedback and with feedback understanding can be checked and interpretations corrected.“*

[Larsson u. Skårman, 2002, S. 1]

Ferner soll nach Zagal u. a. eines der Hauptziele für ein Multiplayer-Spiel sein, dass soziale Interaktion, Wettbewerb und Kooperation ermöglicht und gefördert werden [Zagal u. a., 2000]. Demnach sollten unter anderem folgende Punkte beim Design von Multiplayerspielen berücksichtigt werden (vgl. [Korhonen u. Koivisto, 2007]):

- Das Spiel ermöglicht Kommunikation [Cornett, 2004; Friedl, 2002; Koivisto, 2003; Mulligan u. Patrovsky, 2003; Seay u. a., 2004; Taylor, 2009].
- Es gibt Gründe zum Kommunizieren [Taylor, 2009; Zagal u. a., 2000].
- Das Spiel unterstützt Gruppen [Mulligan u. Patrovsky, 2003; Zagal u. a., 2000].
- Das Design muss abweichendes Verhalten minimieren[137] [Bruckman u. a., 1994; Jensen u. a., 2002].
- Das Design verbirgt die Nebeneffekte des Netzwerks [Chen u. a., 2006; Gautier u. Diot, 1998; Korhonen u. Koivisto, 2007; Pantel u. Wolf, 2002].

Laut Beznosyk u. a. kann zwischen eng- und lose gekoppelten Interaktionen respektive Kollaborationen unterschieden werden [Beznosyk u. a., 2012]. Sofern ein Spiel viel Wartezeit erfordert oder wenn die Aktionen eines Spielers oder einer Spielerin einen jeweils anderen Mitspieler oder eine Mitspielerin betreffen, so wird von enger Kopplung gesprochen. Sofern etwas unabhängiger gehandelt werden kann, wird von loser Kopplung gesprochen. Eng gekoppelte Spiele scheinen hierbei einen höheren Level der Aufregung („excitement“), eine höhere Bindung („engagement“) sowie einen emotionaleren Bezug zueinander zu fördern. Lose gekoppelte Spiele erwecken hingegen den Eindruck ein Gegeneinander zu forcieren. Zusätzliche Interaktionsmöglichkeiten führen zudem zu einem höheren Vergnügen in Spielen. [Beznosyk u. a., 2012]

[136] Mehr Informationen hierzu lassen sich unter dem Begriff der „Medienreichhaltigkeitstheorie“, insbesondere in der ersten Veröffentlichung hierzu aus dem Jahre 1984 [Lengel u. Daft, 1984] finden.

[137] Damit ist gemeint, dass Spielern nicht zu viele Freiheiten ermöglicht werden, sodass es z.B. möglich wird das Hauptziel, die Rettung einer Spielfigur und die damit einhergehende Tötung einer Spielkreatur zu vertauschen.

Nachdem ein grundsätzliches Verständnis dafür geschaffen wurde, auf welche Arten Interaktionen definiert werden können, geht es im Folgenden Kapitel darum, wie sich eine Interaktionen innerhalb des Spektrums der MR verorten und einsetzen lassen.

„Our point is that just because an image 'looks real' does not mean that the object being represented is real, and therefore the terminology we employ must be able carefully to reflect this difference."

(Milgram u. Kishino [Milgram u. Kishino, 1994, S. 7])

KAPITEL 7

Mixed Reality

Anwendungen, die mit dem Begriff *Mixed Reality* umschrieben werden können, bilden gewissermaßen eine Brücke zwischen realer und virtueller Welt (*Virtual Reality*). Wenn das Konzept des „virtuellen Kontinuums", vorgeschlagen von Milgram u. Kishino, zugrunde gelegt wird, so ergibt sich hierdurch ein unscharf abgegrenztes Spektrum (vgl. Abbildung 7.1).

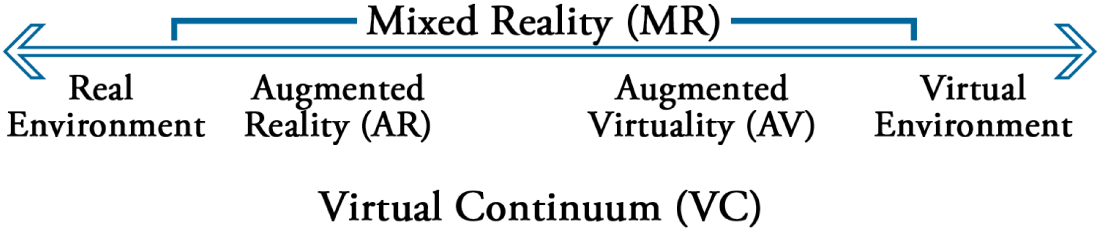


Abbildung 7.1.: Simplifizierte Veranschaulichung des Konzepts vom „virtuellen Kontinuum". Die Mixed Reality wird hierbei unter Verwendung unscharfer Grenzen zwischen der Realität und der (vollständig) virtuellen Realität aufgespannt.

nach [Milgram u. Kishino, 1994]

In diesem Spektrum liegen die realen Umgebungen an einem äußeren Ende, während vollständig virtuelle Umgebungen das andere Ende des Spektrums bilden. Mixed Reality-Anwendungen liegen innerhalb dieses Spektrums und sind überall dort vorhanden, wo virtuelle und reale Umgebungen und/oder Objekte in einem gemeinsamen Kontext präsentiert werden. Mit dem Begriff der *Virtual Reality* ist dabei eine vollständig synthetische Welt gemeint, welche die Partizipierenden umgibt und eine Möglichkeit zur immersiven Interaktion bietet. Diese virtuelle Welt ist nicht notwendigerweise an eine Imitation der realen Welt gebunden; so können beispielsweise durchaus physikalische Gesetze außer Kraft gesetzt werden, gleichfalls die Gesetze der Zeit und des Raumes im Allgemeinen [Milgram u. Kishino, 1994].

Ideen, welche mit den Vorstellungen von Wirklichkeit spielen, sind beispielsweise im Roman *Simulacron-3* von Galouye zu finden, dessen Tenor ebenfalls von dem relativ bekannten Film *The Matrix*[138] aufgegriffen wird [Galouye, 2011]. In diesem Film befindet sich die Menschheit in einer kollektiven Simulation, bei der die Gehirne direkt miteinander vernetzt werden, während sich die realen Körper in groß angelegten Feldern von Kapseln befinden und künstlich ernährt werden. Dabei ist den Menschen nicht bewusst, dass sie in einer Simulation leben: Die Human-Computer Interaction ist in diesem Beispiel zu einer vollständigen Immersion übergegangen und verschiebt die Grenze der Realitätswahrnehmung.

Das Ziel von MR-Anwendungen ist die nahtlose Verschmelzung von realem und virtuellem Raum und positioniert sich zwischen diesen Extrema – idealerweise einhergehend mit einem Design, welches die jeweils sinnvollste und beste Verwendung von Elementen der jeweiligen Welten vorsieht. Dieses Ziel hat mehrere Facetten: Zum einen sollen die damit verbundenen Interaktionen und Anwendungen bereichert werden, während Systemdesigner von den Möglichkeiten beider Welten profitieren wollen und zum anderen wird angestrebt, die für eine Nutzung des Systems notwendige Wissensbasis so klein und zugänglich wie möglich zu halten.

Die Entwicklung zeigt dabei, dass sich Benutzerschnittstellen konsequent weiterentwickeln[139] – weg von primär grafischen Bedienkonzepten, bei denen die eigentliche Interaktion in der Virtualität stattfindet[141], hin zu Schnittstellen, die auch haptische Komponenten berücksichtigen und den Nutzungskontext mit mehr Bedeutung versehen [Hornecker u. Buur, 2006].

Schnittstellen, die reale Interaktionselemente mit digitalen Repräsentationen verbinden, können auch als inhärent hybride Schnittstellen bezeichnet werden[142]. Rein Graphical User Interface (GUI)-basierte Lösungen sind zwar flexibler, stoßen jedoch bei komplexen Funktionen an ihre Grenzen[143]. Hybride Mischungen, bestehend aus physischen und virtuellen Elementen, bieten sich an diesen Stellen an und können die Interaktion einfacher gestalten respektive einfacher *begreifbar* machen. Der Einsatz von verschiedenen Interaktionsmetaphern und Ansätzen ist daher auch immer mit der Frage verbunden, wie Benutzern ein System und damit verbundene Aufgaben möglichst selbsterklärend vermittelt

[138] 1999, [Wachowski u. Wachowski, 1999a]

[139] Inzwischen wird jährlich die Konferenz Computer Supported Cooperative Work (CSCW) veranstaltet, auf welcher im Jahr 2012 ein Workshop mit dem Titel „Mixed Reality Games“ angeboten wurde [ACM, 2013b,a]. Dies zeigt unter anderem, wie sehr MR-Anwendungen in den Fokus der Entwickler und auch Konsumenten gerückt sind. Laut Montola werden seit etwa 10 Jahren[140] MR-Prototypen erstellt und untersucht [Montola, 2011].

[140] zum Stand des Papers

[141] wie beispielsweise beim WIMP-Prinzip

[142] Ein Beispiel hierfür sind Tangible User Interfaces (TUIs). Diese werden in Abschnitt 12.3 auf Seite 221 näher vorgestellt.

[143] Wie beispielsweise die Navigation in dreidimensionalen Modellierungsumgebungen.

respektive die Durchführung einer Aufgabe durch passende Technologie unterstützt werden kann[144]. [Israel u. a., 2012, S. 40]

Um zu verstehen wie sich reale Objekte von digitalen Objekten und letztlich von digitalen Systemen unterscheiden, ist es sinnvoll, diese Begriffe näher zu betrachten.

7.1. Taxonomie der Mixed Reality

Im Gegensatz zur virtuellen Realität ist es bei der Mixed Reality nicht immer scharf abgrenzbar, wann eine Komponente oder Interaktion rein *digital* oder, im Gegensatz dazu, vollständig *real*[145] existiert. Unterscheidungen können hier durch mehrere Ansätze erfolgen, die idealerweise miteinander kombiniert werden, um eine definitive Taxonomie zu erreichen. [Milgram u. Kishino, 1994]

Die erste Möglichkeit der Unterscheidung liegt in der Definition, ob ein Objekt real, d.h. physikalisch erfassbar, vorhanden ist oder nicht [Milgram u. Kishino, 1994, S. 6]. Am Beispiel des 3D-Drucks kann gezeigt werden, dass diese auf den ersten Blick klare Trennung nicht in jedem Fall hinreichend sein muss. Ein Objekt, welches tatsächlich rein virtuell existiert, in Form von Koordinaten und Oberflächeninformationen, kann durch ein dreidimensionales Druckverfahren ebenfalls in ein reales, begreifbares Objekt überführt werden. Ein anderes Beispiel ist der eigene Kontostand auf der Bank: Obwohl die Zahlen nur rein virtuell existieren, haben diese doch eine sehr reale und konkrete Bedeutung, welche eine direkte Auswirkungen auf die Realität hat. Selbst, wenn diese Zahlen an einem Geldautomaten in real greifbare Objekte in Form von Geldscheinen überführt werden, behalten diese dennoch eine überwiegend virtuelle Komponente. Das rührt daher, dass die Geldscheine in diesem Fall stellvertretend für eine bestimmte Arbeitsleistung stehen und sich in bestimme, andere Gegenleistungen umtauschen lassen. Die Frage nach der Realität eines Objektes oder einer Komponente liegt demnach nicht nur ausschließlich in seiner Physis, sondern auch in seiner Bedeutung und Wechselwirkung mit der Realität.

Ein weiterer Ansatz zur Distinktion realer und virtueller Objekte in einer Mixed Reality-Anwendung besteht durch die Betrachtung des jeweiligen Anwendungszwecks. Komponenten könnten als *real* bezeichnet werden, wenn sie vom System ausgewertet werden oder als Interaktionsschnittstelle dienen

[144] vgl. zu verschiedenen Ansätzen der Mixed Reality, Augmented Reality und Tangible User Interfaces auch [Milgram u. Kishino, 1994; Ishii u. Ullmer, 1997; Azuma u. a., 2001].

[145]

> *„What is real? How do you define 'real'? If you're talking about what you can feel, what you can smell, what you can taste and see, then 'real' is simply electrical signals interpreted by your brain."*
>
> [Wachowski u. Wachowski, 1999b, Figur des „Morpheus"]

und demnach „benutzbar“ sind [Israel u. a., 2012, S. 43]. Dies kann ebenfalls eine Bewegungssteuerung oder eine Sprachsteuerung sein, denn letztlich werden die Bewegungen und/oder die Spracheingaben physikalisch von einer Kamera bzw. einem Mikrofon erfasst. Ein Beispiel ist die in diesem Projekt verwendete Steuerung durch Smartphones, welche die Eingaben entgegennehmen und an das System weiterleiten. Ihrerseits können die Smartphones auch Informationen aus dem System übertragen, da sie Dank des Touchscreens, Vibratoren, Sensoren, Mikrofonen und Lautsprecher als ambivalentes Interaktionsgerät ausgelegt sind.

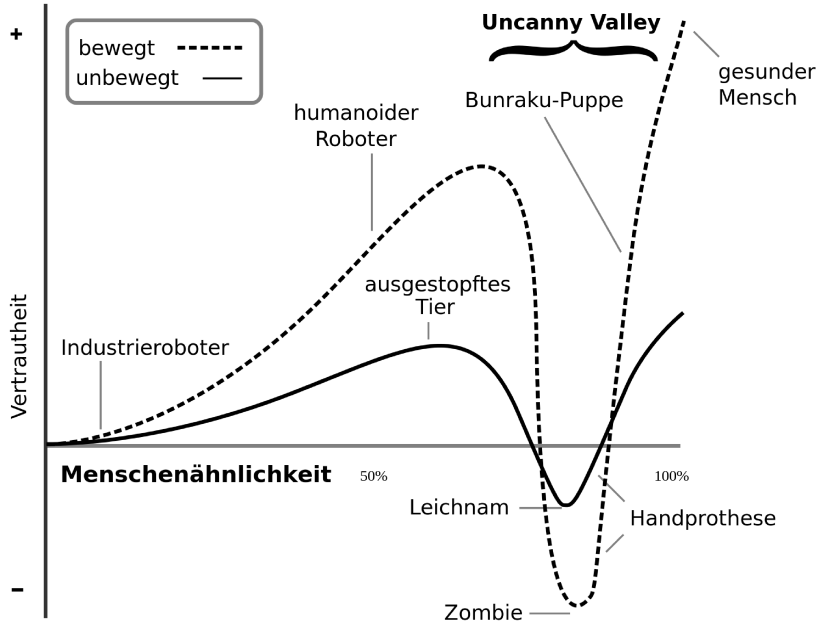


Abbildung 7.2.: Die Abbildung zeigt das als „Uncanney Valley“ bekannte Phänomen der Akzeptanz von humanoid-ähnlichen, künstlich erzeugten Objekten. Eigentlich wird erwartet, dass Objekte linear stetig mit ansteigendem Anthropomorphismus eine ebenso linear ansteigende Akzeptanz erfahren. Stattdessen stellt sich etwa im hinteren Drittel ein massiver Einbruch dar. Das zieht unter anderem die Konsequenz nach sich, dass ein *fast* fotorealistischer Film, der unter Zuhilfenahme von CGI-Techniken erzeugt wurde, als weniger vertraut wahrgenommen wird als ein Zeichentrickfilm, in dem sich Figuren menschenähnlich verhalten.

Grafik von [K, 2008], erstellt nach [MacDorman, 2005].

Die zweite Möglichkeit einer Unterscheidung liegt in der Qualität einer Verbildlichung der Realität [Milgram u. Kishino, 1994, S. 7]. In diesem Zusammenhang soll zunächst der als „Uncanny Valley“[146] bekannte Effekt beschrieben werden (vgl. Abbildung 7.2). Bei diesem empirisch messbaren Phänomen kann gezeigt werden, dass die Akzeptanz einer künstlichen Figur nicht linear stetig steigend mit einem stetig steigenden Anthropomorphismus[147] einhergeht. Dies ist insbesondere bemerkenswert in der Hinsicht, dass Probanden sogar abstrakten, höchstens näherungsweise an Humanoide erinnernde Figuren[148] eine höhere Akzeptanz entgegenbringen können, als mit viel Aufwand errechneten, menschenähnlichen CGI-Modellen. Die Ursachen hierfür werden weiterhin erforscht. Ein mögliches Erklärungsmodell ist jedoch in verschiedenen Erwartungshaltungen der menschlichen Psychologie begründet. Je realistischer ein Objekt dem Menschen scheint, desto eher werden kleinste Fehler wahrgenommen. Oftmals geschieht es nur noch unterbewusst, dass geringfügige Unstimmigkeiten einen Hinweis darauf geben, dass das Objekt nicht „echt“ ist. Ferner können Probanden menschenähnlichen Robotern sogar mit Angst begegnen: *„I dislike it. [It looks] scary and dead“* [MacDorman, 2005, S. 6]). Dies ist bemerkenswert, da die gleichen Probanden nicht immer sagen konnten, ob das, was sie gerade sahen, *kein* Mensch war. Das *Uncanny Valley* kann nur mit überproportional viel Aufwand überwunden werden[149]. [MacDorman, 2005]

Neben der Problematik des Uncanny Valley ist das Problem der *Cornell Box* ein weiterer Indikator für die Schwierigkeit, echten Fotorealismus zu erzeugen (vgl. [Goral u. a., 1984]). Bei dem von den Autoren Goral u. a. erdachten Test wird eine Fotografie mit einem computergenerierten Bild verglichen, welches die fotografierte Szene möglichst exakt nachbilden soll. Das Differenzbild dieser Bilder zeigt dabei die Schwierigkeit, physikalisch korrekte Lichtverteilung zu simulieren respektive nachzubilden.

Die Quintessenz der hier vorgestellten Teilprobleme ist, dass mit Hilfe von Virtualität versucht wird, sich der Realität anzunähern. Wenn alle Teilprobleme gelöst wurden und es gelungen ist, die Realität zu kopieren (oder vorher nicht existente Objekte real erscheinen zu lassen), so kann es jedoch im Anschluss ebenso schwierig sein zu beweisen, dass etwas *nicht* real ist, wie auch Milgram u. Kishino resümieren:

[146] zu deutsch „Unheimliches Tal“, Übersetzung der Autoren

[147] Anthropomorphismus bezeichnet, vereinfacht ausgedrückt, die Menschenähnlichkeit eines Objektes. Exakter formuliert ist ein Objekt besonders anthromorph, wenn es viele humanoide Eigenschaften vereint.

[148] wie beispielsweise Kuscheltiere

[149] Ein populäres Beispiel für einen Anspruch an Fotorealismus ist der Film *Final Fantasy: The Spirits within*[150]. Surman untersucht den Selbstanspruch des Films eines möglichst realistischen Eindrucks[151] und findet mögliche Erklärungen für dessen kommerzielles Scheitern[152]. Auch wenn auch das *Uncanny Valley* nicht namentlich genannt wird, so werden dennoch dessen mögliche Ursachen analysiert.

[150] 2001, [Sakaguchi u. Sakakibara, 2001]

[151] vgl. [Surman, 2003, S. 8ff.]

[152] vgl. [Surman, 2003, S. 6]

> *„Our point is that just because an image 'looks real' does not mean that the object being represented is real, and therefore the terminology we employ must be able carefully to reflect this difference."*

[Milgram u. Kishino, 1994, S. 7]

Die dritte Möglichkeit einer Distinktion ist letztlich ein Vorschlag von Milgram u. Kishino, die Optik zur Hilfe zu nehmen: Ein Bild ist als real zu bezeichnen, wenn es an der Stelle beleuchtet wird, an der es sich *scheinbar* befindet. Dies bedeutet, dass der Einfluss einer Lichtquelle auf ein Objekt physikalisch korrekt dort stattfindet, wo sich das Objekt auch optisch befindet beziehungsweise rezipiert wird. Wird ein Bild jedoch beispielsweise mit Methoden der Holografie in eine reale Umgebung hinein projiziert, ist es als virtuelles Bild zu bezeichnen, da der Einfluss der Beleuchtung nicht mehr stimmt: Ein Objekt wird an Punkt A beleuchtet und anschließend auf Position B projiziert. Reale Bilder können beispielsweise ebenfalls auf einem Bildschirm entstehen und obwohl das Bildes innerhalb seiner Projektion unstrittigerweise virtueller Natur ist, wäre das erzeugte Bild im Sinne der Definition real, da es gleichermaßen auf dem Bildschirm entsteht und auch dort beleuchtet wird.

7.1.1. Zusammenfassung

Reale Objekte im Sinne der Mixed Reality können als materielle, physisch wahrnehmbare Objekte definiert werden, die von einem digitalen System erfasst und ausgewertet werden können und mit diesem in Beziehung stehen. Andererseits können reale Objekte retrospektiv Systemzustände vermitteln, wie es beispielsweise mit Hilfe von Smartphones möglich ist.

Virtuelle Objekte sind von Benutzern wahrnehmbare Objekte, die in einem (digitalen) Computersystem gespeichert sind und mit menschlichen Sinnen wahrgenommen werden können (z.B. visuell über einen Bildschirm). Der Nutzer interagiert nicht direkt mit virtuellen Objekten, selbst dann nicht, wenn die Eingaben über einen Touchscreen erfolgen. In diesem Fall dient der Touchscreen als physikalisches Eingabemedium und vermittelnde Schicht zwischen Realität und Virtualität.

Digitale Systeme letztlich führen reale und virtuelle Objekte in einem Systemen zusammen, in diesem Fall in einem Mixed Reality (MR)-System. MR-Anwendungen können in einem übergangslosen Spektrum, dem *virtuellen Kontinuum* verortet werden[153]. (vgl. auch [Israel u. a., 2012, S. 43])

[153] vgl. hierzu auch [Milgram u. Kishino, 1994] sowie Abbildung 7.1 auf Seite 139

Nachdem mit dieser Taxonomie ein grundlegendes Verständnis dafür geschaffen wurde, wie die Begriffe der Realität und Virtualität mit Mixed Reality-Anwendungen in Beziehung gebracht werden können, ist es naheliegend zu definieren, wie Aufgaben und Funktionen zwischen Mensch und Maschine aufgeteilt werden können. Es mag intuitiv ersichtlich sein, dass es einige Aufgaben gibt, die in der Virtualität umzusetzen sind und wiederum andere, für die ein Ansatz unter Verwendung realer Objekte als der sinnvollere erscheint. Dennoch ist dieser scheinbaren Intuition entgegen zu halten, dass seit über 50 Jahren Versuche der Einordnung existieren, welche Kompetenzen eher im Bereich der Maschinen respektive des *Virtuellen* und welche im Bereich des Menschen beziehungsweise des *Realen* liegen.

Das durch Chapanis u. a. eingeführte Konzept der so genannten „Machines Are Better At - Men Are Better At (MABA-MABA)-Listen“ wurde unter anderem von Hurtienne u. a. aufgegriffen, erweitert und letztlich speziell im Hinblick auf Interaktionen und Mixed Reality-Anwendungen ausführlicher von Israel u. a. erörtert. Diese neuen Listen nennen sich „Digital Is Better At - Physical Is Better At (DIBA-PIBA)“ und geben nicht nur weitergehende Hilfestellung bei der Auswahl konkreter Zuständigkeitsbereiche, sondern helfen insgesamt besser zu verstehen, wie digitale und reale Systeme miteinander in Korrelation gesetzt werden können. Im Rahmen dieser Arbeit sollen diese Listen aufgegriffen und präzisiert werden, um den neuen Begriff der Physical Is Good At - Digital Is Good At (PIGA-DIGA)-Listen einzuführen.

7.2. Das Prinzip von DIBA-PIBA

Der Kern der DIBA-PIBA-Listen liegt in einer (wertenden) Gegenüberstellung und ist unter anderem deswegen mit einem Bindestrich notiert. Dabei werden verschiedene Aspekte der namensgebend physikalischen Welt mit denen der digitalen Welt gegenübergestellt, wobei sich die Thematiken in fünf Abschnitte unterteilen lassen, deren Divergenz wiederum durch einen Bindestrich gekennzeichnet werden kann (vgl. [Israel u. a., 2012, S. 44-45]):

Physische Materie – Binäre Substanz: Physische Objekte unterliegen den Naturgesetzen, während virtuelle Objekte dies nicht tun. Letztere liegen lediglich in binärer Form vor und sind den Algorithmen und physikalischen Grenzen innerhalb des digitalen Systems unterworfen.

Räumliche Ausdehnung – Nicht-räumliche Ausdehnung: Die räumliche Ausdehnung beschreibt den realen Raum um den Nutzer herum, während die nicht-räumliche Ausdehnung in der digi-

talen Welt so verstanden werden kann, dass hier kein physikalisch messbarer Platz verbraucht wird[154] und somit „unendlich groß“ oder auch „unendlich klein“ sein kann.

Räumliche Position – Allgegenwärtigkeit: Reale Objekte befinden sich an einem örtlich beschreibbaren und wiederauffindbaren Ort, während virtuelle Objekte das Prinzip der *Allgegenwärtigkeit* zur Eigenschaft haben: Diese können innerhalb eines digitalen Systems „überall und nirgends“ gleichzeitig vorhanden sein.

Räumliche Ortsbezogenheit – Referenzielle Ortsbezogenheit: Ort, Kontext und Lage bilden bei realen Objekten wichtige Indikatoren zur Bedeutung. Bei virtuellen Objekten ist es eher der Bezug[155] des Objektes zu anderen Objekten; gemeint ist hiermit die referenziell induzierte Identität. Dies bedeutet, dass virtuelle Objekte in ihrer Bedeutung dadurch mitbestimmt oder festgelegt werden, wie sie mit anderen Objekten verknüpft sind.

Analoges Interaktionsalphabet – Digitales Interaktionsalphabet: Die Manipulation realer Objekte erzeugt ein physisches Feedback, beispielsweise durch Verformung, Temperatur, Änderung des Ortes oder des Gewichtes. Virtuelle Objekte hingegen existieren nur symbolisch und müssen daher auch nicht über eine physikalisch sinnvolle Art manipuliert werden. Ein Beispiel der Manipulationsarten von virtuellen Objekten ist die im vorherigen Verlauf dieser Arbeit bereits erwähnte *Metapherhaftigkeit*[156]. Ein virtuell vorhandenes Auto kann beispielsweise per Maus zum Fahren bewegt werden, durch Bewegungssteuerung oder auch durch direkte Eingabe eines simulierten physikalischen Impulses in ein Textfeld.

Die Kernaussagen der detaillierten Beschreibungen der jeweiligen Vor- und Nachteile von realen respektive virtuellen Komponenten, insbesondere beim Einsatz innerhalb von Mixed Reality-Umgebungen, sind in den folgenden zwei Abschnitten ausgearbeitet und zusammengefasst.

7.3. PIGA - „Physical Is Good At“

Mit der Analyse einhergehend, was an physisch vorhandenen Objekten und Systemen „besser“ ist, muss verständlicherweise immer ein Bezug hergestellt werden respektive ein Maßstab angelegt werden, unter welchen Gesichtspunkten ein Vergleich angestrebt wird – und womit. Zu sagen: „Ein Computer ist besser als ein Buch“ hat als alleinstehende Aussage keinen Informationsgehalt. Erst, wenn ein Bezug hergestellt wird, beispielsweise in der Form: „Ein Computer erlaubt die schnelle Indexierung

[154] Von unterschiedlichen Magnetdotierungen in Speichermedien einmal abgesehen.
[155] anders formuliert: die Referenzen respektive *Links*
[156] vgl. Abschnitt 6.1 auf Seite 124

und Durchsuchung umfangreicher Dokumente nach eingegebenen Stichworten und ist daher zur Recherche besser geeignet als ein Buch für die gleiche Aufgabenlösung“ - ergibt sich aus der Aussage eine Kontextinformation und ein Vergleich wird möglich[157].

Die Bezeichnung Physical Is Better At (PIBA), wie unter anderem von Hurtienne u. a. und Israel u. a. verwendet, lässt daher Raum für Präzisierung [Hurtienne u. a., 2008; Israel u. a., 2012]. Ein möglicher Vorschlag soll hier unterbreitet werden: „PIGA - Physical Is Good At“[158] – denn ein Vergleich sollte immer kontextbezogen angestrebt werden: Ein reales Objekt ist nicht „besser“ sondern zunächst einmal *gut* in bestimmten Aspekten; während sich das „besser“ erst durch einen Vergleich ergibt. Der neu eingeführte Begriff der PIGA-DIGA-Listen versucht daher keine Wertung vorwegzunehmen, sondern erst durch Gegenüberstellung zu entwickeln.

Bei der Betrachtung physikalischer Objekte fällt zunächst auf, dass diese inhärent intrinsische Eigenschaften besitzen, welche sich einem Beobachter direkt erschließen oder sich erschließen lassen [Israel u. a., 2012, S. 45]. Dazu können beispielsweise Merkmale wie Beschaffenheit oder Gewicht zählen, jedoch auch Form und Farbe. An dieser Stelle ist der Begriff der „Affordanz“ von Bedeutung, ein Konzept, welches von Gibson entwickelt und erstmalig in seinem „ökologischen Ansatz in der visuellen Wahrnehmung“ vorgestellt wurde [Gibson, 1982]. Affordanzen beschreiben die unmittelbar wahrgenommenen Eigenschaften eines Objekts, unabhängig von Erfahrung oder kulturellem Hintergrund. Schnee beispielsweise ist ein Material, dessen Eigenschaften mess- und reproduzierbar ohne Vorwissen von jedem Menschen herausgefunden werden können[159]: Schnee ist kalt und von weißer Farbe. Er verwandelt sich in Wasser, wenn er zu lange in der Hand gehalten wird (erwärmt wird), kann allerdings durch Ausübung von Druck gehärtet werden. Der Begriff der Affordanz kann in erweiterter Form ebenfalls in der digitalen Welt angewandt werden[160].

Eigenschaften wie die Lage im Raum, Position oder auch der Aggregatzustand lassen sich bei physikalischen Objekten ebenfalls verhältnismäßig einfach manipulieren[161] [Israel u. a., 2012, S. 52]. Anders als bei virtuellen Objekten können reale Objekte auch vollständig aus dem eigentlichen Interaktionsraum entfernt- und an anderer Stelle verwendet werden. Ein Beispiel hierfür wäre ein Stift, den jemand ausleiht und an einem anderen Ort weiter benutzt.

Im Gegensatz zu virtuellen Objekten können reale Objekte zeitgleich als Präsentations- wie auch Interaktionsschicht dienen. Ein anschauliches Beispiel liefern Israel u. a. mit einem Schachspiel: Es stellt

[157] In einer Randüberlegung kann auch festgehalten werden, dass qualitative Aussagen eines Vergleichs sehr stark von der Qualität des gewählten Kontext abhängig sind. Anders ausgedrückt: Mit schlechtem Kontext lässt sich vieles kaputt reden.
[158] ebenso später „DIGA – Digital Is Good At“
[159] Gesetzt dem Fall der Mensch ist gesund und weist alle dafür erforderlichen Körpermerkmale auf.
[160] vgl. Abschnitt 7.4 auf Seite 151
[161] Einzige Beschränkung ist die Physik und das menschliche Leistungsvermögen oder das Vorhandensein eventuell benötigter Hilfsmittel. Ein Schiff kann beispielsweise in den meisten Fällen nur mit Hilfe eines Hafendocks bewegt werden und nicht aus eigener, menschlicher Kraft heraus.

gleichermaßen den Ort der Handlung, die physische Repräsentation sowie den aktuellen Spielzustand dar und verbindet auf diese Weise kognitive und physische Wahrnehmung [Israel u. a., 2012, S. 46]. In diese Überlegung spielt mit hinein, dass reale Objekte direkt voneinander unterschieden werden können. Wahrnehmungsraum und Handlungsraum bilden somit eine Einheit - anders als beispielsweise bei virtuellen Positionsveränderungen eines Mauszeigers, die durch ein reales Objekt angestoßen werden [Israel u. a., 2012, S. 52]. Beim Schachspiel ist intuitiv verständlich, dass keine zwei Figuren auf dem gleichen Feld *ineinander* stehen können. Reale Objekte haben eine klare räumliche Position und müssen sich physikalischen Naturgesetzen beugen. Das ist insbesondere dann von Vorteil, wenn ein reale Aufgabe von mehreren Menschen bearbeitet wird, da sich Zustände einfacher persistieren lassen. Wenn real ein bestimmter Stand erreicht wird, kann dieser leichter von weiteren Menschen adaptiert werden. Das schafft Klarheit von Arbeitsabläufen. Virtuell hingegen kann verteiltes und/oder paralleles Arbeiten gewünscht sein. [Israel u. a., 2012, S. 46-48]

Reale Objekte befinden sich - im Gegensatz zu virtuellen Objekten - in einer direkten Abhängigkeit von äußeren Bedingungen. Schwerkraft, Trägheit, Fliehkraft und andere physikalische Phänomene sind in der realen Welt unmittelbar vorhanden und müssen in einer virtuellen Welt erst aufwändig simuliert werden. Ebenfalls können virtuelle Objekte nicht *für sich* existieren sondern lediglich in einem Kontext, d.h. in einem bestimmten Programm, Betriebssystem und/oder in Verbindung mit bestimmter Peripherie oder mit bestimmten Systemanforderungen. Reale Objekte haben keinerlei Abhängigkeit in ihrer Existenz[162] und können darüber hinaus auch in schwierigeren Umweltbedingungen eingesetzt werden, obwohl auch hier Einschränkungen bzw. Besonderheiten existieren, wie dass Elektronik für extreme Kälte, Wärme oder Flüssigkeiten anfällig ist[163]. [Israel u. a., 2012, S. 29-50]

Anders als bei virtuellen Objekten im virtuellen Raum verhalten sich reale Objekte im realen Raum immer skalar zu den auf sie angewandten Kräften. Auch mit dem Aufkommen von Multi-Touch-Technologie gilt die Aufhebung der Trennung zwischen Wahrnehmung und Handlung nur beschränkt, denn auch hier entspricht die dargestellte Interaktion einer Simulation und ist in ihrem Verhalten beliebig anpassbar. Die reale Wirklichkeit kann somit durch Manipulation virtueller Parameter beeinflusst werden. Im Gegenzug hierzu verhält sich ein virtuelles Ereignis nicht unbedingt proportional zu einer real ausgeführten Handlung[164]. [Israel u. a., 2012, S. 53]

[162] Hiermit ist nicht gemeint, dass es natürlich auch innerhalb der Erde Einschränkungen gibt. Menschen könnten beispielsweise nicht ohne weitere Hilfsmittel durchgängig in der Antarktis überleben.

[163] Diese Anfälligkeit stellt gleichzeitig ein Manko beim Einsatz digitaler und elektronischer Systeme in extremen Umweltbedingungen dar.

[164] Ein einfaches Beispiel hierfür ist die Translation eines Mauszeigers in Abhängigkeit einer Mausbewegung. Die Strecke, die eine Maus über ein Mauspad geschoben wird oder die durch das Rollen eines Trackballs zurückgelegt wird, muss nicht im Millimeterbereich der auf einem Bildschirm zurückgelegten Strecke des Mauszeigers entsprechen.

Dies zieht in Konsequenz ebenfalls nach sich, dass der globale Bezugsrahmen im physischen Raum für alle Teilnehmerinnen und Teilnehmer identisch ist[165], während der Bezugsrahmen im Virtuellen für jeden Teilnehmer durchaus unterschiedlich skaliert werden kann. Unter anderem hat das Auswirkungen auf kollaboratives Arbeiten an digitalen Systemen: Es ist zwar technisch möglich, dass alle Beteiligten eine identische Darstellung und Vorstellung vom gemeinsamen Interaktionsraum vermittelt bekommen, beispielsweise durch ein gemeinsam betrachtetes Display, letztlich obliegt die Kontrolle dieser Möglichkeiten jedoch keinen Naturgesetzen, sondern denen des Systemdesigners [Rogers u. a., 2009]. Wenn das System so ausgelegt ist, dass mehrere Teilnehmer mit verschiedenen Endgeräten auf ein Mixed Reality-System zugreifen, können diese auch unterschiedliche Repräsentationen des gemeinsamen Kontextes erhalten. [Israel u. a., 2012, S. 53]

Abschließend soll an dieser Stelle der Aspekt der räumlichen Ortsbezogenheit erwähnt werden. Reale Objekte tragen semantische Botschaften in sich, welche einerseits aus physikalischen Gesetzmäßigkeiten und andererseits aus Extrapolation von individuellen Erfahrungen zusammensetzt werden können[166]. In Bezug auf reale Objekte lässt sich feststellen, dass gewisse Gesetzmäßigkeiten immer zutreffen. In einer Szene des schon einmal als Beispiel herangezogenen[167] Films *Minority Report* stellt ein Protagonist gegenüber dem Hauptdarsteller in Frage, ob ein System zur Prädiktion von in der Zukunft liegenden Verbrechen immer richtig läge, da die Verbrechen schließlich noch gar nicht geschehen seien. Als Gegenargument lässt der Hauptdarsteller einen Apfel auf einer Fläche hinunterrollen, welcher wiederum vom Gesprächspartner in einer Instinkthandlung aufgefangen wird. Auf die Frage hin, warum dieser den Apfel aufgefangen hätte, erwidert dieser, er wäre sonst zu Boden gefallen. Dieses Beispiel steht für erlerntes und Objekten innewohnendes Verhalten, welches durch Erfahrung weiter verfestigt wurde, bis es sich schließlich zu einer subjektiven Gewissheit formt. Gleichzeitig sind Menschen in der Lage, reale Objekte semantisch zu dem Ort in Bezug zu setzen, an dem diese aufgefunden werden: Selbst, wenn die Bedeutung eines Gegenstandes nicht direkt intuitiv klar wird, kann jedoch der Kontext, beispielsweise das Auffinden des Gegenstandes nur in Badezimmern, einen Hinweis auf den Sinn und die angedachte Nutzung geben. [Israel u. a., 2012, S. 56-56]

[165] Es soll an dieser Stelle darauf hingewiesen werden, dass dieser globale Bezugsrahmen in der Praxis oftmals nicht in vollkommen identischer Weise reproduziert werden kann. Es kann sich als extrem schwierig herausstellen zwei Orte auf der Welt zu finden, die zwar entfernt voneinander liegen, wo aber dennoch exakt die gleichen Bedingungen gelten, wie Luftdruck, Gravitation, molekulare Luftzusammensetzung und die Ausrichtung des Erdmagnetfelds.

[166] Individuelle Erfahrungen können ebenfalls mit virtuellen Objekten gemacht werden und sind nicht auf den physikalischen Raum beschränkt.

[167] vgl. Abschnitt 6.3 auf Seite 133

7.4. DIGA - „Digital Is Good At“

Nach einer Vorstellung der realen Objekte folgt an dieser Stelle die Betrachtung in umgekehrter Weise, ausgehend von der virtuellen Welt und im Hinblick auf Vorteile digitaler Systeme und Komponenten gegenüber den jeweiligen realen Entsprechungen.

Zunächst fällt auf, dass visuelle Eigenschaften virtueller Objekte beliebig verändert werden können – was, je nach Konzeption und Kontext des digitalen Systems, ohne zusätzlichen Aufwand erfolgen kann. Daher ist es insbesondere reizvoll, Repräsentationen einer veränderlichen Umgebung respektive Objekte mit (zur Laufzeit) veränderlichen Eigenschaften zu virtualisieren (vgl. [Spur u. Krause, 1997]). Der Grund für die hohe Flexibilität einer visuellen Darstellung liegt in der algorithmischen Berechenbarkeit und den vielzähligen Eingriffsmöglichkeiten in den Darstellungsprozess innerhalb der so genannten Renderpipeline (vgl. Abbildung 7.3).

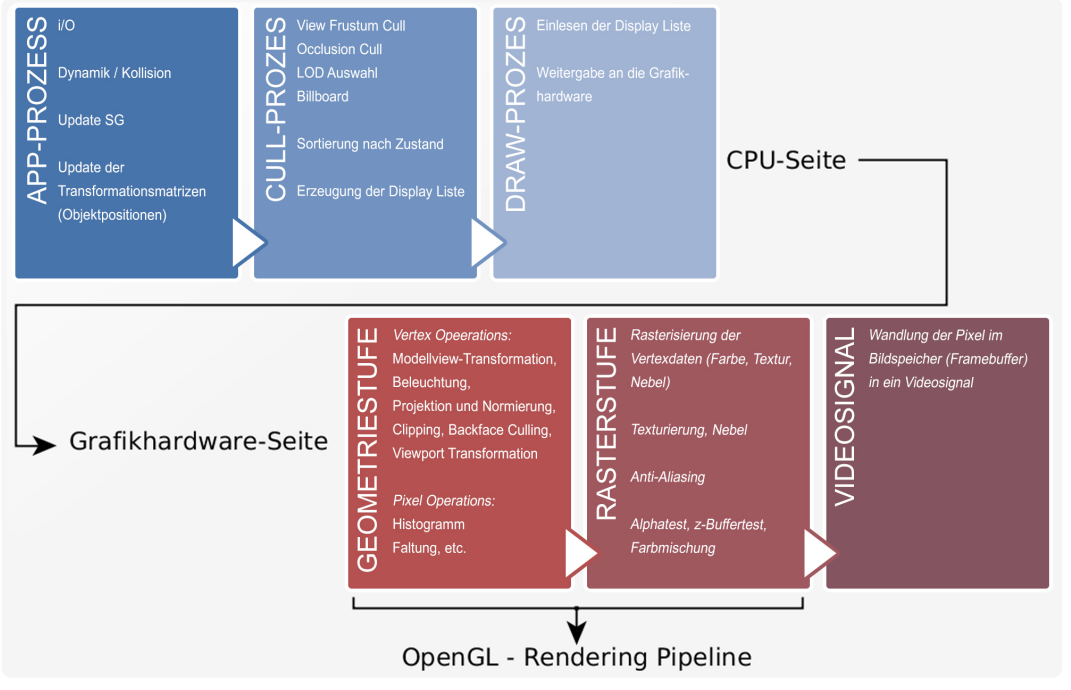


Abbildung 7.3.: Die hier dargestellte Renderpipeline zur Berechnung und Ausgabe digitaler errechneter Bilder zeigt auf, an welchen Punkten in den Berechnungsprozess eingegriffen werden kann.

aus [Heisenberg, 2012d, S. 39]

Auf Seite 148 wurde unter anderem beschrieben, dass physikalische Prozesse und Naturkonstanten in der realen Welt direkt vorhanden sind und in einer virtuellen Umgebung erst aufwändig simuliert werden müssten. Diesem Nachteil steht entgegen, dass *sobald* ein solches Simulationssystem existiert[168], Einwirkungen von Kräften nach Belieben verändert, repliziert und angewandt werden können. Um an dieser Stelle ein besseres Verständnis zu ermöglichen, sei das Beispiel eines ansonsten praktisch unbeweglichen Objekts gegeben: eine Bohrinsel. Die Verlegung einer Bohrinsel respektive der Abbau und Aufbau einer solchen an neuer Stelle stellt einen hohen logistischen Aufwand dar. Die Bewegung in einer virtuellen Umgebung ist dahingegen problemlos möglich. Hierfür wird nichts weiter als eine Interaktionsmechanik benötigt, welche eine solche Bewegung innerhalb eines virtuellen Ozeans ermöglicht, beispielsweise durch den Klick einer Computermaus. [Israel u. a., 2012, S. 58-59]

Oftmals wird beim Design von virtuellen Oberflächen, analog zur physikalischen Welt, ebenfalls von „Affordanzen“ gesprochen. Gemeint sind damit jedoch meist Konventionen und Erfahrungen: Die beispielsweise oft anzutreffende Mechanik von Fenstern auf einem Bildschirm, welche hin- und her geschoben werden können und ein „X“ am oberen Rand besitzen, signalisieren, dass dieses Fenster mit einem Klick darauf geschlossen werden kann. Dies ist jedoch keine Affordanz sondern eine Konvention und ein erlerntes Verhalten; die Eigenschaft und Interaktionsfähigkeit oder allgemein das Erkennen des Buttons als integrationsfähiges Element ist keine intrinsische Eigenschaft, die unmittelbar vorhanden ist [Norman, 1999, S. 40]. Dennoch lässt sich der Begriff der Affordanz aufgreifen und auf digitale Systeme anwenden. Norman nimmt hierfür die Definition von Gibson[169] als Grundlage und erweitert diese um den Begriff der „wahrgenommenen Affordanz“[170]. Gerade im Bereich der digitalen Objekte kann praktisch nur mit wahrgenommenen (d.h. visuellen) Affordanzen gearbeitet werden, die jedoch unabhängig von den tatsächlichen Affordanzen des Systems betrachtet werden müssen. Im Bezug auf berührungssensitive Bildschirme schreibt Norman beispielsweise:

> *„Although all screens within reaching distance afford touching, only some can detect the touch and respond to it. Thus, if the display does not have a touch-sensitive screen, the screen still affords touching, but it has no effect on the computer system.“*

[Norman, 1999, S. 39]

Dies ist ein Aspekt, der beim Design von Interaktionen, die Gebrauch von einer berührungssensitiven Fläche machen, bedacht werden sollte. Vorerfahrungen der Nutzerinnen und Nutzer lassen sich an diesen Stellen nicht aus einer natürlichen Intention ableiten, sondern erfordern persönliche Vorerfahrung und erlernte Konventionen. Neuartige Interaktionsformen im Bereich der Mixed Reality sollten daher

[168] Oder auf vorhandene Systeme zurückgegriffen wird, wie beispielsweise die *Havok*-Technologie [Havok.com Inc, 2014]. Hierbei handelt es sich um eine Software, die auf Physik-Simulationen spezialisiert ist und sich in andere Projekte integrieren lässt.
[169] vgl. auch Abschnitt 7.3 auf Seite 147
[170] engl. „perceived affordances“ [Norman, 1999, S. 39], Übersetzung der Autoren

Informationen aus beiden Welten einfließen lassen: Erlerntes Verhalten (meist im Zusammenhang mit virtuellen Welten) und Affordanzen, die sich aus den intrinsischen Eigenschaften realer Objekte ergeben.

Ein Vorteil digitaler Systeme ist, dass virtuelle Objekte im Allgemeinen schneller verarbeitet werden können als reale Objekte. Die Modellierung eines menschlichen Körpers im Computer beispielsweise ist, je nach gewünschtem Realitätsgrad, vergleichsweise aufwändig. Wenn das Modell mit allen gewünschten Eigenschaften jedoch erst einmal existiert, kann es beliebig kopiert, vervielfältigt, verändert und skaliert werden. Objektvariationen mit vorher bestimmten gültigen Spielräumen sind leicht umsetzbar, beispielsweise durch Methoden des Software-Engineering (so genannte *Konstruktoren*). Der Zustand virtueller Objekte kann persistiert, wiederhergestellt, gelöscht und repliziert werden. Dies erlaubt das Speichern ganzer Systemzustände[171] - was ebenfalls in mehrfacher Hinsicht sinnvoll ist, da Systemnutzern auf diese Weise auch erlaubt sein kann, Fehler zu machen und einen Standardzustand wiederherzustellen. Die Möglichkeit, *gültige* Interaktionsräume zu definieren, bedingt ebenso als Konsequenz, dass alle vom Systemdesigner nicht vorgesehenen Möglichkeiten als *ungültig* betrachtet werden können. Durch diese Einschränkungen können gewünschte Interaktionen forciert werden. In direkter Folge ergibt sich ebenfalls, dass lediglich die Aspekte eines Systems virtualisiert, modelliert oder, allgemein formuliert, beim Designprozess *beachtet* werden müssen, welche im Systemkontext zweckdienlich und notwendig sind. [Israel u. a., 2012, S. 62]

Einer der bedeutenden Vorteile von Virtualisierung ist der praktisch unendlich[172] zur Verfügung stehende Interaktionsraum. Ein Beispiel dafür sind die enorm großen Welten des Spiels *Minecraft* im Vergleich zur Größe der Erde [Marchiafava, 2012; Huang u. Huang, 2014].[173] Virtuelle Objekte existieren nur als Interpretation und können sogar prozedural erzeugt werden. Entsprechend hohe Bandbreite vorausgesetzt, können virtuelle Welten und/oder Interaktionsräume auch kooperativ respektive kollaborativ genutzt werden, da nicht gesamte Objekte übertragen werden müssen, sondern nur die notwendigen Informationen *über* das Objekt. Ein Beispiel hierfür wäre die Translation eines Hauses: Wenn Alice und Bob die Informationen zur Visualisierung des Hauses beide auf ihren Computern vorliegen haben und die gleiche Szene betrachten, dann muss Alice nur die Translationskoordinaten an Bob verschicken, um das Haus auf seinem Computer virtuell zu bewegen. [Israel u. a., 2012, S. 63-65]

[171] beispielsweise Speicherzustände in Spielen

[172] faktisch nur durch den Speicherplatz beschränkt

[173] *Minecraft* ist ebenfalls ein Beispiel dafür, wie Interaktionen erleichtert werden können, wenn physikalische Beschränkungen nicht gelten. So wurden in diesem Sandbox-Spiel schon ganze Raumschiffe oder fiktive Filmschauplätze in „Originalgröße“ nachgebaut [CerberusRCAF, 2011; Fishyyy, 2011]. Eine Vergleichsgrafik der Minecraft-Welt zur Erde ist in [Peel, 2013] zu finden.

7.5. Zusammenfassung

Anhand einiger Beispiele und Gegenüberstellungen sowohl in der digitalen als auch virtuellen Welt wurde herausgestellt, dass kein Medium gegenüber dem jeweils anderen bevorzugt werden sollte. Bewertungen ziehen immer einen Kontext mit ein und idealerweise ergänzen sich die Welten, um die jeweiligen Vor- und Nachteile in einem Gesamtsystem bestmöglich ausspielen zu können. Das Konzept der DIBA-PIBA-Listen wurde an dieser Stelle aufgegriffen, spezifiziert und an geeigneten Stellen exemplarisch mit Anwendungsszenarien versehen, die helfen können, den Einsatz der jeweiligen Technologien besser zu bewerten. Hierfür wird an dieser Stelle der Begriff Physical Is Good At - Digital Is Good At (PIGA-DIGA) vorgeschlagen.

Im nachfolgenden Kapitel wird erklärt, welche narrative Rahmen auf welche Art und Weise in Computerspielen aufgespannt werden können, um die gewonnenen Erkenntnisse miteinander zu verbinden. Die Narration dient damit als Brücke zwischen Entertainment, Realität und Wirklichkeit und gibt der Anwendung von Interaktionen somit einen Kontext, um eine Interaktionsabsicht im Menschen hervorzurufen.

„Games and stories have much in common. Both deal with how we handle fear. Both can teach us about the world and ourselves. Both can challenge us, move us to laughter or tears."

(Sheldon [Sheldon, 2004, S. 3])

KAPITEL 8

Storytelling in Spielen

Der englische Begriff „Story" ist im Zusammenhang mit dieser Ausarbeitung nicht als Begriff „Geschichte" zu verstehen, sondern als „Erzählung" respektive *Narration*. Erzählungen existieren unabhängig von einem Medium und insbesondere vor der Erfindung der Schrift wurden sowohl Informationen als auch Geschichten im Sinne einer *fiktiven Erzählung* ausschließlich mündlich weitergegeben[174] [Assmann, 2007, S. 91ff.]. Selbst solche mündlichen Erzählungen beinhalten Interaktivität: Jede Leserin und jeder Leser kann für sich selbst an dieser Stelle ein Gedankenexperiment wagen. Stellen Sie sich vor, Sie haben einen Freund, eine Freundin oder auch Ihren Partner. Fällt Ihnen ein Moment ein, in dem Sie beide gemeinsam eine Geschichte, welche Sie beispielsweise zusammen erlebt haben, einer dritten Person erzählen und sich dabei gegenseitig in der Erzählung ergänzen?[175]

Auch heute noch können Kinder beim Spielen und Erzählen beobachtet werden, wie sie in Rollen schlüpfen und Geschichten förmlich „erleben". Eine Ausprägung dieser Eigenart, welche auch in erwachsene Zielgruppen hineinreicht, lässt sich in der Form von Rollenspielen finden [Sheldon, 2013, S. 9][176].

Da diese Ausarbeitung die Digitalisierung eines Pen-&-Paper-Rollenspiels zum Inhalt hat, soll auch beim Storytelling ein besonderer Fokus auf die Eigenheiten dieser Spielgattung gelegt werden. Im Fol-

[174] Wobei sich die Inhalte und Wiedergabeform von mündlich überlieferten Darstellungen verständlicherweise von Individuum zu Individuum voneinander unterschieden.

[175] Sollten Sie diesem Gedankenexperiment gefolgt sein, dann war dies eine in diesem Fall unidirektionale Interaktion. Sie hatten die Wahl, ob Sie selbst den Kontext dieses Absatzes mit einer eigenen Erinnerung bereichern wollen oder nicht. Dadurch lässt sich ebenfalls zeigen, dass auch ein scheinbar nicht interaktives Medium wie eine Niederschrift eine Interaktivität beinhalten- oder zu dieser anzuleiten vermag.

[176] Sheldon hat sich im Verlauf der Bearbeitung als aktuelle und vielzitierte[177] Quelle herausgestellt. Mit seinem Buch „Character Development and Storytelling for Games"[178] wurde nach Ansicht der Autoren eine ideale Grundlage gefunden, in welcher viele Aspekte der in diesem Kapitel angesprochenen Thematiken behandelt werden. Deshalb wird im aktuellen Kapitel vergleichsweise häufig auf diesen Autor und sein Werk verwiesen.

[177] vgl. [Google Scholar, 2014]

[178] vgl. [Sheldon, 2013]

genden wird eine Definitionsabgrenzung vorgenommen, um schließlich verschiedene Erzählmodelle, insbesondere mit Bezug auf Computer- und Rollenspiele, gegenüberzustellen.

8.0.1. Definitionsabgrenzung

In ihrem Werk *Rules of Play - Game Design Fundamentals*[179] stellen die Autoren Salen u. Zimmerman verschiedene Definitionen für den Begriff des Spiels gegenüber [Salen u. Zimmerman, 2003, S. 73ff.][180].

Diese Gegenüberstellung resultiert in folgendem, neuen Definitionsvorschlag, mit dem der Anspruch erhoben wird, eine Mehrzahl an den zuvor analysierten Kennzeichen zu vereinen[181]:

> *„A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome."*

[Salen u. Zimmerman, 2003, S. 80][182]

Zugleich beschreiben die Autoren ausführlich, dass deren selbstgewählte Definition in Konflikt mit dem Genre der Rollenspiele steht und auf dieses nicht anwendbar ist. Dies wird darin begründet, dass Rollenspiele zwar die zuvor erwähnten und erläuterten Elemente aufweisen[183] und auch einen Großteil der selbst aufgestellten Definition erfüllen, einen wichtigen Aspekt hierbei jedoch auslassen: ein quantifizierbares Ergebnis. [Salen u. Zimmerman, 2003, S. 81-82].

Rollenspiele werden in ihrem Fortgang durch ihre Teilnehmerinnen und Teilnehmer bestimmt, wobei die Ziele eines jeweiligen Spiels durch Gruppenentscheidungen festgelegt werden und sich durchaus während des Verlaufs eines Spiels ändern können. Es ist hierbei sogar möglich, dass ein Spiel nie endet[184]:

[179] siehe [Salen u. Zimmerman, 2003]

[180] Eine Übersicht mit den Gemeinsamkeiten und Unterschiedenen dieser Gegenüberstellung ist auch im Anhang unter Tabelle B.1 auf Seite xxi zu finden.

[181] vgl. [Salen u. Zimmerman, 2003, S. 80]

[182] In einer Randbemerkung sei festgehalten, dass die Autoren direkt im Anschluss noch eine weitere Definition vorstellen und den Prozess des Game-Designs an sich definieren:

> *„Game design is the process by which a game designer creates a game, to be encountered by a player, from which meaningful play emerges."*
>
> [Salen u. Zimmerman, 2003, S. 80]

[183] Game Master, Regelwerk, Hilfsmittel, Spieldesign

[184] Anhand einer persönlichen Anekdote soll dies verdeutlicht werden: Es gab den Fall einer Rollenspielerin, die mit anderen Leuten gemeinsam regelmäßig innerhalb des *Star Trek*-Universums ein Rollenspiel spielte. Das Setting befand sich demnach im Weltraum. Durch einen Umzug (in der realen Welt) konnte diese Rollenspielerin nun nicht mehr regelmäßig mit ihrer Gruppe zusammen spielen, sodass im Spiel selbst eine Lösung dafür gefunden werden musste. Dies wurde dadurch erreicht,

> *„Role-playing games are structured like serial narratives that grow and evolve from session to session. Sometimes they end; sometimes they do not. Even if a character dies, a player can rejoin as a different character. In other words, there is no single goal toward which all players strive during a role-playing game.“*

[Salen u. Zimmerman, 2003, S. 81]

Streng formuliert würden Rollenspiele somit nach der Definition von Salen u. Zimmerman nicht als Spiele gelten, was offensichtlich eine Fehleinschätzung bedeuten würde. Dieser Konflikt kann aufgelöst werden, wenn betrachtet wird, dass Spielerinnen und Spieler sich ihre Ziele selbst setzen. Somit bilden Rollenspiele einen größeren Rahmen, der das so genannte *Gameplay* ermöglicht und eine Auswahl an Zielen präsentiert, die aufeinander aufbauen und selbst gewählt werden können [Salen u. Zimmerman, 2003, S. 82].

8.1. Storytelling in verschiedenen Medien

Mit der Niederschrift von Erzählungen wurden diese unter den Menschen teilbar und änderten den Adressaten von einem persönlichen Gegenüber zu einer vagen Masse an Unbekannten. Der Aspekt der Interaktivität tritt zunächst in den Hintergrund, auch deswegen, weil die Interaktionsfähigkeit durch die Einbringung des Mediums der Schrift zunächst beschränkt wird. Einer der ersten Versuche, diese Barriere zu durchbrechen, ist die interaktive Erzählung *Afternoon* des amerikanischen Autors Michael Joyce [Joyce, 1987]. Es herrschte und herrscht teils Uneinigkeit darüber, ob es sich bei diesem Werk im Ganzen um eine zusammenhängende Erzählung handelt, da eine klare Abfolge an narrativen Modulen scheinbar nicht gegeben ist, obwohl bei genauerer Betrachtung durchaus ein vorgegebener Standardpfad erkennbar ist [Walker, 1999, S. 111]. Die Art der Erzählung in diesem Werk macht jedoch deutlich, dass die Chronologie einzelner Teile einer Narration nicht zwangsläufig an die Chronologie des Inhalts gebunden ist [Walker, 1999, S. 112].

Eine andere Form der Wiedergabe von Erzählungen ist das Theater. Sowohl Ausprägungen, in denen das Publikum das aufgeführte Stück ohne Mitwirkung ansieht, als auch Varianten, in denen das Publikum mit einbezogen wird und die sogenannte *vierte Wand*[185] durchbrochen wird, sind hier anzutreffen. Computerspiele können mit interaktiven Theaterstücken verglichen werden, denn auch hier existiert die Partizipation des Zuschauers. Mit der Stimmung und Teilnahme der Spielenden verändert sich die

dass der Charakter (der Avatar) in der Handlung des Spiels ins Exil auf einem fremden Planeten emigrierte. Auf diese Weise wurde ein reales Ereignis in die Handlung des Spiels verwoben um die Konsistenz aufrecht zu erhalten.

[185] Zum Begriff der *vierten Wand* sei auf [Bell, 2008, S. 203] verwiesen. Der Begriff erhält im Kontext dieser Arbeit jedoch noch entscheide Bedeutung, vor allem in Abschnitt 13.1 ab Seite 229.

Spielerfahrung und vice versa [Sheldon, 2013, S. 8]. Bei Büchern ist das Medium ein Zweck, um den Inhalt zu vermitteln. Computerspiele besitzen die Möglichkeit, das Medium in den Inhalt mit einzubeziehen und eine Symbiose zu entwickeln [Sheldon, 2013, S. 14]. (Fiktive) Erzählungen möchten die Rezipienten unterhalten, was auf der anderen Seite genau so voraussetzt, dass Rezipienten unterhalten werden möchten. Die Gründe für den Wunsch nach Unterhaltung können vielfältig sein: Einige Menschen möchten in Rollen schlüpfen, andere möchten fremde Orte besuchen oder entdecken und wiederum gibt es einige, die vielleicht Dinge tun möchten, die im realen Leben nicht möglich sind. Mit Hilfe von Computerspielen ist es möglich, viele dieser Wünsche gleichzeitig zu adressieren. [Sheldon, 2013, S. 8-14]

Bei der Entwicklung von Computerspielen existiert zumeist ein Henne-Ei-Problem: Sollte das Spiel an sich oder die Story zuerst entwickelt werden? Sheldon schrieb hierzu:

> *„Gameplay and story work best together when developed together“*

[Sheldon, 2013, S. 159]

Der Autor adressiert damit den Umstand, dass Gamedesign und Storydesign beim Medium der Computerspiele zwar trennbar sein können, nicht aber müssen und - beim erklärten Ziel eine gut durchdachte Geschichte zu erzählen - auch nicht sein *sollten*. An dieser Stelle ist auch eine Parallele zum Medium *Buch* erkennbar, denn obwohl bei Computerspielen eine Interaktivität möglich und erwünscht ist, so kann kontinuierlich die Frage gestellt werden, welche Interaktionen für das Publikum offen stehen sollen [Sheldon, 2013, S. 161]. Analog hierzu besteht die Frage nach der richtigen Wahl des Grundthemas in Verbindung mit der richtigen Art des Spiels. Was narrativ als Film funktioniert muss als Spiel nicht gut sein und teilweise gilt dies auch für gesamte Themengruppen: Bis heute existiert kein nennenswertes Massive Multiplayer Online Role Playing Game (MMORPG) mit Romantik-Thema [Sheldon, 2013, S. 165]. Weiterhin kann beobachtet werden, dass es verhältnismäßig einfach ist, Spielerinnen und Spieler zum Lachen zu bringen oder Angst auszulösen[186], es ist jedoch ungleich schwerer, komplexe Emotionen wie Trauer auszulösen. Dies hängt auch damit zusammen, dass eine Vorliebe für heldenhafte Figuren zu existieren scheint, komplexe Emotionen jedoch tragische Helden benötigen[187] [Sheldon, 2013, S. 25]. Die Charaktere einer Erzählung und eine Erzählung selbst stehen hierbei in direktem Zusammenhang: Normalerweise können Erzählungen nicht ohne Charaktere existieren und werden sogar von diesen getragen [Sheldon, 2013, S. 25].

Abgrenzbar zu den vorausgegangenen Punkten liegt der Sachverhalt bei Rollenspielen, der im Rahmen dieser Arbeit besonders relevant ist: Eine Geschichte wird hier nicht mehr von Charakteren getragen, da diese dort entweder gar nicht oder allenfalls nur sehr diffus vorgegeben sind. Der Spieler respektive

[186] Ohne an dieser Stelle die Komplexität von Freude und Angst in Abrede stellen zu wollen.
[187] Tragisch bedeutet in diesem Fall das eine enge Bindung zwischen Protagonisten und Spielenden geformt wird.

die Spielerin schlüpfen selbstständig in die Rolle der tragenden Charaktere, formen diese im Verlauf der Geschichte und passen sie ebenfalls an Ereignisse selbiger an. Der Game Master wiederum erzählt eine Geschichte, die sich adaptiv zu den von den Spielenden vorher getroffenen Entscheidungen entfaltet und zu den Charakteren passt - somit ergibt sich ein ständiges Wechselspiel aus Aktion und Reaktion.

Die Frage, die sich im Rahmen der Entwicklung von Computerspielen anschließt, ist die nach der richtigen Balance zwischen vorgegebenen und anpassbaren Elementen. Eine vollständig dynamisch erzeugte Welt mit allen Feinheiten, die ein menschlicher Game Master beschreiben kann und sich prozedural selbst erzeugt, wurde bis zur Abgabe dieser Arbeit als noch nicht realisierbar angesehen[188]. Der Einfluss und die Beschränkungen durch das Spieldesign sollten hierbei so gering wie möglich ausfallen, gleichzeitig jedoch nicht das Momentum aus der erzählten Geschichte entfernen. Wenn den Spielenden völlige Freiheit gewährt wird, rückt die Narrativität in den Hintergrund und es wird der Bereich einer Simulation betreten. Diese ist beispielsweise bei den bekannten Titeln *Sims*[189] oder *Second Live*[190] zu beobachten.

Dies bedeutet im Umkehrschluss jedoch nicht, dass Geschichten immer linear sein müssen oder dass die Adaption einer linearen Erzählung über eine lineare Spielmechanik verfügen muss. Streng bewertet kann jede Individualerfahrung beim Durchspielen eines Spiels als linear betrachtet werden, selbst wenn die Spielwelt als solche nicht linear ausgelegt ist, da bei jedem Durchlauf zwangsläufig genau ein Pfad verfolgt wurde – unabhängig davon, wie viele Pfade und Möglichkeiten tatsächlich existieren (vgl. Abbildung 8.1).

Der Begriff der Nichtlinearität muss daher feingranularer betrachtet werden, denn es gibt einen Unterschied zwischen linearer Perzeption und nichtlinear Konzeption einer Erzählung und/oder eines Spiels. In diesem Zusammenhang ist es wichtig, den Begriff des Pfads genauer zu verstehen. Die Idee, Spiele oder auch Erzählungen entlang eines metaphorischen Pfades „abzuschreiten“ leitet sich aus der Lehre des Zen-Buddhismus ab[191] und prägte den Begriff des *goldenen Pfads*. In Pfaden zu denken und zu konstruieren führt stets zum Gedanken der Linearität, während vollständige Nichtlinearität die völlige Abwesenheit von Pfaden vorsieht. [Sheldon, 2013, S. 176-179]

[188] Ein interessanter Ansatz für eine prozedural erzeugte Spielwelt wurde kurz vor Abgabe dieser Arbeit mit der *Upvoid Engine* vorgestellt, einem Projekt von mehreren ehemaligen Studenten der RWTH Aachen. Hier ist die freie Gestaltung einer Spielwelt zur Laufzeit weitgehend möglich, jedoch fehlt auch hier die Möglichkeit eine Situation zu beschreiben und diese in allen Feinheiten und mit endgültiger Wirkung auf das weitere Spielgeschehen zu formulieren und automatisch erzeugen zu lassen [Sauter, 2014].

[189] vgl. [IDG Entertainment Media GmbH, 2000]

[190] vgl. [Casati u. a., 2007]

[191] Exakter formuliert bezieht sich der Gedanke des „Goldenen Pfads“ auf Buddhas Lehre des acht-faltigen Weges zu Erleuchtung [Sheldon, 2013, S. 179].

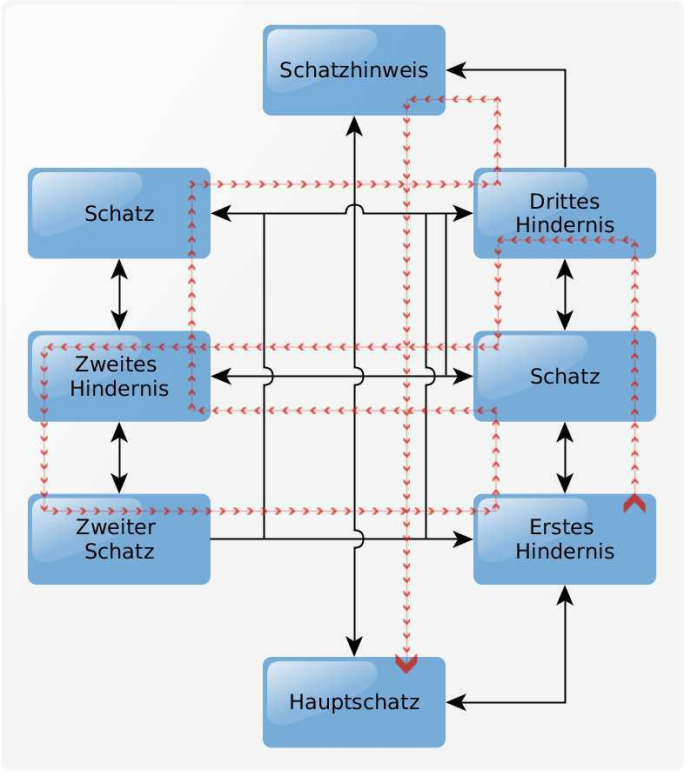


Abbildung 8.1.: Die Abbildung zeigt einen möglichen Ausschnitt aus der Struktur eines Spiels. Obwohl hier viele Wahlmöglichkeiten offen gelassen wurden, die eine Bewegung durch die Spielwelt ermöglichen, kann jede individuelle Spielerfahrung innerhalb des Konstrukts als linear bezeichnet werden. Selbst dann, wenn der Spieler oder die Spielerin mehrere Räume oder Pfade mehrmals besucht, so entsteht am Ende doch genau ein Pfad, der von A nach B geführt hat. [Sheldon, 2013, S. 176] In diesem Beispiel wurde ein Pfad vom ersten Hindernis bis zum Hauptschatz gewählt, der auch das mehrmalige Besuchen bestimmter Abschnitte sowie Kehrtwendungen beinhaltet (dargestellt in roter Farbe, die Pfeile geben die Bewegungsrichtung durch das Spiel aus Sicht des Spielers beziehungsweise der Spielerin an).

nach [Sheldon, 2013, S. 177]

Im Folgenden werden die gängigsten Erzählstrukturen gegenübergestellt und deren spezifische Stärken und Schwächen erörtert, wobei auch immer ein Rückbezug zu der (nicht-)Existenz von Pfaden innerhalb dieser Strukturen hergestellt wird, um die Linearität visuell zu analysieren.

8.2. Erzählstrukturen

Erzählstrukturen, wie sie hier vorgestellt werden, beschreiben verschiedenen Arten, Geschichten zu erzählen. Obwohl es sich bei Computerspielen um ein interaktives Medium handelt, heisst dies nicht, dass eine Geschichte auch interaktiv sein muss oder das Eingreifen des Spielers erlaubt. Erzählungen und das Medium, durch welches sie transportiert werden, sind disjunkt. Dies trifft auch auf die Linearität von Erzählungen zu: Ein lineares Medium kann eine nichtlineare Geschichte erzählen, wie gezeigt werden wird. Ebenso kann ein interaktives Medium eine lineare Geschichte erzählen. Der Fokus in diesen Abschnitten liegt demnach auf verschiedenen Narrationsformen, die sich prinzipiell in und mit jedem Medium anwenden ließen[192]. Computerspiele bieten durch ihre Form lediglich eine elegante Möglichkeit, die gewählte Form der Narration vor dem Rezipienten zu verschleiern. [Pearce, 1994, S. 100-101]

8.2.1. Lineares Storytelling

Lineares Storytelling ist die älteste und am besten erforschte Art, Geschichten zu erzählen (vgl. Abbildung 8.2) [Sheldon, 2013, S. 313ff.].

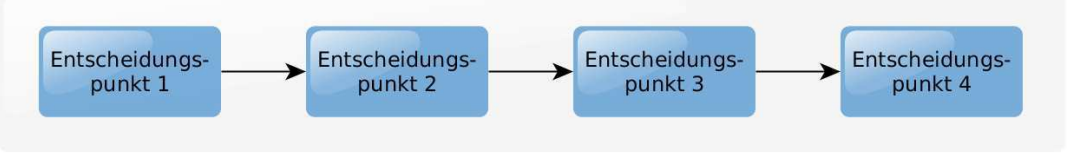


Abbildung 8.2.: Beim linearen Storytelling besteht nur die Möglichkeit, sich in einer fest vorgegebenen Bewegungsrichtung durch eine im Vorhinein durchstrukturierte Geschichte zu bewegen.

nach [Sheldon, 2013, S. 314]

Die klare Struktur des linearen Storytellings garantiert sowohl eine für Entwickler als auch für das Publikum transparente Herangehensweise, während die auktoriale Kontrolle zu jeder Zeit vollständig aufrecht erhalten wird. Es ist möglich, eine nichtlineare Spielerfahrung zu konstruieren, welcher dennoch eine lineare Geschichte zu Grunde liegt. Hierzu können die einzelnen Story-Schritte aus Abbildung 8.2 auch als Welten völliger Spielfreiheit betrachtet werden, die jedoch in klarer Reihenfolge in genauer einer Richtung miteinander in Verbindung stehen. [Sheldon, 2013, S. 313-315]

[192] Bestimmte Mechanismen können lineare Medien nicht bieten, was besonderes bei der Netzform und beim modularen Storytelling von Relevanz ist.

Ein bekanntes Beispiel neuerer Zeit für eine linear erzählte Geschichte ist das Rollenspiel *Skyrim*[193]. Skyrim spielt in einer großen, ans Mittelalter angelegten Fantasy-Welt und ist vollständig frei begehbar. Spielerinnen und Spieler haben beispielsweise jederzeit die Möglichkeit, eine Höhle, einen Berg oder auch ganze Städte zu erkunden. Neben einer Hauptgeschichte existieren viele Nebenschauplätze, wie beispielsweise die Möglichkeit, sich in der Burg der Magier vollständig zum Zauberer ausbilden zu lassen, was wiederum mit mehreren Untermissionen einhergeht. Diese Missionen können ihrerseits jederzeit unterbrochen werden, um beispielsweise an einem völlig anderen Ort der Spielwelt einen Drachen zu töten. Das Spiel erzeugt die Illusion freier Entscheidung und Bewegung – was im Bezug auf das reine *Gameplay* auch korrekt ist. Die eigentliche Rahmenhandlung kann jedoch nur Schritt für Schritt in vorgegebener Reihenfolge und ohne Abzweigungen erspielt werden. Hier wird demnach eine lineare Geschichte eingebettet in ein hochgradig nichtlineares Gameplay. Um den jeweils nächsten Schritt der Erzählung zu erfahren, müssen jedoch so genannte „Nadelöhre" durchlaufen werden – und genau hierbei handelt es sich um die verbindenden Pfeile, die auch in Abbildung 8.2 dargestellt sind.

Lineares Storytelling hat für die Spielentwicklung viele Vorteile, da es gut plan- und vorhersehbar ist, allerdings zwingt es die Menschen gerade durch diese Vorhersehbarkeit in vorgegebene Verhaltensweisen. Ein damit einhergehender Aspekt ist die Limitierung von Entscheidungsfreiheit oder die Größe der Spielwelt – und dies kollidiert wiederum mit dem Wunsch der Menschen nach größtmöglicher Freiheit: [Sheldon, 2013, S. 314-316]

> *„players want freedom of choice and action. Linear storytelling clearly limits their choices and actions."*

[Sheldon, 2013, S. 316]

Bei Spielen ist ein Beschnitt der Freiheit im Gegensatz zum Medium des Films als besonders kritisch anzusehen. Bei Filmen sind die Menschen daran gewöhnt, dass sie keinen Einfluss auf den weiteren Verlauf der Geschichte haben. Die Illusion, dass der Filmstar theoretisch alles tun und sich frei bewegen kann, wird als gegeben wahrgenommen. Dies ist ein wichtiger Unterschied, denn mit der Möglichkeit zur Interaktion und Einflussnahme steigen auch die Anforderungen, die an das eingesetzte Medium zum Transport von Inhalten gestellt werden. [Sheldon, 2013, S. 315]

8.2.2. Verzweigungen (in linearer Ausführung)

Bei der Strukturierung einer Erzählung in Verbindung mit einem Spiel als Verzweigung gibt es zwei gängige Ansätze: Spiele mit Verzweigungen an *einem* Punkt oder Spiele mit Verzweigungen an *mehreren* Punkten (vgl. Abbildung 8.3) [Sheldon, 2013, S. 318].

[193] 2011, *Bethesda Game Studio* [MobyGames.com, 2014b]

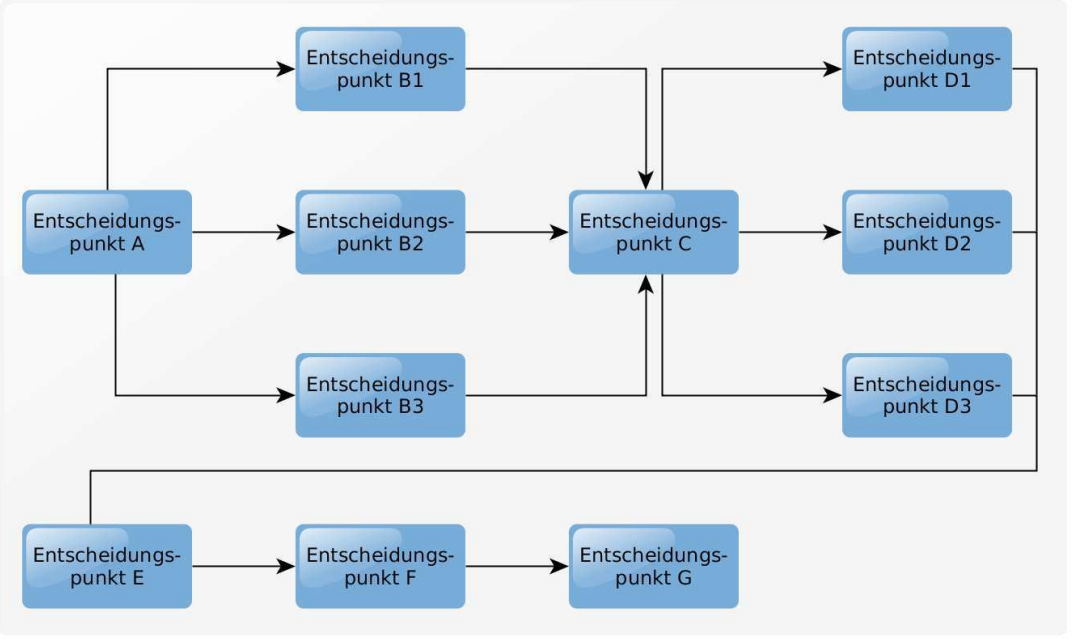


Abbildung 8.3.: Verzweigungen in linearer Ausführung erlauben die minimale Einflussnahme beim Verlauf der Erzählung, wobei die auktoriale Kontrolle niemals aufgegeben wird. Genau betrachtet handelt es sich hierbei um die Parallelisierung mehrerer linearer Erzählstränge, die an bestimmten Entscheidungspunkten miteinander verknüpft werden.

nach [Sheldon, 2013, S. 323]

Eine häufig anzutreffende Variante beim Storytelling in Spielen ist diese, dass eine Geschichte linear schrittweise verläuft und sich am Ende in mehrere mögliche Enden aufteilt. Eine andere Variante ist diese, dass am Anfang des Spiels einer von mehreren möglichen Anfängen gewählt werden kann, beispielsweise um der spielenden Person einen stärkeren Charakterbezug zu ermöglichen[194]. [Sheldon, 2013, S. 318].

Verzweigungen in Spielen ergeben genau dann einen Sinn, wenn alle spielbaren Anfänge respektive Enden einer Geschichte einen gleichen Spielwert besitzen. Eine Verzweigung, die in ein Ende mündet, welches das gesamte vorherige Spiel und alle Bemühungen sinnlos erscheinen lässt, während alle

[194] Das Verfahren mit mehreren möglichen Anfängen wurde beispielsweise im Titel *Dragon Age: Origins*[195] verwendet. Mit der Wahl des Charakters respektive der Charakterklasse änderte sich auch die komplette, spielbare Hintergrundgeschichte und der Anfang des Spiels [Sheldon, 2013, S. 316]. Ein Beispiel für ein Spiel mit mehreren, in diesem Falle nicht gleichwertigen (im Sinne von „guten“) Enden ist *S.T.A.L.K.E.R.: Shadow of Chernobyl*[196].

[195] 2009, *BioWare Corporation* [MobyGames.com, 2013d]

[196] 2007, *GSC Game World* [MobyGames.com, 2013s]

andere Enden das Vorgehen im Spielverlauf belohnen, sollte in Frage gestellt werden. Eine andere Sichtweise wäre die, dass Verzweigungen genau dann sinnvoll gewählt wurden, wenn sich *kein* goldener Pfad erkennen lässt. Der Anreiz, Verzweigungen zu verwenden, muss für die Spielerinnen und Spieler vor allem in einem Wiederspielwert begründet sein, der beispielsweise durch Änderungen in der Erzählung oder im Gameplay motiviert werden kann. Wenn es auch nicht ganz so erfolgreich ist beziehungsweise nicht so häufig verwendet wird wie lineares Storytelling, so gehört das Prinzip der Verzweigungen dennoch ebenfalls zu den ältesten und auch am besten erforschten Erzählstrukturen. [Sheldon, 2013, S. 318]

Da eine Struktur in Verzweigungen immer noch Pfade aufweist, wird auch bei dieser Variante die auktoriale Kontrolle beibehalten – anders ausgedrückt stellt eine Struktur in Verzweigungen nur eine Erweiterung des linearen Storytellings dar. Ein Gefühl von Nichtlinearität wird in diesem Fall nur durch die Illusion der freien Wahl erzeugt, während Entwickler letztlich nur den Mehraufwand zusätzlicher Pfade berücksichtigen müssen. [Sheldon, 2013, S. 321-324]

8.2.3. Netzform (Simple Form der Nichtlinearität)

Die Netzform respektive Netzstruktur einer Geschichte stellt eine simple Variante echter Nichtlinearität dar (vgl. Abbildung 8.4).

Die Netzform lässt Übergänge von verschiedenen Entscheidungspunkten in einer Erzählung zum ersten Mal[197] diffus erscheinen. Es ist beispielsweise in Abbildung 8.4 nicht nur möglich von Entscheidungspunkt *A* über *B3* zum *C2* zu gelangen, sondern auch von *A* über *B4* zu *C3* und von dort erst über *B3* nach *C2*. Ein Fortschritt in der Erzählung ist an dieser Stelle somit erstmalig in mehr als nur eine Richtung möglich. [Sheldon, 2013, S. 324-325]

Das Prinzip der Netzform wird in Computerspielen noch nicht häufig realisiert – falls es dennoch eingesetzt wird, dann verwechseln Spielerinnen und Spieler es oftmals mit dem bereits vorgestellten Prinzip der Verzweigungen[198]. Dass die Netzform gewissermaßen immer noch als Novum betrachtet wird ist insbesondere deswegen erstaunlich, weil das erste Adventure-Spiel überhaupt, *Collosal Cave*[199], eine Netzstruktur vorweist. Das von Will Crowther und Don Wood geschriebene Spiel war weiterhin das erste, welches interaktive Erzählstrukturen realisierte [Sheldon, 2013, S. 312-313]. Bei diesem Spiel konnten die Räume einer gigantischen Höhle erkundet werden.

[197] Im Vergleich zu den in diesem Kapitel zuvor vorgestellten Methoden.
[198] vgl. Abschnitt 8.2.2 auf Seite 162
[199] 1976, *Will Crowther* [MobyGames.com, 2013a]

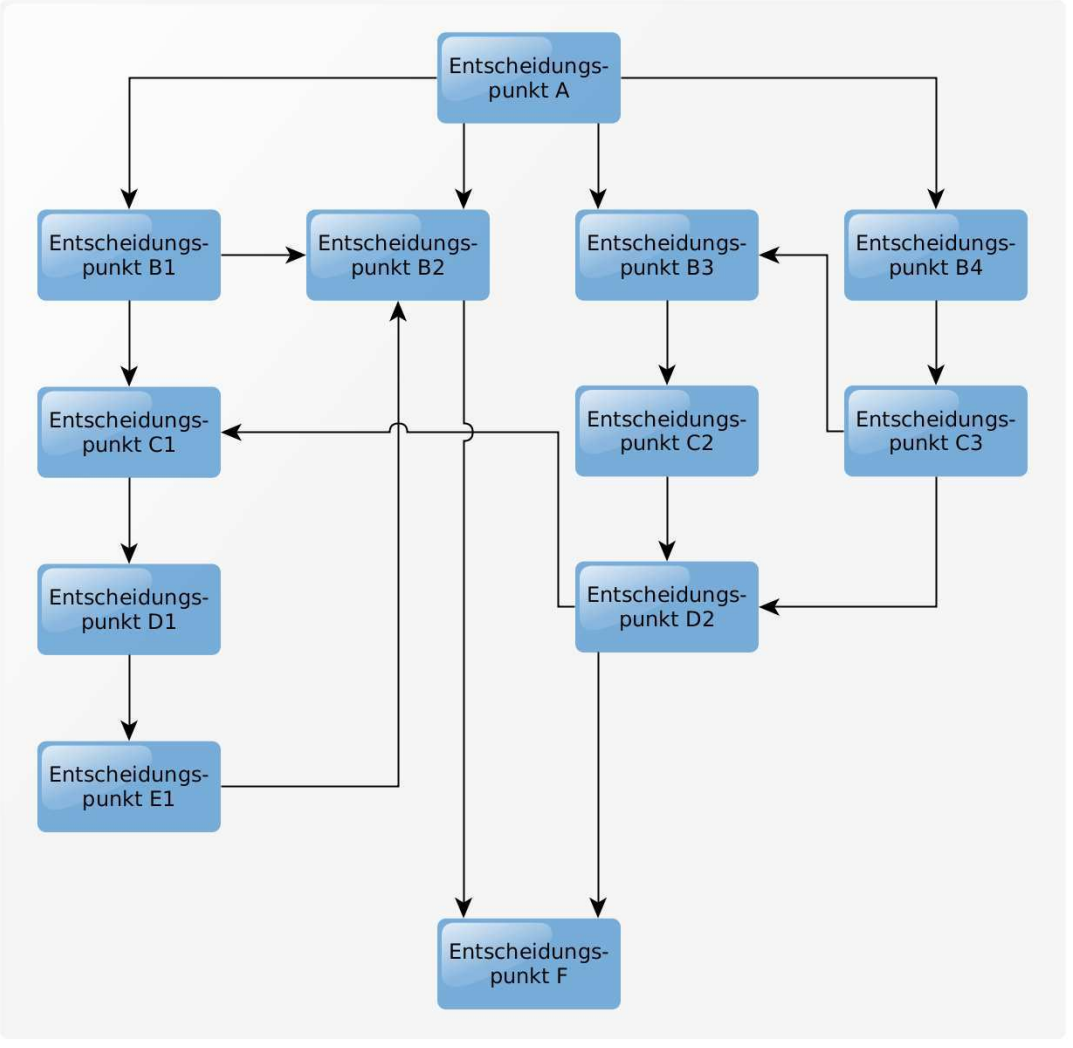


Abbildung 8.4.: Die Netzform gibt an einigen Stellen erstmal die auktoriale Kontrolle auf. Hiermit ist gemeint, dass der Designer einer Erzählung an bestimmten Punkten nicht vorhersehen kann, in welcher Reihenfolge die Geschichte erspielt wird: nach A kommt nicht immer B. Auf diese Weise müssen Vorkehrungen für die Konsistenz der Erzählung in allen Fällen getroffen werden, was wiederum durch die Beschränkung nur auf bestimmte Pfadangebote erleichtert werden kann.

nach [Sheldon, 2013, S. 326]

Die auktoriale Kontrolle wird bei der Netzform zumindest teilweise aufgegeben und es ist vergleichsweise aufwändig, eine kohärente Erzählung zu konstruieren. Die Netzform kann sowohl als Übergang der linearen Erzählmethoden zu vollständig nichtlinearen Erzählungen gesehen werden, als auch als ein Rückzugsraum vom vollständig modularen Storytelling, falls es an gewissen Punkten in einem Spiel zwingend erforderlich ist, dass bestimmte Entscheidungspunkte in jedem Fall besucht werden. [Sheldon, 2013, S. 325-328]

8.2.3.1. Die Hypermedia Novel als Anwendungsbeispiel der Netzform

Ein Anwendungsbeispiel der Netzform stellt die *Hypermedia Novel* dar, eingeführt durch Heiden u. a. und vorgestellt in [Heiden u. a., 2001]. Das als Erweiterung klassischer Erzählkonstrukte erdachte Narrationskonzept sieht in Modulen erdachte Inhalte vor und verfügt darüber hinaus ebenfalls über die Möglichkeit, auf einfache Art und Weise Modifikation an den Inhalten selbst vorzunehmen respektive neue Inhalte hinzuzufügen [Heiden u. a., 2001]. Auf diese Weise entsteht ein dualer Fokus: Einerseits steht der Rezipient einer in Modulen erzählten Geschichte im Mittelpunkt, andererseits wird ein Augenmerk auf die Autoren gelegt, die eine solche Geschichte erzählen wollen, ohne dabei die „Granularität einzelner Module definieren zu müssen“ [Heiden u. a., 2001].

Die Autoren beschreiben, dass das Konzept der Hypermedia Novel grob als eine Erweiterung der *Graphic Novels* betrachtet werden könnte, deren multimediales Auftreten in Form von Texten und Bildern an dieser Stelle zusätzlich um Töne, Videos, Animationen oder auch interaktive Elemente wie kleiner Spiele erweitert werden kann.

Sinneinheiten innerhalb einer Hypermedia Novel bestehen grundsätzlich aus in sich inhaltlich abgeschlossenen Threads (Erzählsträngen, auch „Szenen“ genannt), die in eines oder mehrere sogenannter *Narrationsmodule*[200] gegliedert sind. Innerhalb eines solchen Moduls sind entweder spezifische Erzählschritte gekapselt, welche mit einem oder mehreren Medientypen präsentiert werden oder dienen wiederum als serieller oder paralleler Container für eine weitere Hierarchie von Modulen. Sofern die Module einen (Teil-)Inhalt repräsentieren, sollten die vermittelnden Medien so gewählt werden, dass sie den Inhalt bestmöglich transportieren. [Heiden u. a., 2001]

Aus diesen Szenen und deren Modulen wird anschließend ein Erzählgraph erzeugt, der die auktoriale Kontrolle beibehält (vgl. Abbildung 8.5). Nutzerinnen und Nutzer können an bestimmten Punkten beispielsweise eine Abzweigung nehmen und sich eine alternative Variante desselben Moduls mit anderen Medien ansehen oder auch Nebenschauplätze besuchen. Die Autoren betonen hierbei, dass die Wahlfreiheit zwar im Fokus steht, gleichzeitig jedoch ein goldener Pfad vorhanden ist: Es darf

[200] im Original *Narration Module*, „NarMo“ genannt [Heiden u. a., 2001]

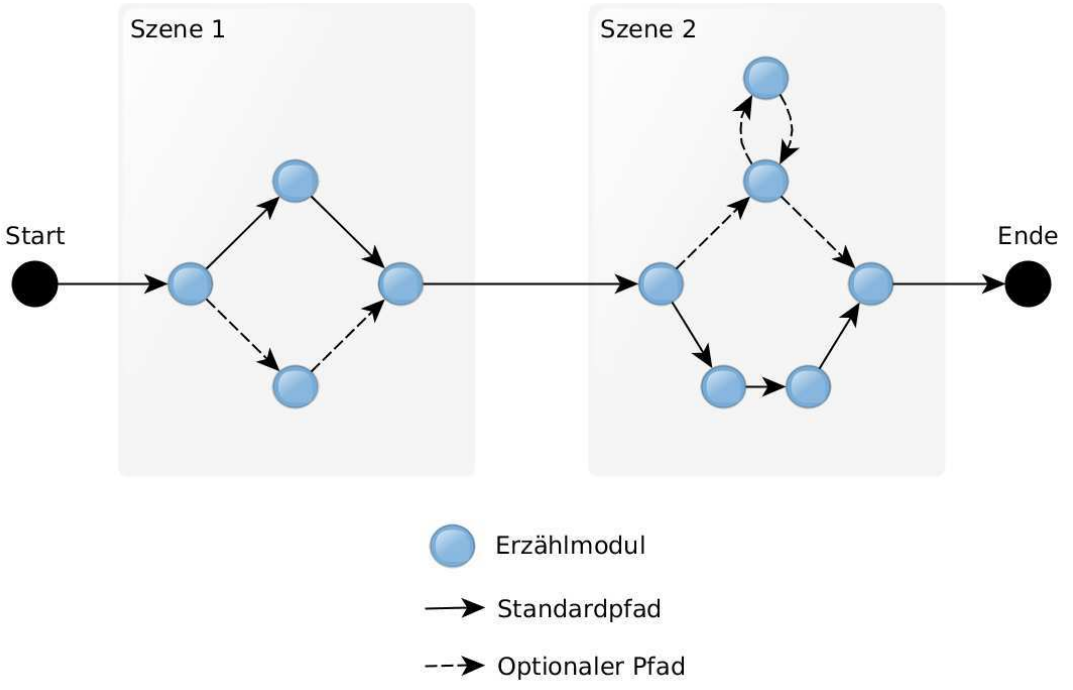


Abbildung 8.5.: Die Abbildung zeigt die Visualisierung des erzeugten Erzählgraphen einer Hypermedia Novel. Die erzählte Geschichte ist in Szenen eingeteilt, welche wiederum über verschiedene Module verfügen. Die Erzählmodule können in einem vorgegebenen Pfad besucht werden; gleichfalls ist es möglich, abweichende (und optionale) Routen einzuschlagen und auch Module auszulassen.

nach [Heiden u. a., 2001]

demnach ein individueller Pfad durch die Geschichte gewählt werden, ein Zwang hierzu besteht jedoch nicht. [Heiden u. a., 2001]

Insbesondere die in [Heiden u. Ostovar, 2006] vorgestellte neuere Version einer Implementierung befähigt den Nutzer oder die Nutzerin zu einem entscheidungsgetriebenen Re-Arrangement der einzelnen Module [Heiden u. Ostovar, 2006]. Aufgrund dieser Möglichkeiten ist die Hypermedia Novel innerhalb der netzbasierten Ansätze zu verorten; das Vorhandensein von Pfaden und die Betonung der Autoren, dass das Vorhandensein eines goldenen Pfades gewährleistet sein muss[201], signalisiert, dass ein vollständig modulares Storytelling[202] nicht angestrebt wurde.

[201] vgl. [Heiden u. a., 2001]
[202] im Sinne einer *Abwesenheit* von Pfaden

8.3. Modulares Storytelling

Alle bisher vorgestellten Verfahren haben eines gemeinsam: Das Konzept beziehungsweise die Idee von *Pfaden*. Es gibt zweierlei Möglichkeiten, sich einer *vollständig* nichtlinearen Erzählstruktur anzunähern. Die erste Möglichkeit besteht darin, beispielsweise die Netzstruktur als Grundlage zu nehmen und so lange mit Pfaden auszubauen bis der Spieler oder die Spielerin in der Lage ist, jeden Erzählschritt von jedem Punkt der Geschichte aus zu erreichen. Die zweite Möglichkeit besteht darin, den Gedanken von Pfaden ad acta zu legen (vgl. Abbildung 8.6). [Sheldon, 2013, S. 329]

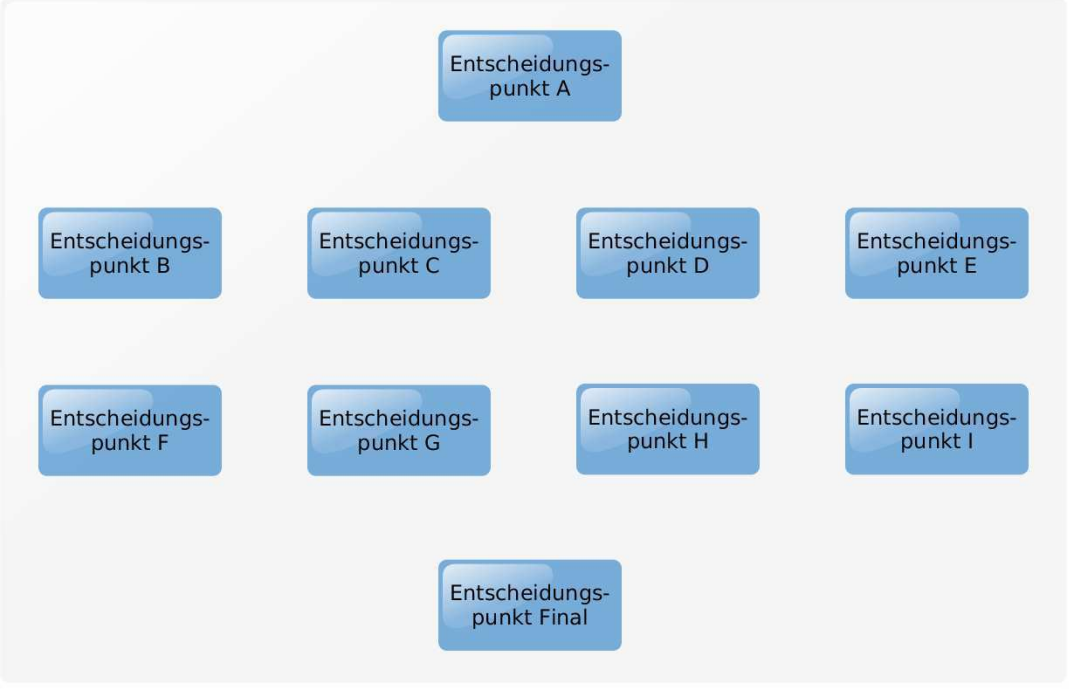


Abbildung 8.6.: Beim modularen Storytelling ist jedes Modul von jedem anderen Modul aus erreichbar. Anfang und Ende sind festgelegt, dazwischen existieren jedoch keinerlei Pfade mehr. Eine andere Variante der Darstellung wäre, einen Pfad von jedem Modul zu jedem anderen Modul zu zeichnen, um die völlige Wahlfreiheit zu signalisieren.

nach [Sheldon, 2013, S. 329]

Beim sogenannten modularen Storytelling wird die zu erzählende Geschichte namensgebend in Module eingeteilt. Ein Modul enthält sowohl persistente (P) als auch dynamische Eigenschaften (D). Die Geschichte kann von einem Modul zu einem beliebig anderen verfolgt werden, wobei Module auch

ausgelassen werden können. Es existiert hierbei nur eine Ausnahme: Jede Erzählung benötigt einen Start[203] und ein Ende[204]. Diesem Gedanken folgend wird immer in einem bestimmten Modul begonnen, während ein anderes Modul als Ende konzipiert wird. Zwischen diesen beiden Fixpunkten kann sich der Spieler oder die Spielerin frei bewegen. [Sheldon, 2013, S. 330-331]

Viele Aspekte eines Spiels oder einer Erzählung können in einem Modul gekapselt werden. Module können dabei jede Größe annehmen, von simplen Entscheidungen und Szenen bis hin zu ganzen Level-Abschnitten, welche in diesem Fall inhärent einen Story-Teil enthalten respektive mit einem solchen verknüpft sind. Dieses Prinzip kann sogar so weit getrieben werden, dass es möglich wird ein vollständiges Spiel als Modul in ein anderes Spiel zu integrieren, wie später anhand eines Beispiels verdeutlicht werden wird[205]. Die Struktur einer modular erzählten Geschichte ist dabei nicht mehr offensichtlich, was sich auch darin äußern kann, dass es nicht mehr auf einfache Weise möglich ist, ein Ablaufdiagramm des Spiels und/oder der erzählten Geschichte zu erzeugen. Der Aufwand, eine modulare Geschichte zu erzeugen, ist im Vergleich mit den anderen vorgestellten Methoden ungleich höher, allerdings steht im Ergebnis die ideale Symbiose zwischen Story und Gameplay, wie Sheldon zusammenfasst:

> *„No other structure is better at integrating story and gameplay than modular."*

[Sheldon, 2013, S. 331]

Die auktoriale Kontrolle wird hierbei nur *scheinbar* vollständig abgegeben, während in der Praxis Möglichkeiten bestehen, diese aufrecht zu erhalten. Ein erster Anhaltspunkt hierfür ist das in jedem Fall vorhandene Anfangsmodul und das Endmodul. Was zwischen diesen Fixpunkten geschehen muss, ist das Momentum einer Geschichte aufrecht zu erhalten und vom Anfang weg und auf das Ende hinzu zu bewegen. Fast alle Schritte zwischen Anfang und Ende können hierbei unabhängig voneinander gehalten werden. Richtig ausgeführt ist das modulare Storytelling von außen nicht als solches zu erkennen (vgl. Abbildung 8.7). [Sheldon, 2013, S. 331-333]

Aus programmiertechnischer Sicht betrachtet haben das modulare Storytelling und die objektorientierte Programmierung vieles gemeinsam. Objekte enthalten Daten und Funktionen, während Module Story (Daten) und damit zusammenhängendes Gameplay (Funktionen) beinhalten [Sheldon, 2013, S. 281]. Um zu errechnen, welches Modul an welchen Vorgänger angeschlossen werden kann, muss der Fortschritt der Spielerin kontinuierlich mitverfolgt werden. Alle Entscheidungen, die während des

[203] den „point of attack" [Sheldon, 2013, S. 330]
[204] „final resolution" [Sheldon, 2013, S. 330]
[205] vgl. Abschnitt 8.3 auf Seite 171 in diesem Kapitel

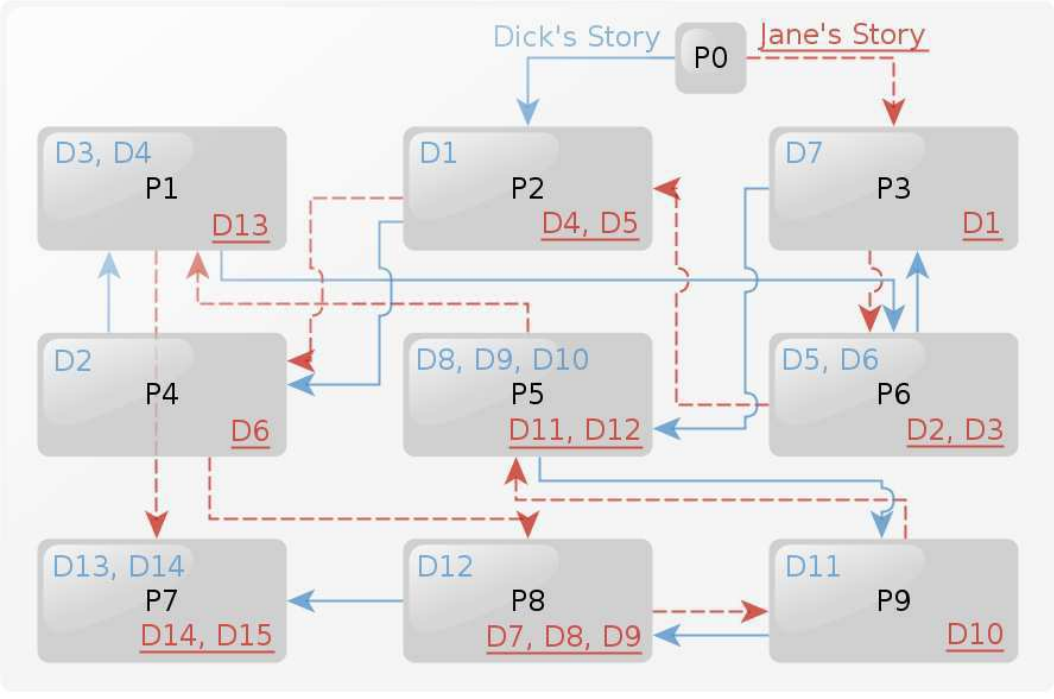


Abbildung 8.7.: Module beinhalten sowohl persistente (P) als auch dynamische (D) Attribute. Orte und Handlungen sind kohärent an ein jeweiliges Modul gebunden, während NPCs beispielsweise auch nicht an ein bestimmtes Modul gebunden sein müssen. In diesem Beispiel ist der Verlauf einer erspielten Geschichte von zwei verschiedenen Personen, Dick und Jane, dargestellt. Beide treffen im Verlauf „ihrer" Geschichte unterschiedliche Entscheidungen und erzeugen auf diese Weise unterschiedliche Pfade. Trotzdem wird in jedem Fall ein Pfad vom Anfang zum Ende beschritten – wobei in diesem Beispiel sogar alle Module besucht werden. [Sheldon, 2013, S. 333-335]

nach [Sheldon, 2013, S. 334]

Spielverlaufs getroffen werden, fließen in die Berechnung mit ein – sollte hierbei ein Modul übersprungen werden, kann es dennoch zu einem späteren Zeitpunkt besucht werden [Sheldon, 2013, S. 336].

Aus narrativer Sicht ist die Funktionsweise des modularen Storytellings nicht zwangsläufig selbsterklärend. Es hat Gründe, warum das lineare Storytelling immer noch am häufigsten eingesetzt wird: Exposition, Klimax und Resolution können fein gesteuert werden, während Orte, Charaktere und Handlungsstränge exakt dosiert und genau zum richtigen Zeitpunkt präsentiert werden. Es lassen

sich jedoch sowohl in linearen als auch in nichtlinearen Medien Beispiele aufzeigen, mit denen die Funktionsweise von modularem Storytelling verdeutlicht werden kann.

Der bekannte japanische Film *Rashōmon*[206] des Regisseurs Akira Kurosawa wird getragen von einer Rahmenhandlung, in der mehrere Personen in den Ruinen des namensgebenden Rashōmon-Tors in der Nähe von Kyoto vor einem Unwetter Schutz suchen. Zum Zeitvertreib sprechen sie über ein Verbrechen, welches fortan in Form von Rückblenden aus Sicht der jeweiligen Protagonistinnen und Protagonisten erzählt wird. Unstrittig ist, dass ein Bandit ein Pärchen, bestehend aus einem Samurai und seiner schönen, jungen Frau, im Wald überfällt, den Samurai fesselt und die Frau anschließend vergewaltigt. Kurze Zeit später wird der Samurai erstochen und der Bandit einige Tage darauf gefasst. Vor einem diffus im Hintergrund verbleibenden und stilisierten Gericht tragen alle Beteiligten ihre jeweilige Version der Ereignisse vor, in der sowohl der Bandit zu Wort kommt, die Frau, der tote Samurai (der durch ein Medium aus dem Totenreich spricht) und letztlich auch ein Holzfäller, der den Tathergang beobachtet hat. Der Holzfäller befindet sich auch unter den Personen der Rahmenhandlung, die in den Ruinen des Rashōmon-Tors Schutz suchen. Letztlich verfolgen alle Beteiligten eigene Motive und es bleibt ungeklärt, wer den Samurai tatsächlich getötet hat.

Der Film beinhaltet somit eine Rahmenhandlung, mehrere Rückblenden, die andere Varianten des selben Ereignisses erzählen und eine Narration, die nicht linear von A über B nach C verläuft, sondern dessen Module in beliebiger Reihenfolge arrangiert werden können. Die Geschichte und ihr Spannungsbogen werden nicht durch die Reihenfolge der Rückblenden beeinflusst und alle Charaktere und Ereignisse weisen eine inhärente Deskription auf, die nicht von jeweils anderen Modulen oder Charakteren abhängig ist.[207] [Sheldon, 2013, S. 309-310].

Ebenso wie beim Medium des Films lassen sich Beispiele für modulares Storytelling in der Literatur[208] oder sogar in der Musik[211] finden.

[206] Rashōmon (1950) [Kurosawa, 1950]

[207] Ein Gegenbeispiel eines auf den ersten Blick nichtlinearen Films ist mit dem Titel *Memento* aus dem Jahr 2000 zu nennen. Dessen Module erschienen zunächst willkürlich, sind jedoch in eine unstrittige lineare Abfolge arrangierbar, was sogar als explizites Feature der DVD-Fassung des Films ausgewiesen ist. Dort besteht die Möglichkeit, den Film in der „richtigen" Reihenfolge abzuspielen.[Nolan, 2000]. Bei *Rashōmon* ist ein solches Re-Arrangement nicht möglich, da kein „goldener Pfad" existiert; alle Module respektive Rückblenden in diesem Film sind voneinander unabhängig (vgl. auch Seite 80)

[208] An dieser Stelle sind die *Canterbury Tales*[209] von Geoffrey Chaucer und *Don Quixote of La Mancha*[210] von Miguel de Cervantes erwähnenswert.

[209] vgl. [Sheldon, 2013, S. 292]

[210] vgl. [Sheldon, 2013, S. 293ff.]

[211] Wolfgang Amadeus Mozart veröffentlichte 1787 ein Stück mit dem Namen *Musikalisches Würfelspiel*. Das Arrangement des Stückes, welches aus insgesamt 252 Einzelteilen besteht, wird im Vorhinein jeweils durch das Rollen von Würfeln festgelegt und erfolgt in der Reihenfolge der Aufführung dadurch willkürlich in einer von vielen tausend Variationen. Es handelt sich somit um eine modulare Komposition, bei der jedes Modul nahtlos zu jedem anderen Modul passt. [Sheldon, 2013, S. 297]

Wie bereits erwähnt wurde[212], können Module in beliebigen Größen vorliegen, wobei sogar die Möglichkeit besteht, ganze Spiele als Module in wiederum andere Spiele einzubetten. Dieses Prinzip kann am folgenden Beispiel einer bereits vorhanden Digitalisierung eines Rollenspiels verdeutlicht werden. Das im Jahre 2012 veröffentliche Science-Fiction Rollenspiel *Borderlands 2*[213] spielt auf einem postapokalyptischen, fremden Planten, bedient sich jedoch gleichzeitig der Elemente des Western [Gearbox Software, 2012]. Das Spiel wird getragen von einer Hauptgeschichte um fremdartige Artefakte und mächtige, uralte Wesen, die gleichzeitig sehr viel Macht verleihen. Im Jahr 2013 veröffentlichte der Hersteller ein Add-On zum Hauptspiel mit dem Namen „Assault on Dragon Keep" [Weber, 2013]. Die grundsätzliche Spielmechanik bleibt in diesem Add-On identisch zur Mechanik des Hauptspiels, allerdings ändert sich die Hintergrundgeschichte und das Spieldesign grundlegend. Kurz zusammengefasst geht es darum, dass ein NPC des „Haupt"-Spiels in die Rolle eines virtuellen Spielleiters[214] schlüpft und fortan ein (wiederum digitalisiertes) Pen-&-PaperRollenspiel *im Spiel* stattfindet. Eine solche Geschichte innerhalb eines Universums kann auch als Modul bezeichnet werden [Bowman, 2010, S. 25]. Das Spieldesign bedient sich somit des Tricks einer weiteren Erzählschicht und beeinträchtigt die Kohärenz des Hauptspiels dadurch nicht. Die Welt wird weiterhin als glaubhaft dargestellt und in ihrer Integrität nicht verletzt [Weber, 2013]. Das Modul kann jederzeit und unabhängig vom umgebenden „Hauptspiel" angewählt werden und existiert als paralleler und in sich geschlossener Container, trägt aber dennoch zur Atmosphäre und Fortentwicklung der Geschichte des Hauptspiels bei.

8.4. Zusammenfassung

Es wurde gezeigt, dass der Begriff der Nichtlinearität ein fließender Begriff ist, dem, je nach Kontext, auch andere Bedeutungen innewohnen. Nichtlinearität kann zum einen auf eine Narration bezogen sein, die in ihrem Aufbau keine zeitlich festgesetzte Abfolge von Ereignissen erfordert. Als Beispiel hierfür wurde der Film Rashōmon herangezogen, welcher eine nichtlineare Geschichte in einem linearen Medium einbettet. Auf der anderen Seite kann Nichtlinearität als Attribuierung für das eingesetzte Medium dienen und meint im Bezug auf Computerspiele einen nichtlinearen Aufbau und die Bewegungsmöglichkeit in mehr als nur eine Richtung[215]. Es ist jedoch ebenso möglich, eine linear erzählte Geschichte in ein nichtlineares Medium einzubetten. Hierbei wurde das Spiel *Skyrim* als Beispiel erläutert.

[212] vgl. Abschnitt 8.3 auf Seite 169
[213] 2012, *2K Games, Inc.* [MobyGames.com, 2013b]
[214] Gleichzusetzen mit dem im Rahmen dieser Arbeit als *Game Master* bezeichneten Element.
[215] Hiermit sind natürlich nicht die Freiheitsgrade im Sinne von kartesischen Achsen gemeint, sondern die Freiheit, sich in multidirekationaler Art und Weise durch eine Spielwelt zu bewegen. Vgl. hierzu auch Abbildung 8.1 auf Seite 160.

Im folgenden Kapitel werden Möglichkeiten vorgestellt, eine Intelligenz zu erzeugen, die auf Basis der von den Spielenden getätigten Eingaben eine Welt dynamisch verändert, sodass eine modulare Geschichte im Sinne eines Rollenspiel umsetzbar wird.

„Anything that gives the illusion of intelligence to an appropriate level, thus making the game more immersive, challenging, and, most importantly, fun, can be considered game AI."

(Bourg [Bourg, 2004, S. 2])

KAPITEL 9

Künstliche Intelligenz

Die Definitionen für künstliche Intelligenz (KI) (engl. artificial intelligence) reichen von einfachem Pathfinding, über den Vergleich mit Individuen[216], welche über Intelligenz verfügen, bis hin zu vergleichsweise sehr komplexen Definitionen, welche ebenfalls ein Lernen und eigenes Schlussfolgern der künstlichen Intelligenz erfordern[217]. Dies liegt zum Teil daran, dass der Forschungszweig der künstlichen Intelligenz nicht ausschließlich im Bereich der Spieleentwicklung existiert, sondern viele verschiedene Anwendungsgebiete, wie unter anderem das der Spracherkennung, Robotik und Mustererkennung, umfasst [Dorn u. Gottlob, 2006, S. 975ff].

Da das in dieser Abschlussarbeit anvisierte Ziel der Realisierung eines Spiels entspricht, ist es möglich, die Definition auf ein spezifisches Aufgabenspektrum einzugrenzen. Die implementierte künstliche Intelligenz kann in Spielen verschiedene Rollen einnehmen, deren Herausforderungen bei der Implementierung zudem in der Art des Genres unterschieden werden können. Nach Fairclough u. a. besitzen die Spieler im Role Playing Game (RPG)-Genre mehr Freiheiten, sodass es hier schwieriger wird, die Konsistenz der Story beizubehalten[218]. Die Koordination der Konsistenz innerhalb eines Spiels ist ebenfalls dem Aufgabenbereich der AI zuzuordnen [Fairclough u. a., 2001, S. 4].

In diesem Kontext stellt sich daher die bereits als Kapitelzitat verwendete Ansicht des Autors Bourg als besonders geeignet heraus[219]:

[216] John McCarthy, Begründer der künstlichen Intelligenz und Gewinner des Touring Awards spezifiziert die Entwicklung von künstlicher Intelligenz folgendermaßen:

> *„Ziel der KI ist es, Maschinen zu entwickeln, die sich verhalten, als verfügten sie über Intelligenz."*

[McCarthy u. a., 2006]

[217] Ein Beispiel für einen solchen Definitionsansatz ist in [Janssen, 2013a] zu finden.

[218] vgl. hierzu auch Abschnitt 2.1 ab Seite 14

[219] Auch andere Autoren, wie bspw. Nareyek teilen diese Meinung [Nareyek, 2004, S. 59ff].

> *„Anything that gives the illusion of intelligence*[220] *to an appropriate level, thus making the game more immersive, challenging, and, most importantly, fun, can be considered game AI.“*

[Bourg, 2004, S. 2]

Von dieser Definition werden sämtliche virtuellen Vorgänge erfasst, beispielsweise auch ein möglichst optimales Ausrichten der Kamera auf die Spielenden, wobei hier genaue Rahmenbedingungen und das Optimum definiert werden müssen[221].

9.1. Künstliche Intelligenz im Bereich der Spieleprogrammierung

Computerspiele unterliegen einem ständigen technischen Wandel, der in der Mehrheit der Fälle mit einer stetig größer werdenden Komplexität einhergeht. Besonders deutlich zeigt sich dies im Bereich der künstlichen Intelligenz. Es existieren neue Herausforderungen in diesem Bereich, welche laut Nareyek durch die zahlreichen Features von modernen Computerspielen bedingt sind, von denen unter anderem die Folgenden zu nennen sind [Nareyek, 2000, S. 1]:

Echtzeit: Es steht lediglich eine kurze Zeit zum Berechnen bzw. Entscheiden zur Verfügung.

Dynamik: Computerspiele bieten eine hochdynamische Umgebung. Dies schließt sowohl veränderliche Spielwelten mit ein als auch Komponenten, welche innerhalb eines Computerspiels ineinandergreifen.

Unvollständiges Wissen: Ein Spielcharakter besitzt lediglich unvollständiges Wissen über die Welt.

Ressourcen: Die Ressourcen des Spielcharakters / der Spielwelt können limitiert sein.

KI ist einer der schwierigsten Bereiche in der Spieleprogrammierung, da intelligentes Verhalten eine subjektive Beurteilung erfährt und somit nicht für jeden Menschen gleich wahrgenommen wird respektive subjektiv die gleiche Qualität erreicht [Jacobs, 2008, S. 207][222]. Auch Nareyek abstrahiert

[220] Genau genommen müsste der Begriff der Intelligenz an dieser Stelle zusätzlich definiert werden. In dieser Ausarbeitung wird der Intelligenzbegriff in Form einer komplexen Übertragungsfunktion verstanden, welche Informationen der Spieler und Spielerinnen oder der Spielwelt automatisch in Reaktionen übersetzt, welche die Intention der Spielenden unterstützen. Dabei ist es unerheblich, in welcher Art und Weise diese Reaktionen errechnet werden.

[221] Mehr Informationen hierzu sind in Abschnitt 23 auf Seite 305 zu finden.

[222] Als Beispiel nennt Jacobs das plötzliche Erscheinen eines Löwen. Das Verhalten von betroffenen Personen variiert sehr stark. Von *ruhig stehen bleiben*, über *weglaufen* hin zum *Verwenden von Gegenständen als Wurfmittel*. Eine dieser Lösungen umzusetzen und dies intelligent zu bezeichnen, ist nicht wirklich sinnvoll, viel mehr muss eine Kombination und durch den Spieler bedingte Auswahl der möglichen Verhaltensarten verwendet werden [Jacobs, 2008, S. 207].

vom eigentlichen Ziel der KI und beschreibt ein Ergebnis, welches möglichst glaubhaft sein sollte und dem Spieler Spaß bereitet, als wesentlichen Aspekt [Nareyek, 2004, S. 60].

Insbesondere die Entwicklung und Verbesserung des Verhaltens der NPCs wird immer komplexer und umfangreicher. Dies reicht heutzutage bis zu einer Analyse des Verhaltens eines menschlichen Spielers und einem damit verbundenen Lerneffekt, sodass auf dieses Verhalten reagiert werden kann. Darüber hinaus können auch Muster beliebig vieler anderer Spieler hinzugezogen werden[223] und mit in die Berechnung einfließen, was das Verhalten des NPCs nachhaltig verbessert und erweitert. Ebenso wird auch das „soziale“ Verhalten und Emotionen von NPCs immer wichtiger, sodass in diesem Bereich viele neue Forschungsergebnisse existieren. Da die hier beschriebene Abschlussarbeit diese Teile nicht benötigt und umfasst, wird an dieser Stelle auf eine weiterführende Erläuterung verzichtet und auf weitere Lektüre, wie z.B. [Jacobs, 2008, Kap. 3], verwiesen.

9.2. Entscheidungsfindung

Im Rahmen dieser Abschlussarbeit gilt es, durch die Digitalisierung des Pen-&-Paper-Konzepts unter anderem den Game Master mittels künstlicher Intelligenz umzusetzen. Die Arbeit des Game Masters umfasst das so genannte *Decision Making*, welches als gleichnamiger Begriff in der Spieleentwicklung exisitert. Dieser Begriff beschreibt die Möglichkeit und Funktionsweise der automatischen Entscheidungsfindung. Die Eingabe eines solchen Systems besteht in der Regel aus einer bestimmten Wissensbasis und die Ausgabe entspricht einer auszuführenden Aktion, welche sich sowohl simpel[224] als auch umfangreich und auf weite Teile der Spielwelt auswirken kann, zum Beispiel in Form des Streamings eines neuen Levelabschnitts (siehe Abb. 9.1).

Auf welcher Basis und in welcher Art diese Entscheidungsfindung stattfindet, hängt von dem gewählten Ansatz ab. Die Quelle des Wissens variiert für gewöhnlich zwischen externem und internem Wissen. Ersteres beschreibt alles, was außerhalb eines Charakters (wenn es um einen solchen geht) vorhanden ist. Allgemein handelt es sich demnach um Wissen über die äußere (Spiel-)Welt, wie beispielsweise den Aktivierungsstatus eines Schalters oder welche Positionen die Objekte und Charaktere in der Spielwelt besitzen. Internes Wissen hingegen beschreibt - zumeist auf Charaktere bezogen - den inneren Zustand selbiger, wie beispielsweise deren Besitztümer oder Ziele. Die anhand dieser Informationen getroffene Entscheidung zieht eine Aktion der KI nach sich, welche wiederum Änderungen am internen oder externen Wissen und damit korrespondierenden Eigenschaften der Spielwelt vornimmt.

[223] Dies kann z.B. durch eine Synchronisierung über das Internet erreicht werden.
[224] Durch Schließung einer Tür in der Spielwelt beispielsweise.

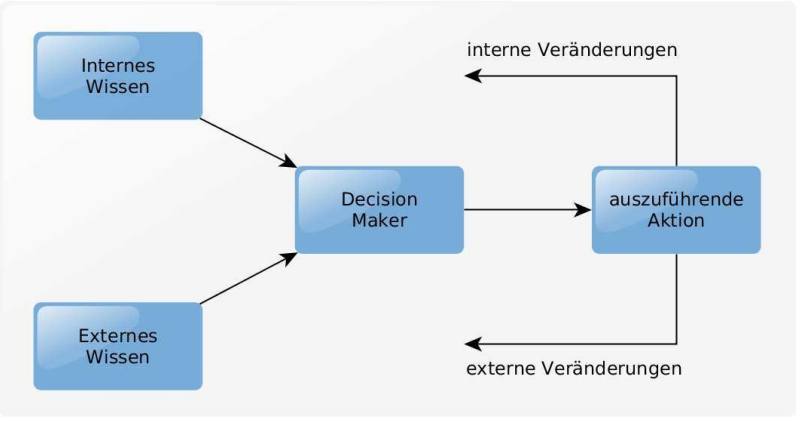


Abbildung 9.1.: In der Abbildung wird schematisch ein vereinfachtes Konzept der Entscheidungsfindung inklusive der integrierten Komponenten dargestellt.

nach [Millington u. Funge, 2009, Abb. 5.2]

Zudem lassen sich Ansätze auf Basis des zeitlichen Entscheidungsvorgangs einordnen. Im Voraus geplantes Verhalten (proaktiv) ist ebenso möglich wie aktives (reaktiv) Verhalten. Ferner kann das Entscheiden durch ein Lernen neuer Zusammenhänge und Fakten zusätzlich erweitert und komplexer gestaltet werden.

Im Folgenden wird eine Auswahl an verschiedenen Entscheidungsfindungsansätzen vorgestellt.

9.3. Ansätze zur Entscheidungsfindung

Im Allgemeinen wird zunächst einmal zwischen zwei grundlegenden Herangehensweisen unterschieden: deterministischen und nichtdeterministischen Ansätzen.

Bei einem deterministischen Algorithmus existiert kein „Unsicherheitsfaktor“. Bei gegebenem Wissen ist der Ausgang, sprich das Ergebnis des Algorithmus, immer vorhersehbar. In der Regel sind diese Arten von Algorithmen einfacher zu implementieren und auch schneller in der Ausführung [Bourg, 2004, S. 3]. Die namensgebende Charakteristik eines deterministischen Ansatzes bringt im Bereich der Spiele jedoch einen erheblichen Nachteil mit sich. Einem Spieler fällt sehr schnell auf, dass gleiche Handlungen immer dieselben Ergebnisse nach sich ziehen. Dies fällt besonders schnell auf, wenn die Spielmechanik eine bestimmte Interaktion forciert. Dem Algorithmus ist es weder möglich, aus vorangegangenen Ereignissen zu lernen, um dadurch sein Handlungsspielraum zu erweitern, noch kann

das Element des Zufalls in die Berechnung mit integriert werden. Hinzu kommt die Tatsache, dass jede mögliche Ausprägung vorher bedacht und vom Programmierer z.B. in Form von zusätzlichen Verzweigungen berücksichtigt werden muss.

Ein nichtdeterministischer Algorithmus zeichnet sich hingegen durch die Existenz eines Unsicherheitsfaktors aus. Dadurch wird das Ergebnis bis zu einem gewissen Grad unvorhersagbar. Im Spielverhalten äußert sich dies durch ein komplexer erscheinendes Möglichkeitsspektrum, wodurch sich der Spieler sowie die Spielerinnen nicht eingeschränkt fühlen und mehr Freiheiten erhalten. Der Nachteil hierbei ist jedoch das erschwerte Testen respektive die erschwerte Fehlersuche, bedingt durch die unberechenbare Zufallskomponente. Hinzu kommt, dass ein echter Nichtdeterminismus nicht implementierbar, sondern nur simulierbar ist [Lang, 2012].

Laut Bourg entsprechen insbesondere einfache, deterministische Verfahren nicht mehr dem aktuellen Stand und werden in Zukunft immer mehr von Ansätzen überholt, welche ein Lernverfahren involvieren. Es soll nicht mehr alles vorhersagbar und damit deterministisch bleiben, sondern über hybride Ansätze bis hin zu vollkommen eigenständigen Systemen gehen [Bourg, 2004, S. 5].

9.3.1. Skripting

Der einfachste Ansatz einer Entscheidungsfindung umfasst die Verwendung von Skripten. Hierbei wird Verhalten von NPCs und Objekten im Allgemeinen fest in deren Coderepräsentation einprogrammiert. Ein Skript beschreibt in diesem Kontext die sequentielle Ausführung von Programmcode. Diese sind verständlich, vorhersagbar, einfach zu implementieren sowie leicht erweiterbar und können mit ein wenig zusätzlichem Aufwand auch von Nicht-Programmierern geschrieben und verwendet werden [Spronck u. a., 2006]. In dem Kontext der Entscheidungsfindungsalgorithmen ist das Skripting jedoch nicht positiv behaftet, sondern als prozedural und fest verdrahtetes Codesegment innerhalb einer gesamten Anwendung negativ intoniert. Es existieren Lösungsansätze wie das so genannte *dynamic scripting*, welche ein fortlaufendes Lernen ermöglichen[225].

In der Literatur (z.B. in [Nareyek, 2000]) wird beim Skripting-Ansatz häufig von *reaktiven Agenten* gesprochen. Es existiert ein fest vorgegebenes Reiz-Reaktionsschema, welches unter bestimmten Bedingungen zu korrespondierenden Aktionen führt. In der Programmierung entspricht dies im Wesentlichen simplen *if-then*-Anweisungen. Für die Realisierung komplexer Umgebungen respektive komplexen Verhaltens werden mehrere Hundert solcher Anweisungen benötigt, um auf möglichst viele Situationen vorbereitet zu sein [Nareyek, 2000].

[225] Weitere Informationen hierzu sind beispielsweise in [Spronck u. a., 2006] zu finden.

9.3.2. Finite State Machines

Finite State Machine (FSM) (dt. endliche Automaten) stellen, im Vergleich zu den weiter unten aufgeführten Ansätzen, ein zwar simples jedoch gleichzeitig mächtiges KI-Konzept dar [DeLoura, 2002, S. 217][Nareyek, 2004], welches in [Fairclough u. a., 2001; Nareyek, 2000] und [Bourg, 2004, S. 4] als die am häufigsten eingesetzte Methode für künstliche Intelligenz in Spielen benannt wird[226]. FSM werden häufig auch in Verbindung mit den später vorgestellten Ansätzen der *Fuzzy-Logik* sowie mit *Spielbäumen* kombiniert, um das Ergebnis z.B. zufälliger zu gestalten [Bourg, 2004, S. 4][Millington u. Funge, 2009, S. 341ff].

Ein endlicher Automat besteht aus einer Menge von endlichen, vorher festgelegten Zuständen und befindet sich zu einem Zeitpunkt immer nur in *einem* dieser Zustände. Die Zustände implementieren das Verhalten des Automaten (in der Spielwelt des Spielobjekts). Zustände können mit definierten Zustandsüberführungen gewechselt werden; diese entsprechen im Kontext von Spielen i.d.R. dem Abgleichen von Variablen[227].

Differenzierter betrachtet existiert in der theoretischen Informatik eine weitere Unterscheidung von endlichen Automaten in *Akzeptoren* und *Transduktoren*.

Akzeptoren werden in [Witt u. Vossen, 2006, Kap. 2] insbesondere für die Akzeptanz von regulären Sprachen vorgestellt. Durch die Eingabe verändert der Automat seinen aktuellen Zustand so lange, bis die Abarbeitung der Eingabe beendet wurde oder keine weiteren Zustandsüberführungen ausgeführt werden können. Ist die Eingabe vollständig abgearbeitet und der Automat befindet sich in einem ausgezeichneten (so genannten *akzeptierenden*) Zustand, so wird die Eingabe akzeptiert, ansonsten verworfen.

Der Automat verarbeitet eine Eingabe in Form einer Zeichenfolge schrittweise. Das heißt, die Eingabe wird iterativ abgearbeitet und ein Zeichen nach dem anderen betrachtet. In diesem Zusammenhang wird ebenfalls von einem nach rechts offenem Eingabeband gesprochen, auf welchem die Eingabe steht. Diese besteht aus einem fest definierten Alphabet (Σ).

Neben dem Eingabeband existiert der Zustandsspeicher. Ein endlicher Automat befindet sich zu jedem Zeitpunkt in genau einem Zustand. Die gesamte Zustandsmenge eines endlichen Automaten wird mit dem Buchstaben Q gekennzeichnet. Jeder endliche Automat besitzt einen ausgezeichneten Startzustand, welcher in der Definition mit q_0 ausgezeichnet wird.

[226] In [Nareyek, 2000] werden diese als auslösbare Agenten beschrieben.
[227] In dem Computerspiele-Klassiker *Pac Man* waren die Geister bereits mit endlichen Automaten realisiert worden. Je nach Verhalten des Spielers wechselten die NPCs ihr Verhalten. [Bourg, 2004, S. 165]

Damit ein Automat seine Zustände wechseln kann, wird eine bestimmte Zustandsüberführungsfunktion δ benötigt. Diese verwendet den aktuellen Zustand sowie das nächste auf dem Eingabeband befindliche Zeichen, um einen Folgezustand zu berechnen. Zur besseren Übersicht der Überführungsfunktion und im Allgemein des gesamten Automaten wird zumeist ein Zustandsdiagramm verwendet (siehe Abb. 9.3).

Sobald alle Zeichen auf dem Eingabeband abgearbeitet wurden und der letzte Zustand als *akzeptierend* ausgezeichnet ist, wird die gesamte Eingabe akzeptiert. Derart gekennzeichnete Zustände werden in der Menge F zusammengefasst.

Ein deterministischer, endlicher Automat wird schließlich folgendermaßen definiert:

$$\mathfrak{A} = (Q,\ \Sigma,\ \delta,\ q_0,\ F)$$

Abbildung 9.2.: Formale Definition eines endlichen Automaten.

aus [Witt u. Vossen, 2006, Kap. 2]

Da es sich hierbei um deterministische Automaten handelt, existiert für jeden Zustand und jedes Eingabezeichen lediglich ein Folgezustand. Im Gegensatz hierzu sind ebenfalls nichtdeterministische Automaten vorhanden, welche mehrere Folgezustände besitzen können. Der Automat folgt dabei einem Weg zufällig und führt ein automatisches Backtracking durch, falls sich dieser Weg als falsch herausgestellt hat.

Transduktoren hingegen generieren anhand von Eingaben Ausgaben. Es findet im Prinzip eine Übersetzung statt. In diesem Zusammenhang werden Transduktoren häufig für Steuerungsaufgaben eingesetzt [Witt u. Vossen, 2006, Kap. 4]. Als weitergehende Literatur wird zuvor genannte Quelle empfohlen.

Bis auf das Akzeptanzkriterium und die vorher definierte Zeichenkette auf dem Eingabeband, entspricht die hier beschriebene Funktionsweise der Akzeptoren ebenfalls dem grundlegenden Konzept bei der Implementierung eines endlichen Automaten für eine künstliche Intelligenz. Die gelesenen Zeichen entsprechen in der Spielwelt ausgeführten Events, welche dem Automaten zugeführt und entsprechend verarbeitet werden. Beim Wechsel *von* und *in* einen Zustand können dabei verschiedene Aktionen und Operationen ausgeführt werden, welche die Spielwelt entsprechend modifizieren. Mit dem zur Verfügung stehenden Zustand können in der Vergangenheit liegende Informationen abgespeichert und bei der Entscheidungsfindung verwendet werden. Auf diese Weise sind auch so genannte *long term* Entscheidungen möglich. Bei dem zuvor vorgestellten Ansatz ist durch simple *if-*

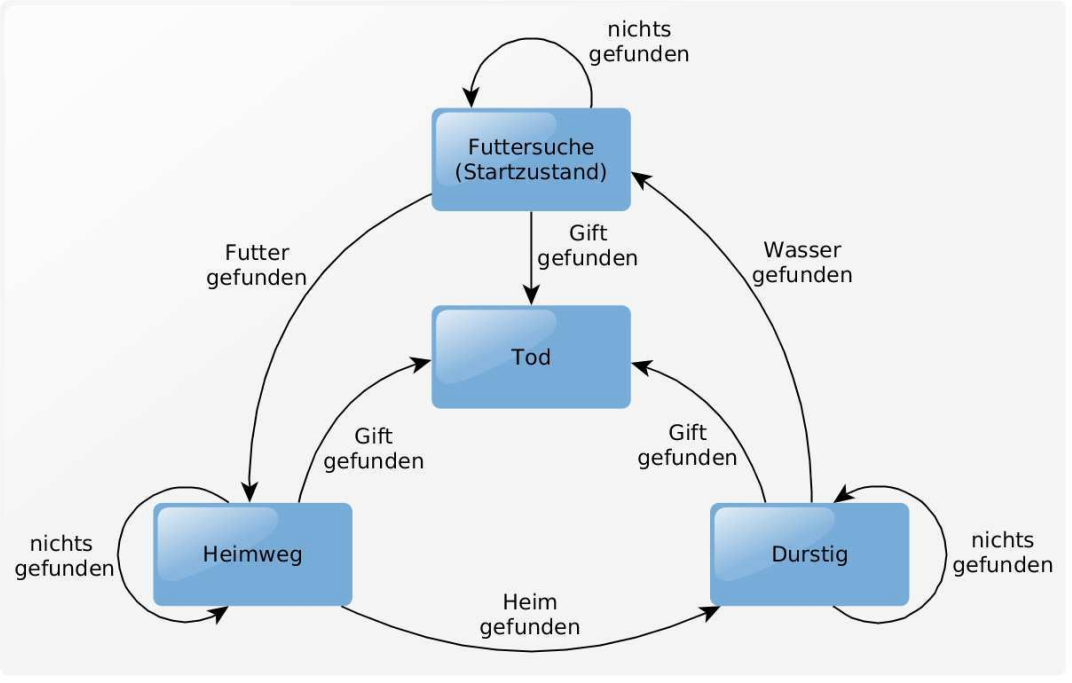


Abbildung 9.3.: In der Abbildung ist die Logik von Ameisen eines gleichnamigen Spiels dargestellt. Diese versuchen sich auf einer Karte vollkommen eigenständig zu bewegen mit dem Ziel Futter zu finden und nach Hause zurückzukehren. Zustände sind in der Abbildung als Rechtecke dargestellt, die Überführungen sind durch Pfeile symbolisiert.

nach [Bourg, 2004, Abb. 9-3]

then-Konstrukte eine Berücksichtigung von länger zurückliegenden Informationen respektive bereits vorangegangenen Entscheidungsfindungen nicht möglich.

9.3.3. Spiel-/Entscheidungsbäume

Das Darstellen von Spielsituationen ist ferner mit Hilfe eines Spielbaums möglich. Hauptsächlich werden Spielbäume für das Berechnen von Spielzügen in Brettspielen eingesetzt [Millington u. Funge, 2009, 647ff]. Knoten repräsentieren hierbei Spielzustände. Die davon abgehenden Kindknoten beschreiben diejenigen Positionen, welche von hier aus mit einem Zug erreicht werden können. Je nach Rechenleistung und zur Verfügung stehender Zeit kann beliebig weit in die Zukunft gesehen und mit Hilfe einer Bewertungsfunktion der beste (gewinnbringende) Pfad gewählt werden. Es wird dabei ver-

sucht, diese Bewertungsfunktion zu maximieren, um die des Gegenspielers zu minimieren.[228]. Da es sich bei einem klassischen Pen-&-Paper-Spiel nicht um ein rundenbasiertes Konzept handelt, wird dieser Ansatz innerhalb dieser Arbeit nicht weiter betrachtet[229].

Eine andere Möglichkeit, Spielsituationen mit Hilfe von Bäumen analysieren zu können, sind so genannte *Entscheidungsbäume* (siehe Abb. 9.4). Diese sind im Vergleich zu den zuvor beschriebenen endlichen Automat in der Komplexität und Abbildungsmöglichkeit eine Stufe einfacher einzuordnen. Die Knoten in einem Baum repräsentieren Fragestellungen, welche je nach Beantwortung in unterschiedliche Teilbäume führen. Die getroffenen Entscheidungen und damit einhergehende Anweisungen sind letztlich am Ende eines Pfades in den jeweiligen Blättern des Baumes zu finden. Entscheidungsbäume werden repetitiv von der Wurzel ausgehend bis zu einem Blatt hin durchlaufen. Die Entscheidungsvielfalt ist sehr beschränkt, da es sich um einen binären Baum handelt. Eine Erweiterung der möglichen Ausgänge ist durch die Einbringung weiterer Verschachtelungstiefen möglich, erhöht jedoch die Komplexität des Baumes [Millington u. Funge, 2009, S. 306]. Hinzu kommt der fehlende interne Status, welcher keine langfristigen Entscheidungsmöglichkeiten zulässt, es sei denn es wird für jeden möglichen Ausgang ein eigener Teilbaum erzeugt, was jedoch in einem sehr großen Gesamtbaum endet. Häufig werden Entscheidungsbäume für Strategiespiele und Entscheidungen auf hoher Ebene verwendet [Nareyek, 2004]. Ebenso ist einer der Haupteinsatzzwecke die Animationssteuerung und das Verhalten von Charakteren [Millington u. Funge, 2009, S. 302].

Aus programmiertechnischer Sicht handelt es sich hierbei ebenfalls um *if-then*-Konstrukte, jedoch um wesentlich komplexere. In höheren Programmiersprachen existieren ebenfalls spezielle Datenstrukturen für die Repräsentation von Bäumen im Allgemeinen, um sowohl die Berechnungsgeschwindigkeit zu erhöhen als auch die Komplexität zu verringern.

Wie bereits in Abschnitt 9.3.2 erläutert, ist eine Kombination mit endlichen Automaten möglich. Zustandsüberführungen werden dabei durch Entscheidungsbäume ersetzt, sodass mehr Komplexität und Flexibilität realisiert werden kann [Millington u. Funge, 2009, S. 341].

9.3.4. Fuzzy-Logik

Traditionelle Entscheidungsfindung stützte sich lange Zeit auf boolesche Aussagen und erlaubte somit die Prüfung einer Kondition auf den Wert *true* oder *false*. Fuzzy-Logik hingegen basiert auf einem Konzept, welches die Verwendung eines Adjektivs ermöglicht. Dadurch ist es möglich, mathematische

[228] bekannt als Nullsummenspiel [Leininger u. Amann, 2007, S. 19]
[229] Dem interessierten Leser wird als weiterführende Literatur unter anderem [Holler, 2009] empfohlen.

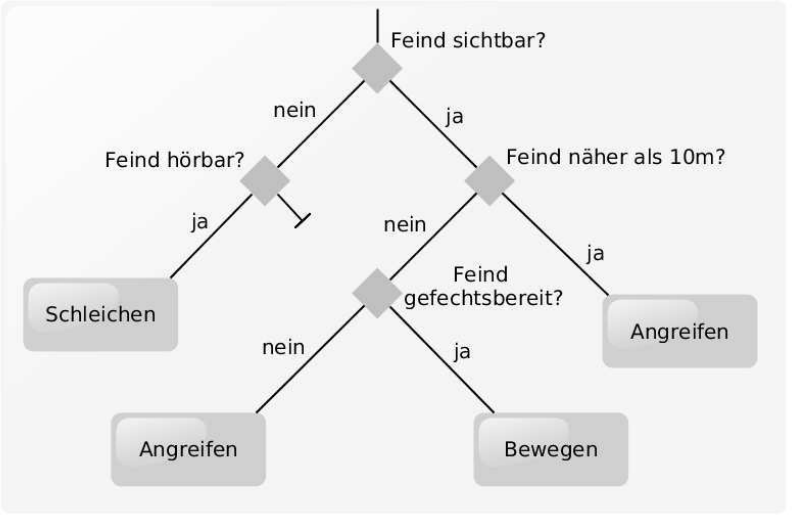


Abbildung 9.4.: Der Algorithmus folgt dem Baum absteigend den Knoten entlang, an welchen Entscheidungen getroffen werden. Die Blätter repräsentieren letztlich durchzuführende Aktionen. Boolesche Operationen wie ein *and* oder *or* entsprechen Kombinationen von Knoten und sind somit ebenfalls abbildbar. Komplexe Entscheidungen lassen sich nur durch tief ineinander verschachtelte Bäume abbilden.

nach [Millington u. Funge, 2009, Abb. 5.4]

Größen wie „ziemlich groß“ oder „mittel“ zu definieren [DeLoura, 2002, S. 309]. Traditionelle Logik basiert zumeist auf diskreten Mengen. Zunächst wird eine Menge definiert und anschließend in Relation zum umliegenden Zahlenbereich gebracht. Bei der Verwendung der reellen Zahlen als Zahlenbereich und einer bestimmte Menge aus diesen (z.B. $\{1.8, 1.9, 2.0\}$), findet eine Abgrenzung der Elemente statt.

Anhand eines Beispiel aus [DeLoura, 2002, S. 310] soll die Funktionsweise der Fuzzy-Logik verdeutlicht werden. Die Menge $\{1.8, 1.9, 2.0\}$ umfasst die Körpergröße von Personen in Metern, welche als „groß“ betitelt werden. Eine Person mit einer Körpergröße von 1.79 *m* ist laut dieser Definition nicht groß, da das untere Intervall nicht erreicht wird. An dieser Stelle kann mit Hilfe der Fuzzy-Logik der Bereich aufgetrennt und etwas wie „ein wenig in der Menge“ oder „fast vollständig in der Menge“ realisiert werden (siehe Abb. 9.5). In dem konkreten Beispiel wäre es möglich, die gemessene Person als „fast vollständig in der Menge“ einordnen.

Mit der Beschreibung von Bourg ergibt sich eine andere, intuitivere Erklärung. Diese stellt einen Vergleich, zu der Art wie Menschen Situationen analysieren, her [Bourg, 2004, S. 188ff]. Menschen

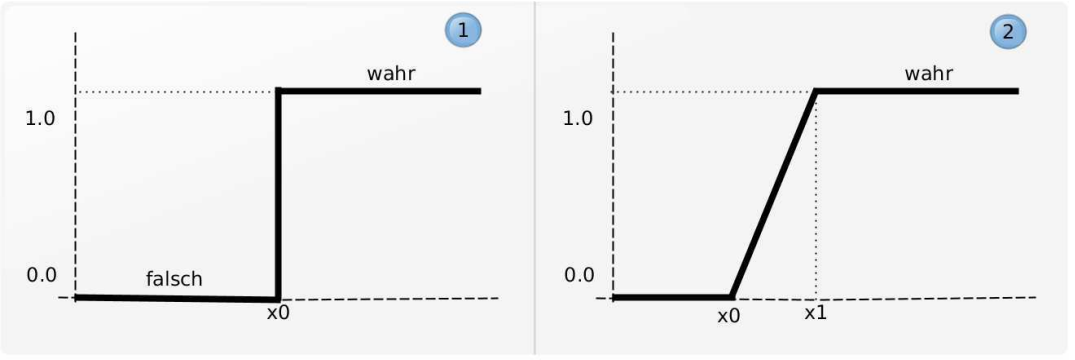


Abbildung 9.5.: In der Abbildung wird zum einen eine Zugehörigkeitsfunktion der booleschen Logik (1) dargestellt und zum anderen eine graduelle Zugehörigkeitsfunktion (2), welche durch Fuzzy Logik möglich wird. Während Werte größer als x_1 eine absolute Zugehörigkeit und Werte kleiner x_0 eine absolute Nicht-Zugehörigkeit symbolisieren, ergeben Zwischenwerte einen gewissen Grad selbiger an.

nach [Bourg, 2004, S. 194]

verwenden unpräzise und generalisierende Fakten, um Entscheidungen zu treffen. Meistens fehlen exakte Daten oder Berechnungsmöglichkeiten, sodass vergleichende Aussagen verwendet werden. Bei der Fuzzy-Logik wird versucht, diesen Ansatz auf den Rechner zur übertragen, sodass nicht mehr auf exakte Werte geprüft wird, sondern auf Adjektive[230].

Laut DeLoura wird der hier beschriebene Ansatz hauptsächlich für menschlicheres Verhalten von NPCs verwendet, weil diese dadurch realistischer nachgebildet werden können [DeLoura, 2002, S. 318].

Fuzzy Logik kann als Erweiterung und gleichzeitig Vereinfachung endlicher Automaten betrachtet werden. Laut Bourg benötigt eine KI, welche diesen Ansatz verwendet, zwischen 50% und 80% weniger Regeln als klassische Ansätze [Bourg, 2004, S. 190]. Es kann demnach weiterhin in der intuitiven Form der Regeln sowie zustandsbasiert gearbeitet werden, jedoch mit einer Verringerung der Komplexität und ohne vorherige Lerndatensätze, wie es bei anderen Systemen, wie neuronalen Netzen beispielsweise der Fall ist.

[230] Es findet keine Werteüberprüfung in Verbindung mit mathematischen Operatoren statt (Energie = 10) sondern beispielsweise so etwas wie „ausreichend Energie“.

9.3.5. Regelbasierte Systeme

Neben den bisher vorgestellten Ansätzen existiert zudem die regelbasierte künstliche Intelligenz. Diese stellt laut Bourg mittlerweile den am weitesten verbreiteten und meist genutzten Ansatz für KI sowohl in der realen als auch virtuellen Welt dar [Bourg, 2004, S. 212]. Besonders in den 1970er und 1980er Jahren war dieser Ansatz sehr weit verbreitet, sodass nahezu jedes bekannte KI-System damit realisiert wurde [Millington u. Funge, 2009, S. 403]. Der Ansatz der regelbasierten künstlichen Intelligenz unterliegt in der praktischen Anwendung im Bereich der Spielentwicklung einer gewissen Fluktuation. Das resultiert laut Millington u. Funge aus einer mangelnden Reputation dieses Ansatzes und der weit verbreiteten Meinung einer schweren Implementierbarkeit [Millington u. Funge, 2009, S. 403].

In der Spielprogrammierung entsprechen regelbasierte Systeme Expertensystemen in einer vereinfachten Form. Diese Art von Systemen sind eine Erweiterung der in den 70er Jahren angestrebten allgemeinen Problemlösungsansätze [Heinemann, 2006][231]. Expertensysteme werden in Szenarien eingesetzt, welche ein umfangreiches, domänenspezifisches Fachwissen erfordern. Die Wissensbasis kann dabei variieren und von realen Benutzern (sogenannten Experten) gepflegt und erweitert werden. Diese besteht zum einen aus Variablen (sogenannte Fakten) und zum anderen aus Regeln. Die Bezüge und Zusammenhänge der realen Welt müssen dem System verständlich dargelegt werden, wofür sich, nach Puppe, die logische Wissensrepräsentation durch Regeln in der Form einer Implikation („Wenn X, dann Y“[232]) besonders eignen [Puppe, 1990, S. 9]. Die Entscheidungsfindung wird mit Hilfe der namensgebenden Regeln umgesetzt. Diese bestehen aus einer Bedingung[233] sowie einer Folgerung[234] und bestimmen das Verhalten des Systems. Für die Bedingung wird auf das in der Datenbank befindliche Wissen zurückgegriffen. Sind bestimmte Bedingungen erfüllt, kann die Basis unter anderem um weitere Fakten erweitert werden. Umfangreiche und komplexe Expertensystemen können darüber hinaus auch eigenständig neue Zusammenhänge erlernen, indem das Konzept der induktiven Inferenz verwendet wird. Als induktive Inferenz wird das Schlussfolgern einer Regel aus vorgegeben Beispielen bezeichnet[235] [Austermann, 2012, S. 2].

[231] bekannt als *General Problem Solver*

[232] JBoss, u.a. Entwickler eines regelbasierten Systems für Java, spezifiziert die Verwendung des „when“ in der englischen Sprache genauer:

> *„Why use „when” instead of „if”?” „When” was chosen over „if” because „if” is normally part of a procedural execution flow, where, at a specific point in time, a condition is to be checked. In contrast, „when” indicates that the condition evaluation is not tied to a specific evaluation sequence or point in time, but that it happens continually, at any time during the life time of the engine; whenever the condition is met, the actions are executed.“*

[Team, 2013]

[233] „left hand side (LHS)“

[234] „right hand side (RHS)“

[235] Häufig verwendet in Intelligenztests, bei denen es eine vorgegebene Zahlenreihe zu vervollständigen gilt. Als Einstiegsliteratur wird dabei z.B. [Konstantinidis u. Kopp, 2013] empfohlen.

Ein Expertensystem ist ein komplexes System, welches aus mehreren Komponenten besteht. Neben der bereits genannten Wissensbasis und der Möglichkeit diese zu erweitern (*Inferenzmotor*), existiert eine Erklärungskomponente und eine Wissenserwerbskomponente[236].

In Spielen wird am häufigsten ein Verfahren namens *forward chaining*[237] eingesetzt. Hierbei wird von einer initialisierten Datenbank ausgegangen, worauf folgend die Regeln durchlaufen und gegebenenfalls ausgeführt werden. Das Zutreffen von Bedingungen führt dazu, dass deren Folgerungen ausgeführt werden, welche in der Regel Änderungen der Spielwelt nach sich ziehen und ferner zu Modifikationen in der Wissensbasis führen können (siehe Abb. 9.6). Alternativ ist es möglich, dass sich durch den Spielverlauf Wissen in der Datenbank ändert. Auf Grundlage der vorangegangenen Änderungen ist es möglich, dass Bedingungen anderer Regeln wahr werden, sodass deren Folgerungen und damit Aktionen ausführbar werden. Neben dem Verfahren existiert zudem eine umgekehrt agierende Herangehensweise namens *backward chaining*. Neben dem jeweils aktuellen Wissenstand existiert ein so genanntes Zielwissen. Durch das Auswerten der Folgerungen, nicht der Bedingungen, wird versucht, umgekehrt eine Überführung der Ziel- in die Ausgangswelt zu finden. Dieses wird jedoch in anderen Bereichen als Spielen eingesetzt[238] [Millington u. Funge, 2009, S. 407].

Die Regeln sind typischerweise nicht ineinander verschachtelt, sodass es möglich ist, dass mehrere Regeln in einem Durchlauf wahr werden. Hierfür existiert die so genannte *Konfliktresolution*. Der einfachste Ansatz ist hierbei vorauszusetzen, dass alle Regeln nach deren Priorität absteigend sortiert sind, sodass die erste zutreffende Regel die wichtigste darstellt und damit sofort ausgeführt werden sollte. Da es dadurch passieren kann, dass die Ausführung in einer Endlosschleife endet, sofern sich die Variablen in der Bedingung nicht ändern, kann dieser Ansatz durch das Einführen zusätzlicher Regelvariablen verhindert werden [Millington u. Funge, 2009, S. 418]. Jede Regel besitzt eine spezielle Variable, welche nach Ausführung umgesetzt wird, sodass ein erneutes Eintreffen nicht ohne weiteres möglich ist.

Weiterhin ist ein listen-basierter Ansatz möglich, in dem alle zutreffenden Regeln in einer Liste sortiert dargestellt werden. Die jeweils erste Regel wird nach einem Durchgang ausgeführt und aus der Liste entfernt, bei erneutem Eintreten wird diese an das Ende der Liste angefügt, sodass weiter vorn befindliche Regeln den Vorzug erhalten. Darüber hinaus kann durch eine spezielle Attribution eine Regel als besonders wichtig deklariert werden, sodass diese stets bevorzugt behandelt wird. Dies geht sogar so weit, dass derzeit ausgeführte Operationen sofort beendet werden, damit diese spezielle Regel unverzüglich ausgeführt werden kann[239]. [Millington u. Funge, 2009, S. 418]

[236] Eine tiefer gehende Einführung in die Thematik der Expertensysteme wird u.a. in [Reif, 2000; Puppe, 1990] gegeben.
[237] Dies wird in der Logik als *modus ponens* bezeichnet.
[238] Anwendung findet das backward chaining z.B. bei automatischen Theorembeweisern oder Inferenzmaschinen.
[239] Im Fachjargon als „Interrupt“ bekannt.

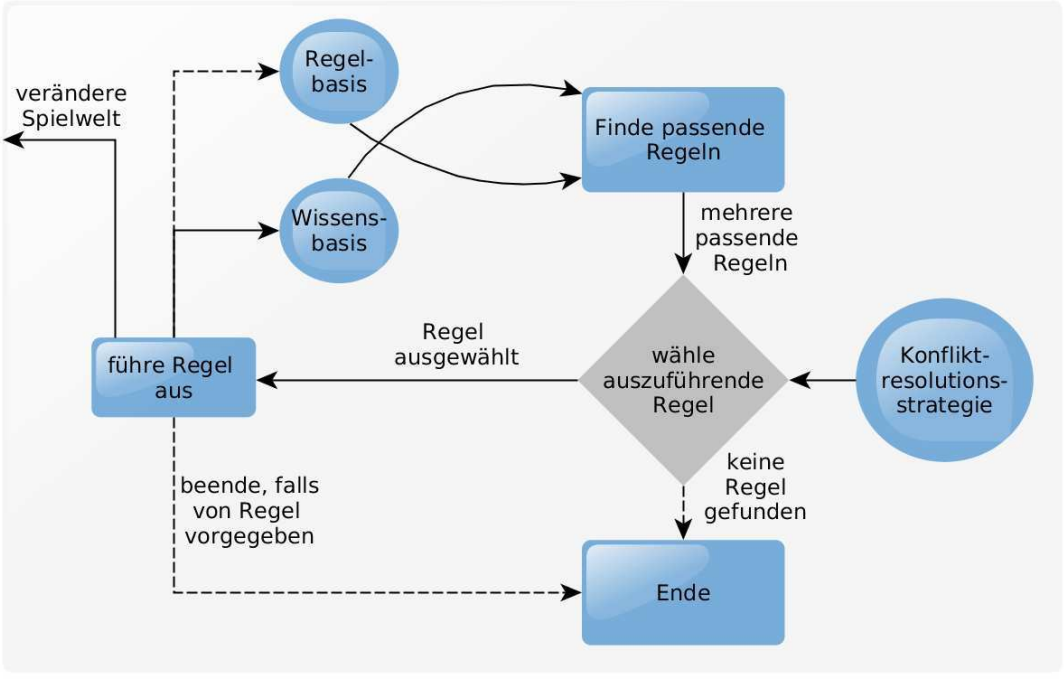


Abbildung 9.6.: In der Abbildung wird der Ablauf eines regelbasierten Systems dargestellt. Die gestrichelten Linien finden in Spielen eher selten Anwendung, da die Abarbeitung lediglich mit Spielende aufhört. Bei regelbasierten Systemen wird jedoch meistens angestrebt eine gewisse Anzahl neuer Regeln und Zusammenhänge zu erschließen, woraufhin die Bearbeitung beendet werden kann. Hauptsächliche Aktionen im Zusammenhang mit Spielen sind die Veränderung der Spielwelt und der Wissensbasis. Ebenfalls ist der gesamte Ablauf hierbei zirkulär zu betrachten. Häufig wird auch ein Event-getriebener Ansatz implementiert, welcher bei Änderungen in der Wissensbasis nicht sämtliche Regeln, sondern explizit möglicherweise betroffene überprüft.

nach [Team, 2013, Abb. 5-2]

Darüber hinaus kann ein regelbasiertes System eine Konfliktresolution jedoch auch in der Art implementieren, dass mehrere Regeln gleichzeitig ausgeführt werden dürfen [Lau, 2011]. Dadurch ergeben sich eine Vielzahl neuer Möglichkeiten, welche durch die bisher vorgestellten Ansätze nicht umsetzbar wären. Weder in einem Baum noch in einem deterministischen endlichen Automaten existieren beispielsweise mehrere gleichzeitig ausführbare Folgezustände.

9.3.6. Neuronale Netze

Neuronale Netze implementieren andere Modelle zum Lösen von Problemen. Es wird nicht auf standardisierte Methoden der deterministischen Modelle zurückgegriffen, sondern auf parallele, verteilte und probabilistische Ansätze [DeLoura, 2002, S. 320ff]. Diese basieren auf einem Netzwerk aus Zellen, welche Lösungen zu Problemen finden, diese analysieren und zueinander in Korrelation setzen können [DeLoura, 2002, S. 320ff].

Neuronale Netze basieren auf dem Versuch der Nachbildung der Funktionsweise des menschlichen Gehirns. Es wird versucht ein vereinfachtes Modell des Neurons nachzuempfinden, mit der Zielsetzung ein lernfähiges System zu erzeugen [DeLoura, 2002, S. 320].

Die Anwendung in Spielen ist komplex, weil eine Integration und Kombination weiterer algorithmischer Lösungen wie deterministischer oder genetischer Ansätze benötigt werden, damit höherwertige Funktionen abgebildet werden können [DeLoura, 2002, S. 321]. Bei den angestrebten Anwendungsfällen handelt es sich um komplexe Szenarien, wie dem des Erinnerungsvermögens von NPCs oder der Klassifikation von Objekten sowie verschiedenen Anwendungsfällen bei der Bilderkennung. Bei einer großen Datenmenge lernt das Netz Zusammenhänge kennen und schafft es so, stetig bessere Ergebnisse zu liefern. Laut Millington u. Funge werden neuronale Netze für Anwendungsfälle präferiert, in denen enorme Datenmengen zur Verfügung stehen und analysiert werden müssen oder ein Lernen benötigt wird [Millington u. Funge, 2009, S. 628ff]. Da der hauptsächliche Einsatzzweck damit von dem in dieser Ausarbeitung benötigten stark abweicht, wird dem interessierten Leser an dieser Stelle weitere Lektüre [Rey u. Wender, 2010; DeLoura, 2002] empfohlen, um einen tiefer gehenden Einblick über die hier nur in Ansätzen wiedergegebene Materie zu erhalten.

9.3.7. Zielorientierte Ansätze

Die in den vorherigen Abschnitten erläuterten Ansätze lassen sich unter der Kategorie *Event-getrieben* vereinigen. Der Vollständigkeit halber wird im Folgenden sehr kompakt eine andere Möglichkeit der Entscheidungsfindung vorgestellt, welche auf *zielorientierten* Systemen[240] basiert. Diese besitzen sowohl ein Modell der Zielwelt, als auch der Ausgangswelt und versuchen anhand einer Folge von durchführbaren Aktionen das Zielmodell zu erreichen. Die Folge, welche ein Erreichen ermöglicht, wird *Plan* genannt. Ein solcher kann durch bestimmte Verfeinerungsstrategien aktualisiert und konkretisiert werden. Das zu Beginn existierende Vorhaben kann während der Laufzeit je nach Gegebenheiten modifiziert werden.

[240] auch als deliberative Agenten bekannt

Diese Art der Entscheidungsfindung findet in vergleichsweise wenigen Spielen Anwendung. Da die Ziele eines Agenten meistens komplex und nicht vollkommen spezifiziert sind, bedeutet diese Methode einen erheblichen Programmieraufwand, welchen sich lediglich große Spielehersteller leisten können [Millington u. Funge, 2009, S. 401]. Diese Art der Entscheidungsfindung ergibt nicht in allen Szenarien Sinn, vorstellbar ist der Einsatz von zielorientierten Ansätzen z.B. bei der Implementierung von NPCs, da diese eigene Ziele verfolgen könnten.

9.4. Zusammenfassung

In diesem Kapitel wurden verschiedene Ansätze zur Entscheidungsfindung einander gegenübergestellt. Eine Entscheidungsfindung ist wichtig, damit ein Game Master im Sinne eines Rollenspiels Entscheidungen der Spielenden auswerten und in neue Spielsituationen überführen kann.

Weiterhin wurde gezeigt, dass der Intelligenzbegriff, gerade im Bereich der Computerspiele, für mehr stehen kann als nur eine wie auch immer geartete Entscheidungsmaschine, die den Spielverlauf beeinflusst[241]. Eine weitere, wichtige Form der Entscheidungsfindung bei Computerspielen ist die Wahl einer richtigen Kamera-Implementierung und die richtige Positionierung derselbigen. Aus diesem Grund werden im folgenden Kapitel einige Kamerasysteme in Spielen vor- und gegenüber gestellt.

[241] vgl. Kapitel 9 ab Seite 176

„All fourteen games reviewed in our study were designed for kids to play together through a shared screen. In these situations, camera set up emerged as an important design component.“

(Seif El-Nasr u. a. [Seif El-Nasr u. a., 2010, S. 255])

KAPITEL 10

Kamerasysteme in Spielen

Die Wahl und Implementierung des Kamerasystems ist einer der wichtigsten Bestandteile jedes digitalen Spiels. Frühere Computer- und Videospiele setzten den Blickwinkel der Rezipienten häufig fix und mussten das Gameplay der Technik unterordnen [Halper, 2003, S. 77]. Dies ist vergleichbar mit der früheren Ära des Films, als Regisseure und Darsteller in ihren Möglichkeiten sehr beschränkt waren: Es gab weder Kamerakräne noch CGI. Die Darsteller waren beschränkt auf ein fixes Bühnenbild, dass in früherer Zeit oftmals als direkte Adaption des Theaters in sehr kurzen Einstellungen aufgenommen wurden, die völlig ohne Schnitt auskamen [Fairservice, 2001, S. 11].

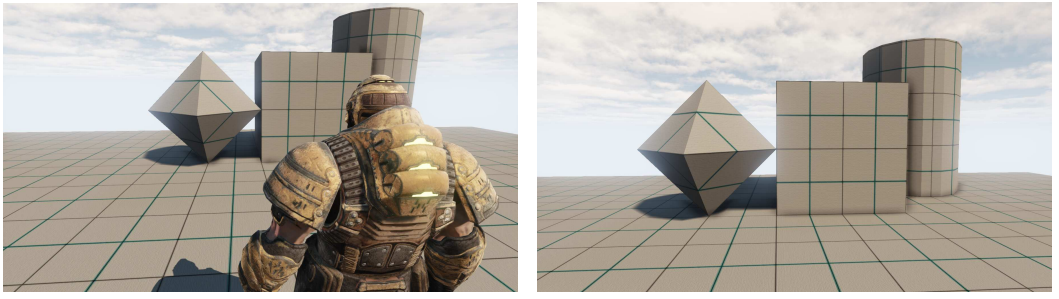
Heutige Erzähltechniken und Kamerasysteme in Spielen ergänzen sich gegenseitig und werden zielgerichtet eingesetzt: Die Technik passt sich der Mechanik und dem Inhalt des Spiels an, nicht umgekehrt. Eine handwerklich gut eingesetzte Kamera kann den Erzählprozess und die Handlung sowohl eines Films als auch eines Spiels genau so vorantreiben, wie eine schlechte Wahl der Kamera das Erlebnis und den Spaß der Spielenden schmälert [Halper, 2003, S. 77]. Die virtuelle Kamera stellt den Blick des oder der Spielenden auf die virtuelle Umgebung dar und beeinflusst die Rezeption in der Rolle als „*Fenster zur Welt*“ [Voyles, 2013, S. 53]. Daher ist es wichtig, die gängigsten verschiedenen Verfahren kurz gegenüberzustellen, um die spätere Designentscheidung sinnvoll begründen zu können.

10.1. Third-Person und First-Person

Bei der *Third-Person-Kamera*, welche auch in Abbildung 10.1(a) zu sehen ist, handelt es sich um die bei Computer- und Videospielen am häufigsten[242] anzutreffende Kamera-Variante. Eines der bekanntesten Franchises mit dieser Perspektive ist die 1996 gestartete *Tomb-Raider*-Reihe[243] sowie in neuerer

[242] Die Webseite `http://Mobygames.com/`, welche sich selbst das Ziel gesetzt hat, alle jemals entwickelten Computer- und Videospiele zu katalogisieren, listet zum Zeitpunkt dieser Arbeit über 18.000 (von insgesamt etwa 75.000) Titel mit Third-Person-Kameraperspektive auf [MobyGames.com, 2013g,t].

[243] 1996, *Core Design Ltd.* [MobyGames.com, 2013x]



(a) Third-Person-Kamera (b) First-Person-Kamera

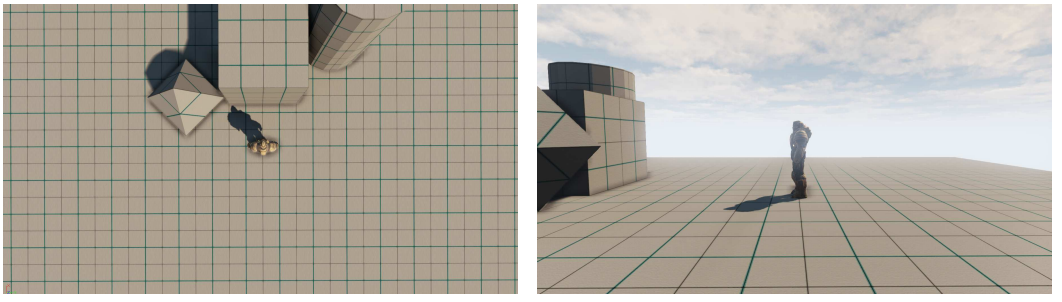
Abbildung 10.1.: Die linke Abbildung zeigt die Third-Person-Kamera-Perspektive, die rechte Abbildung zeigt das Geschehen aus der „Ich“ (auch „Ego“)-Perspektive.

Generation die 2006 gestartete *Gears of War*-Reihe[244]. Bei dieser Ansicht wird das Spielgeschehen oftmals aus der Schulter-Perspektive verfolgt, obwohl das Genre der Third-Person-Spiele eine größere Distinktion zulässt - eine Third-Person-Kamera deshalb grundsätzlich mit einer Schulterperspektive gleichzusetzen wäre nicht völlig korrekt [Thorn, 2012, S. 626]. Diese Kamera-Perspektive eignet sich ebenfalls für eine Story-unterstützende Darstellungsweise, bei welcher der Avatar des oder der Spielenden im Geschehen partizipiert und eine aktive Rolle einnehmen kann [Geddes, 2008].

Von der *Third-Person-Kamera* abzugrenzen ist die so genannte *First-Person-Kamera*, bei der das Spielgeschehen direkt aus der Perspektive eines einzelnen Avatars eingefangen wird. Diese Ansicht ist in Abbildung 10.1(b) dargestellt. Populär wurde dieses Kamerasystem durch Spiele wie *Duke Nukem 3D*[245] oder *Half-Life*[246] und stellt die am zweithäufigsten gewählte Kamera-Perspektive dar[247]. Die Ansicht aus der „Ich“-Perspektive entfernt den Charakter aus dem sichtbaren Bildbereich und ermöglicht ein immersiveres Spielgefühl und ein generell größeres Sichtfeld [Geddes, 2008]. Obwohl sich die Konzepte der *Third-Person-Kamera* und *First-Person-Kamera* sehr ähneln, ermöglichen sie sehr unterschiedliche Konzepte. In der *Third-Person-Perspektive* können Elemente wie beispielsweise ein Deckungssystem leichter verbaut werden, ebenso könnten Veränderungen an der Spielfigur visualisiert werden [Geddes, 2008].

[244] 2006, *Epic Games, Inc.* [MobyGames.com, 2013o]
[245] 1996, *3D Realms Entertainment* [MobyGames.com, 2013e]
[246] 1998, *Valve L.L.C.* [MobyGames.com, 2013q]
[247] Mit knapp 9000 bei `http://Mobygames.com/` gelisteten Spielen, welche diese Ansicht benutzen [MobyGames.com, 2013f].

10.2. Top-Down und Side-Scrolling



(a) Top-Down-Kamera (b) Side-Scrolling-Kamera

Abbildung 10.2.: Die linke Abbildung zeigt die Top-Down-Kamera, die rechte Abbildung zeigt die Side-Scrolling-Kamera.

Im Gegensatz zu den beiden als erstes beschriebenen Systemen sind die *Top-Down-Kamera* und die *Side-Scrolling-Kamera* nicht mehr zwangsläufig an nur eine Spielfigur gebunden, sondern erlauben es im Sinne einer Weitwinkel-Aufnahme, das Spielgeschehen frei einzufangen.

Die *Top-Down-Kamera*, dargestellt in Abbildung 10.2(a), fand hierbei in fast ebenso vielen Spielen[248] eine Anwendung wie die *First-Person-Kamera*, verfolgt jedoch ein grundsätzlich davon verschiedenes Prinzip. Bekannt wurde dieses Kameraansicht durch Titel wie *Legend of Zelda*[249] und *Grand Theft Auto*[250]. Durch die namensgebende Draufsicht wird eine Übersicht geschaffen, bei dem das rezipierte Sichtfeld nicht mehr von der Blickrichtung eines bestimmten Charakters abhängig sein muss, sondern das Geschehen aus einem neutralen Standpunkt betrachtet. Die Technik ist sowohl im 2D- als auch 3D-Bereich anwendbar und ermöglicht den Fokus auf die Gesamtsituation und Umgebung, während Details und eine Personifizierung des Spielerlebnisses durch die größere Distanz zurückgestellt werden[251].

In Abbildung 10.2(b) schließlich ist die so genannte *Side-Scrolling-Kamera* dargestellt. Diese Darstellungsform hat mit der zuvor beschriebenen Top-Down-Ansicht gemein, dass sich mit deren Anwendung ebenfalls auf eine fixe Perspektive festgelegt wird, bei der die Kamera nur innerhalb ihrer Koplanarität verschoben wird. Der Unterschied liegt darin, dass die Einschränkung der Dreidimensiona-

[248] `http://Mobygames.com/` listet knapp über 8500 Spiele mit dieser Ansicht auf [MobyGames.com, 2013i].
[249] 1986, *Nintendo Co., Ltd.* [MobyGames.com, 2013v]
[250] 1997, *DMA Design Limited* [MobyGames.com, 2013p]
[251] *Grand Theft Auto* beispielsweise bedient sich eines stufenlosen Zooms, bei dem die Kamera auf den Avatar zentriert wird und sich immer weiter von diesem weg bewegt, je schneller dieser sich durch die Spielwelt bewegt wird [MobyGames.com, 2013p].

lität hier nicht auf die Fläche, sondern auf die Höhe und die unmittelbare Umgebung des Geschehens festgelegt wird. Dadurch eignet sich das Verfahren besonders gut für das Genre der so genannten *Jump & Runs*, bei denen meist eine Spielfigur durch mehrere Hindernisse und durch besondere Fingerfertigkeit in den nächsten Abschnitt überführt werden muss. Die *Sidecrolling-Kamera* wird etwa halb so häufig eingesetzt wie die *Top-Down-Kamera*, was jedoch nicht zwangsläufig auf technischen Fortschritt und damit verbundenen Alternativen zurückgeführt werden muss, sondern auf die hohe Genre-Abhängigkeit dieser Darstellungsweise[252]. Bekannte Vertreter dieser Gattung sind *Super Mario Bros*[253] und *Mega Man*[254].

10.3. Isometrische Kamera

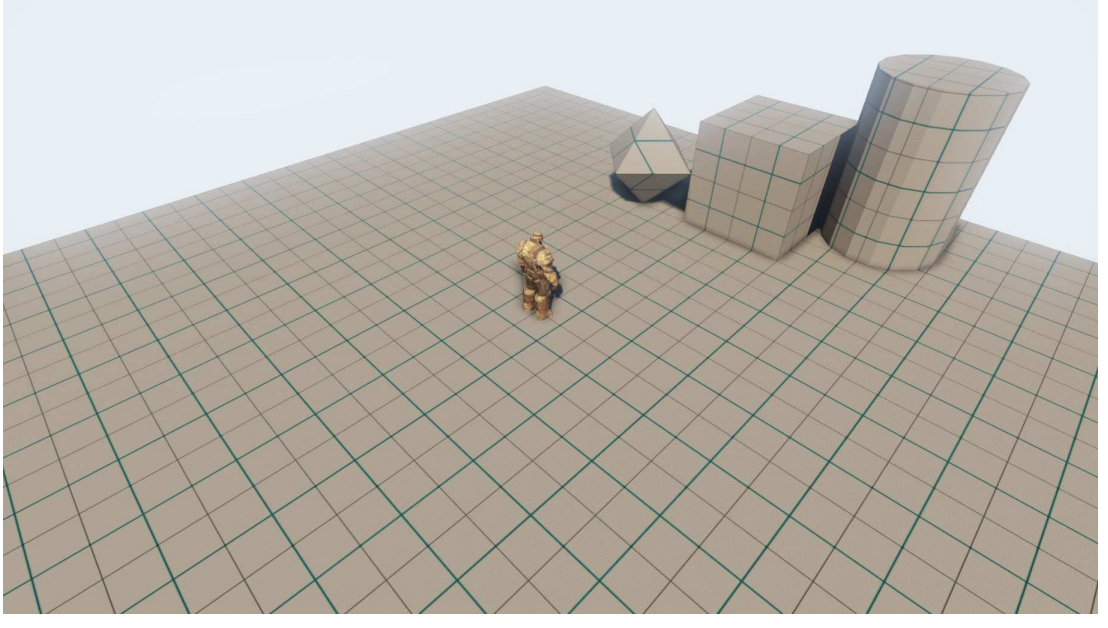


Abbildung 10.3.: Die isometrische Kamera-Ansicht

Eine technische Weiterentwicklung der in vielen Rollenspielen schon erfolgreich eingesetzten Topdown-Ansicht stellt die so genannte *Isometrische Kameraperspektive* dar, welche auch in Abbildung 10.3 zu finden ist. Dieses Kamerasystem vereint die Übersicht einer Topdown-Ansicht mit der gleichzeitigen

[252] Von den insgesamt knapp 5500 Spielen mit dieser Kamera-Art sind etwa 4800 Spiele dem Action-Genre zugeordnet, während nur etwa 160 Spiele dem Rollenspiel-Genre zugeordnet sind [MobyGames.com, 2013m,n].
[253] 1985, *Nintendo Co., Ltd., Systems Research & Development Co., Ltd.* [MobyGames.com, 2013u]
[254] 1987, *Capcom Co., Ltd.* [MobyGames.com, 2013r]

Möglichkeit einer Tiefenwahrnehmung und stellt somit eine Mischung der Aspekte der *Third-Person-Kamera* und der *Top-Down-Kamera* dar. Erstmalig verwendet wurde diese Kameraperspektive im 1982 erschienenen Spiel *Zaxxon*[255]. Ein weiterer nennenswerter Vertreter ist die 1996 gestartete *Diablo*-Reihe[256]. Bei der isometrischen Kamerperspektive handelt es sich mathematisch betrachtet um eine Spezialform der Axonometrie [Hartmann, 2005, S. 13ff.], wobei damit der Eindruck dreidimensionaler Objekte in einen zweidimensionalen Raum hervorgerufen wird [Egenfeldt-Nielsen u. a., 2008, S. 111f.]. Die isometrische Perspektive stellt auf diese Weise eine weniger abstrakte Form der Topdown-Ansicht dar, die den Rezipienten auch erlaubt, die Geometrie und die unmittelbare Umgebung mit mehr Informationen zu erfassen [Perron u. Wolf, 2008, S. 158]. Insbesondere in Rollenspielen findet die isometrische Perspektive überall dort eine Anwendung, wo explizit keine *First-Person-Kamera* gewählt wird. Als Grund hierfür vermuten Egenfeldt-Nielsen u. a. eine Identifikationsmöglichkeit mit den dargestellten Charakteren bei gleichzeitiger strategischer Wahrnehmung der Umgebung [Egenfeldt-Nielsen u. a., 2008, S. 111f.]. Eine Distinktion zur *Third-Person-Kamera* ist nicht immer klar zu benennen und definiert sich hauptsächlich anhand des verwendeten Winkels der Kamera und die Entfernung zum Bezugspunkt. Spiele, die generell über keine Spielfigur verfügen oder Spiele, bei denen das Spielgeschehen mehrerer Avatare gleichzeitig gezeigt werden soll (beispielsweise bei modernen Sportsimulationen) fallen jedoch der isometrischen Perspektive zu. Mit nur knapp 2700 Spielen stellt die isometrische Kameraperspektive die am wenigsten häufig eingesetzte Variante dar [MobyGames.com, 2013h].

10.4. Zusammenfassung

Die hier vorgestellten Kamera-Modelle stellen Konzepte dar, die sich für bestimmte Genres oder Zwecke besonders eignen. Allerdings gibt es durchaus auch hybride Realisierungen, bei denen Ansätze von verschiedenen Kamera-Modellen kombiniert werden. Ein Beispiel hierfür ist die PC-Version des Titels *Dragon Age: Origins*[257]. Im normalen Spielverlauf ist die Spielfigur aus einer Schulterperspektive zu sehen. Dies erlaubt starke Identifikationsmöglichkeiten, da sich die Spielfigur individualisieren lässt und während des Spiels ihr Aussehen stark verändert, weiterhin auch an Gesprächen teilnimmt und der weiteren mehr. Diese Identifikationsmerkmale wären aus einer Ego-Perspektive nicht realisierbar. Wenn die Spielsituation allerdings viele darzustellende Charakter enthält, beispielsweise in einem Kampf, kann die Kamera für eine bessere Übersicht zu einer isometrischen Perspektive herausgezoomt werden. Hieraus wird ersichtlich, dass die Kamera nicht als unveränderliches Sakrosankt,

[255] 1982, *SEGA Enterprises Ltd.* [Perron u. Wolf, 2008, S. 158]
[256] 1996, *Blizzard Entertainment Inc., Blizzard North* [MobyGames.com, 2013c]
[257] 2009, *BioWare Corporation* [MobyGames.com, 2013d]

sondern als Hilfsmittel zur Präsentation dessen eingesetzt werden sollte, was tatsächlich relevant ist und den oder die Rezipienten bestmöglich bei der Ausübung der Spielmechanik unterstützt.

Im nächsten Kapitel werden Grundlagen von Netzwerken vorgestellt, die für die effiziente und Spielunterstützende Verbindung der Spielenden untereinander und mit der Spielwelt von Bedeutung sind. Die Netzwerkgrundlagen bilden somit ein fundiertes Wissen auf einer weiteren Ebene, die gut funktioniert, wenn sie nicht mehr wahrgenommen wird sondern die zu transportierenden Inhalte auf bestmögliche Art und Weise unterstützt.

„One of the most enjoyable things you can do with a computer network is play connected games with friends and family."

(Bradley Mitchell [Mitchell, 2014]*)*

KAPITEL 11

Netzwerkgrundlagen

Beim Spielen von Pen-&-Paper-Rollenspielen in der realen Welt findet sämtliche Interaktion und Kommunikation analog und unter Verwendung physikalischer Spielutensilien statt. Da es Ziel der Arbeit ist, eine Digitalisierung dieser Art von Rollenspielen durchzuführen und zur Interaktion technische Geräte (Smartphones) sowie zur Visualisierung des Spielgeschehens eine vollständig dreidimensionale und virtuelle Welt gewählt wurden, ist es notwendig, diese miteinander zu verbinden. Für diese Verbindung wird ein Netzwerk benötigt, da Daten zwischen den Komponenten gesendet und empfangen werden.

Als kurze Einführung wird eine historische Übersicht von Netzwerken geliefert und selbige in Spielumgebungen kurz dargestellt. In diesem Kapitel wird auszugsweise aus der vorangegangenen Projektarbeit zitiert und mit zusätzlichen Informationen erweitert.

11.1. Netzwerke in Computerspielen

Für die Realisierung eines Multiplayerspiels, welches nicht an einem Gerät über Methoden wie dem Split-Screen-Verfahren umgesetzt wird, ist ein Netzwerk notwendig.

Im Bereich der Spielentwicklung existieren als Basis solcher Netzwerke zwei verschiedene Modelle: Peer-2-Peer[258] sowie Client-Server.

[Ein Peer-2-Peer-Modell zeichnet dadurch aus, dass] die Topologie ein vollvermaschtes Netz darstellt (Abb. 11.1), bei dem jeder Netzwerkteilnehmer mit jedem weiteren Teilnehmer verbunden ist. Ein Peer

[258] In einem Artikel [Mark Terrano u. Paul Bettner, 2001b] erläutern die Entwickler des Spiels *Age of Empires* das Design und die Implementation des verwendeten Peer-2-Peer-Ansatzes sowie die Schwierigkeiten und Auswahlentscheidungen.

steht dabei für die Rolle eines verbundenen Teilnehmers. Ein Spiel, welches auf Basis dieses Modells implementiert werden soll, muss dahingehend abstrahiert werden, dass es lediglich aus einer bestimmten Anzahl an vorgegebenen Zuständen und Überführungen besteht, welche während des Spielverlaufs einen Initialzustand deterministisch verändern. Ein konkreter Spielzug besteht aus dem Ausführen von Aktionen (Zustandsüberführungen), welche den Status respektive Zustand des Spiels (und somit auch die Spielwelt), modifizieren.[259]

Der Client-Server-Ansatz ist das am häufigsten eingesetzte asymmetrische Netzwerkkommunikationsmodell (vgl. Seite 30). Im Gegensatz zum Peer-2-Peer-Ansatz existiert eine Komponente - der Server - welcher allen anderen übergeordnet ist. Alle Clients verbinden sich zum Server, wodurch eine Sterntopologie entsteht (Abb. 11.1).

In diesem Modell implementiert der Server meist einen Service, welcher von Clients genutzt werden kann, sobald diese zum Server verbunden sind. Häufige Aufgaben eines Services sind: den Clients Daten bereitstellen oder Daten der Clients entgegennehmen und weiterverarbeiten. Die Vorteile eines Servers lassen sich unter anderem wie folgt definieren: Eine zentrale Stelle für die Client- und Datenverarbeitung macht es einfacher Benutzer zu überwachen und diese zu administrieren; Hauptberechnungen auf einem leistungsstarken Server ausführen zu lassen ermöglicht es, weniger effiziente Clients, wie mobile Endgeräte, zu verwenden; Der Server kann nicht ausschließlich von Spielclients verwendet werden, sondern zum Beispiel auch von einem Überwachungsclient, welcher aktuelle Statistiken und Geschehnisse anzeigt.[260]

Die ersten Multiplayerspiele waren asynchroner Natur und wurden über Forensoftware gespielt[261]. Ein Server hielt dabei den Spielstatus in Form von Kommentaren fest [Lecky-Thompson, 2008, S. 5]. Dieser Methodik folgte mit *play-by-email*[262] die nächste Möglichkeit, gemeinsam Spiele über das Internet spielen zu können. Um auch leistungsfähigere Spiele gegeneinander spielen zu können, war es gezwungenermaßen notwendig, Computer direkt miteinander zu vernetzen, wodurch ein so genanntes LAN geschaffen wurde. Dadurch existierten bereits in den 1990er Jahren Multiplayerspiele auf Basis einer lokalen Vernetzung [Glenn Fiedler, 2010; China Daily, 2011]. Die Verbreitung von LANs wurde zusätzlich durch die Nicht-Existenz von schnellen Bandbreiten gefördert, da ein Spielen auf weite Distanzen zur damaligen Zeit nicht der Regel entsprach. Erst mit dem flächenmäßigen Breitbandaus-

[259] siehe Abschnitt 2.3.3
[260] siehe Abschnitt 2.3.3
[261] Mehr Informationen und eine Anleitung dazu ist in [wikiHow, 2014] zu finden.
[262] siehe z.B. [Blackmoor, 2005]

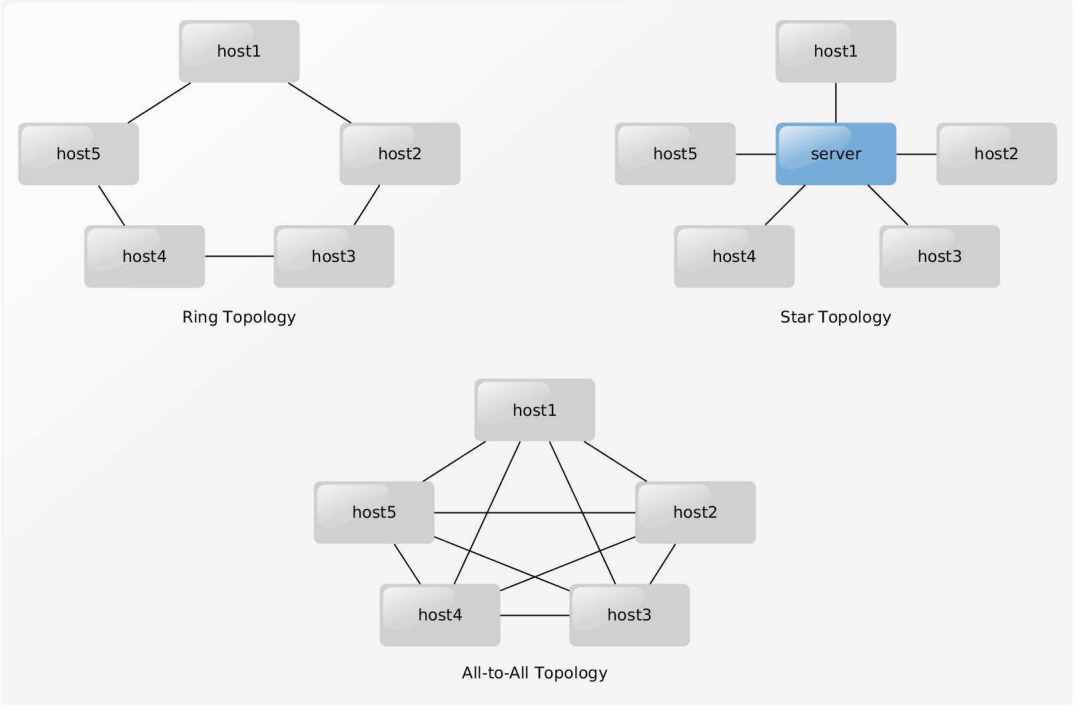


Abbildung 11.1.: Eine Übersicht verschiedener Netzwerktopologien. All-to-All Topology stellt das typische Peer-2-Peer-Modell dar. Das Client-Server-Modell wird durch die Star Topology beschrieben.

nach [Andrew Davison, 2005, Figure 29-1]

bau gab es eine Verschiebung in Richtung heutiger Onlinespiele. Besonders hervorzuheben sind dabei MMOGs[263], welche mittlerweile sehr weit verbreitet sind und selbst in der Forschung immer häufiger betrachtet werden [Bogost, 2004; Ducheneaut u. a., 2007; Fritsch u. a., 2005; Sheldon u. a., 2003; Ducheneaut u. Moore, 2004]. Jedoch fand entgegen der Vermutungen keine vollständige Substitution der LAN- respektive der klassischen Brettspiele statt, denn laut Lecky-Thompson können Onlinespiele den sozialen Charakter dieser Spiele nicht vollständig aufgreifen oder gar ersetzen [Lecky-Thompson, 2008, S. 6].

Kommunikation in Multiplayerspielen ist sehr wichtig[264] [Lecky-Thompson, 2008, S. 59]. Bei der Nutzung von LANs können Spieler direkt miteinander kommunizieren, ebenso bei klassischen Brett- und

[263] Bei MMOGs handelt es sich wie bei den MMORPGs um ein Onlinespiel, welches über das Internet gleichzeitig von mehreren (genauer: von Massen (massively)) Spielern gespielt wird, jedoch muss dieses nicht zwangsläufig ein Rollenspiel sein.
[264] siehe dazu auch Abschnitt 6.4 auf Seite 134

Kartenspielen. In Computerspielen ist mittels Sprachübermittlung zwar häufig eine ähnliche Möglichkeit gegeben, jedoch existiert diese Möglichkeit nicht immer und spiegelt den sozialen Aspekt nicht vollkommen wider. Insofern haben sich verschiedene Techniken etabliert, wie beispielsweise ein Chat mit vorgegebenen Stimmungsindikatoren oder so genannte *Emotes*[265]. Ferner gilt es, die Erwartungen der Spieler zu erfüllen. Insbesondere in virtuellen Welten existiert beim Antreffen von anderen Avataren das Bestreben, mit diesen interagieren zu können [Lecky-Thompson, 2008, S. 59]. Im Rahmen dieser Abschlussarbeit wird die Kommunikation nicht digitalisiert und Aufgrund des Anspruchs, die sozialen Aspekte zu erhalten und zu fördern[266] auf direkte Mensch-zu-Mensch-Kommunikation gesetzt[267].

11.1.1. Probleme

Hauptsächlich existieren bei Netzwerkspielen die folgenden Probleme im Bereich der Netzwerkprogrammierung [Lecky-Thompson, 2008, Kap. 7]:

Latenz (Verzögerung der Datenübertragung): Die Netzwerklatenz[268] beschreibt eine Verzögerung in der Datenübertragung. Laut Lecky-Thompson tritt eine Latenz eher in Verbindung mit TCP als UDP auf [Lecky-Thompson, 2008, Kap. 7]. Diese kann in unterschiedlichen Arten in Erscheinung treten, beispielsweise wie lange ein Client braucht um die Informationen, die vom Server kommen zu verarbeiten oder wie lange der Client braucht, um die Bewegungsbefehle an den Server zu übertragen. Außerdem existiert die Möglichkeit, dass Server-interne Latenzen auftreten, beispielsweise durch Datenbankzugriffe.[269]

Jitter (unzuverlässige Datenrate): Jitter wird im Allgemeinen als eine Genauigkeitsschwankung im Übertragungstakt bezeichnet, wohingegen in der Netzwerktechnik von einer Varianz in der Laufzeit von Datenpaketen[270] gesprochen wird [Nextragen, 2012]. Die Laufzeit steht hierbei für die Latenz, d.h. die Zeit, die ein Paket vom Sender zum Empfänger braucht. In Netzwerkanwendungen wird Jitter als irritierende Erscheinung wahrgenommen [Lecky-Thompson, 2008, S. 164ff]. Ein Lag entsteht durch eine gleichmäßige Latenz. Dadurch, dass letztere jedoch zumeist konstant ist, existiert die Möglichkeit der Vorberechnung und Korrektur durch die Game-Engine [Lecky-Thompson, 2008, S. 165]. Die Schwankungen sind bedingt durch die Netzwerktopologie.

[265] Emotes werden in Onlinespielen verwendet um Emotionen des Spielers durch den Avatar ausdrücken zu lassen. Eine Übersicht lässt sich unter anderem in [GuildWiki.de, 2012] finden.
[266] Präzisierungen hierzu sind u.a. in Abschnitt 6.4 auf Seite 134 zu finden.
[267] Zur weitergehenden Lektüre zum Thema Kommunikation in digitalen Spielen wird zudem unter anderem auf [Ducheneaut u. a., 2007; Ducheneaut u. Moore, 2004; Innocent u. Haines, 2007] verwiesen.
[268] häufig auch als „Lag“ bezeichnet
[269] Tiefer gehende Informationen hierzu lassen sich in [Lecky-Thompson, 2008, Kap. 7] finden.
[270] ungleichmäßige Latenz

Die Latenz kann dadurch einerseits sehr hoch und bemerkbar sowie andererseits sehr gering und kaum bemerkbar sein. Besonders bei TCP äußert sich dies, unter anderem durch die Zusicherung der richtigen Reihenfolge[271], sehr stark.

Datenverlust (unzuverlässige Datenübertragung): Ein Datenverlust kann ebenfalls zu Problemen in Netzwerkspielen führen. Informationen, welche ein Empfänger benötigt, bleiben aus, woraufhin z.B. wichtige Berechnungen nicht durchgeführt werden können. Die Existenz und Ursache des Datenverlustes stehen ebenfalls in Korrelation zur verwendeten Netzwerktopologie[272].

11.2. Netzwerkprotokolle

Jedes Gerät in einem Netzwerk wird durch einen so genannten *Knoten* repräsentiert. Eine Verbindung zwischen diesen wird als *Kante* bezeichnet. In welcher Art und Dichte Kanten auftreten, hängt von der verwendeten Netzwerktopologie und dem Netzwerkprotokoll ab. Ungeachtet dessen besitzt jeder Knoten eine eindeutige Adresse in diesem Netzwerk, über welche beispielsweise Daten empfangen werden können. Im Bereich der Internetanwendungen wird hierfür eine IP-Adresse verwendet.

Durch das Internet Protocol wird ein paketorientierter, verbindungsloser Ende-zu-Ende-Transport ermöglicht. Daten werden in Pakete unterteilt und einzeln übertragen. Es wird weder eine korrekte Reihenfolge, noch Nachrichtenübertragung garantiert. Ein IP-Paket besteht aus einem Header- und Datenbereich. [...] IP unterstützt keine Dienstadressierung, wodurch kein Multiplexing möglich ist.[273]

Aufgrund der Verbindungen zwischen den Knoten ergeben sich verschiedene Kommunikationsrouten, um Daten zu übertragen. Eine neu hinzugekommene Kante kann z.B. zu einem kürzeren Pfad vom Sender zum Empfänger führen, sodass eine korrekte Reihenfolge nicht gewährleistet werden kann. Verschiedene Datenpakete können den Empfänger somit über unterschiedliche Routen erreichen. Zur Kommunikation in der Netzwerkanwendung wird ein Netzwerkprotokoll benötigt.

Ein Netzwerkprotokoll ist ein vereinbartes Kommunikationsprotokoll, unter dessen Verwendung in einem Netzwerk verbundene Geräte Daten austauschen können [Siegmund, 2002]. Dafür existiert eine

[271] Die Eigenschaften von TCP werden in Abschnitt 11.3 auf Seite 204 genauer erläutert.
[272] Nicht zu vernachlässigen ist hierbei ebenfalls die Architektur der Netzwerkspiele, welche Verbindungsabbrüche sowie ein nachträgliches Initialisieren ermöglichen sollte. Für weiterführende Informationen sei hier auf [Snader, 2000, S. 70-98] verwiesen.
[273] siehe Seite 31ff

fest definierte Syntax, welche die Semantik der Kommunikation bestimmt. An einer solchen ist selten nur exakt ein Protokoll beteiligt, viel mehr handelt es sich um ein komplexes Zusammenspiel mehrerer, auf unterschiedlichen Ebenen befindlicher Protokolle. Da jedes Protokoll einen anderen Zweck erfüllt, wird - wie in der Informatik üblich - ein Schichtenmodell zur Einordnung der Protokolle verwendet. Durch die Verwendung eines solchen kann die Komplexität verringert und übersichtlicher gestaltet werden. Komplexe, low-level Funktionalitäten werden durch die darüberliegende Schicht abstrahiert und dem Programmierer in Form eines einfachen Interfaces zur Verfügung gestellt.[274]

Als Standard im Bereich der Netzwerkprotokolle gilt seit 1983 das Open Systems Interconnection Model (OSI). Dieses basiert auf sieben aufeinanderfolgenden Schichten, welche jeweils einen fest definierten Aufgabenbereich sowie Schnittstellen besitzen (siehe Abb. 11.2).

Das OSI-Modell ist insbesondere für Internetapplikationen zu generisch und umfangreich gestaltet, sodass hierfür ein weiteres, standardisiertes Vierschichtenmodell (TCP/IP-Referenzmodell) existiert, welches lediglich das IP berücksichtigt [Plüss, 2004; Harold, 2004]. Dieses ist in Abb. 11.2 in der Spalte *TCP/IP Schicht* dargestellt. Die Netzwerkschicht (auch *Netzzugang* genannt) spezifiziert die Übertragung der zu sendenden Daten. Im Bereich der Internetapplikationen finden hier häufig sowohl Ethernet als auch WLAN eine Anwendung. Die darüberliegende Internetschicht ist für die Übertragung der Pakete innerhalb des Netzwerks an den nächsten Knoten zuständig. Somit fällt auch das Routing in den Aufgabenbereich dieser Schicht. Da es sich hierbei um ein Modell für Internetanwendungen handelt, entspricht das zugrundeliegende Protokoll dem Internet Protocol.

Die Host-to-Host-Schicht wird ebenfalls als Transportebene bzw. -schicht benannt. Diese unterteilt die zu sendenden Daten in Pakete und übergibt diese der Internetschicht.

Auf der Transportebene existieren zwei wichtige Protokolle, welche auf dem IP-Protokoll aufbauen und dieses erweitern: TCP und UDP. Da diese Ports anbieten, was eine Form der Dienstadressierung darstellt [Kipp, 2002], ist es möglich, mehrere Verbindungen gleichzeitig aufrechtzuerhalten. Ein Port ist keine physikalische Einschränkung, sondern auf dem entsprechenden Gerät über Konfigurationsdateien implementiert. [275]

Bei der Verwendung von TCP muss vor Beginn der Datenübertragung zunächst eine Verbindung initialisiert werden, weshalb hierbei auch von einem verbindungsorientierten Protokoll gesprochen wird. Während dieser Phase wird der Pfad von Sender zu Empfänger im Netzwerk festgelegt, sodass u.a.

[274] siehe Seite 30
[275] siehe Seite 32

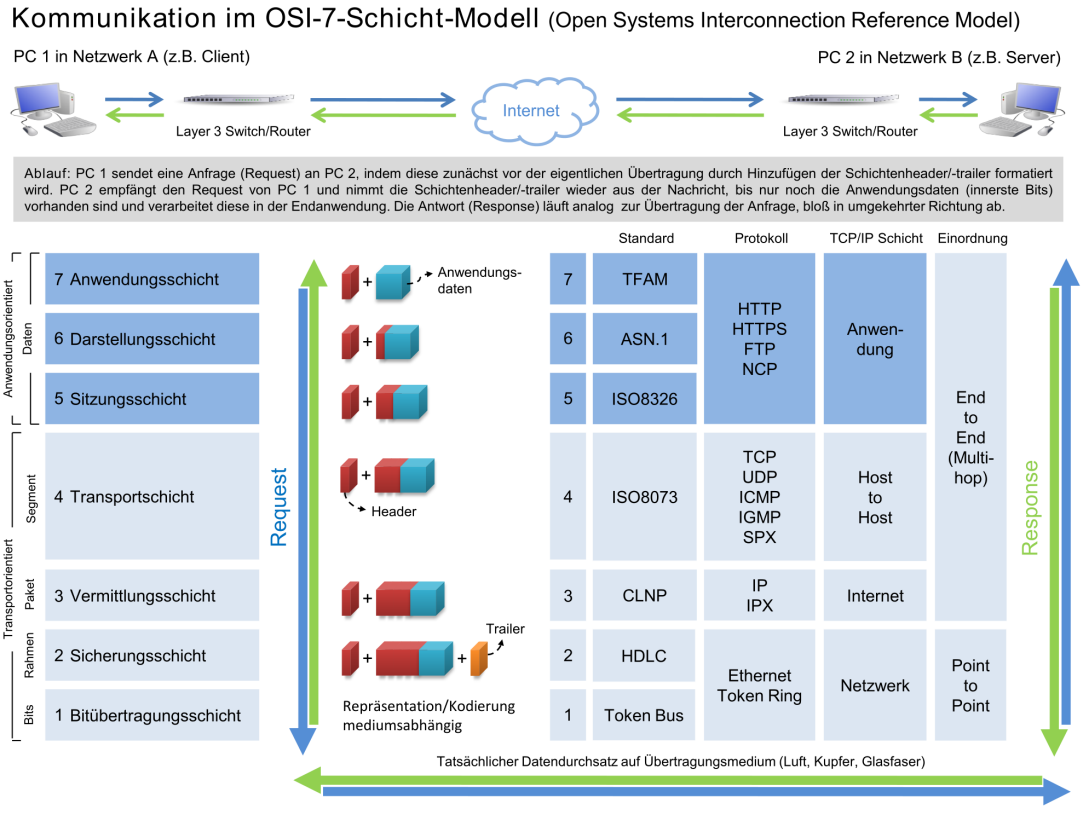


Abbildung 11.2.: In der Abbildung ist sowohl das OSI-Modell als auch ein Sendungs- sowie Empfangsverlauf schematisch dargestellt. Auf der Anwendungsschicht befindet sich die Netzwerkanwendung, welche Daten des Senders (PC 1 oben in der Abbildung) an den Empfänger entsendet. Jede Schicht fügt dem Datensatz einen spezifischen Header hinzu und übergibt diesen an die darunterliegende Schicht. Auf der letzten Ebene findet letztlich die physikalische Datenübertragung statt. Auf der Empfängerseite vollzieht sich das geschilderte Vorgehen analog, jedoch in umgekehrter Reihenfolge. Auf jeder Schicht existieren verschiedene Protokolle, welche nach Fertigstellung der Anwendung prinzipiell beliebig ausgetauscht werden können.

aus [Stoll, 2013]

eine korrekte Reihenfolge gewährleistet werden kann. Im Gegensatz dazu existieren verbindungslose Protokolle (z.B. UDP), bei welchen weder ein vorgegebener Pfad noch eine Verbindung existiert.

Verbindungslose Protokolle ähneln laut Snader der Art des Schreibens, Sendens und Empfangens von Briefen [Snader, 2000, S. 16ff]. Jeder Brief wird einzeln adressiert und abgeschickt. Die Poststelle

behandelt jeden Brief unabhängig und sichert keine Historie zwischen diesen. Ebenfalls wird weder eine Liefergarantie[276] noch eine zeitliche Einhaltung aufeinanderfolgender Briefe gewährleistet. Verbindungsorientierte Protokolle hingegen ähneln Telefongesprächen. Nach dem Wählen einer Rufnummer wird der Anruf aufgebaut, woraufhin das Gespräch folgt. Einzelne Aussagen müssen nicht mit Informationen bezüglich des Empfängers erweitert werden, da dieser explizit feststeht und bis zum Zeitpunkt des Beendens der Verbindung nicht gewechselt wird.

Der größte Nachteil der verbindungslosen Protokolle ist, dass ein gesendetes Paket vom Empfänger nicht quittiert wird, wodurch keine Übertragungszuverlässigkeit zugesichert werden kann [Snader, 2000, S. 15ff]. Aufgrund dieser und weiteren in Abschn. 3.1.5 beschriebenen Gründen, wurde bereits eine Entscheidung zugunsten von TCP getroffen, sodass an dieser Stelle auf eine weitere Betrachtung von UDP verzichtet wird.

11.3. Transmission Control Protocol (TCP)

Seit den 1990er Jahren ist TCP/IP bei der Konzeptuierung von Netzwerken und Implementierung von Netzwerkanwendungen die führende Technologie [Snader, 2000, S. 1][Erik Rodriguez, 2012][277]. Es handelt sich um ein Protokoll auf der Transportschicht. Obwohl es auf dem verbindungslosen Internet Protocol aufbaut, ermöglicht es eine verbindungsorientierte, zuverlässige und stream-basierte Kommunikation. Hierfür fügt TCP laut Snader folgende Dinge hinzu:

- Die Checksumme eines Pakets, wodurch auf der Empfängerseite getestet werden kann, ob der Inhalt bzw. das Paket beschädigt bzw. vollständig ist;
- Sequenznummern, welche eine zeitliche Einordnung der Pakete zulassen, damit ein verlorengegangenes bzw. zeitlich falsches Paket erkannt werden kann und
- einen *Quittungs-* respektive *Neusendungsmechanismus* (*acknowledgement/retry mechanism*).

Ein Paket kann durch die Sequenznummern jeweils in den gesamten Kommunikationskontext gesetzt sowie mit Rückbezügen versehen werden, da eine Art Zustand zwischen den gesendeten Paketen gesichert wird, wodurch eine Zuverlässigkeit gewährleistet werden kann [Snader, 2000, S. 15ff.]. Die Anwendung kann somit nachvollziehen, wann welche Pakete wohin gesendet und ob diese vom Empfänger bestätigt worden sind. Dadurch ist es möglich, nicht erhaltene respektive quittierte Daten erneut zu versenden.

[276] hier existieren mittlerweile Sonderkonditionen
[277] Dies wird in [Harold, 2004] als „lingua franca“ der Netzwerkapplikationen bezeichnet.

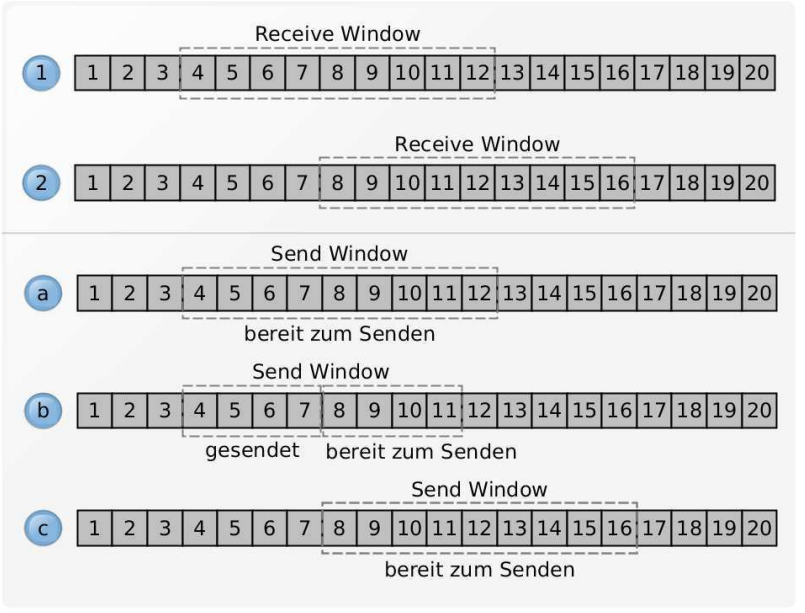


Abbildung 11.3.: Die Abbildung zeigt das Vorgehen des *Send-* sowie *Receive Window*. Ersteres bestimmt die empfangbaren Daten in der Form, dass sowohl das erwartete als auch das maximal empfangbare Paket angegeben werden. Nach erfolgreichem Eintreffen der benötigten Daten, wird das Fenster automatisch weitergeschoben. Das *Send Window* sendet Daten und startet parallel dazu einen so genannten RTO, welcher die Quittierung der Gegenseite kontrolliert und gegebenenfalls eine Neusendung vornimmt. Bei Erfolg (eingehender Quittung) wird das gesamte Fenster ebenfalls weitergeschoben.

nach [Snader, 2000, S. 19ff]

Zur Verwaltung wird hierfür das so genannte *Receive Window* bzw. *Send Window* verwendet (siehe Abb. 11.3) [Snader, 2000, S. 18ff][Lienemann u. Larisch, 2010, Abschnitt 2.4.1]. Ersteres übernimmt die Datenannahme. Dies wird mit Hilfe der Sequenznummer des nächsten erwarteten sowie des maximal empfangbaren Pakets realisiert (1)[278]. Sofern das eingetroffene Paket nicht dem Erwartungswert entspricht, so wird dieses zurückgehalten und nicht an die Anwendung weitergegeben, bis das erwartete Paket eingetroffen ist. Somit wird die Reihenfolge der Pakete hardwareseitig sichergestellt. Sobald die Erwartungen erfüllt wurden, werden die zwischengespeicherten Pakete an die Anwendung weitergegeben sowie das Fenster weitergeschoben (2). Parallel dazu wird eine Quittung (Acknowledgement) an den Sender gesendet, welcher den Empfang quittiert und die nächste erwartete Sequenznummer enthält.

[278] Es steht lediglich ein begrenzter Zwischenspeicher (Buffer) zum Empfang von Daten zur Verfügung.

Das *Send Window* (a) arbeitet auf ähnliche Weise, mit der Ausnahme, dass das hier noch eine weitere Unterteilung in *gesendet, jedoch nicht bestätigt* und *bereit zum Senden* stattfindet (b) [Snader, 2000, S. 19ff]. Sofern Pakete gesendet, jedoch nicht bestätigt wurden, bedeutet dies nicht zwangsläufig das Aussetzen des weiteren Sendevorgangs, allerdings wird parallel dazu ein so genannter Retransmission Timeout gesetzt. Dieser Timer übernimmt, unabhängig von den weiteren zu sendenden Daten, die Überprüfung, ob in einer bestimmten Zeit die bereits gesendeten Daten bestätigt wurden. Sofern dies nicht der Fall ist, werden die vorherigen Daten noch einmal gesendet. Sollten die beim ersten Mal gesendeten Daten dennoch den Empfänger erreichen, so wird das Receive Window unverzüglich weitergeschoben, was beim Eintreffen der erneut und demnach doppelt gesendeten Daten dazu führt, dass diese verworfen werden. Nach erfolgreicher Bestätigung wird das gesamte Fenster auch hier weitergeschoben (c). Um z.B. auf Netzwerkschwankungen reagieren zu können, wird das Zeitlimit des Timers während der TCP-Verbindung adaptiv neu gesetzt. Hierfür wird die Paketumlaufzeit in geglätteter Form verwendet [Microsoft, 2006].

Ferner implementiert TCP standardmäßig einige Methoden, welche die Performanz des Netzwerksprotokolls verbessern. Im Folgenden werden einige der von Hager beschriebenen Methoden und deren Funktionalität näher erläutert [Hager, 2005]. Die Auflistung wird zudem um einige andere TCP-Eigenschaften erweitert.

Slow Start: Die Senderate wird in Abhängigkeit zur Auslastung des Netzwerks dynamisch bestimmt. Der Sender entsendet dabei zunächst ein Paket und wartet auf eine eintreffende Bestätigung. Die Paketanzahl wird daraufhin so lange erhöht, wie keine Pakete verlorengehen und die Bestätigung zeitgemäß eintrifft.

Delayed Ack (verzögerte Quittung): Die Anzahl der entsendeten Pakete kann die Netzwerkverbindung erheblich beeinträchtigen. Aus diesem Grund existiert die Möglichkeit, nicht jeden Paketerhalt zu quittieren, sondern dies eine gewisse Zeit zu verzögern (maximal 0.5 *s*) [Braden, 1989b]. Hiermit wird versucht, ein Charakteristikum von verbindungsorientierten und streambasierten Protokollen ausnutzen, dass regelmäßig und nahezu kontinuierlich Daten gesendet werden. In der Wartezeit könnte zum einen der Fall eintreten, dass ohnehin ein Paket gesendet wird, woran die Quittung angehängt werden kann (so genannte Huckepack-Quittung[279]) oder zum anderen das Eintreffen mehrerer neuer Pakete, welche im Anschluss zusammen bestätigt werden können. Die Wartezeit kann jedoch die zuvor beschriebene Methodik des Slow-Starts beeinträchtigen, sodass nach maximal zwei erhaltenen Paketen unverzüglich eine Quittung gesendet werden muss.

[279] Im Englischen als *piggy-back acknowledgement* bekannt [Tanenbaum u. Wetherall, 2010, S. 211ff].

SACK – selective acknowledgement: Standardmäßig quittiert TCP jeweils die höchste empfangene Sequenznummer, da die Daten in der richtigen Reihenfolge und vollständig vorliegen. Neusendungen umfassen alle Segmente ab dem fehlenden Segment, selbst wenn diese fehlerfrei übertragen worden. Die Möglichkeit der selektiven Quittung ändert dieses Verhalten dem Namen entsprechend ab, indem exakt spezifiziert werden kann, welche Segmentblöcke empfangen worden sind und welche nicht. Somit müssen lediglich die fehlenden Segmente erneut gesendet werden.

TCP-Fenstergröße: Das zuvor beschriebene *Receiving Window* besitzt eine feste Größe und bestimmt wie viele Daten der Sender auf einmal senden kann. Dadurch, dass die heutigen Betriebssysteme diese nicht mehr fest setzen, sondern während des Verbindungsaufbaus aushandeln, existieren für gewöhnliche Anwendungen keine Probleme mehr [Microsoft, 2006]. Standardmäßig wird die Größe auf 16 *KB* respektive 12 Segmente mit jeweils 1.460 *Byte* gesetzt. Unter Verwendung weiterer TCP-Eigenschaften vergrößert sich der Header, wodurch der Datenteil geringer ausfallen kann.

Fast Retransmission (schnelle Neuübertragung): Bei dieser Eigenschaft wird der zuvor erläuterte Timer umgangen. Erhält der Empfänger Daten mit Sequenznummern, welche höher sind als die erwarteten, so kann umgehend darauf geschlossen werden, dass Daten verlorengegangen sind. In dem Fall sendet der Empfänger unverzüglich eine Quittung, jedoch nicht für die gerade erhaltenen Pakete sondern für das erwartete. Somit besteht die Möglichkeit, dass der Sender ein Acknowledgement für ein älteres Segment erhält, was auch diesen darauf schließen lässt, dass ein Segment verlorengegangen ist. Dies führt zum unverzüglichen Neusenden des Segments, unabhängig davon, ob der RTO bereits abgelaufen ist. [Microsoft, 2006]

Nagle Algorithmus: Die Anzahl vieler kleiner im Netz befindlicher Pakete kann den Durchsatz der Verbindung verringern. Der hier beschriebene Algorithmus verhindert das Senden kleiner Pakete, indem diese zwischengespeichert und zu einem großen zusammengefasst werden. Die angestrebte Größe entspricht dabei der MTU. Diese muss jedoch nicht zwangsläufig erreicht werden. Befinden sich beispielsweise keine unbestätigten Pakete mehr im Netzwerk, so wird das bis dato akkumulierte Paket sofort ins Netz entsendet. Weitere Details hierzu sind unter anderem in [Nagle, 1984; Kyle Kingsbury, 2012] zu finden.

Insgesamt münden diese Optimierungen in folgendem Sachverhalt:

Es ist problemlos möglich, mehrere (100+) TCP-Verbindungen über einen DSL-Link zu betreiben, wobei jede produktiv arbeitet. Dies liegt daran, dass eine TCP-Verbindung u.a. die zur Verfügung stehende Bandbreite sowie die Netzwerkkapazitäten berücksichtigt und nicht willkürlich Daten in das

Netzwerk entsendet[280]. Beim Einsatz eines kabellosen Netzwerkes kann sich dies allerdings auch negativ auswirken. Die Wahrscheinlichkeit, dass in einem solchen Netzwerk ein Paket verlorengeht, ist viel höher und führt demnach häufiger zu einer merklichen Empfangsverzögerung, welche in Spielen auch als Latenz[281] bekannt ist. Bei vergleichsweise vielen UDP-Verbindungen käme es zu einem so genannten „verstopfen" [Jacobson, 1988] der Leitung, da hier eine solche Überprüfung nicht existiert.[282]

Eine sinnvolle Nutzung von TCP hängt demnach von dem zugrundeliegenden Vorhaben ab. Wie die Verwendung im Bereich der Spielentwicklung beurteilt wird und mit welchen möglicherweise auftretenden Problemen dies einhergeht, wird im nächsten Abschnitt genauer behandelt.

11.3.1. TCP in Netzwerkspielen

Die Wahl des Netzwerkprotokolls bei der Entwicklung von Netzwerkspielen ist eine fundamental wichtige Designentscheidung. Ursprünglich wurde TCP für unidirektionale und im Netzwerk limitierte Massendatentransfers entwickelt [Chen u. a., 2006]. Die meisten Onlinespiele verwenden hingegen sehr kleine Pakete und einen bidirektionalen Datentransfer. Letzteres stellt mit Hilfe heutiger Programmiersprachen kein Problem mehr dar. Die mit der Verwendung von TCP einhergehenden Funktionen können den Spielfluss gefährden, da beispielsweise nicht jedes Paket zuverlässig und in der richtigen Reihenfolge übertragen werden muss [Chen u. a., 2006].

Die Tendenz der weitaus größten Spieleentwicklercommunity [Betable Tyler, 2012] geht dahin, dass TCP für die meisten Spiele vollkommen zufriedenstellend ist [Watte u. a., 2005]. Laut Harold hat TCP/IP mittlerweile nahezu alle anderen universellen Netzwerkprotokolle effektiv ersetzt. Allerdings gibt es auch Studien [Chen u. a., 2006], welche zu dem Ergebnis gelangen, dass TCP speziell für Massive Multiplayer Online Role Playing Games (MMORPGs) ungeeignet ist. Die Zuverlässigkeit in Netzwerken führt zwangsläufig immer zu einer Erhöhung der Latenz [Andrew Davison, 2005, S. 773].[283]

Dennoch verwenden eine Reihe erfolgreicher Online-Rollenspiele ausschließlich oder zumindest als Fallbacklösung TCP[284]. Der Grund dafür liegt in der Portverwaltung vieler Firewalls, welche standardmäßig die durch TCP genutzten Ports freigeben, da über diese sämtliche Kommunikation im Internet

[280] Für weitere Informationen wird u.a. auf [Kipp, 2002] verwiesen.
[281] Latenz ist die Zeit die gebraucht wird bis die gesendeten Daten vom Adressaten empfangen werden. Das Hauptziel bei Mehrspieler-Onlinespielen ist nahezu immer die Latenz zu minimieren.
[282] siehe Abschn. 2.3.2.1
[283] siehe Seite 33ff
[284] siehe [Chen u. a., 2006, Tabelle 1]

stattfindet [Petlund u. a., 2008; Brosh u. a., 2010]. Die Nutzung anderer Protokolle respektive Ports führt häufig dazu, dass Nutzer diese selber freischalten müssten.

Die Antwort definitiver Vorzüge eines Verfahrens gegenüber dem anderen in allen erdenklichen Situationen kann nicht gegeben werden. Häufig findet eine Unterscheidung in Genres statt, anhand welcher eine Zuordnung vorgenommen wird [GameDev.net, 2004]:

Rundenbasierte Strategiespiele: TCP.

Echtzeitstrategiespiele: TCP (für extreme Szenarien UDP empfohlen).

Rollenspiele: hängt vom Grad der Story und Action ab. Für Story-lastige Rollenspiele wird TCP, für Action-lastige hingegen wird UDP vorgeschlagen.

Actionspiele: UDP.

Ferner existieren für diese Genres unterschiedliche Ansprüche an die Latenz, bevor ein unbeeinträchtigtes Spielen nicht mehr möglich ist respektive der Spielfluss unterbrochen wird. Petlund u. a. geben als maximale Latenz für RPGs 500 *ms* sowie für Echtzeitstrategiespiele 1000 *ms* an [Petlund u. a., 2008][285]. Mark Terrano u. Paul Bettner haben durch eine Variation der Latenz getestet, ab welcher Zeiverzögerung sich Benutzer gestört fühlen [Mark Terrano u. Paul Bettner, 2001b]. Für Echtzeitstrategiespiele wurde herausgefunden, dass eine Latenz von 250 *ms* nicht wahrgenommen wird. Erst ab 500 *ms* hat das Spielverhalten darunter gelitten. Das besondere ist, dass eine kontinuierliche Verzögerung von 500 *ms* als nicht so störend empfunden worden ist, wie eine zwischen mehreren Werten variierende Latenz. Dies wird dadurch begründet, dass Spieler sowie Spielerinnen während des Spielverlaufs eine individuelle Spielgeschwindigkeit entwickeln, worauf mentale Erwartungen an das Spiel aufgebaut werden. Für First-Person-Shooter gelten mit maximal 150 *ms* weitaus geringere Latenzen als Zielsetzung [Armitage, 2001]. Beigbeder u. a. beschreiben, dass Spieler bereits ab 75 *ms* eine gewisse Trägheit des Spiels bemerkten und eine Latenz ab 100 *ms* bereits zu vermindertem Spielspaß führt [Beigbeder u. a., 2004]. Mittlerweile gelten mit einer Latenz von maximal 40 *ms* jedoch weitaus höhere Anforderungen an dieses Genre [Westermark, 2007, S. 35]. Ungleich beständiger gegen hohe Latenzen sind hingegen Rundenbasierte Strategiespiele mit Werten zwischen 1000 und 4000 *ms* [Westermark, 2007, S. 34]. Eine empirische Analyse aus zwölf Spielen aus den oben aufgelisteten Genres führte zu folgenden Ergebnissen [Dick u. a., 2005]: Die in allen Genres für ein unbeeinträchtigtes Spielen akzeptierte Latenz betrug durchschnittlich 80,7 *ms*. Werte ab 150 *ms* (durchschn. 118 *ms*) werden als störend für das Gameplay empfunden.

[285] Auch in [Sheldon u. a., 2003] wurden ähnlich große Werte prognostiziert. Jedoch liegt dies viel mehr an den clientseitigen Reduktionsmethoden, wie eine *Client-Side-Prediction* bspw.

Es existieren verschiedene Ansätze, um die Latenz und den Effekt derselben zu reduzieren. *Dead reckoning*[286] verhindert z.B. unter anderem das so genannte *Warping*[287] von Objekten in Spielen und ist das am häufigste eingesetzte und typische Verfahren, um Latenzproblemen entgegenzuwirken [Wang u. a., 2009; Alexander, 2003]. Im Bereich der Reduktionsmethoden sowie Glättung fanden eine Vielzahl von Forschungen [Lecky-Thompson, 2008, Kap. 7][288] statt, auf welche, aufgrund fehlender Relevanz in dieser Abschlussarbeit, lediglich verwiesen werden kann. Neben den Latenzproblemen in Netzwerkspielen bestehen weitere Herausforderungen, wie z.B. die der Weltkonsistenz [Wang u. a., 2009, Abschnitt 2.2][289]. Bei der Verwendung von TCP ergeben sich jedoch eine Reihe von Möglichkeiten, um den Datendurchsatz und die Effizienz zu erhöhen, sodass sich der Einsatz auch für den Spielbereich eignet. Unter anderem machbar sind[290]:

- Ausschaltung des Nagle-Algorithmus,
- Verwendung großer Zwischenspeicher und
- nicht-blockierender Sockets sowie
- die Ausschaltung der Selective Acknowledgements.

Besondere Wichtigkeit ist bei der Implementierung eines Netzwerkspiels der Antwortzeit des Empfängers zuzuschreiben. Diese wird zusätzlich durch die Pausenzeiten zwischen Paketen beeinflusst, sodass hier eine optimale Balance gefunden werden muss. Laut Wang u. a. liegt die empfohlene Pausenzeit bei der Verwendung von TCP und einem WLAN bei 11 *ms*, um die bestmöglichen Antwortzeiten zu erhalten [Wang u. a., 2009]. Ferner ist es wichtig, dass die Paketgrößen nicht zu stark variieren und in einem bestimmten Intervall liegen. Hierfür empfehlen Wang u. a. Größen zwischen 25 und 95 *Byte* [Wang u. a., 2009].

11.4. Zusammenfassung

Im aktuellen Kapitel wurden essentielle Begriffe der Netzwerktechnik erklärt. Hierbei wurde sowohl herausgearbeitet, welche spezifischen Vor- und Nachteile die Protokolle Transmission Control Protocol und User Datagram Protocol aufweisen und das Zustandekommen störender Phänomene wie *Jitter* und *Latenz* wurden erklärt. Ferner sind Maßnahmen erläutert worden, wie diese Phänomene

[286] Für nähere Informationen wird u.a. auf [Wang u. a., 2009; Alexander, 2003] verwiesen.
[287] Gehen Pakete verloren, kann es sein, dass im nächsten Frame ein Objekt von einer Position auf eine andere springt.
[288] vgl. [Batista u. a., 2001; Aronson, 1997; Aggarwal u. a., 2004; Epic Games, Inc., 2013i; Yahn W. Bernier, 2001; Glenn Fiedler, 2006]
[289] vgl. [Diot u. Gautier, 1999; Vogel u. Mauve, 2001; Zander u. a., 2005; Lin u. a., 2002]
[290] Weitere ausführliche und hardwarenahe Erläuterungen hierzu lassen sich in [Brosh u. a., 2010] finden.

bestmöglich unterdrückt oder so verarbeitet werden können, dass sie nicht mehr als störend empfunden werden.

An dieser Stelle folgt als Übergang eine Übersicht und ein Vergleich verwandter Arbeiten, die in Idee, Konzept oder Ausführung mit dem im Rahmen dieser Arbeit angestrebten System vergleichbar sind.

Teil II.

Verwandte Arbeiten und Konzept

„Everything is related to everything else, but near things are more related than distant things."

(Tobler [Tobler, 1970, S. 236])

KAPITEL 12

Verwandte Arbeiten

Im vorliegenden Kapitel werden Arbeiten und Konzepte aus dem Bereich der Mixed Reality vorgestellt und gegenüber den Zielen dieser Arbeit abgegrenzt. Ebenso werden die Funktionsweisen und Aspekte der *Pervasive Games* und *Tangible User Interfaces* erläutert.

12.1. Mixed Reality und Interaktionen

Mixed Reality-Anwendungen, Kollaborationen und Kooperationen leben in einem engen Verhältnis zueinander. Es ist auffallend, dass Mixed Reality-Anwendungen häufig einen Kontext haben, der mehrere Menschen involviert[291]. Dies kann unter anderem mit der Entwicklung und Verbreitung des Internets zusammenhängen, in deren Folge sich auch die Partizipationsmöglichkeiten rapide veränderten [Sick, 2012, S. 104]. Eines der ältesten Beispiele für kollaborative Interaktion ist das Kunstprojekt mit dem Namen „The worlds first Collaborative sentence" aus dem Jahre 1994. Hierbei handelt es sich um einen kollaborativ über das Internet geschriebenen Satz, der auch als Kunstwerk viel Beachtung fand und zur Diskussion über das Medium Internet als Kunst- und Interaktionsmedium anregte und weiterhin anregt [Whitney Museum of American Art, 2013; Arns, 2007].

12.1.1. Virtuelle Kollaborationsplattform

Sharma u. a. und Bartlett u. Simpson zeigen mit ihren Arbeiten zwei ähnliche Ansätze, eine virtuelle Kollaborationsplattform zu erschaffen. Dabei wird, unter anderem durch Zuhilfenahme von 3D-Engines, ein virtueller Raum bereitgestellt, in dem die Teilnehmenden miteinander in Interaktion treten können. Kollaborationen respektive kollaboratives Arbeiten soll hierbei durch die Bereitstellung

[291] Ein weiteres Beispiel für die Vielfältigkeit in diesem Bereich ist im Anhang unter C.2 auf Seite xxxi zu finden.

von technischen Hilfsmitteln gefördert werden, wie beispielsweise durch die Möglichkeit der gemeinsamen Dokumentenansicht oder die Kommunikation über Voice over IP (VoIP) (vgl. [Bartlett u. Simpson, 2005, S. 3] und [Sharma u. a., 2011, S. 8]). Die realisierten Arbeiten unterscheiden sich in vielerlei Hinsicht von dem in dieser Arbeit umzusetzenden System, da hier explizit als Ziel genannt wird, die Notwendigkeit realer Präsenz für eine Kollaboration zu vermeiden. Gleichzeitig findet eine Durchmischung mit der realen Welt nicht statt, sodass hier nicht von Mixed Reality-Anwendungen gesprochen werden kann.

Abbildung 12.1.: Die Abbildung zeigt die Anwendung *iGameFloor* bei der Benutzung in einer Schule.

aus [Grønbæk u. a., 2007, S. 64]

12.1.2. iGameFloor

Grønbæk u. a. haben eine Spiele-Anwendung namens *iGameFloor* entwickelt, bei der Teilnehmerinnen und Teilnehmer gemeinsam auf einem interaktiven Fußboden kollaborieren (vgl. Abbildung 12.1) [Grønbæk u. a., 2007]. Die interaktive Oberfläche ist hierbei physikalisch durch ein 3m tiefes Loch im Boden realisiert worden und wird per Rückprojektion bestrahlt [Grønbæk u. a., 2007, S. 66]. Neben dem bekannten Computerspiel *Pong* wurden weitere Anwendungen entwickelt, die ebenfalls ein soziales Spielerlebnis fördern sollen [Grønbæk u. a., 2007, S. 67]. Im Gegensatz zu dem im Rahmen dieser Arbeit anvisierten Projektziel finden erweiterte Konzepte der Mixed Reality hier keine Anwendung. Ebenso liegt der Fokus auf dem parallelen Tracking mehrerer Körperteile von verschiedenen Personen, um auf diese Weise eine Art Bewegungsteuerung zu ermöglichen [Grønbæk u. a., 2007, S. 70].

12.1.3. QuakeRunner

Bei dem Projekt *QuakeRunner* [Faust, 2003] wurde das Spiel *Quake 3 Arena* [id Software, 2014] um eine an die Realität angelehnte Steuerung erweitert. Reale Muster, wie die des Laufens, Springens oder Duckens werden durch einen magnetischen Tracker erkannt und in das Spiel übertragen, sodass sich ein Spieler besser mit dem virtuellen Avatar identifizieren kann, physische Auswirkungen am eigenen Körper spürt und damit im Spiel beispielsweise nicht unentwegt *rennen* kann. Das Ziel dieses Projektes war, eine natürliche Benutzeroberfläche für Spiele zu schaffen, indem der reale Raum hinzugezogen und nahtlos in die virtuelle Welt integriert wurde [Faust, 2003].

Das geplante Ziel stimmt mit dem in dieser Arbeit umzusetzenden System zwar in Teilen überein, jedoch handelt es sich um eine andere Herangehensweise. Zum einen findet die Erweiterung lediglich in eine Richtung statt, da von der virtuellen Welt kein Feedback an den Spieler zurückgelangt und zum anderen zielt das System viel mehr auf eine Bewegungsübertragung unter der Verwendung von Gesten und anderen Metaphern. Es findet zwar eine Übertragung der Interaktionen statt, jedoch ist eine vorherige Kalibrierung und Erläuterung der Gesten notwendig, neben dem Faktum, dass kein gemeinsames Spielen ermöglicht wird.

12.1.4. CREWSpace

In der Arbeit von Lutz u. a. wurde ein Konzept entwickelt, mit dem mehrere Nutzerinnen und Nutzer gleichzeitig unter Zuhilfenahme eines Android-Smartphones mit einem Modell interagieren können, welches auf einem gemeinsamen Display angezeigt wird [Lutz u. a., 2012] (vgl. Abbildung 12.2).

Die entwickelte Anwendung *CREWSpace* ermöglicht das Zusammenarbeiten mehrerer Personen an einem zentralen Class Responsibility Collaboration (CRC)-Modell und versucht das Konzept der CRC-Modellierung teilweise zu digitalisieren, um kollaborative, computergestützte CRC-Sitzungen durchführen zu können. Jeder Teilnehmer verfügt über ein Smartphone auf Android-Basis, auf welchem eine Manipulation des Modells vorgenommen werden kann, wobei das Ergebnis wiederum auf einem gemeinsamen Bildschirm angezeigt wird. Einzelne Karten können Nutzern in ihren privaten Arbeitsbereich auf das Smartphone übertragen und dort editiert werden. Die Anwendung übernimmt unter anderem die Zustandssicherung und Regeleinhaltung. [Lutz u. a., 2012]

Beim Requirements Engineering[292] sind sowohl eine CRC-Modellierung als auch eine Anwendungsfallanalyse etablierte Methoden und werden häufig als Gruppenaktivität durchgeführt [Lutz u. a., 2012].

[292] Mit einem Requirements Engineering werden u.a. die Anforderungen an eine Software ermittelt unter Verwendung natürlichsprachiger Spezifikationen. Für einen weiterführenden Einblick wird beispielsweise [Pohl, 2008] empfohlen.

Die CRC-Modellierung basiert auf der Idee, Klassen in Form von Kärtchen darzustellen, welche ebenfalls den Zuständigkeitsbereich und Abhängigkeiten zu anderen Komponenten beinhalten. Diese Kärtchen können herumgereicht, erweitert oder durch Weglegen aus dem System entfernt werden. Insbesondere für die Anwendungsfälle werden verschiedene Akteure verwendet, um das zu entwickelnde System für die verschiedenen Rollen auszulegen und jeder Partei verständlich darzustellen. Bei der Verwendung der CRC-Methode wird eine Art Rollenspiel durchgeführt, in welchem jeder Teilnehmer eine Karte zugewiesen bekommt. Der Reihe nach werden diese analysiert, erweitert oder weitergereicht.

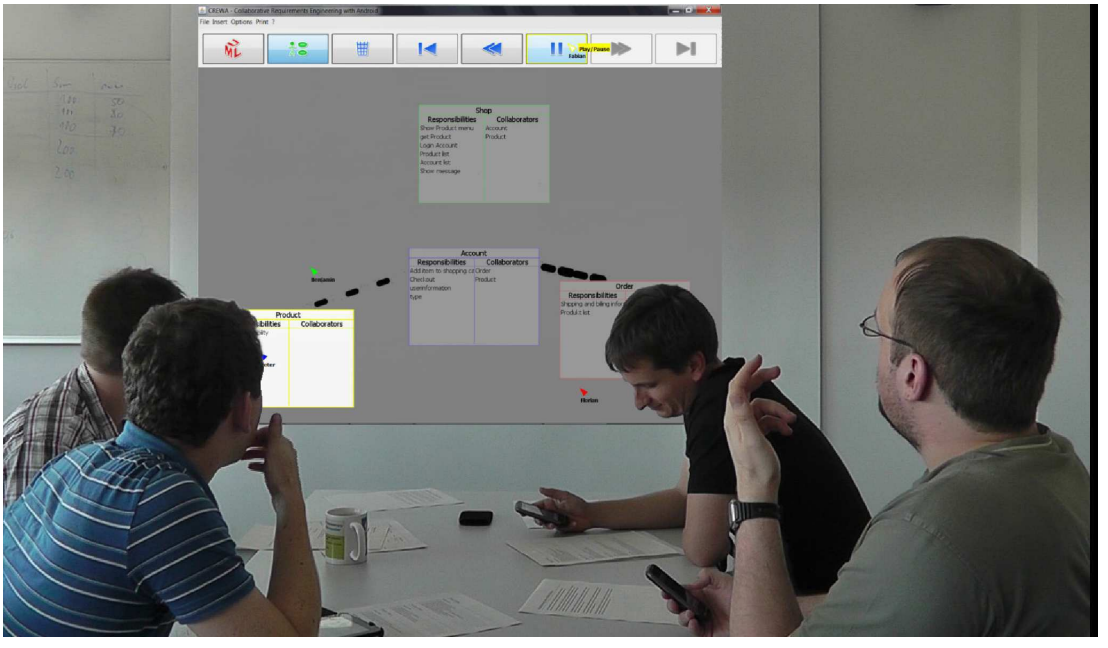


Abbildung 12.2.: Die Abbildung zeigt die gemeinsame Interaktion auf einer Projektionsflächefläche im Kontext der von Lutz u. a. entwickelten Anwendung.

aus [Lutz u. a., 2012, S. 299]

12.1.4.1. Abgrenzung zur Arbeit

Das hier vorgestellte Projekt deckt ein anderes Themenfeld ab[293], als das von den Autoren angestrebte Projektziel: Während im Rahmen dieser Arbeit ein Spiel umgesetzt wird, welches viele soziale Aspekte vereinigt, handelt es sich bei dem Projekt von Lutz u. a. um eine Anwendung ohne Anspruch eines

[293] Es fällt eher in den Bereich der CSCW-Anwendungen.

Unterhaltungswerts. Der Fokus liegt demnach auf der Vereinfachung analoger, traditioneller Arbeitsabläufe und der Ablauf wird nicht in Echtzeit, sondern runden-basiert abgewickelt.

Dementsprechend ist auch keine echte Mixed Reality vorhanden, da diese dem Projekt nicht zweckdienlich wäre. Dahingegen ist ein privater Arbeitsbereich für die Teilnehmenden vorhanden und erlaubt demnach eine Separation.

12.2. Pervasive Games

Eine weitere Forschungs- und Anwendungsrichtung nennt sich „Pervasive Games“. Magerkurth u. a. definieren diese wie folgt:

> *„[a] genre in which traditional, real-world games are augmented with computing functionality, or, depending on the perspective, purely virtual computer entertainment is brought back to the real world“*[294]

[Magerkurth u. a., 2005, S. 1]

Bei Pervasive Games wird die reale Welt eine intrinsische Komponente für das gesamte Spiel, wodurch die Grenzen von traditionellen Spielen überwunden werden. Durch den Verbund der beiden Welten wird die soziale Komponente von traditionellen Spielen in die von Computerspielen übertragen. Einen Schritt weiter gehen die so genannten „(Augmented) Pervasive Location-Based Games“, welche zusätzlich Augmented Reality (AR) einsetzen, um die Umgebung der Spielenden durch virtuelle Objekte zu erweitern und die räumliche und zeitliche Präsenz zu verändern. Anwendungen, die unter diesen Kontext fallen, setzen Lokalisierungsmethoden wie Global Positioning System (GPS) oder WLAN ein, um beispielsweise auf einer Karte zusätzliche Informationen anzuzeigen. [Herbst u. a., 2008; Soute u. a., 2010] Einige Beispiele für Pervasive Games sind:

Can you See me Now [Benford u. a., 2006]: Hierbei handelt es sich prinzipiell um ein Fangspiel, in der sich die Spieler und Spielerinnen auf einer virtuellen Karte sehen, um sie in der Realität zu suchen und zu fangen.

Catchbob! [Nova u. a., 2006]: Dies ist ein Spiel, bei dem digitale Objekte auf Karten der realen Welt angezeigt werden. Anders als beim vorherigen Spiel müssen hier demnach digitale Objekte in

[294] Eine alternative Definition aus [Montola, 2011, 2005, 2009] ist folgende: „expand the magic circle of play socially, spatially or temporally“.

der echten Welt gesucht werden. Diese Objekte können „gesammelt“ beziehungsweise „gefangen“ werden, indem drei Spieler oder Spielerinnen den „Aufenthaltsort“ des Objektes in einer Dreiecksform umstellen.

Savannah [Benford u. a., 2005]: Zu guter Letzt genannt, ist dieses Beispiel als Schulspiel konzipiert worden, bei dem die Spielenden, in diesem Fall Schülerinnen und Schüler als Zielgruppe, die Karte des Schulgeländes als Savanne angezeigt bekommen, auf welcher „Wild“ dargestellt wird. Die Spielerinnen und Spieler repräsentieren die „Löwen“ und müssen zusammenarbeiten, um die (digitale) „Beute“ zu erlegen.

Die Ursprungsidee der Pervasive Games liegt in der Zeit vor der Computerisierung, als Spiele vollständig in der realen Welt verortet wurden. Interaktionen waren hierbei entweder vollständig zwischenmenschlicher Natur, oder bezogen sich vom Menschen auf die physischen Gegenstände des jeweiligen Spiels. Laut Magerkurth u. a. gibt es mehrere Gründe, warum Computerspiele realweltlichen Spielen inzwischen vorgezogen werden. So geben sie den Menschen das Gefühl, Teil einer imaginären Welt zu sein, weshalb Spielziele auch als interaktiver und deshalb als ansprechender empfunden werden. Weiterhin werden Spieler und Spielerinnen durch mehrere Dinge gleichzeitig motiviert: Fantasie, Herausforderungen und Neugierde. Dies sind einige der Hauptelemente, welche zum Spaß eines Spiels beisteuern. [Magerkurth u. a., 2005]

Aufgrund des Umschwungs von der realen in die virtuelle Welt wird mittlerweile wieder versucht, die Spiele vermehrt in die Realität zurückzuführen, allerdings in der Form, dass die Vorteile aus beiden Welten idealerweise kombiniert werden. Pervasive Games sind dabei ein Ergebnis dieser Evolution. Dabei können verschiedene Arten der Pervasive Games unterschieden werden[295], unter anderem existieren *Location-Aware Games*.

Bei Location-Aware-Games geht es darum, den Mobilitätsaspekt in den Fokus zu rücken. Das Spiel wird demnach nicht mehr an einem definierten Ort gespielt, sondern die gesamte Umgebung kann zum Spielfeld erklärt werden. Zu diesem Zweck kann ebenfalls mit Methoden der AR gearbeitet werden oder mit Metaphern, wie beispielsweise das gesperrte Drahtlosnetzwerke als „Gegner“ deklariert werden, während offene Netzwerke verwendet werden können, um eine Armee aufzubauen. So wird ein virtuelles Spiel auf die reale Spielwelt und deren Größe übertragen. Verwendet werden dabei Technologien wie WLAN, GPS, mobiles Internet oder RFID Chips[296] [Magerkurth u. a., 2005, S. 8f.].

Magerkurth u. a. stellen heraus, dass der große Nachteil von Computerspielen heutzutage der ist, dass die soziale Interaktion vernachlässigt wird, genauer formuliert das „Face-To-Face“-Setting [Ma-

[295] Im Anhang C.1 auf Seite xxix sind weitere Arten von pervsaive Games zu finden.

[296] RFID-Transponder lassen sich im Gegensatz zu Barcodes oder QR-Codes vollkommen ohne Sichtkontakt und Berührungen auslesen und erlauben somit eine automatische Identifizierung von Objekten, welche zudem robuster und weniger fehleranfällig ist.

gerkurth u. a., 2005, S. 6]. Daher sei es nur eine Frage der Evolution, dass versucht wird, die Vorteile beider Spielwelten zu vereinen.

12.2.1. Abgrenzung zur Arbeit

Bei pervasive Games stellt die reale Welt eine intrinsische Komponente für das gesamte Spiel dar. Insbesondere Live Action Role Playing Games (LARPs) werden als pervasive Games umgesetzt [Falk u. Davenport, 2004; Tychsen u. a., 2006]. Ein Beispiel für ein populäres Spiel ist *Ingress* von *Google*. Hierbei geht es darum, in der realen Welt an bestimmten Orten Aktionen auszuführen, wofür eine Smartphone-Anwendung zur Hilfe genommen wird [Briegleb, 2013]. Die reale Welt stellt in diesem Fall gleichzeitig das Spielfeld dar. Die Digitalisierung eines Pen-&-Paper-Rollenspiels im Rahmen dieser Arbeit legt den Schwerpunkt auf die digitale Welt, während digitale Ereignisse in die reale Welt zurückgeführt werden und umgekehrt reale Ereignisse in die digitale Welt mit einfließen. Eine Umsetzung als pervasive Game wäre nicht mehr ortsungebundenen und verlässt damit das Spielgefühl eines gemeinsamen Raums und steht in Kontrast zum Konzept eines Pen-&-Paper-Rollenspiels.

12.3. Tangible User Interfaces

Mit Hilfe von Tangible User Interfaces (TUIs) wird versucht, den vom Menschen als natürlich empfundenen Umgang mit der (physikalischen) Welt entgegenzukommen. Traditionelle Benutzerschnittstellen wie Maus und Tastatur oder, allgemeiner formuliert, das WIMP-Prinzip, lassen keine direkte physikalische Manipulation virtueller Welten zu, sondern bauen auf einer Vermittlung auf, bei der häufig Metaphern eingesetzt werden. Dieser Bruch soll durch den Einsatz von Tangible User Interfaces vermieden werden, indem reale, physikalische Objekte zur Interaktion mit einem digitalen System herangezogen werden. [Ishii, 2008, S. 32]

Das Ziel von Tangible User Interfaces ist es somit, die haptischen Interaktionsfähigkeiten des Menschen sinnvoll zu nutzen, um *„digitalen Informationen eine physische Form“*[297] zu geben [Ishii, 2008, S. 35]. Der Rückweg soll genau so funktionieren: Physikalische Eingaben werden in digitale Informationen umgesetzt. Damit findet eine Vereinigung der virtuellen, sozialen und physikalischen Dimension statt. Durch die Integration von Eigenschaften aus der realen Welt wird die Immersion verstärkt. Dies gilt insbesondere für atmosphärische Eigenschaften wie Umgebungsgeräusche oder die Beleuchtung [Magerkurth u. a., 2004]:

[297] „Give physical form to digital information“ - übersetzt von den Autoren aus [Ishii, 2008, S. 35].

> *„Think of a vampire game where the light conditions at the game table directly relate to the danger of being attacked by the undead. Or a spy game, where real world noise makes it difficult to overhear a conversation between non-player-characters in the virtual domain.“*

[Magerkurth u. a., 2004, S. 3]

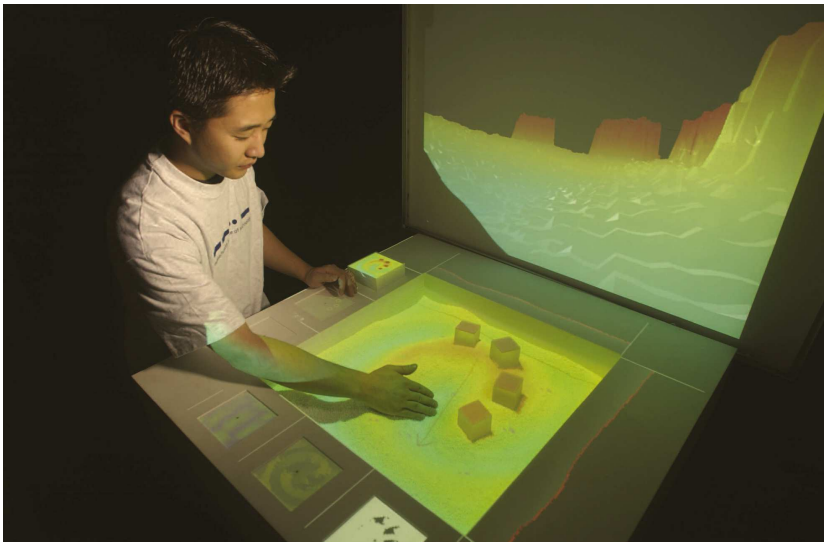


Abbildung 12.3.: Bei diesem Projekt können Benutzer eine Landschaft verändern, indem sie diese mit Hilfe von realem Sand formen. Die Ergebnisse digitaler Berechnungen werden hierbei wiederum in Echtzeit auf die echte Sandoberfläche projiziert.

aus [Ishii, 2008, S. 35]

Ein Beispiel für ein so genanntes organisches Tangible User Interfaces ist das Projekt *SandScape* (vgl. Abbildung 12.3). Bei diesem Projekt geht es darum, Nutzerinnen und Nutzern verschiedene Aspekte von Landschafbildung intuitiv verständlich zu machen. Dabei werden digitale Informationen über die soeben aus echtem Sand geformte Landschaft in Echtzeit auf ebendiese projiziert. Auf diese Weise können komplexe Modellierungsvorgänge und Topografien intuitiv verstanden werden, während die klassische Modellierung einer Landschaft in etablierten 3D-Programmen teilweise sehr aufwändig ist. [Ishii, 2008, S. 36]

12.3.1. Golem Arcana

Das Prinzip der Digitalisierung eines Pen-&-Paper-Rollenspiels lässt sich bei dem in der Einleitung dieser Ausarbeitung bereits erwähnten[298] Umsetzung *Golem Arcana* wiederfinden. Dieses Spiel ist insbesondere deswegen interessant, weil hier einige Überlegungen getroffen wurden, die unabhängig von den in dieser Arbeit[299] bereits zu einem früheren Zeitpunkt verwirklichten Implementierungen in einem kommerziellen Produkt bestätigt werden. Bei diesem Projekt wird ebenfalls die Hilfe von Touchscreens herangezogen und auch hier handelt es sich um die Umsetzung eines Pen-&-Paper- respektive Tabletop-Rollenspiels. Wie bereits erwähnt wurde, können die Spielerinnen und Spieler dabei reale Figuren auf einem Spielbrett aufstellen; teilen ihre Eingaben allerdings einem digitalen Systems mit, unter Verwendung eines per Bluetooth verbundenen Scanner-Stiftes, welcher entsprechende Markierungen auf dem Spielbrett erkennt (vgl. Abbildung 12.4).

Abbildung 12.4.: Die Abbildung zeigt den Eingabestift, mit dem verschiedenen Markierungen eingescannt werden können. Informationen über den anvisierten Gegenstand werden auf dem im Hintergrund zu sehenden Tablet angezeigt.

aus [Harebrained Schemes LLC, 2013a]

Das explizite Ziel des Projektes ist damit die Unterstützung und der erleichterte Zugang zum erfolgreichen Konzept des Pen-&-Paper-Rollenspiels, um den Spielerinnen und Spielern einen Großteil der

[298] vgl. Abschnitt 5.2.3 auf Seite 113
[299] und des vorangegangenen Projektes

sonst notwendigen Vorarbeit und Kenntnisse der Spielregeln abzunehmen. Das Überprüfen von Spielzügen oder die Bewertung von Kampfsituationen wird von der Anwendung übernommen und muss nicht mehr analog und manuell vorgenommen werden. Es handelt sich hierbei um ein Tangible User Interface (TUI) und damit ebenfalls um eine MR-Anwendung.

12.3.1.1. Abgrenzung zur Arbeit

Obwohl die Thematik von *Golem Arcana* viele Ähnlichkeiten zum im Rahmen dieser Arbeit angestrebten System aufweist, lassen sich ebenso einige weitreichende Unterschiede feststellen. Der Entwickler *Harebrained Schemes LLC* behält das Spielprinzip klassischer Rollenspiele weitgehend bei und ergänzt diese um digitale Elemente, während in dieser Ausarbeitung darauf gezielt wird, ein Rollenspiel möglichst sinnvoll und vollständig zu digitalisieren. Das Hauptaugenmerk liegt ebenfalls nicht auf der Schaffung eines sozialen und/oder topischen Raumes, sondern auf der Übertragung des Kampfsystems und der Überprüfung von Regeln mit Unterstützung des Computers. Das Spiel läuft rundenbasiert und nicht in Echtzeit ab und weist ebenfalls Elemente der Tangible User Interfaces auf. Kollaborationen und gemeinsame Interaktion steht nicht im Fokus, vielmehr wird das analoge Spielprinzip der Pen-&-Paper-Rollenspiele um digitale Komponenten ergänzt, ohne den Kern zu verändern.

Insgesamt ergibt sich auf diese Weise eher ein repetitives Vorgehen während eines Spiels, da keine Rückführung virtueller Ereignisse in die reale Welt stattfindet.

12.3.2. KnightMage

Die Autoren des Spiels *KnightMage* wählten ein virtuelles Spielbrett, auf welchem die Manipulationen der virtuellen Spielwelt mit echten physikalischen Interaktionen verschmelzen, um eine soziale Spielerfahrung zu schaffen. Besonders betrachtet wurde die Sinnhaftigkeit der Digitalisierung einzelner Elemente. Beispielsweise sollten die Würfel aus Rollenspielen nicht ohne Weiteres digitalisiert werden, weil viele eine soziale und aufregende Erfahrung damit verbinden oder einen engen Bezug zu eigens mitgebrachten Würfeln besitzen. Ebenfalls sollte nicht das gesamte Spielverhalten digitalisiert werden und automatisch Entscheidungen getroffen werden. Viel sinnvoller sei es, sich als Spieldesigner Möglichkeiten offen zu halten und gewisse Entscheidungen von den Spielern und Spielerinnen selber vornehmen zu lassen. Magerkurth u. a. bieten daher sowohl eine virtuelle als auch physikalische Basis an, welche von den Spielern verwendet werden kann, um eine individuelle und soziale Spielerfahrung wahrzunehmen.

Abbildung 12.5.: Auf der Abbildung ist das berührungsempfindliche Display dargestellt, welche eine Situation aus dem Spiel *KnightMage* darstellt. Auf der Oberfläche werden physikalische Objekte verteilt, welche durch eine Kamera erkannt und in die virtuelle Welt übertragen werden.

aus [Magerkurth u. a., 2004, S. 8]

Hierzu verwendeten sie einen Touchscreen-Tisch, welcher die virtuelle Repräsentation der Spielwelt darstellt und sowohl durch direkte Berührungen als auch durch die Verwendung physikalischer Gegenstände manipuliert werden kann (siehe Abb. 12.5). Mit Hilfe einer über dem Tisch positionierten Kamera werden Bewegungen und Positionierungen realer Gegenstände registriert sowie ausgewertet, woraufhin eine Rückführung in die reale Welt stattfindet. Durch eine erschwerte Objektidentifizierung, z.B. aufgrund unterschiedlicher Lichtbedingungen, existiert ferner die Möglichkeit, Muster auf den Objekten zu verwenden oder die visuelle Identifizierung durch eine Sensortechnologie (z.B. RFID) zu ersetzen. Zusätzlich wurden für persönliche Informationen und Vorhaben mobile Geräte eingesetzt, welche ebenfalls mit der Anwendung gekoppelt wurden. Unter Verwendung dieses Aufbaus wurden verschiedene Prototypen entwickelt, wovon *KnightMage* am ähnlichsten zu dem hier geplanten Vorhaben ist.

Spieler explorieren eine Welt mit Hilfe physikalischer Gegenstände auf der virtuell dargestellten Spielwelt. Das Spielgeschehen wird dabei von einem menschlichen Game Master moderiert. Die Idee war hierbei, die sozialen Aspekte von klassischen Tabletop-Spielen mit den Vorteilen einer virtuellen Spielwelt zu kombinieren, da es besonders in Pen-&-Paper-Spielen häufig an Immersionsfähigkeit und

Visualisierungen fehlt. *KnightMage* beinhaltet kooperative sowie kompetitive Komponenten; zudem werden bestimmte Spielereignisse direkt an einzelne Spieler gesendet, woraufhin diese entscheiden können das Wissen für sich selber zu nutzen oder mit der Gruppe zu teilen.

12.3.2.1. Abgrenzung zur Arbeit

Beim Projekt *KnightMage* werden echte physikalische Gegenstände verwendet; somit handelt es sich um eine Anwendung aus dem Bereich der TUIs. Nicht alle Elemente eignen sich für eine Digitalisierung, sodass bewusst nicht-digitale Komponenten beibehalten werden. Ein Beispiel hierfür sind Spielwürfel, zu denen Spieler oft eine persönliche Bindung haben oder selbst gestaltet sind und eine ganz eigene Interaktion in der realen Welt bilden. Eine digitale Implementierung eines Würfels kann dieses Erlebnis nicht bieten.

Die Entscheidung, den Game Master nicht zu digitalisieren, wird bewusst getroffen und verargumentiert:

> *„While the advancements in the field of AI make actions of virtual entities more and more believable, they still pale against the richness and immersiveness of the social interaction with a skilled human game master or storyteller. On the other hand, putting too much action into the social domain might incumber the creation of interfaces between both domains and lead to gaming styles that are essentially identical to e.g. traditional tabletop role playing games that work very well without any computer support.“*

[Magerkurth u. a., 2004, S. 4]

Letztlich weist diese Anwendung zudem keine Echtzeit-Komponente auf, da das Spielprinzip hier ebenfalls rundenbasiert erfolgt.

„Since the concept of merging physical and cyberspaces covers applicable fields to a great extent, we expect that the interest or attention now paid to MR will spread and increase. “

(Tamura u. a. [Tamura u. a., 2001, S. 70])

KAPITEL 13

Mixed Reality-Konzeption

Die Intention, Menschen erneut zu realer Face-2-Face-Interaktion zu animieren, beruht auf der Beobachtung, dass die steigende Digitalisierung in vielen Lebensbereichen und insbesondere im Bereich der Spielindustrie ein zweischneidiges Schwert ist. Auf der einen Seite bildet die weltweite Vernetzung und Digitalisierung einen erleichterten Zugang zu Informationen, neuen Bekanntschaften und daraus folgend, auch Zugang zu Menschen, mit denen zusammen gespielt wird (vgl. Abbildung 2.5 im Masterprojekt).

Auf der anderen Seite kann beobachtet werden, dass diese Digitalisierung gleichsam isolierend und entfremdend wirken kann, da Barrieren zur Sozialisierung, im Gegensatz zu direkter, zwischenmenschlicher Kommunikation, bidirektional niedriger ausfallen: Kontakte werden leichter hergestellt und leichter beendet. Ein Kurzfilm von deGuzman u. Crawford mit dem Titel „I Forgot My Phone“ veranschaulicht beispielsweise die eher als negativ zu bewertenden Seiten der steigenden Integration von Smartphones in den Alltag [deGuzman u. Crawford, 2013]. Ein prägendes, längeres Zitat von Björk u. a. fasst die Situation folgendermaßen zusammen:

> *„While computer technologies have provided exciting directions for game design and game play, they have also tended to eliminate many of the social aspects of ‘traditional’ game play. With computer games, we no longer need to physically get together with friends and family to play games that involve other people than ourselves. In some sense, computers have turned game play into individual and isolated activities. In a typical computer game, the game and its mechanics are inside a stationary computer, and if we interact with fellow game players, we do so through a computer screen, rather than in a face-to-face, co-located situation.“*

[Björk u. a., 2001, S. 1]

Der Anspruch dieser Arbeit bestand unter anderem darin, die Möglichkeiten moderner Technologien zielgerichtet einzusetzen, um eine reale Interaktion zwischen den Partizipierenden zu erhöhen. Es

lag nahe, als Grundlage für eine Digitalisierung ein Prinzip aufzugreifen, welches die grundsätzlichen Ziele der Arbeit unterstützt respektive in den Vordergrund hebt. Aus diesem Gedanken heraus wurden Pen-&-Paper-Rollenspiele als Digitalisierungsgrundlage ausgewählt, da hier eine enge Absprache und Interaktion zwischen den Teilnehmenden ein zwingend notwendiges Element darstellt.

Nach dieser Grundsatzentscheidung galt es, ein Theorem dafür aufzustellen, wie die Interaktion zwischen den Spielerinnen und Spielern eines realen Pen-&-Paper-Rollenspiels konserviert und mit einer virtuellen Spielwelt kombiniert werden kann. Im Grundlagenteil dieser Arbeit wurde dabei herausgestellt, dass die Konzeption einer Mixed Reality-Anwendung hierfür als geeignet erscheint, da reale Interaktionen tatsächlich in der Realität verbleiben, während sich der digitalisierte Teil der Anwendung um die Aspekte kümmert, welche in der Realität nur schwer realisierbar ist[300].

Laut Magerkurth u. a.; Björk u. a. wird ein digitalisiertes Spiel, unabhängig von der Anzahl der partizipierenden Personen, eher als isolierte Tätigkeit wahrgenommen, da in unnatürlicher Art über Technologie kommuniziert sowie interagiert wird [Magerkurth u. a., 2004]. Ein erster Kerngedanke zur Adressierung dieser Probleme war, dass die Separation der Spielenden durch Technologie aufgebrochen und unterbunden werden muss, damit eine reale Interaktion entstehen kann. Zunächst galt es daher als unabdingbar den Systemaufbau so zu gestalten, dass eine physische Anwesenheit der Spielerinnen und Spieler zur Voraussetzung wird. Ein weiterer Schritt war der, dass den Spielenden ein möglicher Rückzugsort in Form eines eigenen Bildschirmabschnitts vorenthalten wird, da auch hier wieder eine Separation entstehen kann, wenn jeder nur auf den eigenen Bildschirmbereich schaut [Tychsen u. Smith, 2008, S. 119].

Die letzte mit technischen Methoden adressierbare Separation ist die der Spielerinnen und Spieler vom Spiel selbst. Traditionell findet bei Medienformen wie dem Theater, dem Film und auch den Computerspielen eine Trennung zwischen Handlungs- und Rezeptionsebene statt; ein unter dem Begriff der *vierten Wand* bekanntes Phänomen. Reale Interaktion zwischen Menschen sieht eine Einheit von Aktions- und Rezeptionsspäre vor. Ein einfaches Beispiel ist das des gemeinsamen Abends in der Bar: Menschen sitzen um einen Tisch herum und unterhalten sich, wobei Zeit und Ort hier eine Einheit bilden. Ein Gegenbeispiel ist der gemeinsame Besuch eines Theaterstücks, wo die eigentliche (fiktive) Handlung auf der Bühne stattfindet, während von den Publikumsplätzen aus zugesehen wird. Eine Interaktion zwischen den Besuchern findet nicht statt, obwohl alle gemeinsam die gleiche Präsentationsebene rezipieren. Genau an diesem Punkt setzt das MR-Konzept an, unter dessen Verwendung versucht werden soll, diese Grenze zu durchbrechen.

[300] vgl. Abschnitt 7.2 auf Seite 145

13.1. Die vierte Wand als Volumen

Das Prinzip der vierten Wand kann ebenfalls am bereits erwähnten Beispiel des Theaters näher erläutert werden: das klassische Bühnenlayout verfügt dort über drei Wände. Die Rückwand der Bühne bildet die erste Wand, während die rechte und die linke Wand neben der Bühne das Geschehen einrahmen. Bei der *vierten Wand* handelt es sich letztlich um ein imaginäres Konstrukt, welches als Grenze zwischen der Bühne und dem Publikum verstanden werden kann. Die Darstellerinnen und Darsteller agieren so, als wäre diese vierte Wand tatsächlich physikalisch vorhanden, während das Publikum dadurch praktisch nicht existiert. Der Blick auf die Theaterbühne erhält auf diese Weise eine unidirektionale Beziehung, bei der eine echte Interaktion zwischen Publikum und Bühnengeschehen nicht stattfinden kann, da das Publikum per Definitionem nicht innerhalb des Bühnenraums existiert und lediglich von „außen“ zuguckt.

Der Begriff der vierten Wand entstammt demnach ursprünglich einer geometrischen Analogie[301], wird inzwischen jedoch als Synonym für eine unsichtbare Grenze betrachtet, welche die Präsentationsschicht[302] von der Rezeptionsschicht trennt und zwar nicht nur unabhängig vom Theater sondern ebenfalls unabhängig vom eingesetzten Medium im Allgemeinen. Auch ein Hörspiel kann auf diese Weise über eine vierte Wand verfügen.

Aus narrativer Perspektive sollte generell vermieden werden die vierte Wand zu durchbrechen, um die anvisierte Illusion beziehungsweise die Handlungsebene nicht in ihrer Integrität und Glaubwürdigkeit zu verletzen [Sheldon, 2013, S. 186].

Auf der anderen Seite kann das Durchbrechen der *vierten Wand* auch zum absichtlichen Prinzip erhoben werden, was gerade im Bereich der Computerspiele häufig anzutreffen ist [Sheldon, 2013, S. 189]. Das kann sowohl in Form von sogenannten Easter Eggs geschehen, als auch durch direkte Anrede[303]. Ein Beispiel, bei dem das aktive Durchbrechen der vierten Wand zum Prinzip und wesentlichem Teil des Spielerlebnisses erhoben wird, ist der PC-Spieletitel *The Stanley Parable*[304], in dem der Spieler in die Rolle des namensgebenden Charakters „Stanley“ schlüpft. Es ist an einer Stelle beispielsweise möglich die eigentliche Spielwelt zu verlassen, sodass für einen kurzen Moment der Eindruck entsteht, es handele sich hierbei um einen Designfehler, woraufhin eine Erzählerstimme aus dem *Off* mit folgendem Kommentar reagiert:

[301] Es existieren durchaus Theater, bei denen die Bühne an drei Seiten von Publikumsrängen umgeben ist, wofür das *Globe Theater* in London sicherlich als prominentes Beispiel dient [The Shakespeare Globe Trust, 2014]. Den Begriff der *vierten Wand* hier im Wortsinn anzuwenden ist bei diesem Theateraufbau demnach nicht sinnvoll, da die Bühne nur über eine Rückwand verfügt.

[302] beim Theater demnach die Welt *innerhalb* des Theaterstückes

[303] Beispielsweise: „Du spielst schon seit über 5 Stunden am Stück. Möchtest Du nicht mal eine Pause einlegen?“

[304] 2013, *Galactic Cafe* [MobyGames.com, 2013w]

> *"At first Stanley assumed he'd broken the map, until he heard this narration and realized it was part of the games design all along. He then praised the game for its insightful and witty commentary into the nature of video game structure and its examination of structural narrative tropes."*

[Zitat entnommen aus dem Spiel]

Sheldon schreibt zu einer möglichen Designentscheidung, die vierte Wand zu durchbrechen:

> *„If we're going to break the fourth wall, we'd better know what we're doing, and have a damn good reason for doing it."*

[Sheldon, 2013, S. 190]

Mit Bezug auf das von Weiser erdachte Prinzip vom *allgegenwärtigen Computer*[305] kam bei der Konzeption der MR-Aspekte folgender Gedanke auf: Was passiert, wenn die *vierte Wand* statt als *Wand* als *Volumen* betrachtet wird, welches in den realen Raum hinein ragt? Ein Volumen kann nicht mehr durchbrochen werden, wenn es den gesamten Raum ausfüllt. Die Rezipienten befinden sich ständig innerhalb des Volumens, während die virtuelle Realität Teil diese Volumens wird. Die Grenzen zwischen Realität und Virtualität werden aufgehoben, indem die „Bühne" und der „Zuschauerraum" verschmelzen. Im nächsten Abschnitt soll beschrieben werden, wie dieses Volumen unter Verwendung von Smartphones aufgespannt werden kann.

13.2. Das Smartphone als Mixed Reality-Interaktionsgerät

Der derzeitige Erfolg von Smartphones alleine in Deutschland ist statistisch erfassbar: Laut einer Pressemeldung des Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V. (BITKOM) stellen Smartphones inzwischen 96% Mobilfunkmarktes dar [Thylmann u. a., 2013]. Geelhaar sieht die Gründe für den hohen Erfolg unter anderem in dem Bedürfnis des Menschen nach hoher Mobilität in urbanen Räumen. Das Smartphone kann dieses Bedürfnis in vielerlei Hinsicht unterstützten und ebenfalls ein Gefühl von Sicherheit und Autonomie vermitteln [Geelhaar, 2012, S. 196]. Ryan u. Deci unterscheiden hierbei zwischen mehreren zentralen Bedürfnissen des Menschen, welche auch bei der Entwicklung von Smartphone-Anwendungen wiederzufinden sind (vgl. Tabelle 13.1, siehe auch [Geelhaar, 2012, S. 197]). Bei dem ersten Bedürfnis handelt es sich um das nach Autonomie als selbstbestimmtes Individuum, abzugrenzen von der Wortbedeutung der Autonomie als Unabhängigkeit von Mitmenschen. Das zweite Bedürfnis wird als Kompetenz respektive Selbstbewusstsein

[305] vgl. Abschnitt 6.2 auf Seite 124

Bedürfnis	Qualität	Unterstützung	Schutz
Sicherheit	In der Gruppe geborgen sein.	Mit dem privaten und sozialen Netz verbunden sein.	Schutz der Privatsphäre
Soziale Beziehungen	Kommunikation	Unterstützung von erwünschter Kommunikation	Schutz vor unerwünschter Kommunikation
Selbstbestimmtheit	Information (im Sinne der Vermittlung von Inhalten)	Zugang zu benötigten Informationen	Schutz vor unerwünschter Information
Selbstverwirklichung	Spiel	Möglichkeit neuer Spielideen (z.B. GPS gestützte Spiele, kollaborative Spiele)	Zuviel Ablenkung und Konzentration auf die digitale Welt

Tabelle 13.1.: Die Tabelle zeigt, wie Smartphones respektive Smartphone-Anwendungen menschliche Bedürfnisse erfüllen könnten. Insbesondere der Punkt *Selbstverwirklichung* ist im Rahmen dieser Arbeit von besonderer Relevanz, da an dieser Stelle explizit kollaborative Spiele als Beispiel genannt werden. Ein Kollaboratives Spiel unterstützt demnach unter anderem das Bedürfnis des Menschen nach Selbstverwirklichung.

aus [Geelhaar, 2012, S. 197]

klassifiziert, im Sinne einer Erwartungshaltung, dass die eigene Leistung durch andere Menschen anerkannt wird. Das dritte Bedürfnis letztlich ist der Wunsch nach Sozialisierung (vgl. [Geelhaar, 2012, S. 197], [Ryan u. Deci, 2000]).

Anhand der Tabelle 13.1 wird ebenfalls ersichtlich, dass Kollaboration respektive kollaborative Spiele unter Verwendung von Smartphones verwirklicht werden können. Jüngst u. Link stellen heraus, dass zwischenmenschliche Interaktion gerade unter Einsatz physikalischer Objekte gefördert wird. Durch zusätzliche, körperliche Interaktion im Raum ließe sich ebenfalls die *„symbolische, emotionale und organisatorische Ausdruckskraft“* vermitteln beziehungsweise verstärken [Jüngst u. Link, 2012, S. 224]. Um Kollaboration und Sozialisierung respektive einen sozialen Raum [306] entstehen zu lassen, muss ein solcher erschaffen werden, nicht nur technologisch, sondern auch real. Aarseth schreibt, dass Spiele schon per se eine räumliche Komponente aufweisen würden:

> *„The defining element in computer games is spatiality. Computer games are essentially concerned with spatial representation and negotiation, and therefore a classification of computer games can be based on how they represent – or, perhaps, implement – space.“*

[Aarseth, 2007, S. 154]

[306] vgl. Kapitel 5 auf Seite 103

Dieser Raumbegriff kann mit dem Begriff der *vierten Wand* in Einklang gebracht werden und zwar genau dann, wenn die vierte Wand als Volumen betrachtet wird. Wie können Smartphones als Mixed Reality-Gerät definiert werden oder anders ausgedrückt: Wie können Smartphones dabei helfen können, eine Mixed Reality-Umgebung zu erzeugen? Diese Frage soll im Rahmen einer Definition im nächsten Abschnitt beantwortet werden.

13.3. MR-Definition

Um die im vorherigen Abschnitt gestellte Frage beantworten zu können, muss zunächst definiert werden, was unter dem Begriff *Mixed Reality* im Rahmen dieser Arbeit verstanden wird. Die Definition muss hierbei dem Umstand Rechnung tragen, dass die Fähigkeiten und Sensoren eines Smartphones alleine durch ihre Benutzung noch kein Aufspannen einer Mixed Reality-Umgebung hervorrufen. Wenn beispielsweise schon die Vibration eines Interaktionsgeräts als Mixed Reality-Komponente gälte, so müssten daraus folgend nahezu alle moderneren Controller von Spielekonsolen ebenfalls als Mixed Reality-Geräte bezeichnet werden[307]. Eine ähnliche Überlegung kann für die Verknüpfung realer und virtueller Welt getroffen werden, wenn Navigationsgeräte für Autos betrachtet werden. Die Auswertung und Einbettung eines GPS-Signals mit realen Koordinaten zur Anzeige des (virtuellen) Fahrtwegs auf einem Display erfüllt rein technisch die Voraussetzungen, um als Mixed Reality-Anwendung bezeichnet zu werden.

Die Verwertung von realen Daten oder Ereignissen, um virtuelle Ereignisse auszulösen ist demnach alleine nicht ausreichend, um die Anwendung von einer rein virtuellen Existenz in die Wirklichkeit hinein zu verschieben oder umgekehrt. Es ist der *Kreislauf* aus realen Ereignissen, welche Einfluss auf die Virtualität ausüben und gleichsam virtuelle Ereignisse, die Einfluss auf die Realität ausüben. In diesem Sinne kann eine Mixed Reality-Anwendung unter Einbezug des Begriffs der *vierten Wand* wie folgt definiert werden:

Eine Mixed Reality-Anwendung lässt virtuelle Ereignisse die vierte Wand durchbrechen, während gleichfalls reale Ereignisse signifikanten Einfluss auf virtuelle Ereignisse nehmen können und ihrerseits die vierte Wand durchbrechen. Die resultierende Anwendung erzeugt innerhalb ihres Wirkungskreises ein *Mixed Reality-Volumen*.

[307] Das erste Videospiel, welches Vibration als Feedback unterstützte, war das Arcade-Automaten Spiel *Moto-Cross* aus dem Jahr 1976. Der Lenker dieser Motorradsimulation vibrierte, wenn eine Kollision mit einem anderen Fahrzeug stattfand. Inzwischen beherrschen praktisch alle Controller moderner Konsolen sowie die sogenannten *Gamepads* über eine Vibrationsfunktion [Wolf, 2008, S. 39] [The Arcade Flyer Archive, 2014].

Durch die Definition der vierten Wand als Volumen kann diese nicht mehr durchbrochen werden, da es Teil des Raumes wird und ihn ausfüllt. Auf diese Weise entsteht auch eine implizite Abhängigkeit von äußeren Einflüssen, was bedeutet, dass die Umgebung, in der das Spiel gespielt wird, zum Teil des Spiels wird. In Abschnitt 14.3.1 ab Seite 248 wird die Umsetzung der in diesem Kapitel vorgestellten Überlegungen beschrieben, wobei dort Mechanismen zum tragen kommen, welche die hier vorgestellte Definition berücksichtigen. Mit Bezug auf die in Abschnitt 7.1 beschriebenen verschiedenen Mixed Reality-Taxonomien wurde hierbei grundsätzlich der erste Ansatz erfolgt, in dem Sinne, dass reale Einflüsse wie Licht und Ton Auswirkungen in der Virtualität nach sich ziehen.

Die gestellte Frage, wie Smartphones unterstützend eingesetzt werden können, um eine Mixed Reality-Umgebung zu erzeugen, kann nun mit Bezug auf die aufgestellte Definition beantwortet werden. Smartphones stellen ein ambivalentes Interaktionsgerät dar, welches nicht nur über Sensoren, sondern auch über Aktoren verfügt. Auf diese Weise kann ein Kreislauf zwischen der Realität und der Virtualität erzeugt werden, indem Smartphones die *vierte Wand* in beide Richtungen durchbrechen können. Diese bilden damit die Eckpunkte eines *Virtual-Reality-Volumens*.

„After nourishment, shelter and companionship, stories are the thing we need most in the world."

(Philip Pullman [Gillespie, 2013]*)*

KAPITEL 14

Storytelling- und Gameplay-Konzeption

Eine Designentscheidungen mit weitreichenden Auswirkungen war die des zu verwendenden Erzählkonstrukts. Zum einen musste ein Modell gewählt werden, welches den Anforderungen und der Dynamik des Pen-&-Paper-Rollenspiels hinreichend genügt und zum anderen wird mit der Wahl des Erzählkonstrukts gleichzeitig die Wahl des Spielaufbaus getroffen, sofern der Empfehlung von Sheldon gefolgt wird:

> *„Gameplay and story work best together when developed together"*

[Sheldon, 2013, S. 159][308]

Das Gameplay steht dabei in direktem Zusammenhang mit den realisierten Interaktionen und somit dem Ziel dieser Arbeit (vgl. Definition auf Seite 104). In diesem Kapitel sollen daher die Gedankengänge nachvollzogen werden, die letztlich sowohl die Gameplay- als auch die Storytelling-Konzeption begründen.

14.1. Storytelling-Konzeption

In Kapitel 8 wurde herausgearbeitet, dass es verschiedene Paradigmen gibt, die beim Storytelling zum Einsatz kommen können. Diese verschiedenen Ansätze eint, dass sie zunächst unabhängig vom eingesetzten Medium existieren und allesamt über spezifische Vor- und Nachteile verfügen. Es ist für Spielerinnen und Spieler nicht unbedingt apparent, welche Erzählstruktur in einem Computerspiel

[308] vgl. auch Abschnitt 8.1 auf Seite 158

verwendet wird[309], genau so wenig ist es intuitiv verständlich, dass selbst lineare Medien wie Filme oder Bücher eine modulare Struktur beinhalten können. Was interaktive Medien und insbesondere Computerspiele von linearen Medien abgrenzt, ist der Umstand, dass Spielerinnen und Spieler in Echtzeit Einfluss auf ihren *erlebten* Durchlauf eines Spiels nehmen können.

Dem Merkmal der Pfade kommt hierbei eine besondere Bedeutung zu. Ein direktes Verhältnis zwischen der Anzahl an möglichen Pfaden zur Nichtlinearität lässt sich nicht ableiten. Wenn eine Erzählung beispielsweise zehn mögliche Endmodule hat, ist jeder Pfad zu jedem Ende klar definiert, was sich auch nicht durch den Einsatz von hundert oder tausend Endmodulen verändert. Auf der anderen Seite bedeutet das Vorhandensein von Pfaden nicht direkt im Umkehrschluss, dass es sich in jedem Fall um eine lineare Erzählstruktur handelt, wie am Beispiel der *Hypermedia Novel* verdeutlicht werden konnte[311]. Eine gänzliche Abwesenheit von Pfaden ist schließlich beim Einsatz des vollständig modularen Storytellings zu beobachten, wobei diese Abwesenheit gleichsam für die maximal mögliche Anzahl an Pfaden steht: jedes Modul kann von jedem anderen Modul aus erreicht werden.

Pen-&-Paper-Rollenspiele stellen durch ihre Struktur die Anforderung, dass sowohl die erzählte (respektive *erspielte*) Geschichte als auch die subjektive Entscheidungsfreiheit der individuell möglichen Aktionen größtmöglicher Freiheit unterliegen. Hierbei handelt es sich um zwei Anforderungen, welche durch ein modulares Storytelling bestmöglich umgesetzt werden können:

> *„Modular storytelling can help us integrate story and gameplay and create a game environment the player is free to explore."*

[Sheldon, 2013, S. 341]

> *„What are the individual quests if not seperate stories linked thematically to the overall story in an RPG?"*

[Sheldon, 2013, S. 293]

Im nachfolgenden Abschnitt wird daher beschrieben, wie sich modulares Storytelling als Rahmen für eine Erzählung in Computerspielen praktisch umsetzen lässt.

[309] Das bereits erwähnte Spiel[310] *The Stanley Parable* beispielsweise scheint auf den ersten Blick im höchsten Maße nichtlinear aufgebaut zu sein, wenn der dazugehörige Graph jedoch betrachtet wird, zeigt sich, dass das Spiel überwiegend aus simplen Verzweigungen besteht (vgl. Abbildung B.7 auf Seite xxv im Anhang).
[310] vgl. Abschnitt 13.1 auf Seite 229
[311] vgl. Abschnitt 8.2.3.1 auf Seite 166

14.1.1. Modulares Storytelling in Computerspielen

Im modularen Storytelling sind ein Anfangsmodul und ein Endmodul grundsätzlich fest vorgegeben, während sich zwischen diesen Fixpunkten frei bewegt werden kann. Diese Bewegungsfreiheit innerhalb einer Erzählung wird durch die namensgebenden Module erreicht: Abhängig von den Entscheidungen und Aktionen der Spielenden im *aktuellen* Modul kann darauf folgend jedes *andere* Modul erreicht beziehungsweise *errechnet* werden. Es wird hierbei angenommen, dass zwar Module ausgelassen, aber (in ihren Auswirkungen) nicht doppelt besucht werden können. Das bedeutet, dass es durchaus möglich ist, ein Modul mehrfach zu besuchen, die in diesem Modul veränderlichen Parameter jedoch nur einmalig zum Einfluss der weiteren Struktur zum Einsatz kommen können und nicht retrospektiv wirksam sind. Dabei ist ebenfalls möglich, direkt vom Startmodul aus das Endmodul zu erreichen. Diese Freiheit an möglichen Wegen führt zu einem signifikanten mathematischen Problem, welches im Folgenden beschrieben wird.

Das Verhältnis zwischen Anzahl der insgesamt vorhandenen Module und den daraus folgenden möglichen Story-Durchläufen N kann durch folgende Fakultätsgleichung beschrieben werden:

$$N = (\text{Anzahl der Module} - 2)! + 2$$

Abbildung 14.1.: Fakultätsgleichung möglicher Story-Durchläufe. Startmodul und Endmodul sind immer fix, weshalb diese beiden Module von der Gesamtanzahl zu betrachtender Pfade abgezogen werden müssen.

Der Zusammenhang zwischen der Anzahl der eingesetzten Module und N kann hierbei der Tabelle 14.1 entnommen werden.

Es ist offensichtlich, dass sehr schnell sinnvoll umsetzbare Grenzen erreicht werden, wenn die Anzahl der Module auch nur minimal erhöht wird. Wenn davon ausgegangen wird, dass die Erfassung von 24 möglichen Fällen subjektiv als vertretbarer Aufwand wahrgenommen wird, um alle Variationen von insgesamt sechs eingesetzten Modulen zu erfassen, so ergibt dies unter Umständen in recht kurzes Spiel. Um die Spielzeit zu erhöhen, könnten diese Module natürlich künstlich vergrößert werden, sodass ganze Spielwelt-Abschnitte als ein Modul definiert werden. Letztlich zufriedenstellend ist diese Situation jedoch nicht, weswegen zur Realisierung eines vollständig modularen Storytellings eine andere Lösung gefunden werden musste. Diese Lösung wird im folgenden Abschnitt vorgestellt.

Anzahl der Module insgesamt	Fixe Module	Permutationen	N
2	2	0	2
3	2	1	3
4	2	2	4
5	2	6	8
6	2	24	26
7	2	120	122
8	2	720	722
9	2	5040	5042
10	2	40322	40324
⋮	⋮	⋮	⋮
26	2	$\sim 6.2 \cdot 10^{23}$	$\sim 6.2 \cdot 10^{23} (+2)$

Tabelle 14.1.: Die Tabelle zeigt die Anzahl möglicher Story-Durchläufe in Abhängigkeit der Anzahl von eingesetzten Modulen beim vollständig modularen Storytelling.

14.1.2. Verschachtelte Module

Der Lösungsansatz für das Problem der exponentiellen Abhängigkeit der möglichen Story-Durchläufe von der Anzahl der eingesetzte Module ergab sich aus der Idee, dass es möglich ist, Module zu verschachteln (vgl. Abbildung 14.2).

Die Überlegung hinter der Verschachtlung von Modulen verfolgt den Gedanken, dass die Spielerinnen und Spieler sich mit jeder Verschachtelungstiefe immer auf einer Ebene vollständiger Modularität bewegen. Diese Ebenen erfüllen, separat betrachtet, die Anforderung an eine vollständige Wahlfreiheit, was bedeutet, dass jedes Modul auf einer Ebene von jedem anderen Modul aus erreicht werden kann. Ein- und Austrittspunkt jeder Modulebene werden wiederum durch jeweils zwei fixe Module realisiert.

Wenn die Spielerinnen und Spieler nun jede dieser Modulebenen durch das zwangsläufig erreichte Endmodul verlassen, dann steht am Ende jeder Ebene eine *Gesamtentscheidung*, welche der individuellen Abfolge an durchlaufenen Modulen entspricht. Diese Entscheidung kann dazu benutzt werden, auf einer höheren Ebene das nächste Modul zu errechnen, welches wiederum mehrere Untermodule enthält. Wenn dieses Schema verfolgt wird, dann entspricht die Anzahl zu berücksichtigender Entscheidungen einem *additiven* Schema, sofern die Anzahl möglicher Module *pro Ebene* und gleichzeitig die Verschachtelungstiefe begrenzt werden. Eine Tabelle mit einer beispielhaften Unterteilung, bei der die maximale Anzahl der Module pro Ebene auf insgesamt sechs Module begrenzt wurde, ist in Tabelle 14.2 zu finden.

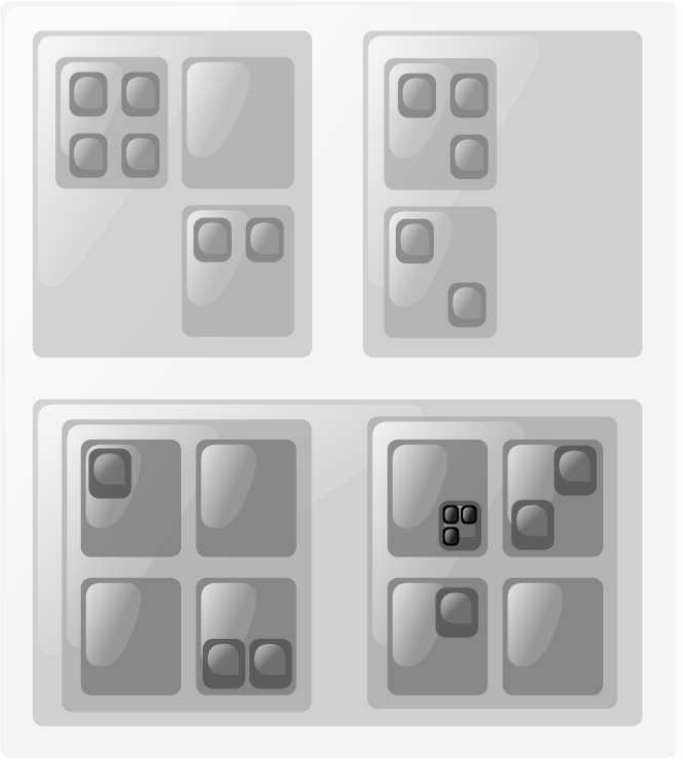


Abbildung 14.2.: Die Abbildung zeigt das Prinzip verschachtelter Module. Jedes Modul wird hierbei durch ein abgerundetes Rechteck repräsentiert, wobei die Sättigung derselben die Verschachtlungstiefe (mit) angibt.

nach [Sheldon, 2013, S. 338]

Bei diesem Beispiel sollte bedacht werden, dass es sich um eine Simplifizierung zur Veranschaulichung handelt und jede Klammer wiederum ein Modul darstellt. Bei korrekter Anwendung mit jeweils einem fixen Start- sowie Endmodul müssten bei dieser Unterteilung zusätzlich Hilfsmodule eingeführt werden, auf deren Darstellung an dieser Stelle verzichtet wurde.

Dennoch wird ersichtlich, dass die Komplexität durch die Unterteilung in Ebenen wesentlich reduziert werden kann. Selbst, wenn von dem Extremfall ausgegangen wird, dass jedes Modul jedes andere Modul beeinflusst und die Reihenfolge der Besuche ebenfalls eine Rolle spielt, bleibt die Komplexität beherrschbar. Es muss an dieser Stelle erwähnt werden, dass bei genauer Betrachtung nicht mehr jedes Modul von jedem anderen Modul erreicht werden kann, sondern dass Nadelöhre erzeugt werden, die zwangsläufig durchlaufen werden müssen, um eine andere Modulebene zu erreichen. In Anbetracht

Anzahl der Module insgesamt	Mögliche Unterteilung	N bei vollständiger Modularität	N bei der Verschachtlung
2	2	2	2
3	1!+2	3	3
4	2!+2	4	4
5	3!+2	8	8
6	4!+2	26	26
7	(4!+2)+1	122	27
8	(4!+2)+2	722	28
9	(4!+2)+1!+2	5042	29
10	(4!+2)+2!+2	40324	30
⋮	⋮	⋮	⋮
26	$(4! + 2) \cdot 4 + 2$	$\sim 6.2 \cdot 10^{23}(+2)$	108

Tabelle 14.2.: Die Tabelle zeigt die Anzahl möglicher Story-Durchläufe in Abhängigkeit der Anzahl von eingesetzten Modulen bei Anwendung der Verschachtlung im Vergleich zum vollständig modularen Storytelling.

der eingesparten Komplexität ist dies jedoch ein zu vernachlässigender Punkt, vor allem auch deshalb, weil weiterhin keine Pfade existieren. Diese werden durch das hier neu vorgestellte Prinzip der Ebenen und Verschachtlung von Modulen ersetzt. Die in diesem Abschnitt vorgestellte Lösung ermöglicht somit eine Umsetzung eines modularen Storytellings[312].

14.2. Hintergrundgeschichte

Für die im Rahmen dieser Arbeit entstehende Implementierung wird eine Hintergrundgeschichte respektive ein Pen-&-Paper-*Universum* benötigt, welches genügend Flexibilität aufweist, um die konkreten Rahmenbedingungen einer Digitalisierung zu erfüllen. Diese Rahmenbedingungen gliedern sich in zwei Aspekte:

1. Die gewählte Hintergrundgeschichte muss grundsätzlich einen modularen Aufbau unterstützen. Dies bedeutet, dass es möglich sein muss, eine zusammenhängende, kurze Geschichte zu erzählen, welche sich in mehrere, kleine Sinnabschnitte unterteilen lässt. Schauplätze im Weltraum mit extremer Offenheit sind für ein solches Vorhaben ebenso wenig geeignet wie sehr strikte Vorgaben mit wenig Spielraum zur Ausgestaltung. Als aktuelles Beispiel für eine extreme Offen-

[312] Dieses Konzept wurde mit Sheldon in einem persönlichen und sehr intensivem Austausch diskutiert und stellt seiner Meinung nach ein weiterhin (vollständig) modulares Konzept dar, wobei der Reinheitsgrad allerdings hinterfragt werden könne.

heit kann der Film *Gravity*[313] herangezogen werden, bei dem eine Routinemission im Weltraum aufgrund eines Unfalls scheitert und fortan die Hauptdarstellerin versucht, einen Weg zurück zur Erde zu finden. Ein überwiegender Teil des Films und seiner Spannung basiert letztlich auf dem Kontrast zwischen einem einzelnen Menschen im Raumanzug im Vergleich zur Größe des Weltraums. Der Versuch, dieses Setting in sinnvoll unterteilte Module zu übertragen, kann sich als schwieriges Unterfangen herausstellen.

2. Bereits im Masterprojekt, auf dem diese Arbeit aufbaut, wurde als visualisierende 3D-Engine des Game-Engine-Clients das UDK und somit die *Unreal Engine* gewählt. Das Unreal Development Kit stellt eine Auswahl an vorgefertigten Elementen bereit, welche einen beschleunigten Aufbau von virtuellen Umgebungen ermöglichen. Diese Objekte lehnen sich überwiegend an alte verfallene Ruinen, Industrieanlagen, Dschungel-Umgebungen und Science-Fiction-Elemente an. Es ist daher sinnvoll, das Universum so zu wählen, dass möglichst viele dieser vorgefertigten Elemente in sinnvoller Art und Weise verbaut werden können. Ein Universum wie das von *Dungeons & Dragons* wäre daher speziell in dieser Arbeit nur mit sehr viel Aufwand realisierbar, da viele Bauteile neu erstellt werden müssten.

Eine Suche nach existierenden Pen-&-Paper-Universen, welche an verfallenen, alten, dennoch technologieorientierten Umgebungen angelehnt sind, führt in den Bereich des *Steampunk* und des *Cyberpunk* oder allgemeiner formuliert in den Bereich der *Science-Fiction Dark Future* (vgl. [Schick, 1991, S. 285]). Die Schauplätze in diesem Genre haben meist gemein, dass eine oftmals dystopische Zukunftsvision der Erde mit verschiedenen Aspekten der Fantasy und/oder der Science-Fiction angereichert werden. Historisch gewachsen ist speziell das Genre des *Cyberpunk* aus dem gleichnamigen Literaturtrend der 1980er Jahre, bei denen eine von Technologie durchdrungene Zukunft von *„großen, amoralischen Konzernen, Mensch/Computer-Neuronalen Schnittstellen und Gewalt“*[314] beherrscht wird. [Schick, 1991, S. 285]

14.2.1. Shadowrun

Der populärste[315] Vertreter der Cyberpunk-Pen-&-Paper-Rollenspiele trägt den Namen *Shadowrun*. Shadowrun spielt im Jahr 2075[316] und verbindet die Genres *Fantasy* und *Cyberpunk*. Damit wird eine Koexistenz von Magie-Elementen und Fabelwesen neben ganz normalen Menschen möglich, welche sich ihrerseits beispielsweise mit Hilfe kybernetischer Implantate modifizieren können. Spielerinnen

[313] 2013, [Cuaron, 2013]
[314] *„vast, amoral corporations, human/computer neural interfaces, and violence“* [Schick, 1991, S. 285]
[315] vgl. [Schick, 1991, S. 285]
[316] Zum Zeitpunkt dieser Arbeit bei Erscheinen des fünften Regelbuchs [Catalyst Game Labs, 2013b, S. 1].

und Spieler können hierbei zwischen verschiedenen *Meta-Rollen* wählen, in denen die eigene Rasse[317] festgelegt wird und fortan im Spiel als Grundlage für den eigenen Charakter dient [Catalyst Game Labs, 2013b, S. 3]. Städte sind zu so genannten *Megaplexen* oder *Metroplexen* zusammengewachsen, in denen die reiche, sehr kleine Oberschicht über eine große Gruppe von Armen herrscht (vgl. Abbildung 14.3) [Cross u. a., 2005, S. 187][Mulvihill, 2014, S. 9]. Beherrscht wird die Gesellschaft von sogenannten *Megacorporations*, die aufgrund ihrer enormen Größe wesentlich mehr Einfluss auf die Weltpolitik nehmen als Staaten oder Regierungen [Mulvihill, 2014, S. 17]. Die übrige Welt ist größtenteils zerstört und verwildert [Mulvihill, 2014, S. 8ff.].

Abbildung 14.3.: Die Abbildung zeigt die mögliche Vorstellung einer als *Megaplex* bezeichneten Stadt aus *Shadowrun*.

aus [Catalyst Game Labs, 2013a, S. 11]

Das Regelwerk berücksichtigt nicht nur die Interaktion mit der Welt, deren Beschreibung und das Verhalten in Kämpfen oder Aufgaben, sondern auch die Wechselwirkung mit der Gesellschaft sowie soziale Hintergründe [Schick, 1991, S. 285]. Spielerinnen und Spieler schlüpfen hierbei in die Rolle von so genannten *Shadowrunnern*, von *Megacorporations* angeheuerte Spezialisten, die offiziell nicht existieren und Aufträge ausführen, um sich selbst oder dem Auftraggeber einen Vorteil zu verschaffen [Mulvihill, 2014, S. 9ff.]. Bei der Eröffnung respektive dem Beitritt zu einem Shadowrun-Erzählstrang kann hierbei auf sogenannte Charakter-*Archetypen* zurück gegriffen werden, die das ansonsten teils langwierige Verfahren einer Spielcharakter-Individualisierung erleichtern. Dies ist auch im Sinne des im Rahmen dieser Arbeit umzusetzenden Systems sinnvoll, bei welchem die Spielerinnen und Spieler zwischen ebendiesen vorgefertigten Klassen wählen können, um den Fokus auf die Kernaspekte der Ausarbeitung zu legen. Bei den vorgefertigten Klassen handelt es sich um:

Decker: Prinzipiell Computerspezialisten, ein anderer Ausdruck für *Hacker*.

[317] Mensch, Elf, Zwerk, Ork oder Troll [Catalyst Game Labs, 2013b, S. 3]

Street Samurai: Physisch durch Implantate verbesserte Nahkämpfer.

Rigger: Mechaniker, die Fahrzeuge und Maschinen zu ihren Gunsten manipulieren können.

Magier: Magische Energie ist in die Welt von Shadowrun zurückgekehrt. Magier können diese Energie für sich nutzen.

[Mulvihill, 2014, S. 10]

Ebenso existiert eine Zeitleiste an Ereignissen, die in der Welt von Shadowrun aufgetreten sind [Wikia.com, 2014b]. Um diese Geschichtsdaten[318] herum lässt sich eine konkrete Handlung konstruieren, welche jedoch nicht mit der Gesamthandlung des Universums interferieren darf. Vergleichbar sind solche Ereignisse mit echten geschichtlichen Daten: Der Absturz des LZ 129 „Hindenburg“ im Jahre 1937 wäre ein realweltliches Beispiel dafür [Waibel, 2012, S. 109]. Dieses reale Ereignis könnte als Anhaltspunkt für eine konstruierte Geschichte dienen, zum Beispiel in der Form, dass die Spieler und Spielerinnen Teil eines Reporter-Teams sind, welche das Ereignis dokumentieren, im Verlauf der Arbeit als Reporter jedoch ins Geschehen eingreifen und Verletzte retten und am Ende als Helden dastehen. So lange die Handlung eines solchen Ereignisses nicht mit der geschichtlichen Integrität der Welt kollidiert, sind derartige Konstrukte zulässig. Die Zeitliste unterstützt den modularen Ansatz des hier entwickelten Systems, da damit sowohl Anhaltspunkte als auch Freiheiten existieren, eine konkrete Handlung in einem größeren Kontext anzusiedeln. Somit ist der Anspruch der Digitalisierung eines Pen-&-Paper-Rollenspiels hinsichtlich der Hintergründe gewährleistet.

14.2.2. Hintergrundgeschichte der Implementierung

Die konkrete Handlung der Implementierung spielt im Jahr 2038. In diesem Jahr gibt es ein Ereignis in der Shadowrun-Zeitleiste, bei dem der Drache *Kaltenstein* den deutschen Hydrogen-Zeppelin *Lothar Späth* über dem Schwarzwald in Deutschland attackiert und zum Absturz bringt [Wikia.com, 2014b; FanPro, 2003].

Der Absturz dieses Zeppelin legt in einem unwegsamen Teil des Schwarzwalds eine bis dato unbekannte, unterirdische Halle (siehe Abbildung 14.4) frei, welche durch einen schnellen Spähtrupp aus der Luft nur spärlich eingesehen werden kann. Die *Saeder-Krupp Schwerindustriegesellschaft*, die größte Megacorporation in der Welt von Shadowrun mit Sitz in Essen[319], beauftragt daraufhin vier Shadowrunner mit den Erkundungen des freigelegten Bauwerks unter strikter Geheimhaltung. Es stellt sich heraus,

[318] sogenannte *Story-Hooks*
[319] vgl. [Immig u. Romer, 1993]

dass die Höhle Teil eines unterirdischen, sehr alten Komplexes ist, der aus verschiedenen Räumen besteht und mit alten Rätseln und Fallen bestückt ist. Was die Shadowrunner schließlich finden, ist ein sogenannter Incubus[320], ein altes vergessenes Wesen, mit ganz eigenen Absichten. An dieser Stelle endet das Spiel.

Abbildung 14.4.: Die Abbildung zeigt eine Echtzeitansicht der eingestürzten Höhle vom Anfang des Spiels.

14.3. Realisiertes Storytelling und Gameplay

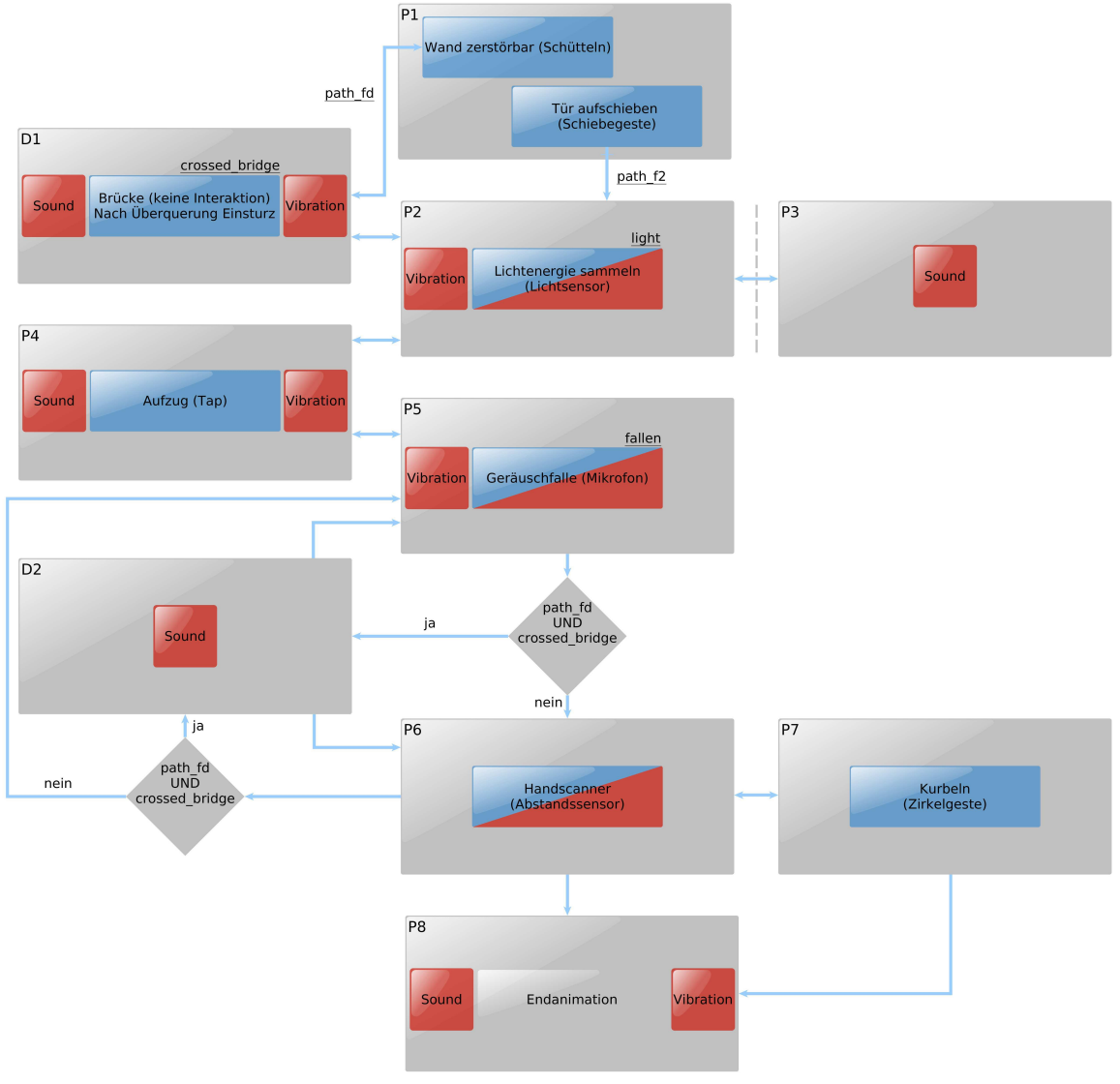
In Abschnitt 14.1 wurde erläutert, auf welche Art und Weise das Konzept des modularen Storytellings im Rahmen dieser Arbeit implementiert werden kann. Sowohl die gewählte Game-Engine[321] als auch die verwirklichte Rule-Based-Engine[322] erlaubt somit einen vollständig modularen Spielaufbau. Um die Zielsetzung der Arbeit mit der Realisierung der Anwendung im Rahmen einer vergleichenden

[320] vgl. [Wikia.com, 2014a]
[321] vgl. Kapitel 2.2.2 ab Seite 24
[322] vgl. Abschnitt 16 auf Seite 257

Evaluation überprüfen zu können, musste der konkrete Spielaufbau bestimmte Entscheidungen erzwingen. Dies stellt sicher, dass bestimmte Module auf jeden Fall immer und auch in angegebener Reihenfolge durchlaufen werden müssen, um an diesen Stellen die Auswirkungen von Mixed Reality-Aspekten zu evaluieren. In direkter Konsequenz ergibt sich dadurch, dass die Modularität aufgebrochen wird und feste Pfade entstehen, was wiederum darin resultiert, dass der realisierte Systemaufbau in eine Netzform zurückfällt.[323]

In der Vollbildgrafik 14.5 ist der realisierte Aufbau des Spiels und der damit zusammenhängenden Module und Interaktionen dargestellt.

[323] Mit anderen Worten: Ohne die Notwendigkeit einer Evaluation müssten auch keine Pfade erzwungen werden und das Endprodukt wäre vollständig modular, unter Zuhilfenahme der Verschachtlung, verwirklicht worden.


P1
Wand zerstörbar (Schütteln)
Tür aufschieben (Schiebegeste)
path_fd
path_f2
D1
crossed_bridge
Sound
Brücke (keine Interaktion) Nach Überquerung Einsturz
Vibration
P2
light
Vibration
Lichtenergie sammeln (Lichtsensor)
P3
Sound
P4
Sound
Aufzug (Tap)
Vibration
P5
fallen
Vibration
Geräuschfalle (Mikrofon)
D2
Sound
ja
path_fd UND crossed_bridge
nein
ja
nein
path_fd UND crossed_bridge
P6
Handscanner (Abstandssensor)
P7
Kurbeln (Zirkelgeste)
P8
Sound
Endanimation
Vibration

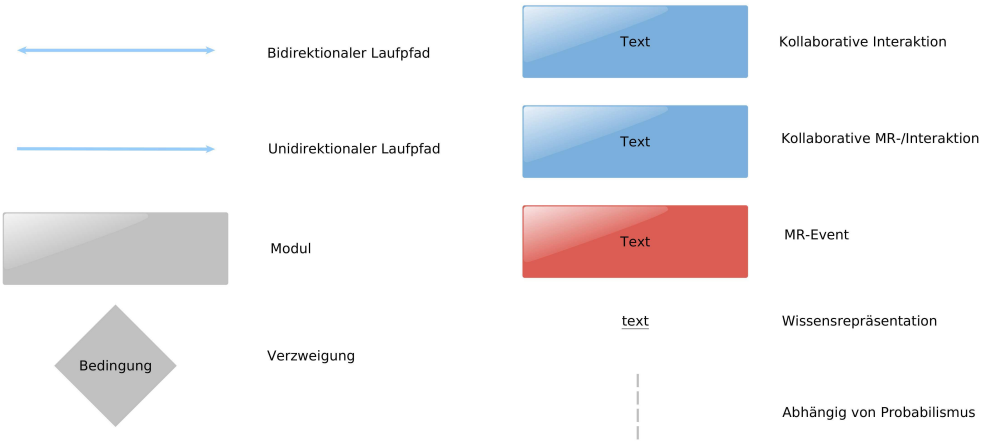

Bidirektionaler Laufpfad
Unidirektionaler Laufpfad
Modul
Bedingung
Verzweigung
Text
Kollaborative Interaktion
Text
Kollaborative MR-/Interaktion
Text
MR-Event
text
Wissensrepräsentation
Abhängig von Probabilismus

Abbildung 14.5.: Die Vollbildgrafik zeigt den Aufbau des in dieser Arbeit verwendeten Erzählkonstrukts. Der konkrete Aufbau und Verlauf ist in Abschnitt 14.3 ausführlich erläutert.

Bei der dargestellten Struktur handelt es sich um eine Variante der Netzstruktur. Dabei wurde ein modularer Aufbau aus insgesamt zehn Modulen als Grundlage genommen. Da es Ziel der Arbeit ist, die reale Interaktion der Partizipierenden zu fördern, muss es auch Gegenstand einer vergleichenden Evaluation[324] werden, ob Mixed Reality-Aspekte tatsächlich vermehrt soziale Interaktionen zwischen den Beteiligten begünstigen. Letztlich schlägt sich dies, wie bereits erwähnt, direkt in der Struktur des realisierten Spiels nieder, welches an bestimmten Stellen Nadelöhre erzwingt, an denen bestimmte Module in vorgegebener Reihenfolge durchlaufen werden müssen. Auf diese Weise konnten die Module P2, P5 und P6 so konzipiert werden, dass die dort vorhandenen Interaktionen einmal mit und einmal ohne Mixed Reality-Aspekte versehen wurden, um eine direkte Vergleichbarkeit des Einflusses ebendieser Aspekte zu ermöglichen[325].

Bei den Modulen handelt es sich gleichsam um Räume[326] die durchlaufen werden und jeweils mit einem Erzählmodul verbunden sind. Modul P3 stellt hier eine Besonderheit dar: Die gestrichelte Linie vor diesem Modul deutet an, dass es mit einem Zufall verbunden ist, ob dieses Modul betreten werden kann oder nicht, beziehungsweise ob der Versuch, die Tür zu diesem Raum zu öffnen, tatsächlich auch gelingt. Hiermit wird ein weiteres Element des Pen-&-Paper-Rollenspiels integriert: Glück. Selbst, wenn alles „richtig“ gemacht wird, kann das Glück respektive der Zufall immer noch eine Rolle spielen und den Plan der Gruppe oder des Individuums zunichte machen[327].

Die Reihenfolge, in welcher die Räume durchlaufen werden, bestimmt ebenfalls die Reihenfolge und den Ablauf der erzählten Geschichte. Hierbei wurde ein nichtlinearer Aufbau realisiert: Es können Räume ausgelassen werden, Zirkel sind möglich und die Bewegung in mehr als eine Richtung ist zulässig. Auf diese Weise wird auch der Wiederspielwert gewährleistet, da in einem Durchlauf niemals das ganze Potential des Spiels ausgereizt – und auch nicht die gesamte Geschichte erlebt wird. Rote Elemente stehen hierbei für Mixed Reality-Elemente oder Mixed Reality-Interaktionen, während blaue Elemente Interaktionen ohne Mixed Reality-Aspekte darstellen. Einige ausgewählte Interaktionsformen werden dabei im folgenden Abschnitt vorgestellt.

[324] vgl. Abschnitt 24.3 auf Seite 328

[325] zu sehen an der diagonalen Unterteilung dieser Module in rote und blue Bereiche

[326] Bis auf eine Ausnahme: Modul D2 beinhaltet lediglich ein Story-Element, welches zwar mit einer Veränderung der Spielwelt verbunden ist, nicht jedoch mit einem „Raum“ im Wortsinn. Herbei handelt es sich um eine Brücke, die abstürzt, sobald sie überquert wurde. An späterer Stelle im Spiel taucht diese Brücke dann wieder als Hindernis auf. Alternativ kann die Brücke auch nicht überquert werden, dann stürzt diese nicht ab und ist später auch nicht als Hindernis vorhanden.

[327] In realen Pen-&-Paper-Rollenspielen wird ein solcher Zufall tatsächlich ausgewürfelt und ist meist obligatorischer Teil vieler Regelwerke.

14.3.1. Interaktionsformen

Bei der Implementierung der Interaktionsformen wurde die Sensoren- und Aktorenvielfalt der Smartphones möglichst vollumfänglich ausgenutzt, um einerseits die Diversität der Interaktionen und damit deren Attraktivität und andererseits eine Feedbackschleife im Sinne einer ausgeweiteten Realität zu ermöglichen.

Da der Fokus dieser Abschlussarbeit besonders auf dem MR-Aspekt liegt, werden im Folgenden lediglich die in diese Kategorie fallenden Formen näher betrachtet. Die Verwendung des Lage- oder Beschleunigungssensors sowie des Gyroskops stellt im Sinne der hier eingeführten Definition[328] keine Mixed Reality-Interaktion dar, da es sich lediglich um eine Funktion des Eingabegeräts handelt und nicht von der Umgebung direkt abhängt[329]. Die Übertragung der Umgebungshelligkeit sowie -lautstärke in und aus der virtuellen Welt ermöglichen hingegen eine echte Ausweitung des Raumes bzw. das Aufspannen eines Volumens nach Definition[330].

Aus diesem Grunde wurde in der ersten, hier näher betrachteten Interaktionsform das Mikrofon der Smartphones als zentrales Element verwendet (vgl. Abbildung 14.5 Element P5). Die den Spielern und Spielerinnen vorgesetzte Situation erfordert ein Schleichen, da ein lautes Geräusch ausreicht, um eine verborgene Falle zu aktivieren. Zu diesem Zweck wird die reale Umgebungslautstärke, für die Spieler nicht erkenntlich, mittels der Smartphones aufgezeichnet und in die Spielwelt zurückgeführt. Die kollaborative Tätigkeit besteht dabei im gemeinsamen Schweigen bzw. der Anpassung der Gesprächslautstärke. Das Überschreiten eines vorgegebenen Schwellenwertes durch eines der Smartphone-Mikrophone ist an das Auslösen eines Spielevents, in diesem Fall der verborgenen Fallen, gebunden. Dadurch wird die im Spiel verwendete Metapher des Schleichens um eine reale, in diesem Fall akustische Komponente erweitert, welche unabhängig von den Spielenden auch durch sonstige Umgebungsgeräusche ausgelöst werden kann.

In Abschnitt 19.2 wird die Umkehrrichtung beschrieben, in der Geräusche der virtuellen Welt in die reale Welt[331] übertragen werden. Durch die Verknüpfung der hier entwickelten Gesamtanwendung mit einem analogen Mehrkanal-Tonsystem ist es zudem möglich, den Spielern durch die Verwendung von Umgebungsgeräuschen der virtuellen Welt ein noch tieferes Gefühl der Immersion zu schaffen.

Die zweite Interaktionform verlagert die Lösung des Problems einer unzureichenden Energieversorgung von der virtuellen in die reale Welt (vgl. Abbildung 14.5 Element P2). Um einen elektronisch gesteuerten Aufzug innerhalb der Spielwelt verwenden zu können, wird Energie benötigt. Für das

[328] vgl. Abschnitt 13.3 auf Seite 232
[329] Eine weitere Überlegung wäre hier die Höhe der Beschleunigung mit in die virtuelle Welt einfließen zu lassen.
[330] vgl. Abschnitt 13.3 auf Seite 232
[331] genauer gesagt zum Spieler hin

Bereitstellen derselben muss eine photovoltaische Vorrichtung aktiviert werden. Da sich die eigentliche Anlage unbeweglich an einer schattigen Stelle befindet, muss das Licht an einer Stelle gesammelt werden, an welcher ein Lichtstrahl durch die Decke scheint. Das Licht kann an dieser Stelle von den Avataren über ihre speziellen, ebenfalls photovoltaisch aufladbaren Anzüge aufgenommen und anschließend in den Generator eingespeist werden. Diese Interaktion des Sammelns wird unter Verwendung des Lichtsensors der Smartphones in die reale Welt übertragen. Kollaborativ gilt es, die Lichtsensoren aller Smartphones ausreichend zu stimulieren, sodass der Aufzug mit genügend Energie versorgt werden kann, um im Spielverlauf voranschreiten zu können. Es handelt sich lediglich um eine unidirektionale Ausweitung des Raumes. Die Steuerung des Lichts in der realen Welt könnte allerdings ebenfalls zur Stimmungserzeugung und damit einem tieferen Gefühl der Immersion führen und ohne viel Aufwand in das bestehende System integriert werden.

Anhand dieser Interaktion kann auch gezeigt werden, wie Mixed Reality-Aspekte, vorbereitend für eine vergleichende Evaluation, substituiert werden können. In der Version ohne Mixed Reality-Komponente müssen die Teilnehmerinnen und Teilnehmer lediglich die Stelle mit dem Licht betreten und drei Sekunden warten. Danach ist das Licht automatisch gesammelt worden.

Abschließend wird der Näherungssensor[332] der Smartphones näher betrachtet (vgl. Abbildung 14.5 Element P6). Innerhalb des Spiels existiert eine Situation, in welcher die Avatare eine ihrer Hände gleichzeitig auf eine vorgesehene Flächen legen müssen, um dadurch eine Tür zu öffnen. Die Tätigkeit der Avatare wird mit dem Umschalten der Smartphones in die korrespondierende Activity[333] in die Realität verlagert. Die Spieler müssen ebenfalls gleichzeitig eine ihrer Hände auf ihr Smartphone legen, um die „Analyse“ zu beginnen, was durch den Näherungssensor des Smartphone ausgelesen wird. Mit dem erfolgreichen Abschluss öffnet sich innerhalb der Spielwelt eine Tür und die Gruppe kann das nächste Modul betreten.

Im folgenden Kapitel werden die Gedanken hinter der Interaktionskonzeption näher vorgestellt. Dabei liegt der Fokus neben einer exakteren Definition des Begriffs „Interaktion“ auf der Metapherhaftigkeit und Kopplung derselben mit der Spielwelt.

[332] Der Näherungssensor wird hauptsächlich in der Telefonansicht verwendet. Das Display wird beim Anlegen des Smartphones an das Ohr ausgeschaltet um mögliche Eingaben während des Telefonierens zu verhindern.

[333] In Abschnitt 2.5 werden die einzelnen Komponenten Androids, unter denen sich auch die Activity befindet, ausführlich erläutert.

„Indeed, the ability of the game to support real-time interaction is even more important to the design of the game than its ability to support real-time communication; interaction is the game."

(Lecky-Thompson [Lecky-Thompson, 2008, S. 60]*)*

KAPITEL 15

Interaktionskonzeption

Im Grundlagenteil dieser Arbeit wurde herausgearbeitet, dass sowohl verschiedene Abstufungen von Interaktionsarten existieren[334] als auch ein nicht scharf abgrenzbares Feld zwischen Kooperationen und Kollaborationen[335].

Das Feld der HCI bezieht sich oftmals auf die Interaktion zwischen einem Individuum und einem Computer(-System). Das Ziel dieser Arbeit sieht eine Förderung der realen, sozialen Interaktion und damit auch Kommunikation zwischen mehreren Personen vor, was unter anderem dadurch erreicht werden soll, dass diese Personen *gemeinsam* mit einem System interagieren. Hierfür müssen verschiedene Konditionen erfüllt werden. Zum einen muss nicht nur eine individuelle, sondern diesem Fall sogar eine gemeinschaftliche Interaktionsabsicht erzeugt werden, woraufhin Interaktionsmöglichkeiten im und mit dem System als solche erkannt werden. Zum anderen müssen die Ergebnisse der Interaktionen in eine neue Interaktionsabsicht münden, um auf diese Weise einen Handlungskreislauf zu erzeugen, welcher den Spielverlauf vorantreibt, indem auf ein größeres Hintergrundziel hingearbeitet wird. In der Art der Interaktionsdurchführung kann ebenfalls eine Unterscheidung in *kooperativ*, *kollaborativ* sowie *konkurrierend* stattfinden.

Die in Abschnitt 6.4 gemachten Definitionen beziehen sich auf eine Granularität, die im Kontext dieser Arbeit zu grobkörnig angesetzt ist, da diese die parallele Existenz mehrerer Ziele ermöglicht. Wird ein Spiel in seiner Gesamtheit betrachtet und ein übergeordnetes, gemeinsam zu erreichendes Ziel definiert, so handelt es sich um ein rein kollaboratives Spiel. Dies ist insbesondere der Fall, wenn die Spielmechanik einen Spieler am Weiterspielen hindert, sobald ein anderer Spieler ausscheidet[336]. Die

[334] siehe Kapitel 6 auf Seite 121ff

[335] siehe Abschnitt 6.4 auf Seite 134

[336] Im Kooperationsmodus des Spiels *Resident Evil 5* [Capcom, 2014] führt der Tod eines Spielers zum Scheitern des gesamten Spiels. Hingegen ist das Ausscheiden eines Spielers im Spiel *Alien Swarm* [Corporation, 2014] mit keinen weiteren Konsequenzen verbunden. Der/Die verbliebene/n Spieler können das Level unter den Augen der beobachtenden, ausgeschiedenen Spieler weiterspielen.

Existenz eines Gesamtziels fördert jedoch nicht explizit das zuvor genannte Ziel der sozialen Interaktion, da einzelne Hindernisse im Spiel auch individuell gelöst werden können. Da ein Ziel dieser Ausarbeitung jedoch darin besteht, die Nutzer des Systems zum Kommunizieren anzuregen und ein soziales Miteinander zu schaffen, müssen die Begriffe der Kollaboration und Kooperation auf die Ebene der reinen Interaktion heruntergebrochen und demzufolge neu definiert werden.

Es wird daher eine neue Definitionsebene eingeführt, in der sich auf zeitlich quantifizierbare Interaktionen beschränkt wird. Diese sollen mit den Attributen *kollaborativ* und *kooperativ* ausgestattet werden können. Im Folgenden sind diese Definitionen zu finden:

Eine Interaktion beschreibt die Änderung eines Systemzustandes durch das Aufbringen einer bestimmten Leistungsmenge durch eine oder mehrere Personen. Diese wird unter Verwendung von Smartphones in der realen Welt induziert, um gewünschte Ereignisse in der virtuellen Welt auszulösen. Mit Abschluss der Interaktion geht der Erhalt von virtuellen Rückmeldungen einher, welche in der realen Welt rezipiert werden.

Eine *kollaborative* Interaktion bezeichnet die **obligatorische** Zusammenarbeit mehrerer Personen an derselben Interaktion, mit der Absicht ein gemeinsames, gesetztes Ziel zu erreichen. Die benötigte Leistungsmenge **muss** in gleichen Teilen erbracht werden. Fehlende Einzelleistungen **sind nicht** durch Leistungen anderer Teilnehmer kompensierbar.

Eine *kooperative* Interaktion bezeichnet die **fakultative** Zusammenarbeit mehrerer Personen an derselben Aufgabe, mit der Absicht ein gemeinsames, gesetztes Ziel zu erreichen. Die benötigte Leistungsmenge **muss nicht** in gleichen Teilen erbracht werden. Fehlende Einzelleistungen **sind** durch Leistungen anderer Teilnehmer kompensierbar.

Die Unterscheidung zwischen *kollaborativ* und *kooperativ* wird hierbei im Hinblick auf die Kopplung und Verbundenheit der Spieler untereinander erreicht. Bei Kollaborationen handelt es sich um eine bidirektionale Abhängigkeit der Spieler untereinander. Ein Erreichen des Ziels ist nur möglich, wenn alle einen Teil zum Gesamten beitragen, wodurch alle voneinander abhängig sind. Kooperationen hingegen äußern sich durch eine unidirektionale oder sogar durch eine nicht existente Abhängigkeit. Das Erreichen eines Ziels kann beispielsweise durch Mehraufwand eines Einzelnen dennoch erreicht werden, wenn auch in längerer Zeit.

15.1. Kopplung und Metapherhaftigkeit von Interaktionen

Bei Pen-&-Paper-Rollenspielen handelt es sich per se bereits um ein gruppenorientiertes, gemeinsam spielbares Genre, dessen Grundlage die Kommunikation und soziale Interaktion innerhalb der Gruppe

bildet. Individuelle Ziele besitzen einen eher untergeordneten Stellenwert und werden zumeist nicht verfolgt. Eine möglichst genaue Abbildung auf diese Eigenschaften lässt sich durch die Verwendung von kollaborativen Interaktionen erreichen. Diese ähneln weitgehend dem Konzept der *eng gekoppelten Interaktionen* der Autoren Beznosyk u. a.[337], sodass die daraus resultierenden Folgen ebenfalls zutreffen. Diese Art der Interaktionen führt unter anderem zu erhöhter Kommunikation und zur Förderung des emotionalen Bezugs untereinander [Beznosyk u. a., 2012], was wiederum das soziale Gefüge stärkt und sich somit hervorragend für die Digitalisierung des Pen-&-Paper-Konzepts und die im Rahmen dieser Arbeit gesetzten Ziele eignet. Auch nach Korhonen u. Koivisto kommt Kollaborationen im Hinblick auf Kommunikation eine besondere Rolle zuteil, da diese häufig für Gesprächsstoff sorgen und somit eine Kommunikation direkt fördern [Korhonen u. Koivisto, 2007].

Ferner werden mit der Unterstützung verschiedener Rollen[338] und dem Lösen von gemeinsamen Aufgaben respektive dem Erreichen gemeinsamer Ziele weitere empfohlene Kriterien[339] eingesetzt, welche als Eigenschaften mit dem größten Einfluss auf Zusammenarbeit und Kommunikation gelten[340]. Zusätzlich zu dem Einsatz der hier definierten kollaborativen Interaktionen sollte demnach das gesetzte Ziel erreicht und eine soziale sowie kommunikative Spielgrundlage geschaffen werden.

Ebenfalls wurden die Designhinweise der Autoren Korhonen u. Koivisto für erfolgreiche Multiplayerspiele beachtet, was sich unter anderem in folgenden Punkten äußert: Das Spiel ermöglicht Kommunikation und es gibt Gründe zum Kommunizieren sowie die Unterstützung eines Gruppengefühls[341]. Speziell der letzte Punkt wird mit der ausschließlichen Verwendung von kollaborativen Interaktionen und der Herstellung eines Face-2-Face-Settings forciert.

Eine Besonderheit besteht zudem in der Art der Interaktionsdurchführung. Das verwendete Konzept sieht vor, dass die Interaktion von allen Teilnehmern und Teilnehmerinnen auf ihrem Interaktionsgerät einzeln, jedoch zeitlich betrachtet *gleichzeitig* durchgeführt werden muss[342]. Eine exakte Gleichzeitigkeit kann physikalisch schon nicht erreicht werden, sodass zusätzlich evaluiert werden muss, welcher zeitliche Rahmen zum einen tolerierbar und zum anderen erfüllbar ist. Begleitend zur durchgeführten Implementierung wurden diese Interaktionen getestet, was zu der Designentscheidung führte, die maximal erlaubte Zeitdifferenz auf eine Sekunde zu setzen, damit auch technisch nicht so versierte Spieler und Spielerinnen unterstützt werden. Das hier entwickelte Konzept ähnelt dem der *synchronisierten Partneraktionen* der Autoren Goh u. a., welches jedoch erst im Nachhinein ausfindig gemacht

[337] Zu finden ist dieses in [Beznosyk u. a., 2012] oder kurz zusammengefasst in Abschnitt 6.4 auf Seite 134 dieser Ausarbeitung.
[338] im Original als „Ergänzungsprinzip“ beschrieben [Seif El-Nasr u. a., 2010]
[339] Die Autoren Seif El-Nasr u. a. haben verschiedene Entwurfsmuster herausgearbeitet, welche für das Entwickeln eines kollaborativen Spiels besonders wichtig sind [Seif El-Nasr u. a., 2010].
[340] siehe Abschnitt 6.4 auf Seite 134
[341] Die vollständige Auflistung ist auf Seite 137 zu finden.
[342] Ausführlicher wird die technische Grundlage dabei in Abschnitt 19.1 auf Seite 277 erläutert.

wurde [Goh u. a., 2011]. In diesem Forschungsprojekt wurde festgestellt, dass diese Art der Interaktionsdurchführung einen erheblichen Einfluss auf kollaboratives Lernen besitzt, sogar so weit, dass ein Lerneffekt festgestellt worden ist. Insbesondere bei Kindern mit Down-Syndrom konnte ein „Abgucken“ der Handlung von anderen Spieler festgestellt werden [Goh u. a., 2011]. Hilfreich erscheint es im Allgemeinen jedoch für alle schwächeren Spieler, da diese das Vorgehen anderer durch einfaches Abgucken verinnerlichen und daraufhin nachahmen können, um somit das gemeinsame Ziel zu erreichen.

Für die konkrete Wahl der einzelnen Interaktionen wurde versucht, die in Abschnitt 6.1 beschriebene Metapherhaftigkeit einzusetzen[343]. Das folgende Beispiel soll hierbei als nähere Erläuterung dienen. Für das Aufschieben einer virtuellen Tür[344] soll versucht werden eine entsprechende Metapher zu finden, welche das Vorgehen intuitiv und verständlich abbildet. Da als Interaktionsgeräte Smartphones eingesetzt werden und eine Steuerung über Gesten die Norm bildet, bietet sich die Verwendung einer solchen als Übertragung an. Durch eine Wischgeste auf dem Smartphone wird das Aufschieben einer virtuellen Tür simuliert. Hierbei handelt es sich zwar nicht um eine Metapher im engeren Sinne, jedoch wird die geistige Brücke zwischen „eine Schiebtür muss aufgeschoben werden“ und „um die Schiebetür im Spiel aufzuschieben, muss über das Display gewischt werden“ erleichtert. Insbesondere dadurch, dass Wischgesten bei der Nutzung von Smartphones beispielsweise auch als Metapher für das Blättern einer virtuellen Zeitung verwendet werden, ist eine Verständnis für diese Übertragungsfunktion bereits gegeben und wirkt für die Spieler vertraut[345]. Es wird versucht, das erlernte Verhalten in der virtuellen Welt beizubehalten und optimal zu übertragen, damit das Eingabegerät nicht als zusätzliche Barriere, sondern möglichst als unterstützendes Medium wahrgenommen wird. Die Wahl von Smartphones und Verwendung üblicher Gesten sowie Steuerungskonzepte wird auch durch folgendes Zitat bestätigt:

> *„Touch and physical interaction are among the fundamental ways in which humans understand the world and effect changes in it. Haptic feedback therefore offers an advanced interaction concept for games.“*

[Faust u. ho Yoo, 2006, S. 1]

Haptisches Feedback ist darüber hinaus sehr wichtig für Spiele, da dadurch der taktile Sinn der Menschen angesprochen-, eine Berührung virtueller Objekte realisiert- und der Realismusgrad gesteigert

[343] Konkrete Beispiele zu Mixed Reality-Interaktionen sind ebenfalls in Abschnitt 14.3.1 auf Seite 248 zu finden.
[344] vgl. Abschnitt 5.2.1 und Abbildung 5.3 auf Seite 109
[345] In einem aktuellen Forschungsprojekt wurde sogar herausgefunden, dass die heutige Touchscreen-Bedienung frühkindlichen Bewegungsmustern entspricht [Wilkens, 2013]

wird. Durch die Nutzung des Vibrators der Smartphones wird zwar kein komplexes haptisches Feedback ermöglicht, jedoch reicht dies aus, um die Perfomanz der Schnittstelle zu erhöhen. [Faust u. ho Yoo, 2006, S. 8]

Sowohl Interaktionen als auch sonstige relevante Events werden in der Spielwelt platziert und durch das Betreten von Avataren ausgelöst[346], hierbei handelt es sich um einen Teilaspekt der künstlichen Intelligenz. Bei einigen der Events handelt es sich um Story-Entscheidungen, welche kurz- oder langfristig zu Konsequenzen der erlebten Geschichte oder innerhalb der Spielwelt führen sollen, wofür ein optimaler Entscheidungsfindungsansatz gefunden werden muss. Diese und weitere KI-Faktoren werden daher im Folgenden genauer behandelt.

[346] Erläutert wird dieses entwickelte Konzept in Abschnitt 18.2 auf Seite 273.

„Eine Künstliche Intelligenz fällt meistens nur dann auf, wenn sie nicht richtig funktioniert."

(Thomas Stein, technischer Direktor bei Related Designs und Chef-Programmierer des Aufbau-Strategiespiels Anno 1701 [Klinge, 2008]*)*

KAPITEL 16

Konzeption der künstlichen Intelligenz

Wie im korrespondierenden Grundlagenkapitel[347] bereits erörtert wurde, übernimmt eine künstliche Intelligenz in Spielen mehrere Aufgaben. Im Rahmen der hier entwickelten Anwendung betrifft dies neben der Digitalisierung des Game Masters und der damit einhergehenden automatischen Entscheidungsfindung ebenfalls die gesamte Intelligenz in der Spielwelt, die Kamera-Intelligenz mit eingeschlossen. Im Folgenden werden daher die getroffenen Designentscheidungen begründet.

16.1. Wahl des passenden Entscheidungsfindungsansatzes

Die Wahl eines passenden Entscheidungsfindungsansatzes[348] für den zu digitalisierenden Game Master gilt als eine der größten Herausforderungen dieser Ausarbeitung. Grundlegend gilt es, für diesen ein Event-basiertes Verfahren einzusetzen, da das Spielverhalten sowie die durchgeführten Aktionen der Spieler im Voraus nicht vorhersehbar sind und zudem kein exaktes Ziel existiert. Ein zielorientiertes System stellt sich daher als vernachlässigbar und ungeeignet heraus.

Unter dem Gesichtspunkt des zur Verfügung stehenden Probabilismus stellen sich nichtdeterministische Ansätze prinzipiell als richtige Wahl heraus, da diese einem der wichtigsten Elemente in Pen-&-Paper-Rollenspielen entsprechen: dem Würfel. Ebenso variieren die Spielverläufe in Pen-&-Paper-Rollenspielen, sodass kein Spiel mehrmals exakt gleich verläuft, was ebenfalls einem nichtdeterministischen Schema entspricht. In der zur Verfügung stehenden Ausarbeitungszeit lassen sich jedoch nicht ausreichend viele Interaktions- sowie Entscheidungssituationen und Storymodule implementieren, welche die Vorteile eines nichtdeterministischen Algorithmus hinreichend ausnutzen würden.

[347] siehe Kapitel 9 ab Seite 175
[348] In Abschnitt 9.3 auf Seite 178 wurden verschiedene Ansätze zur Entscheidungsfindung vorgestellt.

Darüber hinaus ist die Tatsache, dass nichtdeterministische Ansätze sich lediglich simulieren, nicht jedoch implementieren lassen, mit weiteren Restriktionen und Besonderheiten verknüpft, was unter anderem zu schlechter Test- und Wartbarkeit führen kann. Aus den genannten Gründen wurden nichtdeterministische Algorithmen nicht weiter betrachtet. Die Zufallskomponente soll jedoch nicht vernachlässigt werden und wird mit Hilfe anderer Methoden Einzug erhalten (siehe Abschnitt 19.1). Im Folgenden werden daher die einzelnen Gesichtspunkte zugunsten respektive zuungunsten deterministischer Algorithmen erläutert.

Skripte sind statischer Natur und tendieren dazu, sehr lang und komplex zu werden[349]. Dies führt nach Spronck u. a. im Allgemeinen zu zwei Problemen:

1. Wegen der vergleichsweise hohen Komplexität ist die Wahrscheinlichkeit von vorhandenen Schwächen höher.[350]

2. Die statische Natur führt dazu, dass nicht auf unvorhergesehene Spieltaktiken respektive -aktionen reagiert werden kann.

[Spronck u. a., 2006, S. 2]

Besonders der zweite Punkt ist ausschlaggebend, um für die Realisierung eines nahezu offenen Spielerverhaltens im Rahmen von Pen-&-Paper-Rollenspielen keinen Script-Ansatz zu wählen. Zudem ist keine langfristige Entscheidungsfindung möglich, da lediglich ein proaktives Handeln stattfindet und kein Zustandsspeicher existiert.

Endliche Automaten führen zur Problematik, dass das Verhalten für Spieler vorhersagbar wird und eine Entscheidungsfindung auf Basis zeitlich weit zurückliegender Geschehnisse sehr schwer abbildbar ist[351]. Zustände können zwar genutzt werden um Entscheidungen und Aktionen zwischenzuspeichern, jedoch ist es damit zwingend notwendig für jede mögliche Konstellation von Spieldurchläufen einen eigenen Zustand anzulegen und Überführungen zwischen all diesen Zuständen vorzunehmen. Die Zustandsmenge entspricht damit der Fakultät aller vorhandenen Entscheidungsmöglichkeiten, sodass häufig von einer „Explosion des Zustandsraumes“ gesprochen wird. Damit wird die Möglichkeit einer gut wartbaren und simple modifizierbaren Anwendung ausgeschlossen. Ferner wäre der initiale Aufwand für eine solche Konstruktion sehr hoch.[352]

[349] Bei der Verwendung einer prozeduralen Programmierweise, wird dies auch *Spaghetticode* genannt [SourceMaking, 2013]. Siehe Abschnitt 9.3.1 auf Seite 179

[350] Dies ist in Spielen zusätzlich problematisch, da diese Schwächen ausgenutzt werden können, um NPCs zu schlagen oder umgehen.

[351] siehe Abschnitt 9.3.2 auf Seite 180

[352] Eine ähnliche Problematik wurde bereits in Abschnitt 14.1.1 auf Seite 237 im Zusammenhang mit dem Konzept des modularen Storytellings erläutert.

Entscheidungsbäume führen zur selben Problematik. Sie bilden zusammenhängende Entscheidungen respektive zusammenhängendes Verhalten ab[353]. Da ein Baum hierarchisch aufgebaut ist, wird lediglich der aktuelle Knoten und davon abgehende Kindknoten betrachtet. Damit eine möglichst große Entscheidungsfreiheit gewährleistet werden kann, ist es notwendig, von jedem Knoten aus neue Teilbäume zu verwenden, welche jeweils wiederum alle Möglichkeiten enthalten und anbieten. Viel eher eignet sich ein Entscheidungsbaum für die Realisierung eines Charakterverhaltens, da dieses meistens in Phasen wie Ruhe, Angriff und Flucht unterteilt ist, wobei jeweils nur eine Phasen-Überführung in eine jeweilige andere existiert [Millington u. Funge, 2009, S. 306]. Ferner besteht ein weiteres Problem von Entscheidungsbäumen darin, dass eine kleine Änderung des gewünschten Verhaltens bzw. der zur Verfügung stehenden Möglichkeiten enorme Auswirkungen auf den gesamten Baum haben und dieser im schlimmsten Fall vollkommen neu konstruiert werden müsste [Lau, 2011].

Fuzzy-Logik findet am häufigsten dort einen Einsatz, wo Entscheidungen auf Basis von subjektiven, unpräzisen oder nur sehr vage formulierten Informationen getroffen werden sollen[354] [Bourg, 2004, S. 190]. Mit diesem Ansatz kann die duale Einschränkung dadurch aufgehoben werden, dass eine Aussage nicht ausschließlich einem klassischen Wahrheitswert entspricht, sondern ein Grad der Zugehörigkeit angegeben werden kann[355]. Die Aktionen der Spieler resultieren, zumindest in der hier angestrebten Digitalisierung des Pen-&-Paper-Konzepts, jedoch in quantisierbaren Ergebnissen, sodass dieser Ansatz ebenfalls ausscheidet.

Eines der ausschlaggebenden Argumente für die Wahl eines **regelbasierten Systems**[356] war folgende Aussage von Bourg:

> *„One advantage of rule-based systems is that they mimic the way people tend to think and reason given a set of known facts and their knowledge about the particular problem domain.“*

[Bourg, 2004, S. 212]

Für den zu digitalisierenden Game Master, dessen Rolle im originalen Pen-&-Paper-Konzept ebenfalls von einem menschlichen Spieler wahrgenommen wird, ist diese Art der künstlichen Intelligenz geradezu prädestiniert. Der Ablauf des Spiels soll aufgrund des gewählten modularen Ansatzes[357] nicht

[353] siehe Abschnitt 9.3.3 auf Seite 182
[354] siehe Abschnitt 9.3.4 auf Seite 183
[355] Normalerweise existieren feste Intervalle in denen eine Abfrage entweder wahr oder falsch ergibt. Mit dem Einsatz der Fuzzy-Logik ist es jedoch möglich, dass ein Ergebnis z.B. lediglich 50% wahr ist. Anders ausgedrückt: Werden den beiden klassischen Ausgangswerten die Zahlen 0 und 1 zugewiesen, so sind nun auch Ergebnisse zwischen diesen beiden absoluten Zahlen möglich.
[356] siehe Abschnitt 9.3.5 auf Seite 186
[357] Konkret handelt es sich um eine Netzform. Eine ausführliche Entscheidungsbegründung und damit weitergehende Informationen sind in Kapitel 14 ab Seite 235 zu finden.

vorgegeben sein, sodass eine adaptive Entscheidungsfindung auf Basis zugrundeliegender Informationen benötigt wird. Dies kann mit Hilfe eines regelbasierten Systems umgesetzt werden, wodurch sich zusätzlich die Möglichkeit ergibt, dass nicht gezwungenermaßen simple *if-then*-Konstrukte verwendet werden müssen. Hinzu kommt die Tatsache, dass die Bedingungen auch komplexe Vergleiche, z.B. auf Basis von Zeichenketten, beinhalten können. Insbesondere das Verwenden von Strings ermöglicht laut Bourg das Auslagern der KI, sodass auch Nicht-Programmierer neue Regeln anlegen und bestehende Systeme verwalten können [Bourg, 2004, S. 226]. Ausreichend viele Skriptsprachen stehen hierfür mit Prolog[358] und Lisp[359] bereits seit Jahren zur Verfügung [Bourg, 2004, S. 226]. Somit ist es ebenfalls möglich, sowohl die Regeln als auch die Wissensbasis ohne Anpassungen der Spiellogik modifizieren zu können, wodurch die gesamte Wartbarkeit der Anwendung gesteigert wird [Bourg, 2004, S. 214].

Der mit Abstand größte Vorteil ist jedoch der Folgende: Es werden weder Pfade vorgegeben, noch werden die Entscheidungsmöglichkeiten für den Spieler erkennbar eingeschränkt. In der Konfliktresolution kann ein Ansatz verwendet werden, welcher alle eintreffenden Regeln gleichzeitig ausführt, sodass auf unvorhergesehene Ereignisse und Kombinationen reagiert werden kann. Befindet sich ein Spieler beispielsweise in einem bestimmten Modul, so würde dies bei der Verwendung eines endlichen Automaten oder Entscheidungsbaums bedeuten, dass sich das System in einem bestimmten Zustand oder Knoten befindet und von dort aus nur eine bestimmte Anzahl an Übergängen definiert ist. Aufgrund der Unabhängigkeit der Regeln voneinander kann der Spieler beispielsweise den Rückweg antreten und zuvor besuchte Module genauer untersuchen, was dazu führen kann, dass sich neue Möglichkeiten ergeben. Dies äußert sich bei der Verwendung eines regelbasierten System dahingehend, dass eine andere Regel wahr wird und deren Konsequenz ausgeführt wird. Mehrere Regeln treffen in diesem Szenario gleichzeitig zu und werden bearbeitet. Somit kann von unvorhergesehenem bis zu eigenartigem Spielverhalten auf nahezu jede Gegebenheit ordnungsgemäß reagiert werden.

Im Rahmen dieser Abschlussarbeit wurde ein eigenes, regelbasiertes System auf Basis von Java implementiert, da vorhandene Frameworks und Lösungen zu komplex und unflexibel implementiert sind. Die häufigsten Probleme waren, dass notwendige Funktionen kaum bis gar nicht vorgesehen waren. Essentiell waren dabei die folgenden zwei Punkte: Das Ändern der Wissensbasis von außerhalb zum einen und das Ausführen von individuellen Aktionen zum anderen, wobei letztere nicht ausschließlich das Finden neuer Regeln und Fakten oder die Modifizierung der bestehenden Wissensbasis beinhalteten. Eine ausführlichere Betrachtung der Implementierung und vorangegangenen Problematik lässt sich in Abschnitt 19.3.2 finden.

[358] *Prolog* gilt als die wichtigste logische Programmiersprache [Kastens, 2011] und beeinflusste viele Entwicklungen in der Informatik.

[359] Lisp gilt als eine der ältesten Programmiersprachen und wird insbesondere für symbolische Berechnungen verwendet. Ebenfalls besitzt die diese eine lange Tradition im Bereich der künstlichen Intelligenz [von Henke, 2009].

Um Kollaborationen als solche zu ermöglichen ist es wichtig, eine geeignete Kameralogik zu implementieren, da diese als Hauptfaktor kollaborativer Handlungen gilt[360]. Insofern behandelt der folgende Abschnitt die Wahl eines passenden Kamerakonzepts und die damit einhergehenden Möglichkeiten sowie Restriktionen.

16.2. Konzeption des Kamerasystems

Bei der Auswahl des Kamera-Typs war es wichtig, dass in der virtuellen Welt kein separierender Raum entsteht, um die Spielenden in einer gemeinsamen Interaktionssphäre zu halten. Viele Spielprinzipien und Umsetzungen, insbesondere im Bereich der Konsolenspiele, nutzen den Effekt der räumlichen Separation bewusst, um den Spielenden eine gewisse Autarkie einzuräumen. So ist es beispielsweise bei Rennspielen durchaus hilfreich, wenn jede/r Spielende einen eigenen Bereich auf dem Ausgabemedium zugewiesen bekommt, um das eigene Fahrzeug unabhängig von der Position der jeweils anderen immer im Blick zu haben[361]. Auf der anderen Seite sorgt diese Independenz für einen sich selbst verstärkenden Effekt: Durch den Fokus auf den „eigenen“ Bildbereich ist, insbesondere bei steigendem Engagement, die Gefahr gegeben, dass sich die virtuelle Isolation ebenfalls auf die reale Interaktion auswirkt. Die Spielenden halten sich in diesem Fall zwar zusammen in einem Raum auf und spielen miteinander, sind in ihrer Wahrnehmung jedoch voneinander separiert.

Der zweite Leitgedanke bei der Auswahl des richtigen Kamera-Typs war die aktive Unterstützung des MR-Ansatzes bei gleichzeitiger Berücksichtigung des sozialen Ansatzes des Pen-&-Paper-Rollenspiels. Wie bereits erläutert wurde[362], sind Pen-&-Paper-Rollenspiele in ihrem Wesen demokratisch[363] organisierte Systeme. Dies bedeutet im Kontext der digitalisierten Anwendung, dass die Wahl des Kamera-Typs Mehrheitsentscheidungen folgt und nicht den Entscheidungen eines einzelnen Spielers, wie es beispielsweise passieren könnte, wenn ein Mitglied der Spielenden als „Leitfigur“ deklariert würde und die Kamera technisch den Bewegungen dieser einen Spielfigur folgte. Ein solches Prinzip ist bei-

[360] In einer Studie von den Autoren Seif El-Nasr u. a. wurde als hauptsächlicher Grund, welcher zum Scheitern eines Zusammenspielens führt, ein schlechtes Kameradesign genannt [Seif El-Nasr u. a., 2010].

[361] Dieses Verfahren ist auch unter dem Namen „Splitscreen“ bekannt. Teilweise wird sogar die Ansicht vertreten, dass Splitscreen-Spiele aus der Not heraus erfunden wurden, dass eine komfortables Zusammenspiel sonst anders nicht möglich war. Bevor sich das Internet in der breiten Masse durchsetzte, waren Splitsceen-Spiele oftmals mit die einzige Möglichkeit, ein gemeinsames Spielen zu realisieren. Allerdings unterstützen auch viele moderne Spiele diese Technik weiterhin. [TheKhoaNguyen, 2010].

[362] vgl. Abschnitt 2.1 ab Seite 14

[363] Alleingänge respektive egoistische Entscheidungen, die einem demokratischen Prinzip zuwiderlaufen, sind zwar grundsätzlich möglich, sofern diese im konkreten Fall nicht explizit durch das zugrundeliegende Regelwerk eingeschränkt werden, führend jedoch meist nicht zum Erfolg.

spielsweise im bereits erwähnten[364] PC-Spieletitel *Dragon Age: Origins*[365] zu finden, wo die Kamera sich immer nach der derzeit ausgewählten Spielfigur richtet[366].

Die Auswahl erfolgte daher durch selektive Betrachtung der in Kapitel 10 vorgestellten Alternativen. Während die *First-Person-Kamera*[367] den wohl detailliertesten Blick auf die Spielwelt ermöglicht, findet dieses Verfahren im Rahmen dieser Arbeit schon alleine deshalb keine Anwendung, da es dem Konzept einer kollaborativen und gleichberechtigten, demokratischen Spielgemeinschaft zuwiderläuft. Die aktuelle Perspektive ist in diesem Fall klar einem einzigen Avatar zugeordnet und der Blick auf die Welt würde alleine durch die Bewegungen eines einzigen Gruppenmitglieds gesteuert werden. Das gleiche Problem, wenn auch in etwas abgeschwächter Form, tritt bei der *Third-Person-Kamera*[368] auf: auch hier wird eine Spielfigur als Referenz definiert. Im Unterschied zur *First-Person-Kamera* ist die Kamera hier jedoch, je nach Implementierung, etwas weiter von der Gesamtszene entfernt und erlaubt die gleichzeitige Darstellung aller am Spiel beteiligten Figuren. Um die Kreation einer Leitfigur jedoch zu vermeiden, scheidet auch dieses Verfahren aus.

Eine *Top-Down-Ansicht*[369] ist prinzipiell gut dafür geeignet, mehrere Avatare gleichzeitig mit einer Kamera einzufangen. Aus heutiger Sicht betrachtet ist die Top-Down-Ansicht allerdings eher als technische Notwendigkeit entstanden (gegenüber einer bewussten Entscheidung) und tritt mit den Möglichkeiten moderner Computer- und Videospielkonsolen hinter anderen Darstellungsformen zurück. Leistungsschwache Geräte der vergangenen und heutigen Zeit können die Rechenleistung für die Berechnung von dreidimensionalen Umgebungen nicht oder nur eingeschränkt zur Verfügung stellen. Mit Hilfe der isometrischen Darstellung lässt sich der Eindruck einer Räumlichkeit aus der Vogelperspektive auch in 2D-Engines gut kaschieren, da dieser mit steigender Entfernung der Kamera zum Zielobjekt verloren geht[370]. Die Bedeutung der Top-Down-Ansicht in der modernen Spielentwicklung geht generell jedoch immer weiter zurück und wird stetig zunehmend von anderen Darstellungsformen ersetzt[371].

[364] vgl. Abschnitt 10.4
[365] 2009, *BioWare Corporation* [MobyGames.com, 2013d]
[366] vgl. [Matschijewsky, 2009; MobyGames.com, 2013d]
[367] vgl. Abschnitt 10.1 auf Seite 191
[368] vgl. Abschnitt 10.1 auf Seite 191
[369] vgl. Abschnitt 10.2 auf Seite 193
[370] Dieser Effekt ist gut nachvollziehbar an der *Pokémon*-Reihe, die langsam von einer (technisch bedingten) rein zweidimensionalen Erstversion bis hin zu einer echten 3D-Darstellung in der (bislang) letzten Generation evolviert. Für diesen Prozess wurden fünfzehn Jahre benötigt [Smith, 2010].
[371] Während im Jahre 1993 noch 511 Neuerscheinungen mit der Top-Down-Ansicht zu verzeichnen waren, sind im Jahre 2003 nur noch 366 Titel zu verzeichnen. Im Jahre 2013 beläuft sich die Zahl der Neuerscheinungen (zum Zeitpunkt dieser Arbeit) lediglich auf noch 130 Neuerscheinungen [MobyGames.com, 2013j,k,l] .

Ein Rollenspiel mit *Side-Scrolling-Kamera*[372] in Verbindung mit einer 3D-Engine umzusetzen ist zwar prinzipiell möglich, jedoch nicht zweckdienlich[373]. Die Mechanik des stark beschränkten Bildausschnitts versteckt kontinuierlich Teile der Spielwelt und konterkariert somit das Genre des Rollenspiels, bei des es darum geht, den Spielenden möglichst *viele* Informationen zur Verfügung zu stellen, um auf Basis dieser Informationen eine Entscheidung zu treffen und die erspielte Geschichte weiterzuentwickeln.

Abbildung 16.1.: Die Abbildung zeigt eine Szene der Konsolenversion von *Diablo 3*. Hier können bis zu vier Spielende gleichzeitig in einer isometrischen Perspektive auf einem gemeinsamen Bildschirm interagieren.

aus [Gaskill, 2013]

Zu guter Letzt ist die *isometrische Kameraperspektive* zu betrachten. Die Konsolenumsetzung der neusten Iteration der *Diablo*-Reihe beispielsweise verfolgt einen isometrischen Ansatz, bei dem die Kamera nicht an eine spezielle Spielfigur gebunden ist, sondern einer Gruppe von Spielenden gemeinsam folgt, die alle auf einem gemeinsamen Bildschirm dargestellt sind (vgl. Abbildung 16.1).

Wie bereits in Abschnitt 5.2.3 herausgearbeitet wurde[374], stellen Videospielkonsolen prinzipbedingt eine Plattform dar, die in ihrem Aufbau denen der Pen-&-Paper-Rollenspiele ähnelt respektive eine Analogie zu diesen ermöglicht. Bei diesem konkreten Spieletitel handelt es sich im Kern ebenfalls um

[372] vgl. Abschnitt 10.2 auf Seite 193
[373] Ausnahmen bestätigen die Regel: Ein Beispiel für ein Side-Scrolling-Rollenspiel ist mit [Chucklefish, 2014] zu finden.
[374] vgl. auch Abbildung 5.4 auf Seite 114

ein Rollenspiel, weshalb in diesem Fall auch auf Seiten des Spielprinzips alle Voraussetzungen für eine unterstützende Mechanik vorhanden sind: Die Konsole erlaubt durch ihren Aufbau die gleichzeitige Partizipation mehrerer Spielenden mit Blick auf ein gemeinsames Ausgabemedium. Die Spielwelt wird ebenfalls in einem gemeinsamen Kontext präsentiert, bei dem keine Separation stattfindet, während das gewählte Kamera-System sich demokratisch im Sinne eines Rollenspiels verhält und nicht einer speziellen Spielfigur zuordenbar ist. Die isometrische Kameraperspektive stellt somit eine moderne Erweiterung der Top-Down-Ansicht dar.

Fazit

Aus den hier angebrachten Gegenüberstellungen ergibt sich, dass für die Realisierung eines digitalen, dreidimensionalen Rollenspiels das isometrische Kamerasystem alle Voraussetzungen erfüllt und der Grundidee der anvisierten MR-Anwendung damit am nächsten kommt. Als ebenfalls wichtiger Faktor von Multiplayerspielen wird das Design des Netzwerks genannt, welches sämtliche Nebeneffekte den Spieler verbergen muss, sodass im Folgenden die Wahl des zugrundeliegenden Netzwerk- sowie Übertragungsprotokolls und zu beachtende Kennzahlen erläutert werden.

„One of the most enjoyable things you can do with a computer network is play connected games with friends and family."

(Bradley Mitchell [Mitchell, 2014]*)*

KAPITEL 17

Netzwerkkonzeption

Als Netzwerkprotokoll wurde bereits im vorangegangenen Projekt eine Entscheidung zugunsten von TCP getroffen. Es existieren zwei unterschiedliche Datenarten innerhalb der Anwendung: wichtige Daten, welche übertragen werden müssen (z.B. Spielentscheidungen) und unwichtige, welche verlorengehen können, da diese in kurzen Intervallen gesendet werden und veraltete Daten überschreiben (Bewegungsinformationen). Da TCP eine garantierte und sortierte Lieferung gewährleistet, stellte sich die Wahl von TCP als optimale Lösung heraus. Ferner wurde der Nagle-Algorithmus ausgeschaltet, welcher kleine Pakete vor dem Sendevorgang bis zur MTU zusammenfasst, um den ansonsten viel zu großen TCP-Header auszugleichen. Da in der hier konzipierten Anwendung sehr kurze Sendeintervalle und kleine Pakete vorhanden sind, stellte sich dies als die optimale Lösung heraus, um die Reaktionszeit erheblich zu verbessern.[375]

Bezüglich der Latenz ergeben sich im Allgemein besonders hohe Anforderungen, da die Spieler und Spielerinnen das Smartphone als Eingabegerät verwenden, die Ausgabe jedoch auf einem zentralen Display stattfindet. Die Übertragung der Eingaben zu dem System erfolgt drahtlos und stellt damit eine große Herausforderung an die Netzwerkimplementierung, da besonders bei responsiven Anwendungen ein geringer Zeitverlust angestrebt werden muss, um die Präsenz und das Spielgefühl (*Engagement*) aufrecht zu erhalten[376]. Generell gilt, dass das Ziel einer niedrigen Latenz bei der mobilen Entwicklung eine der größten Herausforderungen darstellt [Wang u. a., 2009]. Das Ergebnis, dass Spieler sehr tolerant auf Latenz und Jitter in Multiplayerspielen reagieren, muss in diesem Fall vernachlässigt werden, da keine Prädiktion der Eingaben erfolgt und somit die Latenz in direktem Zusammenhang mit den getätigten Eingaben steht [Fritsch u. a., 2005]. Die drahtlos angebundenen Smartphones sollten sich daher idealerweise wie ein lokales Eingabegerät verhalten, welches direkt und kabelgebunden an einen Computer angeschlossen ist. Obwohl es sich bei dem im Rahmen dieser Arbeit umgesetzten Projekt um ein Rollenspiel handelt, ähnelt die Handhabung mehr einem Actionspiel. Deshalb werden

[375] siehe Abschnitte 3.1.5f

[376] vgl. Abschnitt 24.1.1 auf Seite 320

für den weiteren Verlauf der Arbeit Latenzwerte für die Eingabe angestrebt, die im Action-Genre als Richtwerte dienen.

Wie auch in [Wang u. a., 2009] beschrieben, ist es nicht möglich, die Spielkonsistenz und das -verhalten durch Zahlen und statistische Analysen herauszufinden, sondern lediglich durch die Beurteilung der Spieler selbst. Aufgrund dessen werden die verwendeten Werte und Konfigurationen anhand persönlicher Tests ausgewählt und anschließend einer Evaluation unterzogen.

Wie in der Auflistung in Abschnitt 11.3.1 aufgezeigt wird, stellt sich TCP als empfohlenes und ausreichendes Netzwerkprotokoll für Rollenspiele mit Story- anstelle von Action-lastigen Elementen dar. Da es sich bei der in dieser Abschlussarbeit realisierten Anwendung weder um eine Internetanwendung, noch um ein *Massively* Multiplayer Online Game handelt, können diverse Nachteile von TCP vernachlässigt werden. Bei der Wahl von UDP müsste zudem eine eigene Schicht auf der Anwendungsebene hinzugefügt werden, welche verlorene sowie verspätet eingetroffene Pakete und Verbindungsabbrüche jeglicher Art handhabt.

WLAN verbindet Geräte mittels Hochfrequenztechnik und ermöglicht hohe Bandbreiten, Bluetooth hingegen erlaubt lediglich geringere Übertragungsraten und ist zudem komplexer in der Handhabung, wodurch sich das zwingend benötigte Client-Server-Modell ungleich schwerer implementieren ließe[377]. In [Wang u. a., 2009] wurden verschiedene mobile Datenübertragungsmechanismen in Verbindung mit einem Mehrspieler-Online-Spiel getestet. Das Ergebnis spiegelt die Erwartungen wider, dass sich ein WLAN hierfür besonders gut eignet[378]. Unterschiede bei der Verwendung von TCP respektive UDP sind nicht aufgefallen, was jedoch durch das Spiel und dessen Implementierung selber bedingt sein kann. Weiterhin wird das geplante Projekt ortsbezogen sowie vorzugsweise in geschlossenen Räumen stattfinden, da die gesamte Anwendung auf einem zentralen Rechner läuft. Insofern stellt sich die Verwendung eines mobilen Datenübertragungsmechanismus über das Mobilfunknetz eines Netzbetreibers[379] zum einen als schwierig (Dämpfung in geschlossenen Räumen) und zum anderen letztlich als unnötig heraus.

Mit diesem Kapitel endet der Konzeptionsteil. Im folgenden Teil der Arbeit werden die vorgestellten Überlegungen anhand konkreter Beispiele erläutert, um auf diese Weise einen Einblick in die Systemrealisierung zu erhalten.

[377] Es existieren auch Versuche [Knutsson u. a., 2004] P2P-Ansätze für MMOs zu entwickeln, welche jedoch an vielen Stellen nach wie vor scheitern.

[378] Lecky-Thompson beschreibt den Einsatzzweck eines WLANs zudem als Möglichkeit für „echtes Networking“ [Lecky-Thompson, 2008, S. 10].

[379] wie beispielsweise EDGE oder HSPA/HSDPA

Teil III.

Realisierung

„Application architecture is the organizational design of an entire software application, including all sub-components and external applications interchanges.“

(WISEGEEK [wiseGEEK, 2014]*)*

KAPITEL 18

Übersicht der Anwendungsarchitektur

Die konzeptionierte MR-Anwendung besteht aus verschiedenen Komponenten, sodass für einen ersten Überblick zunächst die verwendete Anwendungsarchitektur grob skizziert wird, um daraufhin eine Komponenten-basierte Detailbetrachtung vorzunehmen.

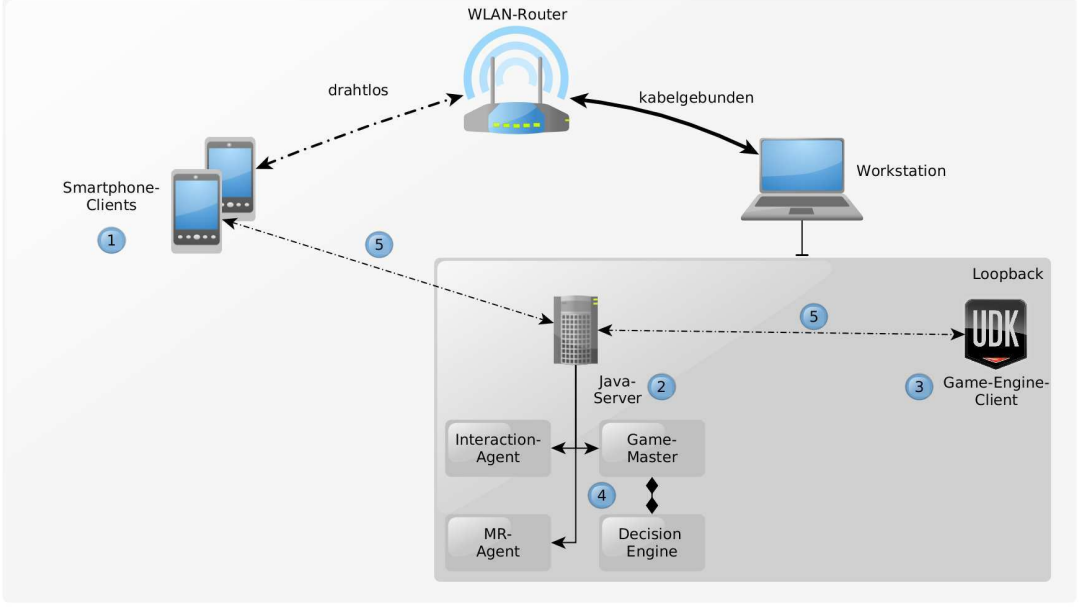


Abbildung 18.1.: Die Abbildung zeigt die Anwendungsarchitektur der umgesetzten Anwendung mit allen involvierten Komponenten.

Bei der in Abb. 18.1 dargestellten Gesamtanwendung handelt es sich um eine multi-threaded Client-Server-Architektur. Die Spieler und Spielerinnen verwenden Smartphones (1) um Avatare in der Spielwelt zu steuern und mit selbiger interagieren zu können. Es werden jedoch nicht ausschließlich Eingaben über die Geräte getätigt, sondern ebenfalls Ausgaben, zum Beispiel in Form von haptischem Feedback oder Sound, verwendet. Die Besonderheit hierbei ist, dass virtuelle Aktionen in die reale Welt übertragen werden und umgekehrt. Es handelt sich nicht um eine rein virtuelle Erfahrung, viel mehr wird die Virtualität um Teile der Realität erweitert, wodurch sich eine Verschmelzung der beiden Welten ergibt. Das implementierte Spiel ist somit nicht nur von der Nutzung der Eingabegeräte abhängig und findet isoliert für jeden Spieler einzeln sowie in einer rein virtuellen Welt statt, sondern ist zusätzlich abhängig von äußeren Einflüssen[380]. Der Implementierung der einzelnen Interaktionen und des Steuerungskonzepts ging die Konzeption einer robusten Anwendungsarchitektur inklusive Kommunikationsverhalten voraus, ebenso wie die Erweiterung der vorhandenen Verbindungsroutine.

Die Spielwelt wird mit Hilfe einer Game-Engine (3), in diesem Fall dem UDK, realisiert. Durch die Platzierung bestimmter Objekte[381] in der Spielwelt, lassen sich sowohl die Interaktionsorte, als auch die Story-relevanten Punkte festlegen. Die Auswirkungen bestimmter Spielentscheidungen werden direkt durch RPCs vom Server respektive dem Game Master in der Spielwelt ausgeführt. Für die Bereitstellung mehrerer Avatare und die Möglichkeit diese über korrespondierende Eingabegeräte steuern zu können, waren grundlegende Änderungen am UDK-Code notwendig. Ebenso aufwendig war die Implementierung der konzeptionierten Kameralogik.

Die beiden zuvor beschriebenen Komponenten bilden Clients, welche sich zu einem zentralisierten Server (2) verbinden und über diesen miteinander kommunizieren (5)[382]. Der Server übernimmt nicht nur jegliche Verbindungs- und Kommunikationsaufgaben in der Anwendung, sondern beherbergt zudem weitere Komponenten (4), welche unterschiedliche Aufgaben übernehmen. Neben dem Game Master, welcher die Berechnungen von Spielsituationen und die Entscheidungsfindung vornimmt, existieren ein spezieller Interaction- sowie MR-Agent. In den nachfolgenden Abschnitten folgt eine Detailbetrachtung aller beschriebener Komponenten sowie des eingesetzten Kommunikationsprotokolls.

[380] Beispiele zu der Art der eingesetzten Interaktionen wurden in Abschnitt 14.3.1 auf Seite 248 bereits beschrieben. Eine ausführliche Betrachtung des MR-Aspekts ist in Kapitel 13 ab Seite 227 zu finden.

[381] Es handelt sich hierbei um die im Rahmen dieser Ausarbeitung entwickelten *placeable Actors*, welche in Abschnitt 18.2 auf Seite 273 genauer behandelt werden.

[382] Es findet keine direkte Kommunikation statt, da dies den Einsatz eines P2P-Ansatzes vorausgesetzt und demzufolge zu einer verminderten Funktionalität und erhöhtem Datenverkehr geführt hätte. Detaillierter werden diese Entscheidungen in den Abschnitten 3.1.5 und 3.1.6 betrachtet.

18.1. Kommunikationsprotokoll

Für die Kommunikation ist ein kompaktes, von allen involvierten Komponenten unterstütztes und sowohl von Menschen als auch Maschinen les- und wartbares Datenformat benötigt worden. All diesen Anforderungen entspricht unter anderem die JSON, welche daher und aus Gründen der Serialisierbarkeit sowie der Möglichkeit der einfachen Konstruktion sowie Skalierbarkeit verwendet worden ist.

Für die gesamte Kommunikation innerhalb der Anwendung wird ein so genanntes *Nachrichten-Konzept* verwendet. Dieses gibt ein festes Schema von Nachrichten vor, welches alle wesentlichen Daten, die letztlich vom Empfänger benötigt werden, umfasst [DeLoura, 2002, S. 217]. Zudem existiert die Möglichkeit zusätzliche, wichtige Informationen nachgelagert in Nachrichten einzubetten, sodass ein Logging selbiger eine ausführliche Analyse zulässt. Die Möglichkeit das Nachrichten-Konzept in der beschriebenen Form zu erweitern, kann als größter Vorteil dieses Ansatzes betrachtet werden. Anhand des Nachrichtenflusses und der zusätzlichen Informationen in den einzelnen Nachrichten lässt sich jede in einen Kontext einbetten und kann somit beispielsweise sowohl zum Nachvollziehen von Fehlern, als auch zur Erstellung eines Flowcharts verwendet werden.

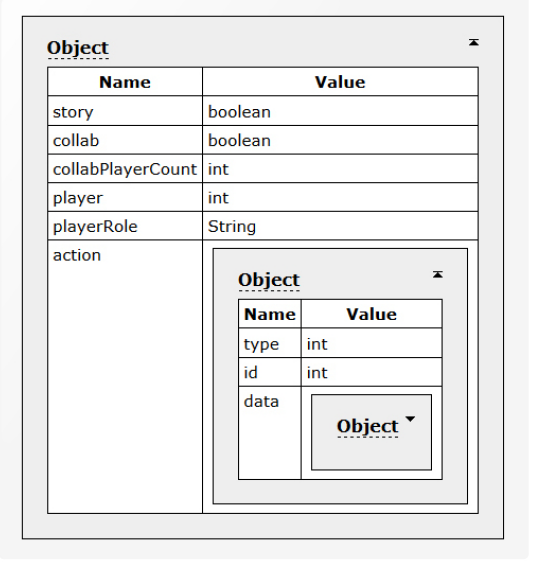


Abbildung 18.2.: Jede Nachricht wird in Form eines JSON-Objekts realisiert und versendet. Die Abbildung repräsentiert beispielhaft einige der nutzbaren Attribute und zielt viel mehr auf die Darstellung der einzelnen Hierarchiestufen.

In Abb. 18.2 ist beispielhaft der Aufbau einer Nachricht dargestellt. Es existieren drei Hierarchiestufen, welche unterschiedliche Arten von Informationen beinhalten. Das äußerste Objekt speichert allgemeine Angaben über den betroffenen Spieler (*player, playerRole*) sowie die Story-Relevanz (*story*). Ebenfalls können hier Informationen zur Art der Interaktion hinterlegt werden (z.B. *collab, collabPlayerCount*). Unterschieden wird jede Nachricht jedoch hauptsächlich anhand des *type*-Attributs in der zweiten Ebene. Neben anderen existieren hier: *Movement, Interaction, MR-Event* sowie *StoryAgent-Knowledge* und *StoryAgent-Action.* Zur weiteren Unterscheidung der Interaktionsart kann zudem das *id*-Attribut verwendet werden. Die letzte Hierarchiestufe bildet das *data*-Attribut. Dieses beinhaltet je nach Nachrichtentyp unterschiedliche Werte und kann zudem flexibel um zusätzliche Angaben (Debugging, Logging) erweitert werden. Bei Bewegungsnachrichten enthält das data-Attribut beispielsweise Koordinatenangaben.

Dadurch, dass nicht jedes Objekt alle Elemente beinhalten muss und zudem weitere Attribute möglich sind, eignet sich das *Builder-Pattern* in diesem Zusammenhang besonders. Dieses Pattern stammt aus der Kategorie der Erzeugungsmuster und ermöglicht es, die Konstruktion eines Objekts zu kapseln, sodass die Herstellung aus mehreren Schritten bestehen und variieren kann (siehe Beispiel 18.1) [Freeman u. a., 2005, S. 614ff].

```
1  JsonBuilder jb = new JsonBuilder()
2     .type(Header.ControlCommands.DISABLE_INTERACTIONVIEW);
3  JsonObject jo  = jb.buildJson();

4  new JsonBuilder().type(
5              Header.ControlCommands.REFRESH_PLAYERCOUNT
6          ).player(androidClientsize)
7          .id(androidClientsize)
8          .dataJson(
9              "{'pnames':"+pnamesVar+", 'roles':"+rolesVar+"}"
10         ).buildJson();
```

Quelltext 18.1: Durch das Builder-Pattern ist es möglich die Kreation eines Objekts in getrennten Schritten vorzunehmen. Technisch wird dies durch einzelne, aneinander gehangene Methodenaufrufe realisiert.

Die Methodennamen stellen dabei die Attribute aus dem Nachrichtenschema dar und sind damit intuitiv verwendbar. Erst mit dem abschließenden Bauauftrag (Zeile 3 und 8), wird das Objekt in der richtigen Konstellation und dem Schema entsprechend zusammengebaut sowie zurückgegeben. Wann das

Objekt letztlich kreiert werden soll ist zeitlich nicht festgelegt, sodass eine Konstruktion weder davon, noch von dem konkreten Direktor[383] abhängig ist.

Alle beteiligten Kommunikationsteilnehmer (Clients sowie Server) besitzen eine zentrale Anlaufstelle, welche die ankommenden Nachrichten entgegennehmen und entsprechend verarbeiten. Hierbei findet zunächst eine Unterscheidung anhand des Nachrichtentyps statt, gefolgt von der auszuführenden Routine. Zur Kommunikation zwischen der Spielwelt und dem Server sowie damit einhergehend mit den Smartphone-Clients werden spezielle, in der Spielwelt platzierbare Objekte verwendet, welche im folgenden Abschnitt näher betrachtet werden.

18.2. Trigger-Objekte

Ein wichtiger Aspekt der Implementierung der Anwendung war die Art der Informationsvermittlung von der Spielwelt zur Decision Engine und anderen Komponenten sowie in umgekehrter Richtung. Ein Event-basierter Ansatz erlaubt hierbei eine Flexibilität im Aufbau des Codes und der Spielwelt, die bei einer statischen Informationsübermittlung nicht gegeben wäre. Um die Events empfangen zu können, musste eine Art Observer-Pattern implementiert werden[384].

Im Rahmen der Anwendungsentwicklung wurde ein spezielles Objekt, ein so genannter *placeable Actor*, konzipiert und implementiert, welches das Verhalten eines Triggers beinhaltet und erweitert. In der Spielwelt kann diese Art von Objekten mit Hilfe des UDKs vollkommen frei platziert und für einen bestimmten Anwendungszweck konfiguriert werden (siehe Abb. 18.3). Trigger verfügen über einen einstellbaren Umgebungsradius[385], dessen Betreten sowie Verlassen das Auslösen einer Nachrichtensendung an den Server oder die Benachrichtigung anderer Spielobjekte zur Folge hat. Die platzierbaren Objekte verfügen demnach über einen oder mehrere Beobachter (so genannte „Observer“), welche automatisch bei Neuerungen oder Ereignissen informiert werden.[386]

Innerhalb eines Triggers können die im vorangegangenen Abschnitt erläuterten Attribute einer Nachricht belegt werden. Je nach Einsatzzweck können hier unterschiedliche Konstellationen entstehen. Nach dem Erhalt der Nachricht wird diese zur weiteren Verarbeitung analysiert und an einen respektive mehrere Nachrichtenagenten delegiert.

[383] Als Direktor wird in diesem Entwurfsmuster diejenige Klasse oder Komponente genannt, welche das Objekt konstruiert.

[384] Das Observer-Pattern ist ein so genanntes *Gang of Four (GOF)*-Entwurfsmuster aus der Kategorie der Verhaltensmuster. Das Ziel ist eine Eins-zu-viele-Abhängigkeit zwischen einem Subjekt und vielen Beobachter-Objekten so zu definieren, sodass bei Änderungen des Zustandes des Subjektes alle Beobachter-Objekte benachrichtigt werden [Freeman u. a., 2005].

[385] siehe Abschnitt 19.2 auf Seite 282

[386] Im Gegensatz dazu existiert die Möglichkeit eines Polling-Ansatzes, welcher in der Art implementiert wird, dass Objekte in der Spielwelt kontinuierlich bei der Game-Engine oder einem übergeordneten Verwaltungsobjekt nachfragen ob es Neuigkeiten gibt, welche sie betreffen, um daraufhin die erwarteten Informationen zu erhalten.

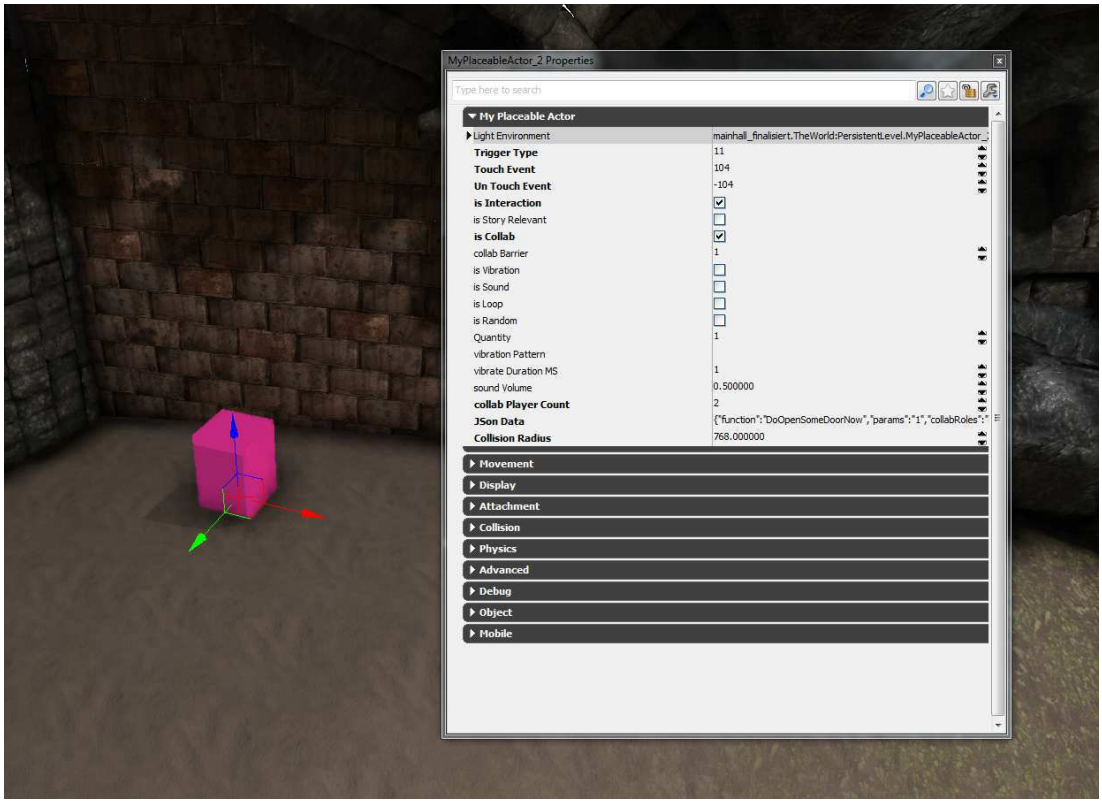


Abbildung 18.3.: Die Abbildung stellt ein im UDK platziertes Trigger-Objekt und dessen geöffnete Eigenschaften dar. Der Eintrag „is Story Relevant“ wird beispielsweise dafür verwendet, um den Game Master darüber zu informieren, dass an dieser Stelle ein Event ausgelöst wurde, welches zur Änderung der erzählten Story führen bzw. eine Modifikation in der Spielwelt nach sich ziehen könnte. Im folgenden Verlauf der Arbeit wird daher auch von Story-relevanten Ereignissen gesprochen.

„Don't blame the messenger because the message is unpleasant."

(Kenneth Starr [BrainyQuote, 2014]*)*

KAPITEL 19

Nachrichtenagenten

Sowohl der Game-Engine-Client als auch die verbundenen Smartphone-Clients senden Nachrichten an den Server, welche entsprechend verarbeitet und in einigen Fällen an die jeweils anderen Clients weitergeleitet werden müssen. Im Allgemeinen existieren zwar verschiedene Nachrichtentypen, welche den Inhalt bestimmen, jedoch unterscheidet sich dieser sowie die auszuführende Routine ebenfalls an der Art des Senders. Insofern existieren auf Seiten des Servers zwei unterschiedliche Empfangseinheiten, welche die Vorverarbeitung vornehmen: eine für den Game-Engine-Client und eine weitere für die Smartphone-Clients. Die Verarbeitung übernehmen je nach Typ daraufhin die zuständigen Agenten.

Eingehende Nachrichten des Game-Engine-Clients werden, wie in Abb. 19.1 dargestellt, verarbeitet. Unabhängig von der Story-Relevanz findet eine Unterscheidung der Nachricht anhand des Typen statt. Hauptsächlich betrachtenswert sind der *Interaction-Agent* sowie der *Game Master*, gefolgt vom *MR-Agent*. Der zusätzlich dargestellte *Message-Agent* findet derzeit noch keine Anwendung. Geplant war ein gezieltes Versenden von Nachrichten an einzelne Clients, beispielsweise um einen bestimmten Spieler über den Fund eines Gegenstands zu informieren bzw. diesen mit Zusatzinformationen zu versehen. Je nach Story-Relevanz wird die erhaltene Nachricht zusätzlich an den Game Master weitergeleitet, welcher die benötigten Informationen extrahiert und an die Decision Engine weiterleitet. Dies ist wichtig, da prinzipiell jede Aktion zu einer Änderung der Wissensbasis führen - und sich somit auf die Spielwelt auswirken könnte.

Nachrichten der Smartphone-Clients besitzen derzeit den Typ *Movement* oder *Interaction*. Zuerst genannter Typ führt zur Sendung aktueller Bewegungskoordinaten an den Server, welche zur weiteren Verarbeitung um den Absender erweitert und unmittelbar an den Game-Engine-Client weitergeleitet werden[387]. Letzterer führt ebenfalls zu einer Delegation, jedoch an den *Interaction-Agent*. Der *MR-Agent* wird ausschließlich durch den Game-Engine-Client mit Nachrichten versorgt. Dies liegt darin

[387] Aus diesem Grund existiert hierfür auch kein eigener Agent.

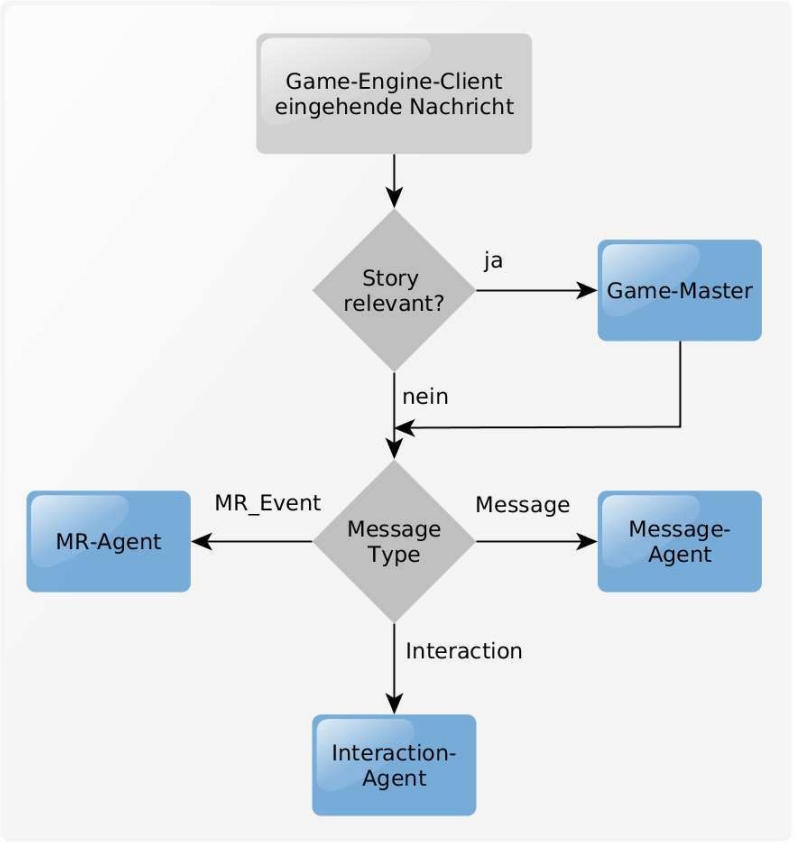


Abbildung 19.1.: Eingehende Nachrichten des Game-Engine-Clients werden anhand des Typs und der Story-Relevanz unterschieden und an unterschiedliche Agenten delegiert.

begründet, dass auf die reale Welt erweiterte Ereignisse in der Spielwelt in Form von Triggern platziert und beim Betreten durch Avatare ausgelöst werden. Der Game Master wird mit sämtlichen die Decision Engine betreffende Informationen betraut und übernimmt sowohl die Analyse der erhaltenen Daten als auch die Ausführung von Reaktionen, welche eine Folge des Eintreffens von Bedingungen in der DE sind.

19.1. Interaction-Agent

Der Interaction-Agent übernimmt auf Serverseite die Überwachung einer durchführbaren respektive durchzuführenden Interaktion. Die Auslösung eines Triggers, welcher eine Interaktion beinhaltet, führt dazu, dass eine Interaktionsnachricht an den Server gesendet wird. Diese besteht unter anderem aus folgenden Eigenschaften:

Type beinhaltet in diesem Fall den für eine Interaktion definierten Identifikator. Um eine zusätzliche Unterscheidbarkeit in der Art des Senders auszumachen[388] werden hier zwei verschiedene Werte verwendet, einer für den Game-Engine- und ein weiterer für die Smartphone-Clients.

Collab enthält eine Angabe darüber, ob die Interaktion in kollaborativer Art (nahezu gleichzeitig) ausgeführt werden muss, oder ob es beispielsweise reicht, wenn ein Spieler bzw. eine Spielerin die Interaktion ausführt. Interaktionen können auch so gestaltet sein, dass eine korrekte Ausführung davon abhängig ist, dass keiner der Nutzer versagt. Ein konkretes Beispiel aus der konzipierten Anwendung sieht so aus, dass zu einem bestimmten Zeitpunkt die Mikrophone aller Mitspieler freigeschaltet werden und das Ziel ist, so wenig Geräusche wie möglich zu produzieren. Das Ausschlagen eines Mikrophons führt zum Versagen des gesamten Teams. Dem gegenüber existieren Interaktionen, welche kein Versagen vorsehen, sondern lediglich durch ein nahezu gleichzeitiges Ausführen erfolgreich beendet werden können.

CollabPlayerCount gibt an, wie viele Mitspieler für diese Interaktion benötigt werden und sich demnach im platzierten Trigger befinden müssen. Mit dem Erreichen der geforderten Anzahl ist die Möglichkeit zur Interaktion gegeben und die Nutzer werden darüber informiert.

CollabRoles beinhaltet eine Auflistung benötigter respektive zugelassener Spielerrollen für diese Interaktion. Ein „all“ ist ebenfalls möglich. Ein Betreten anderer Rollen hat keine Auswirkung.

CollabBarrier steht für die erlaubte Zeitdifferenz zwischen den einzelnen Durchführungen der Interaktionen der jeweiligen Spieler.

Quantity gibt die Anzahl der Durchführungen einer Interaktion an. Einige Interaktionen erfordern ein mehrmaliges Durchführen, da damit eine sukzessive Animation einhergeht oder um den Grad des Realismus zu erhöhen. Eine Wand lässt sich beispielsweise nicht durch einfaches Ausführen einer Einschlaggeste zum Einsturz bringen.

[388] Beim Ankommen der Nachrichten auf Serverseite ist eine eindeutige Unterscheidbarkeit aufgrund physikalischer Eigenschaften (eigener Netzwerksockets) zwar bereits gegeben, aus Gründen der späteren Analyse bietet sich die Unterscheidbarkeit in Form von nicht-kontextsensitiven Parametern jedoch an.

Randomness beinhaltet eine Angabe über die Verwendung eines Probabilismus zur Berechnung des Erfolgs einer durchgeführten Interaktion. Da der Zufall in Pen-&-Paper-Rollenspielen einen wichtigen Aspekt bildet, galt es diesen ebenfalls abbilden zu können.

Function gibt die Funktion an, welche nach abgeschlossener Interaktion durch einen Remote Procedure Call vom Server direkt ausgeführt werden soll.

Params enthält Parameter, welche der auszuführenden Funktion mitgegeben werden sollen. Speziell im Hinblick auf das Öffnen von Türen in der Spielwelt ist es eher suboptimal für jede Tür eine eigene Funktion zu definieren, sodass das Verwenden von Parametern zur Spezifizierung geeigneter erscheint.

ID beinhaltet die Art der Interaktion. Eine Unterscheidung der Interaktionsformen findet unter Verwendung verschiedener Identifikatoren in Form von Zahlen statt. Dadurch, dass Trigger betreten und verlassen werden können, wird hierfür ein zusätzlicher Unterscheidungsmechanismus benötigt. Anstatt zwei verschiedene Identifikatoren zu verwenden, wird mit Hilfe eines Vorzeichens ein eindeutiges Merkmal eingeführt. Korrespondierend zur mathematischen Bedeutung wird auch hier ein positives Vorzeichen für das Betreten (Inkrementierung der Spieleranzahl im Trigger) und ein negatives für das Verlassen (Dekrementierung) eines Triggers verwendet.

Player zeigt mittels Identifikator den ein- respektive ausgetretenen Spieler an. Jeder Spieler besitzt eine eindeutige ID innerhalb der gesamten Anwendung.

Playerrole gibt entsprechend die Rolle des ein- respektive ausgetretenen Spielers an. Der UDK-Client bezieht sich beim Betreten des Triggers durch einen Spieler dessen Eigenschaften und übertragt diese automatisch in dieses und das vorangegangene Feld.

Die Verarbeitung des Interaction-Agent, welche in Abb. 19.2 dargestellt ist, orientiert sich dabei an zwei wesentlichen Faktoren: der Interaktions-ID sowie der Client-Art. Erstere signalisiert dem Agenten, ob ein Objekt für diese Art der Interaktion bereits vorliegt. Ist dies nicht der Fall und es handelt sich zudem um eine positive ID (b) und der eingetretene Spieler besitzt eine der angegebenen Rollen, so wird ein spezielles, die Interaktion betreffendes Objekt angelegt (c). Dieses enthält fortan alle notwendigen Informationen der Interaktionsform. Neben der Spieleranzahl, den erlaubten Rollen und einem Hinweis auf Notwendigkeit einer gleichzeitigen Ausführung, wird die nach Abschluss auszuführende Funktion inklusive Parametern zwischengespeichert sowie die aktuell verbundenen Spieler, identifizierbar durch die eindeutige ID.

Das Betreten des Triggers durch einen weiteren Spieler führt zur Sendung derselben Interaktionsnachricht, jedoch mit den spezifischen Eigenschaften des neuen Spielers (ID und Rolle). In dem Fall muss das Interaktionsobjekt nicht neu angelegt, sondern entsprechend modifiziert werden (e). Da es

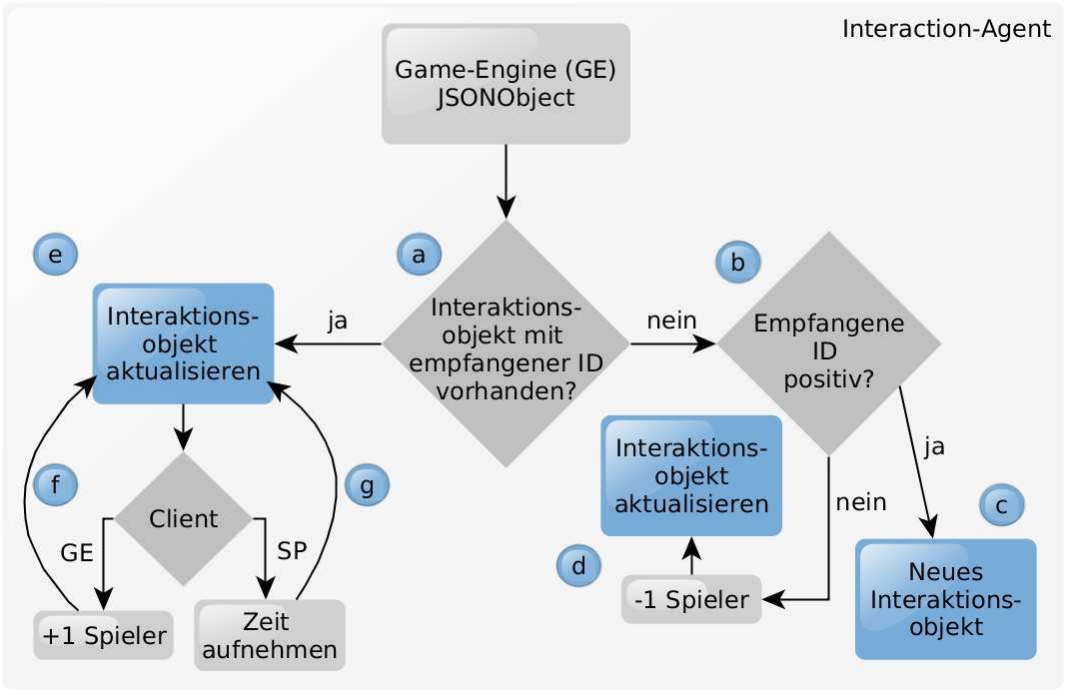


Abbildung 19.2.: Die Verarbeitung des Interaction-Agent orientiert sich an der Interaktions-ID sowie dem Sender der Nachricht. Entsprechend sieht die Verarbeitungsroutine anders aus und mündet entweder in der Überwachung der durchzuführenden Interaktion oder in der Ermöglichung einer Interaktion. Letztere äußert sich dadurch, dass die Smartphone-Clients über die Möglichkeit zum Interagieren informiert werden und erst nach einer Bestätigung, diese durchführen können, da keine erzwungenen Handlungen stattfinden sollen. *GE* steht in der Abbildung für Game-Engine, *SP* hingegen für Smartphone.

sich hierbei um eine Nachricht der Game-Engine handelt wird die Kante f besucht, was zu einer Inkrementierung der Spieleranzahl innerhalb des Interaktionsobjekts führt. Zusätzlich findet an dieser Stelle eine Überprüfung der Spielerrolle statt, ebenso wie ein Abgleich der aktuell registrierten und benötigten Spieler. Mit Erreichen der erforderten Anzahl, wird ein Steuersignal an alle involvierten Smartphone-Clients gesendet, welche daraufhin den Benutzer sowohl visuell als auch haptisch auf die Möglichkeit zu einer Interaktion hinweisen (vgl. Abbildung 19.4). Das Verlassen eines Triggers führt dazu, dass der rechte Pfad zu *b* verfolgt wird. Die Prüfung an dieser Stelle führt dazu, dass das Interaktionsobjekt dahingehend aktualisiert wird, dass der ausgetretene Spieler entfernt und die Spieleranzahl dekrementiert wird (d).

Mit der Bestätigung des Interaktionshinweises auf Seiten der Smartphones wechseln selbige unverzüglich in die korrespondierende Interaktionsansicht, welche die ID aus dem zuvor gesendeten Steuersignal extrahiert und zwischenspeichert. Das Ausführen der eigentlichen Interaktion auf dem Smartphone führt gleichermaßen zum Senden einer Interaktionsnachricht. Diese wird ebenfalls an den Interaction-Agent weitergeleitet. Die Nachricht beinhaltet das Type- sowie das zuvor extrahierte ID-Attribut. Damit lässt sich eine Zuordnung zur durchgeführten Interaktion herstellen, sodass in Abb. 19.2 der linke Teilbereich erreicht wird. Dadurch, dass es sich bei dem Absender dieses Mal um einen Smartphone-Client handelt wird eine andere Aktion ausgeführt. Diese Funktionsweise wird im Folgenden genauer erläutert.

Falls es sich bei der Interaktion um eine kollaborative Tätigkeit, im Sinne der nahezu gleichzeitigen Ausführung einer Aktion handelt, werden die Zeitpunkte der Interaktionsdurchführung zur weiteren Verarbeitung herangezogen. Die Interaktionsnachricht der Smartphone-Clients erreichen zu einem Zeitpunkt X_n den Server. Dieser wird, mitsamt der Spieler-ID, an das Interaktionsobjekt weitergegeben und abgespeichert. Sofern für alle involvierten Smartphones ein Zeitpunkt vorhanden ist, wird die in Abb. 19.3 dargestellte Berechnung durchgeführt.

$$t = |max(X_0, \ldots, X_n) - min(X_0, \ldots, X_n)|$$

Abbildung 19.3.: Maximale Zeitdifferenz von Zeitpunkten in der Menge $X_0, \ldots, X_n$ in Sekunden.

Durch die Verwendung des *CollabBarrier*-Attributs kann die maximale Zeit manuell gesetzt geändert werden. Ist dies nicht der Fall, wird die eruierte Standardzeitspanne von einer Sekunde verwendet[389]. Das Unterbieten der erlaubten Zeitdifferenz führt zur Ausführung der vorher definierten Funktion dieser Interaktion in der Spielwelt und zum Schließen der Interaktionsansicht auf den Smartphones. Ist ein Versagen möglich, so reicht das Erhalten einer Interaktionsnachricht von einem der Smartphone-Clients zur Ausführung des zuvor geschilderten Verhaltens. Zusätzlich besteht durch die Verwendung des *Randonmess*-Attributs die Möglichkeit, einen Probabilismus mit einfließen zu lassen. Innerhalb von Pen-&-Paper-Rollenspielen existiert mit dem Würfel ebenfalls ein Zufallselement, welches als zentrales Element in diesem Genre gilt. Der gewählte regelbasierte Ansatz ist aufgrund seiner deterministischen Natur dafür nicht geeignet, sodass hiermit eine alternative Möglichkeit geschaffen worden ist. Ferner ist es laut Klinge zudem essentiell, dass eine KI den Spieler immer mit neuen Verhaltensweisen überrascht [Klinge, 2008]. Es handelt sich in der realisierten Form lediglich um eine simple 50:50 Chance, welche jedoch aufgrund ihrer Realisierung als eigenständige Komponente leicht um komplexere Berechnungsverfahren erweitert werden kann.

[389] Wie in Kapitel 15 ab Seite 251 erläutert.

Der zwischengelagerte Schritt, die Interaktion nicht automatisch bei Erreichen der benötigten Spieleranzahl zu starten, führt jedoch zu einer schwerwiegenden Problemstellung. Das Bestätigen des Hinweises auf dem Smartphone führt, wie bereits beschrieben, zum unverzüglichen Wechseln in die korrespondierende Interaktionsansicht. Nutzen beispielsweise drei von vier involvierten Spielern die Interaktionsmöglichkeit und der letzte verlässt daraufhin den Trigger, gelangt dieser nie in die Interaktionsansicht, was wiederum dazu führt, dass die Berechnung der maximalen Zeitdifferenz nie erreicht werden kann. Insofern war es notwendig, das Austreten eines Spieler aus dem Trigger zusätzlich um die Funktionalität zu erweitern, die anderen im Trigger befindlichen Spieler über einen solchen Umstand, in der Art zu informieren, dass die Interaktionsansicht bei diesen automatisch wieder geschlossen wird, sofern diese geöffnet war.

Abbildung 19.4.: In der Abbildung wird die Standardansicht der Smartphone-Clients dargestellt, welche die Bewegung der Avatare ermöglicht. Der rechte Bereich, welcher eine Carbon-Textur beinhaltet, wird zur Steuerung der Avatare bzw. der Cursor der Spieler verwendet[390]. Das Berühren des „Run“-Buttons führt dazu, dass der Avatar sich zu dem in der Spielwelt platzierten Cursor bewegt. Der darunter befindliche Button signalisiert dem Spieler, dass hier eine Interaktionsmöglichkeit besteht. Das Erscheinen desselben geht mit einer Vibration einher, die den Nutzer auch haptisch darauf hinweisen soll, dass interagiert werden kann. Das Betätigen des Buttons führt zum unmittelbaren Wechsel in die korrespondierende Interaktionsansicht. Ein Beispiel ist in Abbildung 21.2 dargestellt.

19.1.1. Smartphone- und Game-Engine-Clients

Auf Seiten der Clients existieren keine speziellen Agenten für die Bearbeitung respektive Überwachung der durchzuführenden Interaktionen. Der Game-Engine-Client stellt Funktionen sowie Parameter zur Verfügung, welche nach Abschluss einer Interaktion direkt per Remote Method Invocation (RMI) aufgerufen werden können und somit zur Modifikation der Spielwelt führen. Insofern wird hier

kein zusätzlicher Logikbereich benötigt. In welcher Art und Weise diese beiden Komponenten kombiniert werden, steht dem Spielweltdesigner daraufhin vollkommen frei. Das Öffnen einer Stahltür kann beispielsweise entweder durch das vorherige Durchführen einer Wischgeste auf den Smartphones oder durch das einfache Betreten des Triggers durch einen Avatar ausgeführt werden.

Die Smartphone-Clients sind ähnlich konzipiert. Eine Reihe von Interaktionsansichten sind in der Anwendung vorgesehen und können durch das Senden bestimmter Nachrichtenkonstrukte ausgelöst werden. Die Überwachung und Bewertung einer Interaktion übernimmt jede Ansicht selber und beinhaltet damit lediglich sämtliche, die Interaktion betreffende Logik, was zu einer hohen Kohäsion führt. Das hat den Vorteil, dass die Komponenten unabhängig voneinander getestet sowie eingesetzt werden können und insgesamt stabiler sowie weniger fehleranfällig sind. Ebenfalls besteht die Möglichkeit, den Initialisierungsnachrichten Parameter für die anzuzeigende respektive durchzuführende Interaktion mitzugeben. Dadurch existiert zusätzlich die Möglichkeit, Permutationen der fest vorgegebenen Interaktionsarten zu erstellen, womit sich eine größere Vielfalt und damit erweiterter Spielspaß erreicht werden kann. Der Wechsel aus und in Interaktionsansichten wird ausschließlich über den Server gesteuert, da eine kollaborative Ausführung nur von diesem kontrolliert und bestätigt werden kann.

19.2. MR-Agent

Bei der als MR-Agent bezeichneten Komponente handelt es sich um eine zwischen dem Server und den Clients verteilt arbeitende Komponente, die einen Zuständigkeitsrahmen für sämtliche MR-Aspekte der Anwendung, namentlich die Haptik- sowie Sound-Trigger, bildet.

Die von dieser Komponente implementierte Logik ist abzugrenzen von sogenannten Mixed Reality-Interaktionen, welche MR-Aspekte als Teil einer Interaktion implementieren. Letztere werden vom Interaction-Agent behandelt, dessen Funktionsweise im vorangegangenen Abschnitt genauer beschrieben wurde.

Im Gegensatz zum Interaction-Agent werden die Hauptaspekte der Logik des MR-Agents von den Clients implementiert. Die Nutzung der Soundausgabe sowie des Vibrators auf den Smartphones führt zum Aufspannen eines Raumes zwischen der virtuellen und realen Welt und implementiert auf diese Weise die vierte Wand als Volumen[391]. Insbesondere das Erweitern von Geräuschen auf die Eingabegeräte derjenigen Spielerinnen und Spieler, deren Avatare sich in der Spielwelt in der Nähe einer bestimmten Soundquelle befinden, verschiebt auf diese Weise die Eckpunkte des Handlungsraums.

[391] Eine Definition und nähere Erläuterung, speziell zur vierten Wand, ist in Kapitel 13 ab Seite 227 zu finden.

Hierbei wird die Entfernung des Spielers zur Quelle für die Berechnung der Lautstärke verwendet (vgl. Abbildung 19.5). Ein ähnliches Verhalten war auch für die Vibrationsintensität und die Entfernung zu einem Objekt geplant, konnte jedoch aufgrund des dafür notwendigen Implementierungsaufwands nicht in einem vertretbaren Zeitrahmen realisiert werden[392]. Alternativ hierzu wurde als Demonstration des grundsätzlichen Konzepts die Möglichkeit geschaffen, wahlweise das Gerät über einen definierten Zeitraum in konstanter Stärke vibrieren zu lassen oder ein Vibrationsmuster zu verwenden.

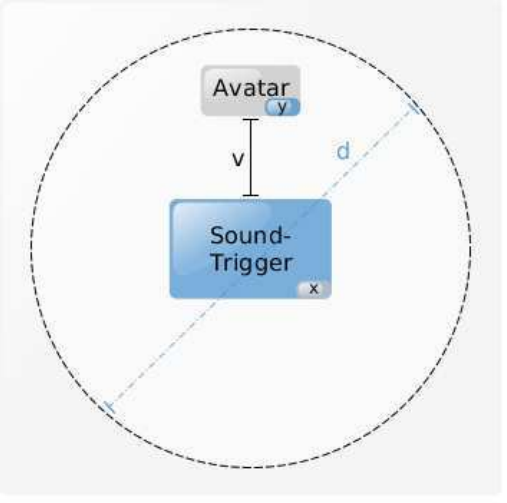


Abbildung 19.5.: Die Berechnung der Lautstärke *vol* wird mit Hilfe der Positionen des Sound-Triggers (x) sowie des Avatars (y) und dem Umgebungsradius respektive Durchmesser (d) des Trigger-Objekts durchgeführt.

Der Server dient in diesem Anwendungsszenario ausschließlich als Nachrichtenverteiler zwischen der Spielwelt sowie den Interaktionsgeräten und übernimmt keinerlei weiterführenden Aufgaben. Die Smartphone-Anwendung wiederum implementiert, wie auch bei den Interaktionen, eine eigene Schnittstelle, welche von außen angesprochen werden kann und entsprechend auf Eingaben reagiert. Hierfür wird in der Anwendung eine bestimmte Anzahl an Sounds vorgehalten, welche durch die Angabe eines Identifikators, einer Lautstärke sowie dem Hinweis auf ein automatisches Wiederholen abgespielt werden können. Für die Funktionsweise war es notwendig, die bereits erwähnten[393] Trigger-Objekte zu erweitern.

[392] Die grundlegende Programmierschnittstelle Androids bietet diese Funktionalität nicht an. Mit Hilfe einer externen Bibliothek könnte die Funktionalität nachgerüstet werden.
[393] vgl. Abschnitt 18.2 auf Seite 273

Sound-Trigger werden durch die Verwendung eines speziellen Attributs für das UDK entsprechend ausgezeichnet. Anstatt der Sendung *einer* Benachrichtigung beim Betreten sowie Verlassen des Umgebungsradius wird *kontinuierlich* die Entfernung des Avatars zum Trigger-Objekt, unter Verwendung der in Abb. 19.6 dargestellten Formel, berechnet. Hierfür werden die Positionskoordinaten des Sound-Triggers (x) sowie des Avatars (y) herangezogen und in Abhängigkeit zum eingestellten Radius[394] gesetzt. Für eine weitere Bearbeitung ist es zuvor notwendig, die Länge des Ergebnisvektors zu berechnen. Da die Lautstärke vol mit abnehmender Entfernung zum Triggerzentrum steigen soll, ist eine Invertierung des Ergebnisses notwendig.

$$vol = 1 - ((|\vec{y} - \vec{x}|)/\frac{d}{2})$$

Abbildung 19.6.: Berechnung der Lautstärke einer Soundausgabe auf einem Smartphone.

Mit dem Betreten eines Sound-Triggers, welcher den MR-Aspekt verwendet, wird der betroffene Smartphone-Client dazu angewiesen, die benötigte Sounddatei in einer Endlosschleife abzuspielen. Die Startlautstärke beträgt einen Wert von null und wird kontinuierlich entsprechend der Entfernung des Avatars zur Quelle erhöht oder verringert. Damit der Smartphone-Client den Sound nicht auch nach Verlassen des Umgebungsbereichs endlos mit einer Lautstärke von null weiterspielt, wird beim Verlassen die Beendigung explizit forciert. Neben diesem eher komplexen Anwendungsfall können auch simplere Fälle abgedeckt werden. Bei dieser Form löst das Betreten eines Triggers das einmalige Abspielen eines Soundeffekts mit angegebener Lautstärke auf dem Smartphone aus.

19.3. Game Master

Der Game Master übernimmt, wie in Pen-&-Paper-Rollenspielen üblich, die Kontrolle über die gesamte Spielwelt und überwacht den aktuellen Spielstatus. Die Spieler werden innerhalb des Spiels unbewusst Situationen ausgesetzt, deren Bewältigung den weiteren Spielverlauf beeinflussen. Das Wählen von Pfaden in der Spielwelt führt zum Beispiel zu Änderungen in der erzählten Geschichte. Diese Ereignisse werden in der Anwendung als Story-relevant bezeichnet und verfügen daher über ein spezielles Attribut in Nachrichten. Sofern dieses gesetzt ist, übernimmt der Game Master die weitere Verarbeitung und nutzt zur Berechnung des neuen Zustands respektive der daraus resultierenden Folgen (Konsequenzen) einer Entscheidung die Decision Engine. Wie bereits in Kapitel 16 beschrieben wurde, soll dafür ein regelbasiertes System verwendet werden. Im Folgenden wird sowohl die Funktionsweise

[394] In Abb. 19.5 als Durchmesser d dargestellt.

des zugehörigen Agenten, als auch die Funktionalität der künstlichen Intelligenz innerhalb der DE beschrieben.

19.3.1. Agent

Der Agent dient als Vermittler zwischen Server und den damit verbundenen Clients sowie der Decision Engine. Dafür existieren Schnittstellen in beide Richtungen. Story-relevante Nachrichten können über eine Schnittstelle an den Agenten weitergegeben und von diesem daraufhin auf relevante Informationen hin analysiert werden. Die extrahierten Daten werden in die Wissensbasis der DE eingetragen. Hauptsächlich existieren dabei zwei Arten von Operationen: das Modifizieren von vorhandenem und das Hinzufügen von zusätzlichem Wissen. Beide Methoden stoßen eine Regelüberprüfung an. Werden Bedingungen durch Regeln erfüllt, so verwendet die Decision Engine zur Ausführung der korrespondierenden Aktionen ebenfalls den Agenten. Hierfür stellt dieser eine zweite Schnittstelle bereit. Die entsprechenden Anweisungen werden in einer Nachricht verpackt und unter Verwendung des Servers an die Empfänger versendet. Durch die Nutzung des Agenten als zentrale Anlaufstelle lässt sich das Verhalten der einzelnen Komponenten exakt auf einen Aufgabenbereich ausrichten und so möglichst effizient und fehlerresistent gestalten. Insbesondere die DE kann somit beispielsweise von verschiedensten Intelligenzansätzen repräsentiert werden. Ferner ermöglicht der Agent die Hinzunahme weiterer Entscheidungsfindungsansätze oder anderer Komponenten, zum Beispiel für die Ausgabe der Daten in Form eines Flowcharts.

Neben der beschriebenen Funktionalität beinhaltet der Agent ebenfalls die zu erzählende Geschichte. Aufgrund des modularen Aufbaus liegt die Geschichte in unabhängigen Textpassagen vor. Mit dem Abschluss des Spiels wird die aktuelle Wissensrepräsentation in der Decision Engine ausgelesen, woraufhin die Textpassagen entsprechend zusammengestellt und als Textnachricht ausgegeben werden. Durch diese weitere Entkopplung kann die Geschichte unabhängig von der Konzeption der Spielwelt modifiziert werden.

19.3.2. Decision Engine

Die Decision Engine wurde unter Verwendung der Programmiersprache Java als regelbasiertes System implementiert. Entsprechend der Funktionsweise eines solchen Systems besitzt die DE sowohl Regeln als auch eine Wissensbasis sowie eine Konfliktlösungsstrategie. Die Regeln bestehen aus einer Bedingung sowie einer Folgerung. Erstere setzt sich der Einfachheit halber als Aussagenverknüpfung aus atomaren Aussagen und den logischen Konstanten „und“ oder „nicht“ zusammen. Die Folgerung beinhaltet Anweisungen, welche nach Erfüllung der Bedingung ausgeführt werden sollen. Möglich ist

hierbei das Ändern von Wissen in der Wissensbasis sowie das Verändern der Spielwelt unter Zuhilfenahme des Agenten oder eine Zusammensetzung der beiden Handlungen.

Theoretisch wäre es möglich gewesen, die Regelüberprüfung endlos durchführen zu lassen, um unmittelbar über das Eintreffen einer Regel informiert zu werden. Da eine Erfüllbarkeit jedoch ausschließlich durch eine Veränderung der Wissensbasis ermöglicht wird, was wiederum durch das Verwenden von Methoden der Decision Engine geschieht, findet eine Überprüfung der Regeln im Anschluss an jede Modifikation statt. Der Vorteil liegt in der einfacheren Implementierbarkeit sowie in der besseren Überwachung der einzelnen Vorgänge. Der Agent speist die Regeln in das System und startet selbiges nach vollständiger Instanziierung.

Alle Regeln werden iterativ in deren atomaren Bestandteile zerlegt[395], welche daraufhin mit der Wissensbasis abgeglichen werden. Eine Regel gilt zu Beginn der Bearbeitung als erfüllt. Sobald eine atomare Aussage nicht erfüllbar ist, wird die Bearbeitung dieser Regel abgebrochen und entsprechend als unerfüllt markiert. Das Eintreffen einer Bedingung und damit einer Regel führt nicht zum Abbruch der gesamten Verarbeitung, sodass es möglich ist, dass mehrere Regeln erfüllt und damit eine Reihe von Aktionen ausgeführt werden. Nach dem Ausführen dieser wird die korrespondierende Regel für alle weiteren Zyklen deaktiviert. Eine Ausnahmeregelung ist nicht vorgesehen, kann jedoch ohne größeren Aufwand nachgerüstet werden. Dieses Verhalten wurde aus dem Grunde gewählt, da jede Aktion die Spielwelt in der Form verändert, dass ein erneutes Ausführen ohnehin keinen Effekt haben würde, mögliche Fehler hierbei jedoch nicht ausgeschlossen werden können.

Dadurch, dass es sich bei der künstlichen Intelligenz um keine abgeschlossene, sondern von außen modifizierbare und dynamische Einheit handelt, ist es prinzipiell möglich, die Regelbasis ebenfalls von außen erweitern respektive modifizieren zu können. Aufgrund der fehlenden Möglichkeit bestehender Systeme und Lösungen wurde sich für eine Eigenentwicklung entschieden.

[395] Als „atomar“ wird in der Aussagenlogik eine nicht weiter unterteilbare Aussage definiert. Dieser Teil wird als *Literal* bezeichnet.

*„Ein Server ist ein Programm, das einen **Dienst (Service)** anbietet. Im Rahmen des **Client-Server Prinzips** kann [...] der Client [...] diesen Dienst nutzen. Die Kommunikation zwischen Client und Server ist abhängig vom Dienst [...].“*

(Alex [Alex, 2014])

KAPITEL 20

Server

Der Server bildet die Basis der gesamten Anwendung. Sowohl die gesamte Netzwerkkommunikation als auch unter anderem die Entscheidungsfindung und Überprüfung der Interaktionen sind als Teilmodule des Servers realisiert worden. In den vorangegangenen Abschnitten wurden bereits die wesentlichen Module des Servers näher erläutert, sodass in diesem Teil lediglich einige netzwerktechnische Funktionen respektive Erweiterungen betrachtet werden.

Benötigt wurde unter anderem ein Lobby- und damit einhergehendes Verbindungssystem, sodass alle Teilnehmer sich in einer beliebigen Reihenfolge verbinden können und darüber benachrichtigt werden auf wie viele respektive welche Spielkomponenten (Spieler, Game-Engine) noch gewartet wird, bevor das Spiel starten kann. Zudem musste allgemein die Möglichkeit geschaffen werden, dass mehrere Clients sich mit individuellen Eigenschaften verbinden können und in einer geeigneten Verbindungsroutine inklusive Statusbildschirm behandelt werden.

20.0.3. Beliebige Verbindungsreihenfolge

Im zugrundeliegenden Prototyp[396] existierte aus Gründen der Einfachheit eine fest vorgeschriebene Verbindungsreihenfolge. Alle Smartphone-Clients mussten vor dem Starten des Game-Engine-Clients mit dem Server verbunden sein, damit das System ordnungsgemäß funktionierte. Das Nicht-Einhalten dieser Reihenfolge führte zu einem Fehler, welcher im Folgenden beschrieben und dessen Zustandekommen in Abb. 20.1 visualisiert wird.

Das Heranziehen der Ausgabestreams an erster Stelle (a) ist sinnvoll und erfüllt hauptsächlich Performanzanforderungen. Werden die Ausgabestreams erst nach der Verarbeitung der erhaltenen Nachricht des Game-Engine-Clients initialisiert, verzögert sich das Senden auf Serverseite und eine Latenz wird

[396] der aus dem Masterprojekt hervorgegangene

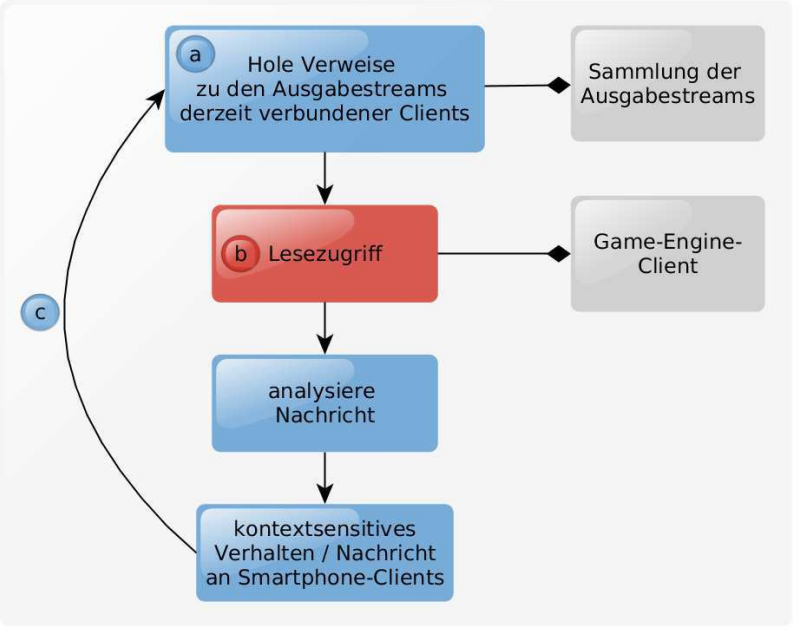


Abbildung 20.1.: In der Abbildung wird das serverseitige Verhalten für den Eingabestream des Game-Engine-Client dargestellt. Hier kommen alle Nachrichten an, welche die Spielwelt an den Server respektive die Smartphone-Clients sendet.

deutlich. Der Server dient als Vermittlungszentrale von Nachrichten aller Clients untereinander. Eine Verzögerung an dieser Stelle würde den Erfolg des gesamten Systems gefährden.

Der zweite Schritt der Ausführungsschleife (b) führte jedoch zu einem schwerwiegenden Fehler. Dadurch, dass es sich um blockierende Sockets handelt, führte dies zum Stoppen der Ausführung an dieser Stelle, da gewartet wurde bis etwas im Eingabestream des Game-Engine-Clients angekommen ist. In der Zwischenzeit ist es jedoch möglich, dass sich weitere Smartphone-Clients verbinden, deren Ausgabestreams sich nicht in der zuvor angelegten Sammlung befinden. Bewegten sich die Spieler daraufhin zu einem Interaktionspunkt im Spiel, welcher ein Event auslöst, sendete der Game-Engine-Client eine Nachricht, welche den Lesezugriff ausführen ließ. Die Benachrichtigung an dieser Stelle interagieren zu können erhielten lediglich zuvor verbundene Spieler. Erst im neuen Schleifendurchlauf (c) befanden sich alle verbundenen Clients im Pool und konnten die Nachricht erhalten. Prinzipiell wäre es möglich gewesen, dass alle Spieler sich aus dem Eventbereich entfernen und diesen daraufhin neu betreten um so zu dem gewünschten Verhalten zu kommen, allerdings stört dies den Spielfluss und ist zudem für Spieler und Spielerinnen nicht nachvollziehbar.

Im Zuge des Refactorings wurde dieser Fehler durch das Hinausverzögern und Auslagern der Game-Engine-Client-Routine behoben. Erst nachdem alle Smartphone-Clients verbunden und alle Initialisierungen vorgenommen worden sind, wird ein Lesezugriff auf den zuvor genannten Client benötigt. Insofern ist das Starten der Ausführungsschleife zeitlich nach hinten verlagert worden, sodass sich im ersten Durchlauf zwangsläufig alle Clients im Pool befinden.

20.0.4. Lobbyfunktionalität

Die Steuerung der Lobbyfunktionalität übernimmt der Server in Form von speziellen Befehlen (Nachrichtentypen), welche an die Clients gesendet werden.

In Abb. 20.2 ist der Workflow des Servers dargestellt. Da verschiedene Clientarten existieren, beinhaltet der Server eine Art interne Poolverwaltung. Alle eingehenden Verbindungsanfragen (a) werden zunächst generalisiert behandelt und temporäre Verbindungsvariablen erstellt. Jeder Client sendet eine Kontrollnachricht mit Informationen welcher Ausprägung dieser angehört. Ebenfalls enthalten sind der übergebene Spielername und die gewählte Rolle. Falls Rollen nur einmal existieren dürfen, findet zunächst ein Abgleich mit den bereits existenten Verbindungen und damit anderen Rollen statt (b), um darauffolgend die Verbindung abzulehnen oder weiter zu behandeln (c), was zur Dekrementierung des Rollenkontingents führt. Sofern der Client akzeptiert wurde, finden die sachgemäßen Initialisierungen und die Einordnung in eine der beiden Kategorien (d,e) statt.

Nach diesem Vorgehen wird dem verbundenen Client ein Steuersignal gesendet, welches diesen dazu veranlasst, in einen Wartebildschirm zu wechseln[397]. Der Game-Engine-Client verfügt in seiner Funktion als Hauptbildschirm über eine detailreichere Ansicht, welche die Spieler darüber informiert, wer bereits verbunden ist[398] und auf wie viele Mitspieler noch gewartet wird, bis das Spiel gestartet werden kann.

Nachdem alle benötigten Spieler und der Game-Engine-Client verbunden sind, wird ein weiteres Steuersignal, in Form einer Nachricht, an alle Clients gesendet, welche letztlich zum Starten des Spiels führt. Ferner wird an dieser Stelle die in Abschnitt 20.0.3 beschriebene Ausführungsschleife für den Game-Engine-Client gestartet.

[397] Die Smartphone-Clients verweilen dabei in der in Abb. 21.3 (3) dargestellten Anzeige.
[398] Dies geschieht durch die Anzeige der Namen, der verbundenen Spieler.

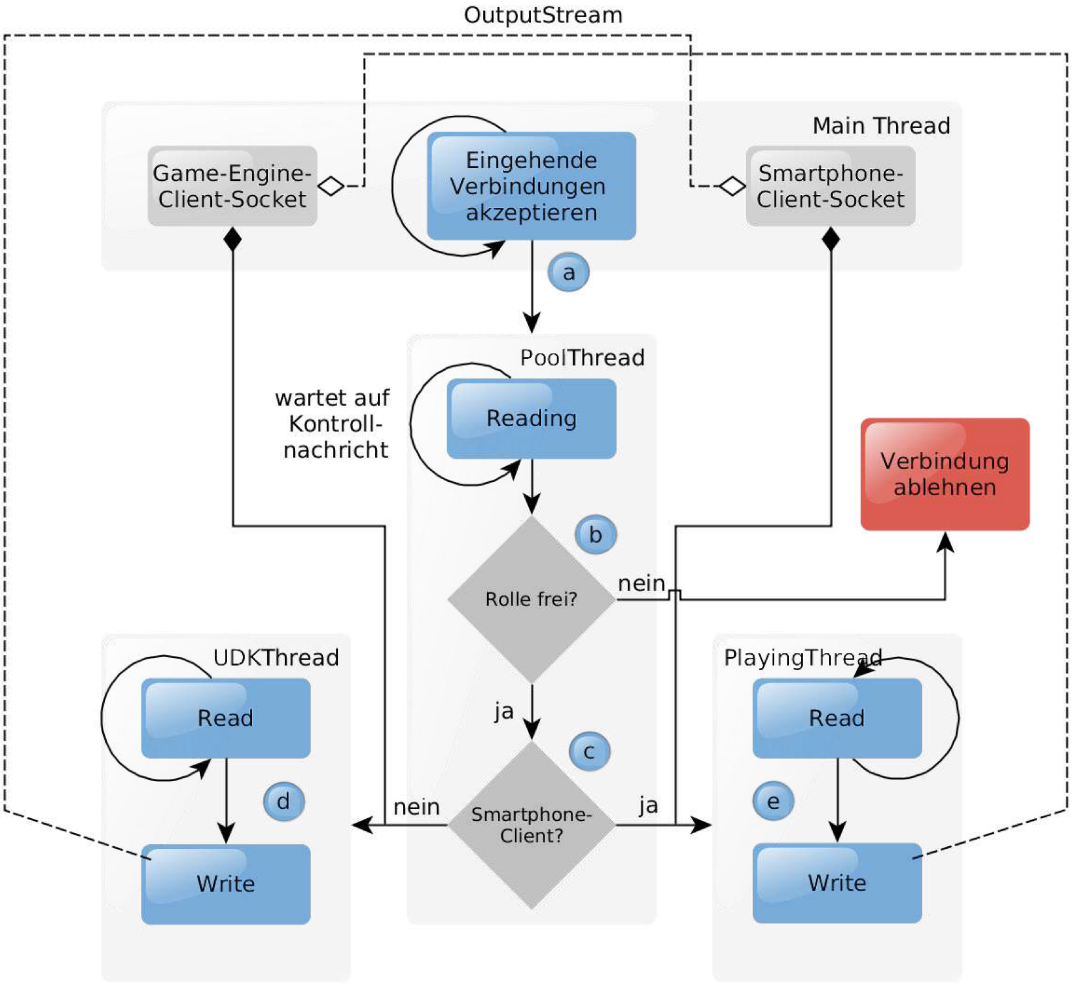


Abbildung 20.2.: Der Server speichert und verwaltet alle eingehenden Verbindungsanfragen sowie darüber hinaus auch alle Threads und Kommunikationsdaten.

„Smartphones sammeln neue Funktionen und Fähigkeiten, als wären sie Staubsauger."

(Sander [Sander, 2011])

KAPITEL 21

Smartphone-Clients

Die Smartphone-Clients wurden drahtlos mit dem Server verbunden und basieren auf dem Betriebssystem Android[399]. Die Wahl dieser Verbindungsart gegenüber einer ebenfalls möglichen Kabelverbindung liegt darin begründet, dass die Anforderungen eines möglichst hohen Engagements und der angestrebten Diffusion der realen und virtuellen Welt eng mit einer möglichst hohen Bewegungsfreiheit der Nutzer verbunden sind. Ein Smartphone besitzt zu wenige Stimuli für einen ausreichend hohen Immersionsgrad, sodass es wichtiger ist, das Engagement und den Spaß eines Spielers so hoch wie möglich zu halten. Dies bedeutet wiederum, dass minimale Latenzzeiten benötigt werden. Eine kabelgebundene Anbindung stand deshalb nicht zur Auswahl, da dies störend empfunden wird und zudem die Ortsungebundenheit der Spieler beeinträchtigt hätte.[400]

In Abbildung 21.1 ist die grundlegende Architektur der Smartphone-Anwendung dargestellt. Den offiziellen Konventionen folgend existieren für alle sichtbaren Teile der Anwendung verschiedene Activities (f), welche wiederum aus unterschiedlichen Views und damit auch Widgets (h) bestehen um bestimmte Aufgabenbereiche zu übernehmen. Somit wird beispielsweise jede Interaktionsform mittels eigener Activity umgesetzt. Allen Activities gemein ist die Tatsache, dass diese Daten an den Server senden müssen. Ob sich dabei um Bewegungsdaten oder andere Arten von Daten handelt, hängt von der zugrundeliegenden Funktionalität ab. Beim Starten des Verbindungsaufbaus wird daher zusätzlich ein Hintergrundservice (g) erstellt, welcher die gesamte Netzwerkfunktionalität kapselt und benötigte Funktionen nach außen freigibt. Die Referenz zum Service wird jeweils von der aktuell sichtbaren Activity gehalten und beim Wechsel mit übergeben. Somit wird das Prinzip der losen Kopplung und hohen Kohäsion ausgenutzt. Dadurch, dass der Service ebenfalls im Hauptthread der Anwendung ausgeführt wird, besitzt dieser mehrere Threads, welche die notwendigen Aufgaben letztlich ausführen.

[399] siehe Abschnitt 2.5
[400] siehe Abschnitt 3.1

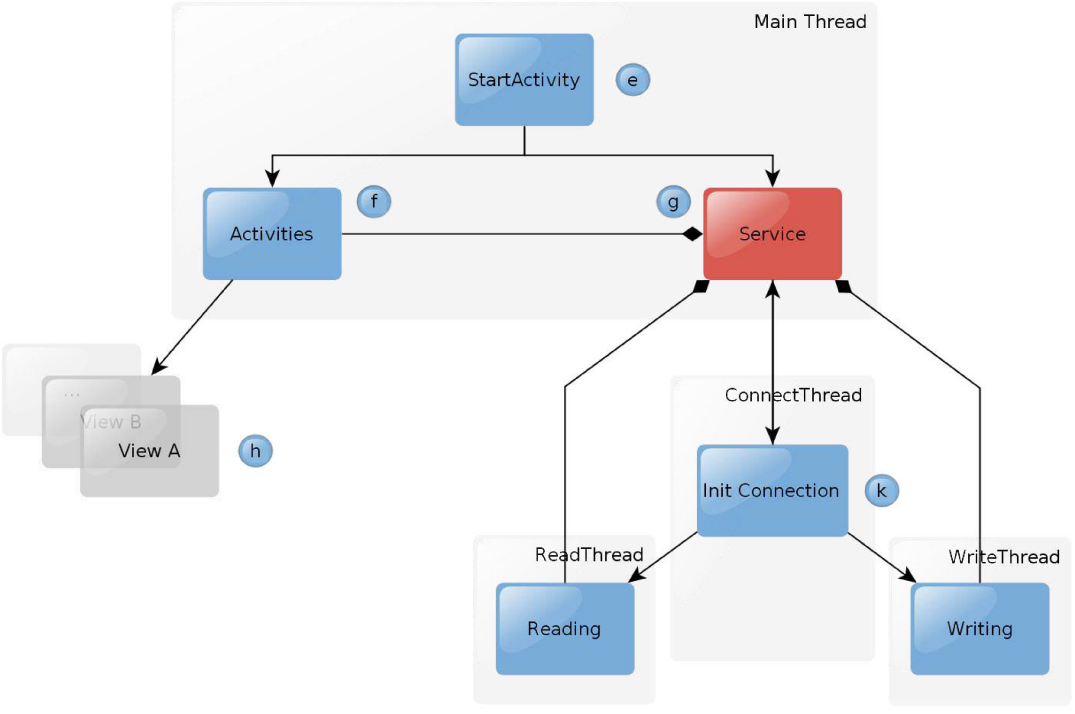


Abbildung 21.1.: Die Architektur folgt den Konventionen des Betriebssystemherstellers Google.

21.0.5. Kommunikation innerhalb der Anwendung

Der Hintergrundservice musste im Rahmen der geplanten Änderungen am meisten erweitert werden. Die gesamte Kommunikationslogik, insbesondere durch die Hinzunahme der Lobbyfunktionalität, erforderten weitreichende Änderungen. Für die gesamte Kommunikation werden *Intents* sowie entsprechende *Receiver* verwendet[401]. Implizite Intents werden dort eingesetzt, wo die Empfängerkomponente nicht bekannt ist oder wenn einer oder mehreren beliebigen Komponenten Daten gesendet werden sollen. Explizite Intents werden in der Regel zum Aufrufen einer anderen Activity verwendet, da der Empfänger hierbei bekannt ist. Durch die Verwendung eines so genannten *Bundles* kann der Intent mit weiteren Informationen ausgestattet werden. Korrespondierend zu den verschiedenen Nachrichtentypen innerhalb der Gesamtanwendung werden verschiedenartige Intent-Typen auf Seiten Androids verwendet.

[401] Wie Abschnitt 2.5.2 erläutert wird, werden Intents für die Verbindung zwischen Activities und sonstigen Komponenten verwendet.

In der Anwendung werden sowohl implizite als auch explizite Intents eingesetzt. Erstere werden verwendet um Activities empfangene Daten des Servers zukommen zu lassen ohne diese explizit aufzurufen. Die Soundausgabe sowie die Vibrationen ereignen sich hauptsächlich in der Bewegungsansicht und sollen im Hintergrund stattfinden, ohne das ein Wechsel des Kontextes stattfindet. Zu diesem Zweck können Receiver eingesetzt werden, welche auf den Empfang entsprechender Intents reagieren und die enthaltenen Daten weiter verarbeiten können. Explizite Intents werden beispielsweise für das Wechseln in die Interaktionsansichten verwendet, da der Empfänger hierbei bekannt ist und der Start einer bestimmten Activity notwendig ist.

21.0.6. Interaktionsansichten

Wie bereits erläutert, werden für die verschiedenen Interaktionen sowie für die Hauptaktivität, der Bewegungssteuerung, eigene Ansichten verwendet. Diese sollen sowohl zur Erweiterung der Virtualität dienen, als auch dem Nutzer intuitiv verständlich aufzeigen, was hier getan werden muss. Anhand des in Abb. 21.2 dargestellten Beispiels soll dies näher erläutert werden.

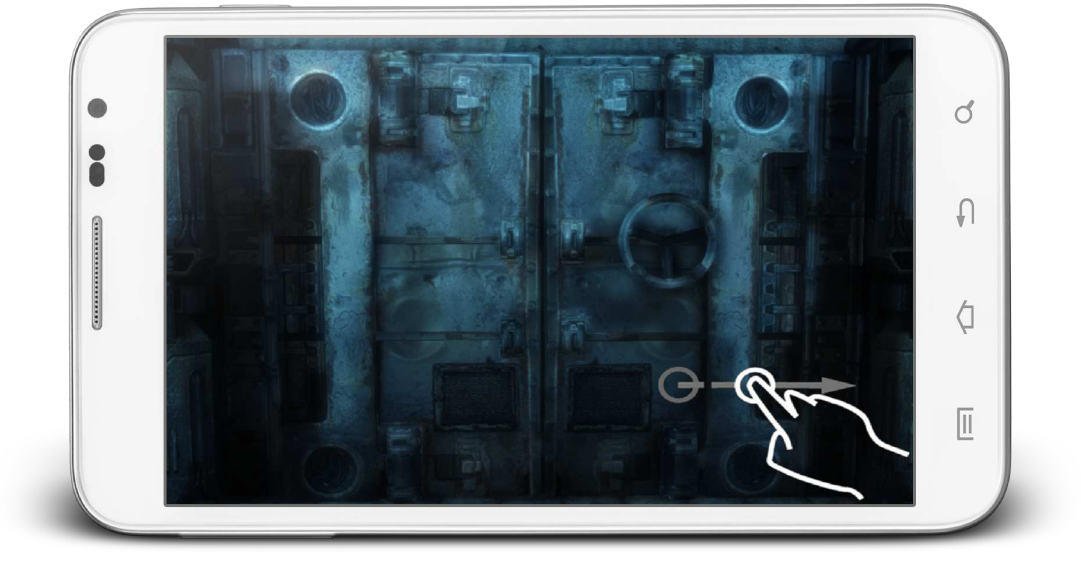


Abbildung 21.2.: Die Grafik zeigt eine Interaktionsansicht auf dem Smartphone.

Für den ersten Aspekt werden Spielansichten des betroffenen Interaktionsziels verwendet. Stehen die Avatare im Spiel vor einer Stahltür, mit welcher sich interagieren lässt, so wird diese aus der Ich-Perspektive auf das Smartphone übertragen und angezeigt. Zur Erreichung des zweiten Aspekts wir

das Bild mit einem zusätzlichen Gesten-Symbol überlagert. In diesen konkreten Fall handelt es sich um eine Wischgeste, welche durch das unten im Bild befindliche Symbol angedeutet wird. Alle verwendeten Interaktionsansichten folgen diesem Schema, um ein einheitliches und in sich geschlossenes Steuerungskonzept zu etablieren.

21.0.7. Erweiterung des Anmeldemechanismus

Auf Seiten der Smartphone-Clients gab es einige Restrukturierungen und eine Codeüberarbeitung vorzunehmen. Hierzu zählen die Erweiterung des Anmeldemechanismus und die damit einhergehende Lobbyfunktionalität. Diese beiden Punkte sind aus diversen Gründen vonnöten gewesen und werden daher im Folgenden genauer erläutert.

Der Anmeldemechanismus mit der zur Verfügung stehenden Möglichkeit der Eingabe einer Server-IP sollte um die Eingabemöglichkeit eines Spielernamens sowie einer zu spielenden Rolle erweitert werden (siehe Abb. 21.3 (1)). Der Name soll zur Individualisierbarkeit der Avatare und als Kennzeichnung für die Spieler verwendet werden. Die Rolle hingegen nimmt einen wichtigen Stellenwert im Rahmen der zu entwickelnden Anwendung ein.

Mit der Wahl einer entsprechenden Rolle gehen gewisse Fähigkeiten respektive Einschränkungen einher. Ein physisch gut aufgestellter Charakter kann beispielsweise im Spiel ausschließlich für körperliche Interaktionen wie das Wegschieben eines Objektes eingesetzt werden, wohingegen technisch versierte Charaktere die Möglichkeit besitzen, Schaltzentralen bedienen können. Für diese verschiedenen Interaktionsmöglichkeiten müssen seitens der Smartphone-Clients entsprechende Views realisiert werden.

Um das gewünschte Ziel zu erreichen, ging es im ersten Schritt darum, die Verbindungsroutine dahingehend zu ändern, dass die benötigten Parameter mit übertragen und auf Serverseite geprüft werden. Insbesondere bei den Rollen besteht die Möglichkeit, dass diese im gesamten Spiel nur ein Mal existieren dürfen, worüber die Spieler informiert werden müssen. Dadurch, dass ein JSON-Objekt als Übertragungsformat verwendet wird, ist eine Erweiterung der gesendeten Daten ohne Weiteres möglich. Auf der Empfängerseite müssen diese gesondert abgefragt sowie weiter behandelt werden. In Abb. 21.3 ist der Ablauf des Verbindungsaufbaus schematisch dargestellt.

Durch das Bestätigen des Buttons werden die eingegebenen Informationen an den Server gesendet, welcher die Verbindung entgegennimmt und eine Prüfung vornimmt. Sofern die Rolle bereits vergeben ist, wird dem Benutzer ein entsprechender Hinweis angezeigt und die Verbindung abgelehnt (2). Erst wenn alle Daten ordnungsgemäß eingegeben und akzeptiert werden, gelangt der Spieler zu einer neuen Ansicht, dem Wartebildschirm (3).

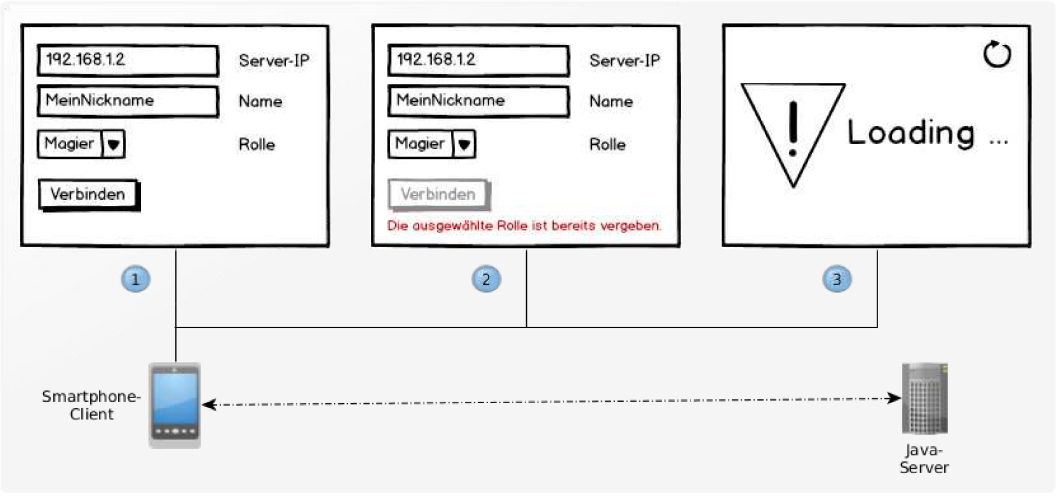


Abbildung 21.3.: Verbindungsaufbau der Smartphone-Clients.

Ferner fehlte es dem vorangegangenen Prototypen an einer geeigneten Lobbyfunktionalität in Form eines Wartebildschirms, welcher die Spieler und Spielerinnen darüber in Kenntnis setzt, dass Mitspieler oder der Game-Engine-Client fehlen und das Spiel noch nicht gestartet werden kann. Nachdem ein Spieler den Anmeldedialog abgeschlossen hatte, bekam dieser bereits die funktionsfähige Bewegungsansicht angezeigt. Daten wurden ab diesem Zeitpunkt an den eventuell noch nicht vollständig initialisierten Server respektive Game-Engine-Client gesendet. Dies führte unter anderem zu folgenden Problemen:

- das Netzwerk wird unnötig belastet;
- die Daten können von dem Server nicht regulär verarbeitet werden, was eventuell zu Fehlern führt;
- die Daten behindern den Game-Engine-Client in dessen Initialisierungsphase und
- den Spielern fehlt ein konkretes Feedback, da die bereits angezeigte Bewegungsansicht zu keiner Funktionalität führt.

Im Zuge des Refactorings wurde der Smartphone-Client um einen Wartebildschirm inklusive technischer Restriktionen erweitert. Nach erfolgreichem Verbindungsaufbau erfolgt vom Server die Sendung eines Steuersignals, in Form einer Nachricht, welches die Ansicht wechselt und das Senden von Nachrichten zunächst blockiert. Erst nach erfolgreichem Verbinden aller benötigter Komponenten und ab-

geschlossener Initialisierung wird ein weiteres Steuerungssignal an die Smartphone-Clients gesendet, welches auf die Bewegungsansicht umschaltet und zudem den Datenfluss freigibt.

*„What **is** the Unreal Engine, exactly? In short, it's a system that organizes your „assets" - your characters, artwork, props, weapons, music, sound effects, voiceovers, etc. - into a visually stunning interactive environment."*

(Busby u. a. [Busby u. a., 2010, S. 14])

KAPITEL 22

Game-Engine-Client

Der Game-Engine-Client ist für die Darstellung der Spielwelt zuständig und besteht aus einer Anwendung, die mit dem Unreal Development Kit entwickelt wurde[402]. Die Verbindung des UDK mit dem Server findet über das Loopback-Device statt, um die Anwendung ebenfalls als Client im Sinne der verwendeten Client-Server-Architektur verwenden zu können. Durch die Entscheidung diese beiden Komponenten auf der gleichen Maschine laufen zu lassen, wird zusätzliches Datenaufkommen und damit möglicherweise einhergehende Paketverluste verringert respektive ausgeschlossen. Zusätzlich ergibt sich durch diese Designentscheidung die Möglichkeit, Server und Game-Engine-Client ohne Mehraufwand aufzutrennen und auf verschiedenen Maschinen laufen zu lassen. Auf diese Weise könnten mehrere, sogar verschiedene Game-Engine-Clients parallel verwendet werden, beispielsweise für die gleichzeitige Nutzung einer Leinwand und eines zentralen Displays als Ausgabemedien.

Damit die Logik zwischen Game-Engine und Server entkoppelt und nicht doppelt vorgehalten wird, teilt der Game-Engine-Client dem Server vor der Ausführung einer Interaktion in der Spielwelt Informationen über die korrespondierende Funktion inklusive Parametern mit. Bei korrekter Ausführung der Interaktion werden per RPC direkte Manipulation in der Spielwelt angestoßen. Auf diese Weise kann der Game-Engine-Client jederzeit modular und unabhängig vom Server erweitert werden, da seine zugrundeliegende Logik und der Funktionsumfang jeweils zur Laufzeit und in der aktuellen Version an den Server kommuniziert werden; er liefert damit im Kommunikationsprotokoll Informationen über seinen eigenen Funktionsumfang[403].

In diesem Kapitel werden spezifische Probleme und Lösungen näher beschrieben, die bei der Implementierung des Game-Engine-Clients auftraten. Dabei wird zunächst auf die Realisierung des Gameplays eingegangen, bei der die Möglichkeit geschaffen werden musste, mehrere Avatare gleichzeitig von einer Instanz der Engine darstellen zu lassen und die Kontrolle über diese zu ermöglichen. Ein weiteres komplexes Feld öffnete sich auf Seiten der Kamera-Implementierung. Hier musste eine Möglichkeit

[402] vgl. auch Abschnitt 2.2.2 auf Seite 24
[403] vgl. auch Abschnitte 3.2.5 bis 3.2.7

gefunden werden, eine intelligente und sinnvolle Art der isometrischen Darstellung zu realisieren, welche das Geschehen auf unterstützende Art einfängt und darstellt[404].

22.1. Grundlegende Implementierung des Gameplays

Die liegende *Unreal-Engine* des UDK wurde in ihrer ersten Version im Jahr 1998[405] zusammen mit dem Ego-Shooter *Unreal*[406] veröffentlicht. Obwohl mit dem UDK in seiner heutigen Form die Möglichkeit besteht, beliebige 3D- und sogar 2D-Spiele zu entwickeln, ist die Unreal-Engine nach wie vor primär auf First-Person respektive Third-Person-Spiele ausgerichtet[407]. Dies schlägt sich sowohl in der ausgelieferten Code-Struktur als auch in den Demonstrationsspielen nieder, bei denen es sich um Ableger der Ego-Shooter-Reihe *Unreal Tournament* handelt.

Die Anforderung an den Game-Engine-Client sieht vor, dass mehrere Spieler *gleichzeitig* mit einer Instanz der Game-Engine interagieren können, was mit anderen Worten bedeutet, dass mehrere Avatare gleichzeitig auf *einem* Ausgabegerät sichtbar sind und von mehreren Leuten gesteuert werden können.

Die Konstruktion der Unreal-Engine bietet zwar eine umfangreiche Unterstützung für Multiplayer-Spiele, hierbei wird jedoch davon ausgegangen, dass jeder Avatar von einer eigenen Instanz der Engine gesteuert wird. Ein Splitscreen-Verfahren kann zwar mit verhältnismäßig einfachen Mitteln auf einer physikalischen Maschine realisiert werden, indem mehrere Instanzen der Engine logisch getrennt verschiedene Bereiche eines Ausgabemediums ansteuern[408], jedoch stellt dies aus zwei Gründen keine Option dar, die näher in Betracht gezogen wurde: Zum einen ist ein Splitscreen-Verfahren unerwünscht, da es die Spielenden innerhalb der Spielwelt voneinander trennt[409] und zum anderen werden damit tatsächlich zwei Instanzen desselben Spiels auf einer Maschine gestartet, was einen erheblichen Overhead für die zu implementierende Netzwerkkommunikation darstellen würde. So müsste nicht nur der Server mit einem Game-Engine-Client kommunizieren, sondern mit mehreren Clients gleichzeitig und diese wiederum untereinander.[410]

Die strikte Auslegung der Game-Engine auf nur einen „kontrollierenden Menschen“ schlägt sich ebenfalls in der Logik der zugrundeliegende Codebasis nieder. Die Unreal Engine ist so aufgebaut, dass die Signale eines Eingabegeräts die Bewegungen genau eines Avatars durch die Spielwelt steuern und

[404] siehe Kapitel 23 ab Seite 305
[405] vgl. [Busby u. a., 2010, S. 2]
[406] vgl. [Busby u. a., 2010, S. 2]
[407] Umgangssprachlich ausgedrückt: Die Wurzeln der Unreal-Engine als Ego-Shooter-Grundgerüst lassen sich schwerlich leugnen.
[408] vgl. [wiki.beyondunreal.com, 2009]
[409] Nähere Informationen hierzu befinden sich unter anderem im Glossar-Eintrag auf Seite 359.
[410] siehe Seite 87f

damit gleichzeitig Rotation und Translation an die dazugehörige Kamera-Klasse übermittelt. Grundsätzlich ergaben sich so drei Teilprobleme respektive Anforderungen:

- Nicht ein Mensch steuert einen Avatar, sondern viele Menschen steuern viele Avatare.
- Nicht ein Mensch steuert über die Bewegungen des Avatars eine Kamera, sondern viele Menschen steuern gemeinsam über die Bewegung ihrer individuellen Avatare eine *gemeinsame* Kamera.
- Als Interaktionstechnologie wird das Konzept einer Computermaus repliziert. Jedoch gibt es nicht nur eine Maus, sondern in Abhängigkeit zu der Anzahl der Spielenden n Mäuse mit n darzustellenden Mauszeigern.

Die Grundlage zur Lösung dieser Probleme wurde im Masterprojekt gelegt, auf dem die vorliegende Arbeit und der damit verbundene Prototyp aufsetzen. Bei dieser Basisimplementierung war die Anzahl der Spielenden auf insgesamt zwei begrenzt, um die grundsätzliche Möglichkeit einer Lösung zu demonstrieren (vgl. Abbildung 22.1 „Vorher“). Als erstes wurde die Kamera von der standardmäßigen First-Person-Perspektive in eine isometrische Ansicht überführt.

Daraufhin ist ein Bewegungskonzept realisiert worden, bei dem ein, über das Netzwerk vom Server empfangener, Datenstrom die Koordinaten in X- respektive Y-Richtung für die darzustellenden Mäuse liefert. Diese Koordinaten wurden in Welt-Koordinaten umgerechnet, um am jeweiligen Schnittpunkt mit der Weltgeometrie perspektivisch korrekt eine Markierung zu platzieren. Bei dem „ersten“ Spieler handelte es sich um den repräsentativen Avatar innerhalb des UDKs, an welchen die gesamte Logik gebunden ist. Ein Tod dieses Hauptcharakters führt demnach zur Beendigung des Spiels. Ebenfalls veränderte die Bewegung dieses Avatars auch die Position der Kamera, da der Bezugspunkt selbiger von der Position dieses Hauptcharakters abhing. Die Darstellung eines zweiten Avatars und die Schaffung einer Möglichkeit, diesen durch einen weiteren realen Spieler steuern zu lassen, führte zu einigen Problemen. Da es nicht möglich war, der Game-Engine einen weiteren, lokalen Spieler hinzuzufügen, musste eine andere Möglichkeit gefunden werden. Die Lösung war die Integration eines sogenannten Bots[411], eines für gewöhnlich durch den Computer gesteuerten Charakters. Für diesen Bot konnte die Steuerung, mit Hilfe eines Smartphones, durch einen weiteren Menschen realisiert werden, indem die Intelligenz des Bots darauf angepasst wurde, die durch die Smartphones induzierten Steuerungsbefehlen zu befolgen.

Die Integration eines „echten“ Spielers erzeugte einige Restriktionen, welche eine sinnvolle Implementierung und Ausweitung auf mehr als zwei Spielende im Rahmen dieser Arbeit verhinderten. So wäre erforderlich gewesen, dass jeder weitere Mitspieler eine tatsächliche, durchnummerierte Klasse

[411] vgl. NPC

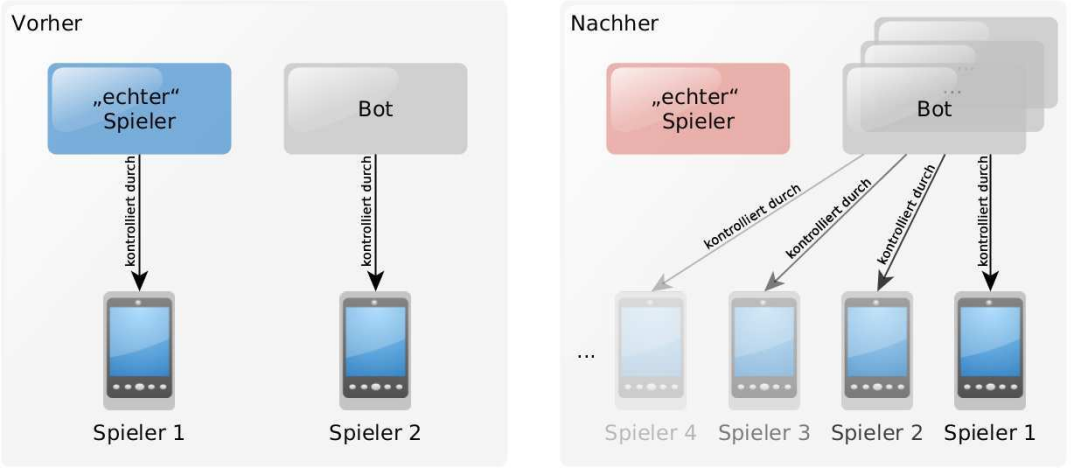


Abbildung 22.1.: Die Abbildung visualisiert das Prinzip der Spieler-Replikation vor und nach der Überarbeitung des Codes. Im Masterprojekt war das System auf eine Gesamtanzahl von zwei Spielern begrenzt, wobei ein Spieler die Kontrolle über die obligatorische „echte“ Spieler-Repräsentation des UDK übernahm, während ein weiterer Spieler einen Bot steuerte. Nach der Überarbeitung des Codes war sowohl die Sichtbarkeit als auch die Kollision des „echten“ Spielers in der Game-Engine abgeschaltet, während die Spielenden nur noch Bots durch die Spielwelt leiten. Dieser obligatorische, „echte“ Spieler existiert nach wie vor, verfügt jedoch über keine wahrnehmbare Repräsentation in der Spielwelt mehr und ist deshalb im rechten Teilbild rot hinterlegt.

im Code erhält. Auf diese Weise wäre viel redundanter Code entstanden, der das System einerseits ausgebremst und andererseits massiv limitiert hätte, da die Anzahl der Spielerinnen und Spieler im Vorhinein fest einprogrammiert hätte werden müssen. Ein weiteres Gegenargument, diesen Ansatz unhinterfragt aufzugreifen und weiterzuverfolgen, ergab sich durch den Umstand, dass sich ein „echter“ Spieler im Code anders verhält als ein „Bot“. Das ist beispielsweise für das Auslösen von Interaktionen von Bedeutung, wo eine ständige Überprüfung hätte stattfinden müssen, ob ein „Bot“ oder der (einzige) „echte“ Spieler einen Interaktionsbereich betreten hat.

Da das Unreal Development Kit vorsieht, dass zwangsläufig die Repräsentation eines echten Spielers in der Spielwelt vorliegen muss, wurde hierfür zunächst eine triviale wie auch effektive Lösung gewählt: Es gibt einen echten Spieler, dessen Kollisionen und Sichtbarkeit abgeschaltet wurden, wodurch die wahrnehmbare Repräsentation dieses Spielers innerhalb der Spielwelt faktisch eliminiert wird. Anschließend ist eine komplette Neuentwicklung der Steuerungs- und Repräsentationslogik als sich selbst erweiternde und modulare Struktur vorgenommen worden: Sobald sich ein neuer Spieler

verbindet, wird automatisch eine Spielfigur samt dazugehörigem Mauszeiger erschaffen und in die Verwaltung des Spiels übernommen. Die Repräsentationen der Spielenden im Code finden somit an zentraler Stelle statt, während gleichzeitig die Maximalanzahl möglicher Spieler faktisch nur noch durch die Ressourcen des Host-Systems begrenzt wird (vgl. Abbildung 22.1 „Nachher“).

22.2. Realisierung der Navigation

Nachdem eine Lösung dafür erdacht wurde, wie mehrere Spielerinnen und Spieler in sinnvoller Art und Weise unterstützt werden können, musste ein Konzept gefunden werden, die Bots[412] mit einer gewissen Intelligenz auszustatten, um die Navigation in der virtuellen Welt zu ermöglichen und zu unterstützen. Wegfindung ist eine der am häufigsten benötigten und wiederkehrenden Aufgaben in Spielen und wird vom UDK auf mehrere, verschiedene Arten unterstützt. Einer der trivialsten, gleichzeitig jedoch auch effektivsten Möglichkeiten stellt hierbei die Nutzung von sogenannten Wegpunkten dar (vgl. Abbildung 22.2).

Abbildung 22.2.: Die Abbildung zeigt einen Ausschnitt des Wegpunktnetzes.

[412] vgl. vorangegangener Abschnitt

Wegpunkte können innerhalb des UDK manuell gesetzt werden und dienen als Anhaltspunkt für begehbare Wege innerhalb der virtuellen Welt. Da die eingesetzten Bots per se keine „Vorstellung“ von der umgebenden Level-Geometrie haben, kann unter Zuhilfenahme von Wegpunkten eine Art Netz konstruiert werden, dessen Linien fortan als Bewegungspfade dienen, wobei die Wegpunkte selbst als Knoten des Netzes dienen (vgl. [Kopp, 2011, S. 50]). Durch die Knotendichte kann das Netz verfeinert werden, was eine „natürlichere“ Navigation erlaubt – auf der anderen Seite sollten jedoch nur so viele Wegpunkte wie nötig eingesetzt werden, um jeden Ort, der von einer künstlichen Intelligenz betreten werden kann, abzudecken, da auf diese Weise weniger Berechnungsschritte für die eigentliche Wegfindung durchgeführt werden müssen. Dem Anspruch, möglichst wenige Knoten verteilen zu müssen, kommt dabei zugute, dass die Wegpunkte an sich kein starres Konstrukt darstellen, sondern gewisse Informationen über die sie unmittelbar umgebende Geometrie mit abspeichern. Auf diese Weise kann nicht nur zwischen einzelnen Wegpunkten navigiert werden, sondern auch zu ihnen naheliegenden Punkten. [Epic Games, Inc., 2014b]

Jeder mit dem System verbundene Spieler erzeugt nicht nur einen Avatar und einen korrespondierenden Mauszeiger, sondern eine komplette Instanz einer State Machine, bei dem der jeweilige Avatar den aktuellen Bewegungsstaus abspeichert: Bewege ich mich, suche ich gerade einen Weg zu einem Ziel oder: Wenn ich einen Weg zu einem Ziel gefunden habe, zu welchem Wegpunkt muss ich mich als nächstes bewegen, um zum Ziel zu gelangen?

Letztlich unterstützt das realisierte Steuerungsprinzip zwei Modi: die automatische Suche eines Weges und eine direkte Steuerung. Bei der direkten Steuerung, die durch ein dauerhaftes Drücken des „Run“-Buttons[413] auf den Smartphones initiiert wird, wird kontinuierlich die Position des zu einem Avatar zugehörigen Mauszeigers abgefragt und der Avatar selbst angewiesen, eine Animation abzuspielen und sich zu dieser Position zu bewegen. Dieser Versuch findet auch dann statt, wenn sich ein geometrisches Objekt zwischen der aktuellen Position und der des Mauszeigers befindet und der Avatar dadurch kontinuierlich „gegen eine Wand läuft“. Zwar wäre dieses Manko leicht zu beheben, benötigt jedoch zusätzliche Rechenzeit in Form von periodischen Raycasts[414] und wird daher an dieser Stelle nicht weiter beachtet. Bei der automatischen Suche, die durch kurzes Antippen des „Run“-Buttons angestoßen wird, versucht die korrespondiere Wegfindungsautomat des jeweiligen Avatars einen Weg zu finden, der zu einem Wegpunkt in der Nähe des jeweiligen Mauszeigers führt. Wird ein solcher Weg gefunden, wechselt der Avatar in einen automatischen Modus und bewegt sich von Wegpunkt zu Wegpunkt, bis die Distanz zum anvisierten Zielort eine fest vorgegebene Distanz unterschreitet[415].

[413] siehe Abb. 19.4 auf Seite 281

[414] Raycasting steht, vereinfacht ausgedrückt, als Begriff für ein einfaches Abtasten einer dreidimensionalen Szene aus gegebener Position unter Verwendung eines oder mehrerer Strahlen.

[415] Diese Distanz lässt Varianzen zu, die in der realen Welt etwa drei bis zehn Metern entsprechen würden.

Der Vorteil in der Verwendung von Wegpunkten liegt hauptsächlich in der absoluten Kontrolle des Designers, welche Orte der virtuellen Welt tatsächlich „durchsuchbar“ gestaltet werden und welche nicht. Dies kann hilfreich sein, um bestimmte Pfade zu erzwingen, was insbesondere sinnvoll ist, wenn gewisse Trigger-Zonen in jedem Fall durchlaufen werden sollen. Es existiert allerdings eine weitere, effiziente und weitaus modernere Variante des Pathfinding im UDK, die gerade für vollständig modulare Umgebungen nicht nur sinnvollerweise, sondern auch notwendigerweise zum Einsatz kommen müsste. Dabei handelt es sich um sogenannte *Navigation Meshes*[416], deren Erzeugung eine automatische Analyse der Levelgeometrie vorausgeht. Navigation Meshes replizieren das Prinzip des Marching-Cube-Algorithmus[417] und nutzen große, zusammenhängende Polygone für weite, offene Flächen und werden in Nachbarschaft filigraner Geometrie feiner unterteilt [Epic Games, Inc., 2014b]. Dieses Prinzip kam im Rahmen des realisierten Prototyps jedoch nicht zum Einsatz, da diese Technologie ungleich mehr initialen Testaufwand erfordert, weil ein Großteil der Kontrolle über mögliche gefundene Wege abgegeben wird. Da der Fokus dieser Ausarbeitung nicht unbedingt auf einer perfekten Navigation der Avatare durch eine möglichst komplizierte Umgebung liegt, wurde der Ansatz des Pathfinding, wie beschrieben, im Sinne des implementierten Spiels als ausreichende Lösung angesehen.

[416] Eine ausführliche Erklärung der Navigation Meshes ist in [Epic Games, Inc., 2014b] zu finden.
[417] Eine vereinfachte Darstellung mit Abbildungen ist bei Interesse ebenfalls in der Dokumentation des UDK unter [Epic Games, Inc., 2014b] zu finden.

„Implementing a camera other than first person usually requires having the camera offset from the player mesh, among other aspects that may need tuning.“

(Epic Games, Inc. [Epic Games, Inc., 2014a])

KAPITEL 23

Realisierung des gewählten Kamera-Systems

Wie in Abschnitt 16.2 begründet wurde, eignet sich die Umsetzung einer isometrische Darstellung für die Zwecke der anvisierten MR-Anwendung insbesondere in Hinblick auf die Digitalisierung eines Pen-&-Paper-Rollenspiels am ehesten. In diesem Abschnitt wird daher die konkrete Umsetzung des Kamerasystems beschrieben, wobei auch gezeigt wird, dass die technische Realisierung in direkter Art und Weise mit dem demokratischen Spielprinzip des Pen-&-Paper-Rollenspiels verknüpft ist.

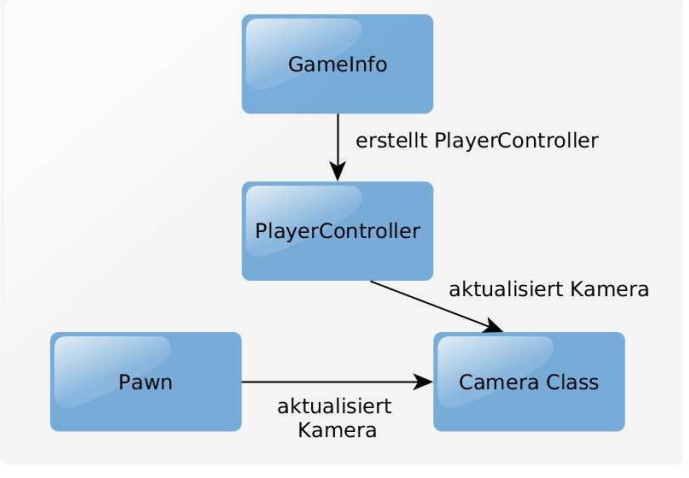


Abbildung 23.1.: Die Abbildung visualisiert die Verschachtlung und Rolle der beteiligten Klassen, die im UDK einen Einfluss auf die Kamera-Berechnungen nehmen. Die Klasse *GameInfo* ist hierbei als Startpunkt und Hauptklasse des gesamten Spiels zu betrachten.

Eine isometrische Kameraperspektive ist vom Fokus auf eine bestimmte Spielfigur unabhängig[418] und erlaubt auf diese Weise die Implementierung einer Übersicht, welche auch mehrere Spielende erfassen kann. Da sich die Spielenden jedoch innerhalb der Spielwelt fortbewegen, benötigt die Kamera einen kontinuierlich errechneten Bezugspunkt, dem diese folgen kann. Daher musste zunächst eine Möglichkeit gefunden werden, diesen Bezugspunkt mathematisch zu bestimmen. Hierbei wurden auch die Erkenntnisse aus Abschnitt 6.4 berücksichtigt, in denen gezeigt wird, dass die Kamera technisch unterstützend wirken soll und die Spielenden nicht behindert.

Das Kamerasystem des UDK setzt sich aus drei involvierten Komponenten zusammen: *Camera*, *Pawn*[419] und *PlayerController*. Diese Komponenten treten miteinander in Wechselwirkung, um die eigentliche Kamerabewegung und eventuell damit verbundene Spezialeffekte zu steuern [Epic Games, Inc., 2013b]. Durch die historisch bedinge Standard-Ansicht der *First-Person-Kamera*[420], welche im UDK verwendet wird, müssen diese drei Komponenten verändert respektive erweitert werden, um eine neue Kameraperspektive zu implementieren. Die Zusammenhänge zwischen den im Folgenden beschriebenen Komponenten wurden ebenfalls in Abbildung 23.1 visualisiert.

Die Klasse „Camera“ stellt die Repräsentation des View Frustum dar und bildet damit die Code-technische Repräsentation des Blicks auf die Spielwelt. Die Position und Rotation der Kamera definiert, was auf dem Wiedergabemedium gerendert und tatsächlich zu sehen ist [Epic Games, Inc., 2013b]. An dieser Stelle ist es ebenso möglich, verschiedene Parameter der Kamera festzulegen, wie das Bildseitenverhältnis, das Field of View (FOV) oder verschiedene Spezialeffekt wie Post-Processing oder Kamera-Animationen.

Die Klasse „Pawn“ repräsentiert den Charakter der Spielwelt im Code. Durch die tatsächliche Position der Spielfigur in der Spielwelt kann die Kameraposition beeinflusst werden, beispielsweise kann die Translation des Spieler-Modells in eine Translation der Kamerabewegung übersetzt werden.

Die Klasse „PlayerController“ letztlich hält sowohl eine Referenz auf die *Camera* als auch auf den *Pawn* vor. Hier werden die Eingabesignale, beispielsweise Tastaturkommandos oder Mausbewegungen, verarbeitet und sowohl Position als auch Rotation des mit dem *PlayerController* verbundenen *Pawns* aktualisiert [Epic Games, Inc., 2013b]. In der Voreinstellung bestimmt die Rotation des Avatars ebenfalls die Kamera-Rotation. Durch das strikte Vererbungsprinzip der Unreal-Engine ist es daher nötig, diese Funktionalität zu überschreiben, damit die Kamerarotation nicht mehr durch die Bewegungen einer Spielfigur bestimmt wird [Epic Games, Inc., 2013b].

[418] im Gegensatz zur *First-Person-Perspektive* beispielsweise
[419] englischer Ausdruck für „Spielfigur“ [LEO Dictionary Team, 2013]
[420] vgl. Abschnitt 2.2.2 auf Seite 24

Es gibt es zwei gängige Wege im UDK, um eine neue Kamera-Funktionalität zu implementieren. Die erste Methodik sieht dabei vor, eine bestimme Funktion[421] zu erben und zu modifizieren, woraufhin grundlegende Modi und einfache Manipulationen der Kameralogik ermöglicht werden. Die zweite Methodik sieht vor, eine komplett eigene Kamera-Klasse zu definieren und dabei gewünschte Funktionalitäten zu erben und erweiterte Funktionalitäten hinzuzufügen [Epic Games, Inc., 2013b]. Zuletzt genannter Weg erfordert tiefere Eingriffe in die Code-Struktur des UDK, erlaubt jedoch eine völlige Freiheit der Umsetzung und gibt gleichzeitig den Zugriff auf tiefere Code-Strukturen, wie Kamera-Animationen und Post-Processing-Effekte, frei. Um alle Anforderungen einer isometrischen und gleichzeitig demokratischen Kamera[422] umzusetzen, musste auf diese zweite Möglichkeit zurückgegriffen werden. Insgesamt bestehen damit folgende Anforderungen an das Kamerasystem:

- Das implementierte Kamerasystem soll isometrisch sein.
- Das System soll sich „demokratisch“ im Sinne eines Pen-&-Paper-Rollenspiels verhalten und die Technik dem Spielprinzip unterordnen. Dies schließt einen festen Bezugspunkt in Form einer bestimmten Spielfigur aus.
- Ferner soll sich das System adaptiv an die Spielsituation anpassen: Wenn mehr Übersicht geboten ist, muss die Kamera aus dem Spielgeschehen heraus zoomen, andererseits muss selbige für Detail-Ansichten an das Geschehen herangeführt werden.

Der erste Punkt wurde im vorangegangenen Projekt unter Verwendung einfacher Matrizenmultiplikationen bereits umgesetzt, sodass der folgende Abschnitt einen Fokus auf den aufwändigeren Teil der Umsetzung legt, namentlich die Berechnung eines im Sinne des Pen-&-Paper-Rollenspiels demokratischen Kamera-Bezugspunkts.

23.1. Berechnung des Kamera-Bezugspunkts

Das Ziel war es, ein demokratisches Berechnungsprinzip zu realisieren. Das bedeutet, dass der errechnete Point of Interest (POI) letztlich den mehrheitlichen Richtungsentscheidungen der Gemeinschaft folgt, während alle Avatare im Bildbereich eingefangen werden.

Abbildung 23.3 visualisiert unter anderem das Ergebnis einer einfachen Methode, bei welcher der Bezugspunkt (Punkt **b**) als absolute Mitte berechnet wird. Hierfür dient eine finite Menge von Punkten $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$ in $\mathbb{R}^n$, als Berechnungsgrundlage. Der Mittelpunkt kann somit durch eine simple Gleichung bestimmt werden (vgl. Abbildung 23.2) [TR Jain, 2006, Abschnitt 1.3]:

[421] namentlich die *CalcCamera*-Funktion[Epic Games, Inc., 2013b]
[422] Was unter einer „demokratischen“ Kamera zu verstehen ist, wird in Abschnitt 23.1 auf Seite 307 genauer erläutert.

$$\mathbf{C} = \frac{\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_k}{k}$$

Abbildung 23.2.: Berechnung des absoluten Mittelpunkts einer finiten Menge von k Punkten $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$ in $\mathbb{R}^n$.

nach [TR Jain, 2006, Abschnitt 1.3]

Der auf diese Weise errechnete Punkt minimiert die euklidische Distanz zwischen sich selbst und jedem Punkt der übergebenen finiten Menge. Die für die Berechnung notwendige Punktmenge kann den Koordinaten aller vorhandener Spielerpositionen entnommen werden und verlangt keine spezielle Ordnung. Abbildung 23.3 illustriert jedoch gleichzeitig ein Problem, welches mit dieser simplen Abstraktionsmethode einhergeht: Ein auf diese Weise ermittelter Bezugspunkt ist nicht demokratisch.

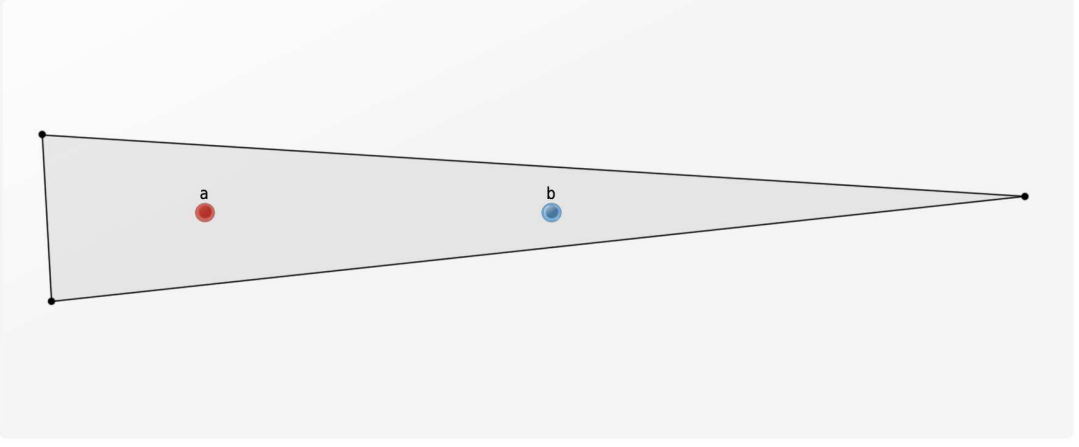


Abbildung 23.3.: Die Abbildung visualisiert die Unterschiede bei der Berechnung eines Kamera-Bezugspunkts in Abhängigkeit der gewählten Methode. Wenn die Eckpunkte des Dreiecks als Spielerpositionen aus der Draufsicht betrachtet werden, so stellt Punkt **a** einen Balance-Punkt dar, dem ein gewisses demokratisches Verhalten attestiert werden kann, bei dem die Position der Mehrheit der Spielenden die Kamera in ebendiese Richtung „zieht“. Dem gegenüber steht der mit einfachen Methoden errechnete Punkt **b**, der einfach nur die Mitte zwischen allen Positionen darstellt.

Wenn drei Spielende in einem asymmetrischen Dreieck zueinander stehen[423], liegt der errechnete Bezugspunkt der Kamera in der absoluten Mitte zwischen allen Positionen, da dort die Distanz zu den

[423] In Abbildung 23.3 stellen die Eckpunkte des Dreiecks die Spielerpositionen dar.

Eckpunkten minimiert wird. *Wünschenswert* wäre allerdings ein Punkt, bei dem eine Mehrheitsentscheidung (in diesem Fall die Entscheidung für eine bestimmte Richtung) die Errechnung des Bezugspunktes beeinflusst. Wenn zwei von drei Spielenden in eine Richtung gehen, während sich der dritte Charakter in die Gegenrichtung bewegt, sollte die Mehrheit der Spielenden der Kamera eine Art Gewicht verleihen und den Bezugspunkt in diese Richtung verschieben (Abb. 23.3 Punkt **a**).

Eine solche Gewichtung lässt sich durch die Uminterpretation der Position der Spielenden von einer Liste an Punkten in die Eckpunkte eines konvexen Polygons erreichen. Der geometrische Schwerpunkt eines sich selbst nicht schneidenden Polygons mit n Eckpunkten (x_0, y_0), $(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$ ist definiert als Punkt (C_x, C_y), für den die in Abbildung 23.4 dargestellten Formeln gelten [Bourke, 1988].

$$C_{\mathrm{x}} = \frac{1}{6A} \sum_{i=0}^{n-1} (x_i + x_{i+1})(x_i\ y_{i+1} - x_{i+1}\ y_i)$$

$$C_{\mathrm{y}} = \frac{1}{6A} \sum_{i=0}^{n-1} (y_i + y_{i+1})(x_i\ y_{i+1} - x_{i+1}\ y_i)$$

Abbildung 23.4.: Die dargestellten Formeln zeigen die Berechnung des geometrischen Schwerpunkts (auch *Centroid*) eines sich selbst nicht schneidenden, im Uhrzeigersinn aufsteigend sortierten Polygons in Abhängigkeit vom Flächeninhalt[424].
nach [Bourke, 1988]

Diese Formeln setzen ein im Uhrzeigersinn sortiertes, sich selbst nicht schneidendes Polygon voraus. Aus dieser Anforderung resultieren zwei Entscheidungen. Die erste Entscheidung ist die, dass die Umsetzung mit dieser Berechnungsmethode auf vier Spielende begrenzt werden sollte, weil ein durch vier Eckpunkte begrenztes Polygon immer eine konvexe Hülle bildet.

Eine Berechnung des Kamera-Bezugspunkts für mehr als vier Spielende kann ebenfalls realisiert werden, indem das aufgespannte Polygon auf Überschneidungen untersucht- und in sich nicht überschneidende Polygone zerlegt wird. Dieser Ansatz erfordert jedoch einen erheblichem Mehraufwand und würde im Rahmen dieser Arbeit keine neuen Erkenntnisse zu Tage fördern. Für die Demonstration der zugrundeliegenden Konzepte werden vier Spielende daher als ausreichende Anzahl angesehen, weshalb ab fünf Spielenden auf die triviale Centroid-Methode zurückgegriffen wird, welche keinen

[424] Wobei der Flächeninhalt A des betrachteten Polygons hierbei definiert ist als:

$$A = \frac{1}{2} \sum_{i=0}^{n-1} (x_i\ y_{i+1} - x_{i+1}\ y_i)$$

[Bourke, 1988]

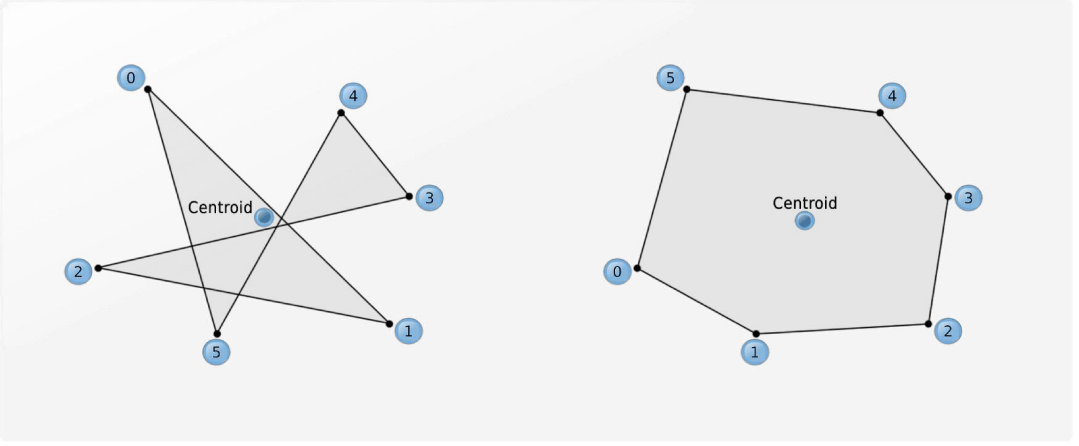


Abbildung 23.5.: Die Abbildung zeigt die Problematik, dass eine unsortierte Liste an Eckpunkten dazu führen kann, dass diese keine konvexe Hülle mehr aufspannen.
nach [Shiffman, 2011]

demokratischen Bezugspunkt (mehr) liefert. Die andere sich aus den Formeln ergebende Konsequenz war die, dass beachtet werden muss, dass sich die Spielenden unabhängig voneinander frei bewegen können, was letztlich auch die Positionsliste ständig durchmischt. Das heißt, dass obwohl eine Liste von Punkten vorliegt, welche die Eckpunkte eines konvexen Polygons bilden, können diese so sortiert sein, dass die Bildung einer konvexen Hülle fehlschlägt.

Diese Problematik ist in Abbildung 23.5 illustriert, bei der ersichtlich wird, dass die Reihenfolge, in der die Eckpunkte intern vorliegen, bei der Definition der konvexen Hülle eine wesentliche Rolle spielt. Aus diesem Grund wurde zunächst der Algorithmus aus Formel 23.2 implementiert, um die absolute Mitte zwischen allen Eckpunkten zu berechnen. Anschließend wurde ausgehend von diesem Punkt ein Einheitsvektor aufgespannt. Dadurch ist es möglich, die Positionen eines jeden Spielenden[425] als Endpunkt für weitere Vektoren zu bestimmen. Durch einen Winkelabgleich zum Einheitsvektor lässt sich anschließend eine Sortierung der einzelnen Eckpunkte vornehmen (vgl. Abbildung 23.6).

Da die Sortierung der Eckpunkte auch im negativen Bereich stattfinden kann, wenn sich alle Spielenden im negativen Bereich des karthesischen Welt-Koordinatensystems aufhalten, wird die resultierende Positionsliste nicht zwangsläufig im Uhrzeigersinn aufsteigend ausgeben. Nach der Sortierung muss deshalb ein Test auf die Reihenfolge gemacht werden, um die Positionen gegebenenfalls umzusortieren. Dies ist möglich, indem das Vorzeichen der berechneten Fläche des Polygons beachtet wird. Ist das Vorzeichen positiv, so sind die Vektoren gegen den Uhrzeigersinn angeordnet. In diesem Fall

[425] und somit jeden Polygon-Eckpunkts

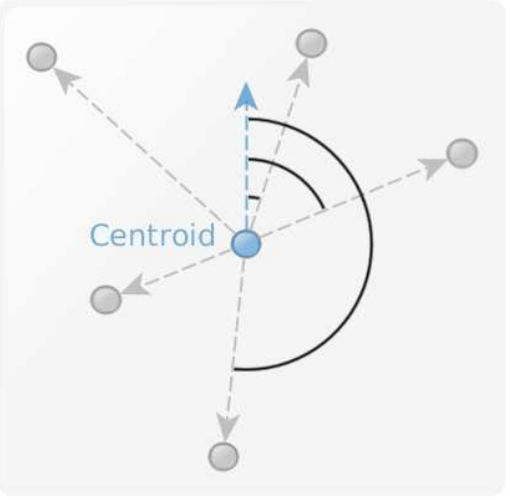


Abbildung 23.6.: Die Abbildung zeigt eine Möglichkeit, die Eckpunkte eines Polygons so zu sortieren, dass nach der Sortierung eine unterbrechungsfreie, konvexe Hülle gezeichnet werden kann. Dies wird über einen Umweg erreicht, indem jeweils die Winkel zwischen einem Hilfsvektor (blau), ausgehend vom absoluten Mittelpunkt (blau) der Punktwolke und allen weiteren Vektoren bestimmt werden, welche sich zwischen dem Mittelpunkt und den jeweiligen Eckpunkten aufspannen lassen. Die Winkel können nach Betragsgröße sortiert werden und geben damit die Reihenfolge der Punkte im Polygon an.

muss das Polygon umsortiert werden. Sind alle Voraussetzungen erfüllt, d.h. sind die Positionen der Spielenden aufsteigend mit dem Uhrzeigersinn sortiert, kann der Algorithmus aus Formel 23.4 angewandt werden, um den tatsächlichen Schwerpunkt des von den Spielenden aufgespannten Polygons zu bestimmen. Dieser Punkt wird als Bezugspunkt der Kamera gesetzt und folgt damit Mehrheitsentscheidungen (vgl. Abbildung 23.3 Punkt **a**).

23.2. Zoom und Kamera-Kollision

Zu diesem Zeitpunkt verfügte die Kamera noch über zwei Probleme: Einerseits befand sich diese immer in einer fixen Entfernung vom Geschehen und andererseits bestand die Gefahr, dass die Kamera plötzlich eine Wand durchbricht, wenn sich die Spielenden einer solchen nähern.

Das Problem der Entfernung konnte dadurch gelöst werden, dass in jedem Berechnungsschritt zusätzlich überprüft wurde, welcher Spieler sich am weitesten entfernt vom Kamera-Bezugspunkt aufhält.

Diese Entfernung bestimmt als Variable letztlich die Distanz der Kamera zum Bezugspunkt. Hierbei wurde eine logarithmische Funktion gewählt, um die Kamera stärker abzubremsen, wenn sich diese näher am Geschehen befindet. Das Zusammen- respektive Auseinanderlaufen von Avataren führt dadurch zur automatischen Zoom-Anpassung durch die Kamera, sodass sowohl eine Detailbetrachtung als auch ein Blick aufs Detail ermöglicht wird, wenn die Spielenden dies wünschen.

Das andere Problem bestand darin, die Kamera dafür zu sensibilisieren, dass es eine Umgebung gibt, welche die Sicht auf die Spielenden verdecken kann. Dieses Problem wurde auf die Art gelöst, dass bei der Konstruktion der Spielwelt zusätzlich spezielle Volumina angegeben werden können, innerhalb derer sich die Kamera bewegen darf. Beispielsweise soll die Kamera nicht von jedem Grashalm oder kleineren, geometrischen Objekten innerhalb des Levels aufgehalten werden, sondern nur wenn die Gefahr droht, die tatsächlichen Level-Grenzen zu überschreiten. Wird eine Kollision mit diesen Volumina registriert, verbleibt die Kamera an der letzten berechneten Position und geht in einen Modus über, bei dem lediglich die Rotation und Neigung stetig so angepasst wird, dass der Kamera-Bezugspunkt weiterhin in der Blickmitte verbleibt. Sobald sich die Spielenden weit genug von den Grenzen des Volumens entfernt haben, fällt die Kamera in das Standard-Verhalten zurück und verfolgt die Spielenden in einem isometrischen Modus. Abbildung 23.7 visualisiert das realisierte Ergebnis, insbesondere mit Bezug auf die Berechnung des Schwerpunkts.

23.3. Verbindung von Interaktionen, Story und Gameplay

Nachdem in den vorhergehenden Abschnitten erläutert wurde, auf welche Weise die essentielle Spielmechanik zur effektiven Unterstützung mehrerer Spielerinnen und Spieler implementiert wurde, soll in diesem Abschnitt eine kurze Übersicht darüber gegeben werden, wie der Game-Engine-Client als Schnittstelle zwischen Story, Gameplay und Interaktionen fungiert.

Im Grundlagenteil wurden verschiedene Ideen vorgestellt, wie Aufgaben zwischen Realität und Virtualität verteilt werden können (vgl. Abschnitt 7.2). Hierbei wurde insbesondere ersichtlich, dass eine der Stärken der Virtualität in der einfachen Manipulierbarkeit und Darstellung von (veränderlichen) Umgebungen liegt. In seiner Designation als visueller Endpunkt einer Interaktion dient der Game-Engine-Client somit unter anderem als Lieferant für ein Feedback, welches den Spielenden signalisiert, wie und ob eine durchgeführte Interaktion Ereignisse innerhalb der Spielwelt nach sich zieht.

Für die grundsätzliche Mechanik der Fortbewegung und Interaktion wird dabei jeder Spielerin und jedem Spieler eine Repräsentation eines (virtuellen) Mauszeigers zugeordnet, während eine individuelle Farbgebung einen schnellen Wiedererkennungswert zusichern soll (vgl. Abbildung 23.8).

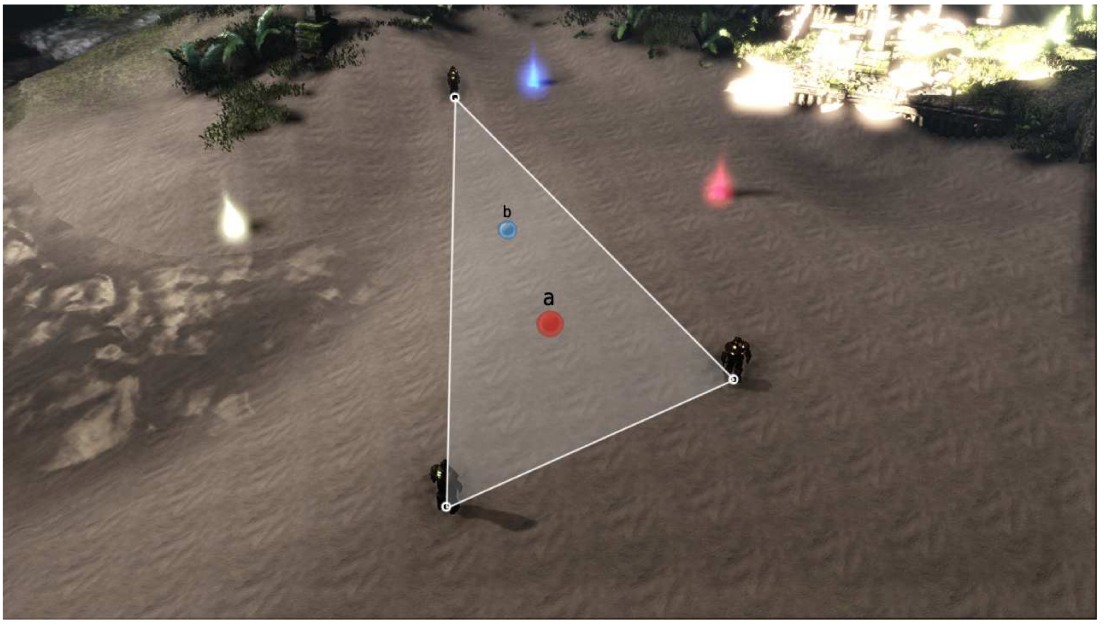


Abbildung 23.7.: Die Abbildung zeigt, wie sich die in den vorangegangenen Abschnitten beschriebenen Konzepte in der Implementierung auswirken. Mit Bezug auf Abbildung 23.3 wird hier verdeutlicht, dass die Kamera vorrangig die beiden Spieler im Vordergrund einfängt (Punkt a), während Punkt b als absolute Mitte zwischen allen Avataren ein ungewichtetes Ergebnis produzieren würde.

Für die Translation der Steuerung leistet ein Differenzvektor zwischen dem jeweils letzten Berührungspunkt und der ab dieser Berührung zurückgelegten Distanz einen Wert, der einer quantisierten Entfernung auf dem Display des Smartphones entspricht. Dieser Wert wird anschließend über das Netzwerk übertragen und skalar aufgewertet, um die zurückgelegte Distanz des durch die Bewegung gesteuerten Mauszeigers auf dem Bildschirm zu errechnen.

Die Manipulation eines Mauszeigers zur Positions- beziehungsweise Richtungsangabe stellt eine indirekte Steuerung dar, die in diesem Fall nicht nur eine konkrete Richtungsangabe ermöglicht, sondern auch weitere Interaktionen, wie das „Zeigen“ auf Objekte. Dies wird beispielsweise verwendet, um mehr Informationen über ein Objekt in der Spielwelt zu erhalten. Eine direkte Steuerung, wie sie beispielsweise mit Hilfe von Gamepads erfolgen könnte, würde keine Interaktion mit der Welt unabhängig von der Position des Avatars zulassen. Um das Beispiel der Mixed Reality-Interaktion aufzugreifen, in der an einer Stelle des Levels Licht gesammelt werden muss[426], können die Spielerinnen und Spie-

[426] vgl. Abschnitt 14.3.1 auf Seite 248

Abbildung 23.8.: Die Abbildung zeigt sowohl eine konkrete Interaktionsszene des Spiels als auch die Visualisierung der Spielernamen über den jeweiligen Avataren und die dazugehörigen Mauszeiger. In diesem Fall bewegt der Spieler „Antony“ seinen Cursor über einen Generator, woraufhin eine Art Tooltip eingeblendet wird. Die Mauszeiger dienen damit sowohl zur Fortbewegung als auch als Interaktionswerkzeug.

ler sowohl mit Hilfe des Mauszeigers navigieren als auch durch das Zeigen auf den Generator einen Informationstext anzeigen lassen (vgl. Abbildung 23.8). Hierbei wird auch deutlich, wie das Gameplay die Story durch Interaktionen unterstützen kann und umgekehrt: Die Gruppe benötigt einen funktionierenden Generator, um das nächste Modul betreten zu können. Dazu muss Licht gesammelt werden, wozu eine Stelle des entsprechenden Raumes beleuchtet wird, während an anderer Position der Generator steht. Die Interaktion wird mit Hilfe der Smartphones ausgeführt. Die Logik übernehmen sowohl der Interaction-Agent[427], als auch der Game Master[428]: Alle Komponenten des Systems treten somit in Wechselwirkung, um letztlich die Story und das Spiel voranschreiten zu lassen.

Ein anderes Beispiel für die Integration von Story und Spielmechanik ist das Einschlagen der Wand in der Starthalle. Zunächst ist die Interaktion auf bestimme Spieler-Rollen beschränkt, da hier körperliche Kraft benötigt wird[429] (vgl. Abbildung 23.9). Dadurch werden Rollenspiel-Elemente berücksichtigt,

[427] siehe Abschnitt 19.1 auf Seite pagerefsec:InteractionAgent
[428] siehe Abschnitt 19.3 auf Seite 284
[429] Diese Einschränkung wird mit Hilfe der konzipierten Trigger-Objekte vorgenommen und vom Interaction-Agent überwacht.

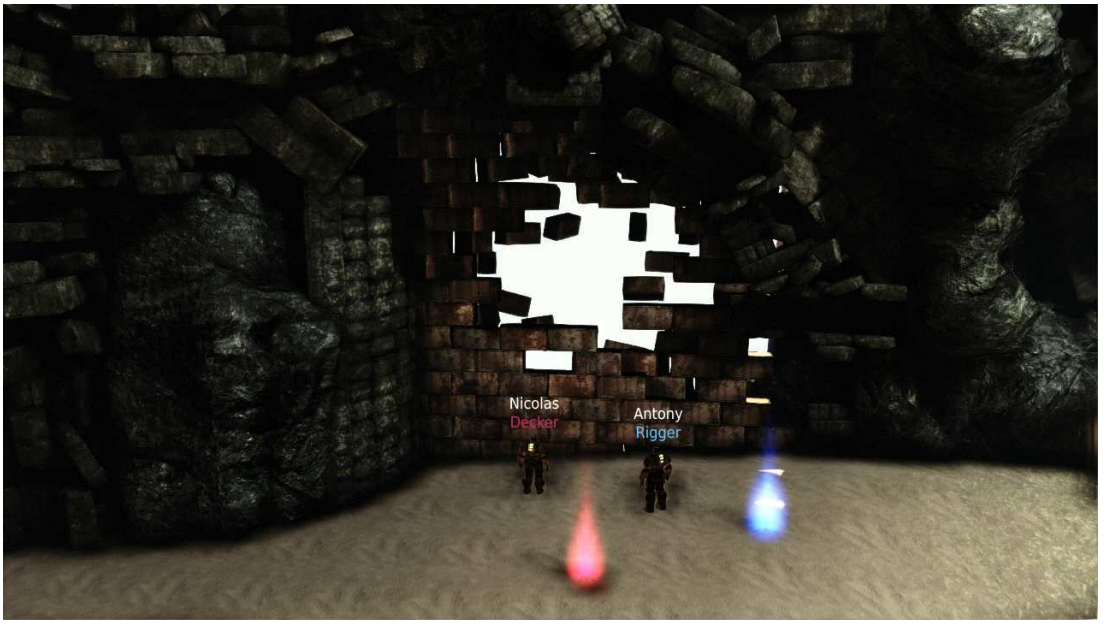


Abbildung 23.9.: Die Abbildung zeigt eine Interaktion bei der Spieler und Spielerinnen durch wiederholtes Schütteln ihrer Smartphones eine Wand einhauen müssen. In diesem Fall wurden die ersten Schläge bereits durchgeführt, sodass wesentliche Teile der Wand schon zum Einsturz gebracht wurden. Auf den Smartphones wird eine entsprechende Interaktionsansicht angezeigt.

nach denen sich Gruppenmitglieder entsprechend ihrer Fähigkeiten im Idealfall ergänzen[430]. Das Einschlagen an sich wird mit Hilfe der Physik-Engine des UDK visualisiert und erzeugt bei jedem durch die Smartphones ausgeführten „Schlag“[431] eine visuelle Rückmeldung und ebenfalls Geräusche auf den Smartphones der Interagierenden. Wurde die Interaktion erfolgreich durchgeführt, kann die Gruppe das nächste Modul betreten.

Mit diesem Abschnitt endet der Realisierungsteil. Der nachfolgende, letzte Teil der Arbeit beinhaltet neben einer Evaluation der technischen Aspekte des implementierten Systems, auch eine vergleichende Gegenüberstellung einer Systemvariante einmal mit- und einmal ohne Mixed Reality-Aspekten. Dieser Evaluation schließt sich im Rahmen einer Zusammenfassung nochmals ein komprimierter Überblick über das gesamte System an, um alle erwähnten Komponenten und Ansätze Revue passieren zu lassen.

[430] Dies entspricht dem in Kapitel 15 ab Seite 251 beschrieben Ergänzungsprinzip, wodurch Spieler zu Kollaborationen angeregt werden.
[431] Physikalisch entspricht ein solcher Schlag einem Schütteln des Smartphones.

Teil IV.

Evaluation und Konklusion

„[An evaluation is] a structured process of assessing the success of a project in meeting its goals and to reflect on the lessons learned."

(Pritchard [Pritchard, 2014])

KAPITEL 24

Evaluation

Im Laufe der Arbeit wurde deutlich, dass es nicht nur sinnvoll sondern auch notwendig ist, sowohl das anvisierte Ziel dieser Arbeit als auch die dafür konstruierte Netzwerktechnologie kritisch zu hinterfragen und auf Leistung, Benutzbarkeit und Praktikabilität hin zu untersuchen.

In diesem Kapitel sind daher zwei Evaluationen zu finden. Zum einen wurde das im Masterprojekt bereits in Teilen entwickelte Netzwerkprotokoll untersucht, um mögliche Engpässe ausfindig zu machen und notfalls zu korrigieren. Zum anderen wurde eine vergleichende Evaluation durchgeführt, um die Auswirkungen von Mixed-Realty-Aspekten auf die soziale Interaktion und die damit zusammenhängende Kommunikation zwischen den Spielenden zu ermitteln.

24.1. Eingesetzte Netzwerktopologie

Wie bereits erwähnt, wurde in einem vorangegangenen Projekt der technische Prototyp für das hier angestrebte System erstellt. Dies hatte unter anderem den Grund, möglichen Problemstellungen, insbesondere in Bereich der Netzwerkimplementierung, frühzeitig entgegenwirken zu können. Im Rahmen dieses Projekts wurde keine Evaluation der Netzwerktoplogie und des technischen Systems durchgeführt, sodass diese hier nachgeholt werden muss, um die Skalierbarkeit, Stabilität und Leistungsfähigkeit der Gesamtanwendung sicherstellen zu können. Die Evaluation findet jedoch auf Basis der hier entwickelten Anwendung und dem erweiterten Kommunikations- und Netzwerksprotokoll statt. Das Ziel dieser Evaluation war daher an dieser Stelle kritisch zu hinterfragen, ob die konstruierte Topologie den Ansprüchen der Anwendung genügt oder ob Nachbesserungen notwendig wären oder sind.

In Abb. 24.1 sind verschiedene Netzwerktopologien dargestellt, welche die Verbindung zwischen den einzelnen Komponenten unterschiedlich handhaben. Die Wahl der verwendeten Technologie[432] (1)

[432] Eine ausführliche Beschreibung der Wahl ist in den Abschnitten 3.1.5 und 3.1.6 zu finden.

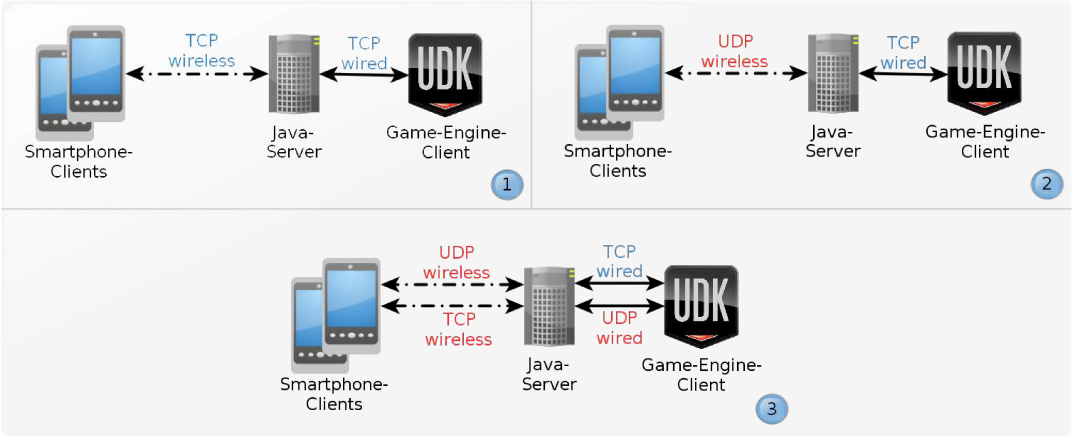


Abbildung 24.1.: Vergleich verschiedener Netzwerktopologien. Neben der derzeit verwendeten Netzwerktopologie (1) werden zwei Alternativansätze dargestellt: hybrid (2) und parallel (3).

könnte zu Problemen führen, sodass ein zukünftiger Wechsel auf eine der in Abb. 24.1 dargestellten Alternativansätze (2,3) notwendig wäre. Ein hybrider Ansatz (2) würde beispielsweise verschiedene Netzwerkprotokolle, namentlich UDP und TCP, für unterschiedliche Kommunikationswege verwenden. Die hier existenten bidirektionalen Verbindungen umfassen zum einen die Verbindung der Interaktionsgeräte zum Server und zum anderen die Verbindung der Game-Engine zum selbigen. Ein paralleler Ansatz (3) hingegen hält für jeden Kommunikationsweg zwei Verbindungen mit jeweils einem der beiden Protokolle offen.

24.1.1. Vorgehen

Der durchgeführte Test beinhaltete das Spielen einer prototypischen Spielwelt durch vier Spieler. Die Spielwelt, dargestellt in Abb. 24.2, bestand lediglich aus einer frei begehbaren Welt sowie einer kollaborativen Interaktion, welche abschließend als Ende eines jeden Testdurchlauf durchgeführt werden sollte. Die Spielzeit jedes Tests betrug fünf Minuten. Die Zeit liegt darin begründet, dass genug Daten gesammelt werden können und eventuell auftretende Schwankungen des Netzwerks ebenfalls erfasst werden. Jeder Testdurchlauf wurde dreimal, jeweils mit verschiedenen Sendeintervallen, durchgeführt: 42 *ms*, 21 *ms* sowie 11 *ms*. Die Sendeintervalle beschreiben die Zeitpunkte in denen Bewegungsdaten der Smartphone-Clients an den Server gesendet werden. Die Auswahl der 42 *ms* als Standardwert liegen, wie in Kapitel 17 begründet, an den zu erreichenden Latenzwerten eines First-Person-Shooters,

welche zwischen 40 und 75 *ms* liegen. Außerdem gilt, um ein hohes Engagement in responsiven Anwendungen erreichen zu können, dass Latenzen eine Zeit von 50 *ms* nicht überschreiten dürfen [Michael Abrash, 2012]. Darin begründet liegt die Mindestanzahl von 20 Updates pro Sekunde je Spieler. Hinzukommend zu der Tatsache, dass das menschliche Auge Bewegungen ab 24 Updates pro Sekunde als flüssig wahrnimmt, wurde ein Sendeintervall von exakt 42 Millisekunden gewählt[433].

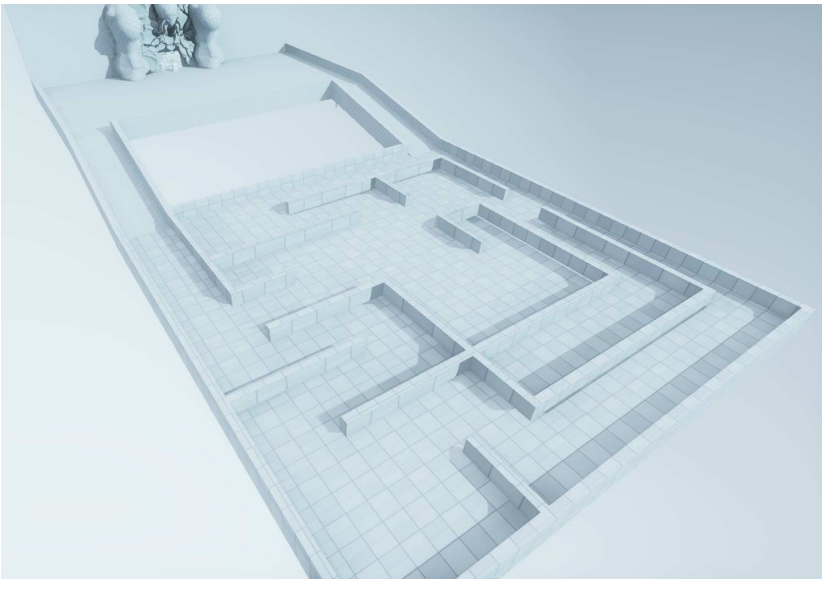


Abbildung 24.2.: In der Abbildung wird die Spielwelt dargestellt, welche für die Evaluation des Netzwerkverkehrs verwendet wurde. Erst zum Ende der fünf Minuten Spielzeit sollte das Labyrinth verlassen und zur oberen Ebene gelaufen werden, wo die abschließende Interaktion stattfand.

Die Auswahl der letzten Option mit 11 *ms* liegt in folgenden Ursachen begründet:

- Der Datendurchsatz wird im Vergleich zur Standardeinstellung von 42 *ms* nahezu vervierfacht und ermöglicht einen guten Stresstest.
- Ferner entsprechen, wie in Abschnitt 11.3.1 beschrieben, 11 *ms* dem Optimum um die besten Antwortzeiten zu erhalten[434].

Die Anzahl der übertragenen Interaktions- und Story-Events nehmen einen vernachlässigbaren Anteil im gesamten Übertragungsvolumen ein, sodass diese nicht weiter betrachtet worden sind. Durch die Verwendung von TCP als zugrundeliegendes Protokoll, wird eine Zustellung von diesen Events in

[433] weil $1000/24 = 41,\overline{6}$
[434] zumindest nach Aussagen von Wang u. a. in [Wang u. a., 2009]

	42 *ms*	**21 *ms***	**11 *ms***
durchschn. Zeit zwischen zwei Paketen	43,2 *ms*	21,6 *ms*	11,4 *ms*
Jitter	unerheblicher Jitter (siehe Anmerkung)	unerheblicher Jitter (siehe Anmerkung)	unerheblicher Jitter (siehe Anmerkung)
Paketverlust (in %)	0	0	0
Sendeleistung insg. / je Smartphone-Client (pro Sekunde)	11,2 *KB* / 2,8 *KB*	22,6 *KB* / 5,65 *KB*	41 *KB* / 10,25 *KB*
Sendeleistung insg. / je Smartphone-Client (gesamte Spielzeit)	3,28 *MB* / 0,82 *MB*	6,6 *MB* / 1,65 *MB*	12 *MB* / 3 *MB*
Paketgrößen	51-55 Bytes (99,99%)	51-55 Bytes (99,99%)	51-55 Bytes (99,99%)
Paketumlaufzeit	zwischen 41 *ms* und 44 *ms*	zwischen 20 *ms* und 22 *ms*	zwischen 10 *ms* und 13 *ms*
Datendurchsatz	konstant	konstant bis auf das Ende	konstant bis auf das Ende
Anmerkungen	Pakete, welche Anweisungen zum Öffnen der Tür beinhalten, weisen hohe (10 *s*) Delta-Time auf.	Pakete, welche Anweisungen zum Öffnen der Tür beinhalten, weisen hohe (4 *s*) Delta-Time auf.	Pakete, welche Anweisungen zum Öffnen der Tür beinhalten, weisen hohe (4 *s*) Delta-Time auf.

Tabelle 24.1.: Tabellarischer Vergleich verschiedener Sendeintervalle

jedem Fall auf Protokollebene sichergestellt, was für diese Anwendungszwecke alleinig entscheidend ist, sodass eine genauere Betrachtung im Rahmen der Evaluation ohnehin nicht notwendig ist. Zur Analyse des Netzwerks wurde das freie Programm Wireshark verwendet.

24.1.2. Ergebnisse

Um ein aussagekräftiges und von äußeren Einflüssen unabhängiges Ergebnis zu erhalten, sind die Tests dreimal am Tag zu verschiedenen Uhrzeiten über drei Tage verteilt durchgeführt worden[435]. Der aufgezeichnete Datenverkehr kumuliert alle verbundenen Smartphone-Clients, insofern war eine Überprüfung der Latenz und des Sendeintervalls eines einzelnen nur über einen Umweg durch die Verwendung der TCP-Protokolleinstellungen von Wireshark in Kombination mit diversen Filtern möglich, wodurch die „Time since previous frame in this TCP Stream“ berechnet werden konnte. Die aus dieser Kennzahl erhaltenen Zeiten müssen den in dem entsprechenden Testdurchlauf verwendeten Sendeintervallen entsprechen. Hier auftretende Ungereimtheiten können Indikatoren für ein Latenzproblem sein.

Neben diesem simplen Wertevergleich wurden verschiedene Kennzahlen und Diagramme erstellt, welche in Tabelle 24.1 tabellarisch dargestellt und nachfolgend erläutert werden.

Die errechnete Kennzahl „durchschnittliche Zeit zwischen zwei Paketen“ sollte dem im korrespondierenden Testdurchlauf verwendeten Sendeintervall entsprechen, was jedoch unter anderem aufgrund von Netzwerkschwankungen wertmäßig nicht exakt gelingen kann. Dennoch kann eine Abweichung

[435] Aufgrund der Entwicklung des Systems in den Räumlichkeiten der Hochschule Bonn-Rhein-Sieg, hängt die Netzwerkauslastung sehr stark von der Mitbenutzung und Anzahl anderer verbundener Clients ab.

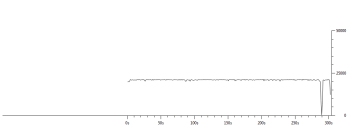
hier als Indikator für mögliche Problemstellungen dienen. Zur weiteren Analyse kann die Paketumlaufzeit[436] herangezogen werden. Neben der Tatsache, dass diese bei der Verwendung von Transmission Control Protocol das Zeitintervall angibt, in welchem verlorengegangene Pakete erneut gesendet werden sollen, lassen sich abweichende Werte direkt auf Latenzprobleme zurückführen. Ein verspätet gesendetes Antwortpaket kann viele Ursachen haben, zum Beispiel eine hohe CPU-Last beim Sender. Bei keinem der Testdurchläufe lassen sich bei dem Ergebnis der beiden zuvor genannten Kennzahlen erhebliche Abweichungen feststellen[437], was einem sehr guten Ergebnis entspricht.

Bei genauerer Betrachtung der Paketumlaufzeiten fallen die ungewöhnlich hohen Werte auf, da bei WLAN-Übertragungen im lokalen Netzwerk rund 10 *ms* als normaler Wert gelten [Lüders u. Winkler, 2006]. Die hohen Werte liegen darin begründet, dass der Berechnung der Paketumlaufzeit ein komplexes Berechnungsverfahren zu Grunde liegt, welches durch die Verwendung der SACK-Eigenschaft[438] und der damit einhergehenden Quittung erschwert wird. Eine Empfangsbestätigung wird erst nach jedem zweiten empfangenen Paket gesendet, welches beispielsweise im ersten Testdurchlauf nach 42 *ms* der Fall ist.

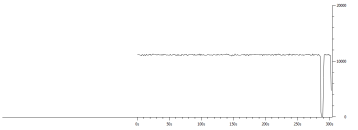
Wie in Abbildung 24.3 erkennbar ist, existiert in jeder der Varianten ein vernachlässigbarer Jitter[439]. Am Ende der Diagramme ist jedoch eine Schwankung zu erkennen. Dies liegt nach weitergehenden Untersuchen darin begründet, dass an dieser Stelle die kollaborative Interaktion stattgefunden hat. Der Ausführung geht eine Pause voraus, in der sich die Spieler absprechen respektive versuchen nachzuvollziehen, was hier getan werden muss. In der Zwischenzeit findet keine Bewegung und somit auch kein Datenverkehr statt. Hinzukommt ebenfalls, dass die gesendeten Daten um ein Vielfaches größer sind, da diese Anweisungen für die Game-Engine enthalten. Dies widerspricht dem Vorhaben der Logikentkopplung und bringt zudem einige Netzwerkprobleme mit sich. Anhand dessen lies sich Verbesserungspotenzial im eingesetzten Kommunikationsprotokoll erkennen, welches umgehend umgesetzt worden ist.

Weiterhin fällt auf, dass kein Paket verlorengegangen ist, was einen erheblichen Anteil an den guten Testresultaten ausmacht. Ein Paketverlust führt bei der Verwendung von TCP zu einem kurzen Aussetzen der Datenübertragung, da ein Ersatzpaket angefordert wird damit die garantierte Reihenfolge eingehalten werden kann. Dies hätte erhebliche Auswirkungen auf die Anwendung, da ein solches Blockieren des Sendens direkt mit einem Stoppen der Steuerungsmöglichkeit der Avatare einhergehen würde, was zu einem zwischenzeitlichen Totalausfall der Anwendung führen würde.

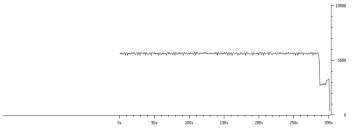
[436] Round Trip Time (RTT)
[437] siehe erste und siebte Zeile in Tabelle 24.1 auf Seite 322
[438] siehe Abschnitt 11.3 auf Seite 204 für eine nähere Erläuterung
[439] Die eingezeichneten Linien in den Diagrammen schwanken kaum.



(a) 11 *ms* Sendeintervall, 40 Events



(b) 21 *ms* Sendeintervall, 20 Events



(c) 42 *ms* Sendeintervall, 10 Events

Abbildung 24.3.: Jitter des Prototypen unter Verwendung verschiedener Sendeintervalle. Im Anhang unter Abbildung B.4, Abbildung B.5 und Abbildung B.6 sind die Sendeintervalle auch noch einmal in größerer Darstellung zu finden.

Darüber hinaus befinden sich die Paketgrößen in dem geforderten, optimalen Intervall von 25 bis 95 *Byte* (siehe Abschnitt 11.3.1). Zusätzlich wird dadurch einem Speicherüberlauf und verlorengegangenen Paketen entgegengewirkt.

Da die gewählten Intervalle jeweils einen Faktor von zwei auseinanderliegen, entspricht die gemessene Sendeleistung ebenfalls diesem Verdopplungsschema. Bei 42 *ms* beträgt der Gesamtdurchsatz für die gesamte Spielzeit und vier Spielende etwa 3,28 *MB*, bei 21 *ms* bereits 6,6 *MB*. Bei der Verwendung eines Sendeintervalls von 11 *ms* beträgt das aufkommende Datenvolumen im Upload pro Spieler und Sekunde lediglich 10,25 *KB*. Prinzipiell wäre es dadurch sogar möglich, WLAN als Übertragungsart durch GPRS zu ersetzen [Wang u. a., 2009]. Problematisch könnten sich beim Einsatz von GRPS allerdings die Paketumlaufzeit sowie die Antwortzeit auf das System auswirken. Zudem ergeben sich bei diesen Paketgrößen zu große Latenzen unter Verwendung von GPRS [Wang u. a., 2009]. Es werden lediglich 0,0085% der zur Verfügung stehenden Bandbreite ausgenutzt, was durch zusätzliche Optimierungen weiter reduziert werden könnte. Beispielsweise werden kontinuierlich Bewegungsdaten gesendet, selbst wenn sich die Werte zu den vorangegangenen nicht geändert haben.

Die Entkopplung des Game Masters und Game-Engine-Clients vom Server bewirkt nicht nur eine erhöhte Performanz und verringerte Netzwerklast, sondern führt auch zu strukturellen und organisatorischen Vorteilen. Dadurch, dass sich der Client und Server auf demselben Host befinden, entsteht keinerlei Netzwerkverkehr, wodurch ebenfalls keine Paketverluste, Verzögerungen oder Out-of-Order Sendungen existieren (vgl. Abbildung 24.4) [Snader, 2000, S. 5]. Jede Komponente kann beliebig ausgetauscht werden und zudem in unterschiedlichen Programmiersprachen entwickelt werden. Lediglich die Netzwerkschnittstelle muss vorhanden sein und entsprechend funktionieren.

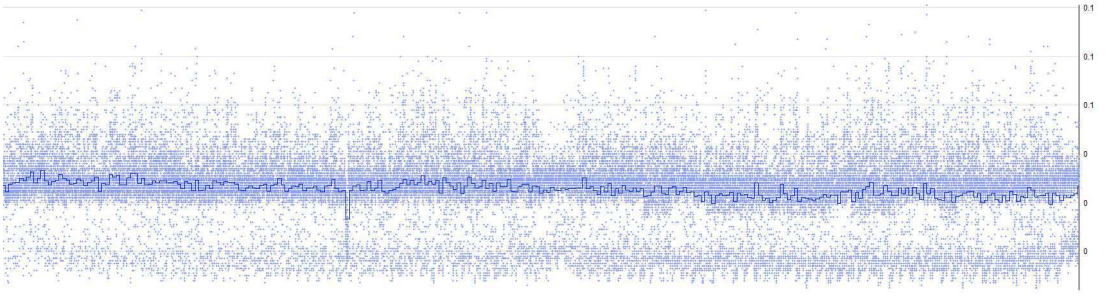


Abbildung 24.4.: Die Grafik zeigt die Ergebnisse eines Ping auf den localhost über einen Zeitraum von fünf Minuten und mit einer Paketgröße von insgesamt 128 Byte. Der zeitliche Verlauf ist auf der X-Achse aufgetragen. Die Zeit zwischen den Paketen betrug jeweils 5 *ms*. Es fand kein Paketverlust statt und die durchschnittliche Paketumlaufzeit betrug im Mittel weniger als 0.1 *ms*. Die Paketumlaufzeiten der einzelnen Pakete werden in der Abbildung als Punkte dargestellt.

Die Betrachtung der theoretischen Ergebnisse zeigt auf, dass keinerlei Bedarf zum Wechsel der technologischen Grundlage besteht. Das technische Grundgerüst weist keinerlei gravierende Mängel auf, welche die Stabilität und damit den Erfolg der Anwendung in irgendeiner Weise gefährden könnten. Ein hybrider Ansatz würde zwar den Datenverkehr und -nutzen verbessern können, der allerdings für die Umsetzung benötigte Zeitaufwand steht in keinem Zusammenhang zu der möglicherweise gewonnenen Leistung. Ebenfalls bringt eine Änderung des Sendeintervalls keinerlei Vorteile mit sich, sodass die voreingestellten 42 *ms* beibehalten werden.

24.2. Infrastrukturelle Problemstellungen

Neben der rein theoretischen, wertebasierten Evaluation gilt es jedoch auch wahrnehmungspsychologische Aspekte zu betrachten. Laut Lecky-Thompson besteht die Möglichkeit, dass Spielern ein Spiel langsam vorkommt bzw. nicht existierende Latenzen wahrgenommen werden [Lecky-Thompson, 2008, Kap. 7]. Die Autoren dieser Ausarbeitung, welche die Datensätze für die Evaluation der Netzwerktopologie zusammengetragen haben, nahmen Verzögerungen wahr, welche nicht durch die Kennzahlen und Auswertungen erfassbar waren. Die eigentlichen Verzögerungen spielten sich in einem sehr geringen Zeitfenster ab, waren jedoch aufgrund des gewählten Interaktionskonzepts (insbesondere bei der Steuerung[440]) durchaus spürbar. Sobald ein beliebiges Paket verlorengeht, stoppt die gesamte Verarbeitungskette, da hier die Eigenschaften des Transmission Control Protocol aktiv werden

[440] siehe Kapitel 17 ab Seite 265

und ein verlorengegangenes Paket versucht wird nachzusenden, bis eine Bestätigung dafür erhalten wurde. Dies lässt insbesondere die Steuerung der Avatare kurzzeitig spürbar aussetzen. Zudem schienen diese Verzögerungen mit der Testumgebung zusammenzuhängen, da diese lediglich beim Einsatz in den Räumlichkeiten der Hochschule Bonn-Rhein-Sieg auftraten. In einem nachgelagerten Schritt wurde daher eine weitere Ursachenforschung auf höherer Ebene vorgenommen.

Neben der Ausschaltung des Nagle-Algorithmus existieren grundsätzlich weitere Möglichkeiten (siehe Abschnitt 11.3) den Datendurchsatz und die Funktionsweise TCPs auf bestimmte Anwendungszwecke hin zu optimieren. Dadurch, dass die Evaluation gute Ergebnisse lieferte und keine Problemstellen in diesem Bereich hervorbrachte ist auf weitere Optimierungen verzichtet worden. Die Problematik der Sendungsblockade aufgrund verlorengegangener Paketen kann ohnehin, bedingt durch die low-level Implementierung auf OSI-Schicht 4[441], nicht behoben werden[442]. Als einzige Möglichkeit bietet sich hier der Umstieg auf User Datagram Protocol an, welcher mit Hilfe weiterer Anwendungslogik diese Szenarien verhindern und dennoch die benötigte Funktionalität anbieten kann. Der Umstieg würde jedoch den Umfang der Arbeit bei weitem übersteigen und zudem kein gutes Aufwand-/Nutzenverhältnis liefern. Ferner kann ein möglicher Erfolg nicht garantiert werden, denn wie in einem Test von Snader gezeigt wurde, verhält sich UDP weder im LAN noch Rechner-intern wirklich besser, da in beiden Umgebungen sehr viele Paketverluste auftreten [Snader, 2000, S. 65ff]. Bei der Verwendung von UDP müssten viele der durch TCP bereits nativ angebotenen Eigenschaften nachgebaut werden, z.B. Mechanismen, welche Paketverluste erkennen und eine Neusendung vornehmen[443]. Ferner beurteilt Snader dies wie folgt:

> *„If an Application requires the reliability of TCP, it is possible that TCP is the best solution. In the first place, it is unlikely that TCP functionality duplicated in a user application will be as efficient as the real thing.“*

[Snader, 2000, S. 67][444]

[441] siehe Abb. 11.2 auf Seite 203

[442] In einem Netzwerk existieren neben allen kontrollierbaren Problemstellungen weitere in viel tieferliegenden und für den Anwender respektive Programmierer nicht erreichbaren Ebenen. Die Blockierung durch eine automatische Neusendung ist auf sehr niedrigem Level implementiert, was die Performanz dieser Funktionalität ausmacht. Zudem handelt es sich insbesondere bei dieser um eine TCP auszeichnende Eigenschaft.

[443] Wenn die Logik programmatisch nachgeholt wird, gibt es sehr viele Kontextwechsel auf Betriebssystembasis. Zunächst muss in den Anwendungsbereich gewechselt werden, in welchem das Neusenden beauftragt wird. Selbiges benötigt einen erneuten Wechsel in den Kernel, welcher das Umkopieren und letztliche Senden vornimmt, wonach zurück in den Anwendungsbereich gewechselt wird, um den Retransmission Timeout zu setzen. Dieser wird jedoch auf Betriebssystembasis gesetzt, sodass ein weiterer Wechsel in den Kernel stattfindet. Da TCP eine hardwareseitige Realisierung anbietet und die gesamte Funktionalität im Kernel handhabt, wird kein einziger Kontextwechsel benötigt. [Snader, 2000, S. 68]

Insgesamt betrachtet können die benötigten Wechsel des Kontext, welche sehr zeitintensiv sind, je nach Einsatzzweck, dem Netzwerkverkehr sowie Sendeintervall und ebenfalls der eigentlichen Programmierung, zu einer schlechteren Lösung führen.

[444] Die Quintessenz und der Tipp Snaders sind:

Da Problemstellen an diesen Komponenten ausgeschlossen werden konnten und weitere Optimierungen einerseits nicht realisierbar waren sowie andererseits der Erfolg fraglich gewesen wäre, gab es lediglich die Art der Datenübertragung, in diesem Fall das drahtlose Netzwerk, weiter zu analysieren.

Nach Betrachtung der umliegenden Drahtlosnetzwerke wurde deutlich, dass der Paketverlust und die damit einhergehende Neusendung, welche zu den Latenzproblemen führen, keineswegs in der Wahl des Netzwerksprotokolls begründet liegen, sondern viel mehr an der Menge der gleichzeitig existierenden sowie sich gegenseitig interferierender Netzwerke [Subramanian u. a., 2006][Wang, 2011, Kap. 19]. Drahtlosnetzwerke besitzen aufgrund der höheren Wahrscheinlichkeit Pakete zu verlieren ein eigenes Retransmission-System [IEEE, 2012, S. 834ff]. Dies führt dazu, dass einzelne Paketverluste auf der Transport- sowie Anwendungsschicht, so wie sie im Rahmen des hier entwickelten Systems erfolgen, weder spür- noch messbar gewesen wären. Das WLAN würde an einer solchen Stelle eingreifen und dem Paketverlust entgegenwirken [IEEE, 2012, S. 834ff].

Eine Möglichkeit, um eine Interferenz mit anderen Drahtlosnetzwerken zu umgehen besteht darin, den eigenen Übertragungskanal in Abhängigkeit zu den bereits existierenden möglichst isoliert zu wählen [Computer Bild, 2011]. In den Räumlichkeiten der Hochschule Bonn-Rhein-Sieg wird fast das gesamte zur Verfügung stehende Spektrum und damit mögliche Frequenzkanäle durch aufgespannte WLANs belegt. Ein Wechsel in einen anderen Frequenzkanal lieferte somit weniger erfolgreiche Resultate.

Darüber hinaus besteht die Möglichkeit, den Frequenzbereich zu wechseln. Die verschiedenen WLAN-Standards verwenden entweder den 2,4-Gigahertz- (z.B. g-Standard) oder 5-Gigahertz-Bereich (z.B. a-Standard)[445]. Dadurch, dass der fünf Gigahertz-Bereich derzeit eher selten genutzt wird und zudem bis zu 23 Drahtlosnetzwerke ungestört nebeneinander betrieben werden können[446] [Huch, 2010], stellt sich diese Variante als aussichtsreichste Alternative dar. Ein Wechsel verbesserte die Resultate erheblich und genügte dem Anspruch dieser Abschlussarbeit nun vollends. Ferner treten Paketverluste unter Verwendung eines drahtlosen Netzwerks allgemein häufiger auf und können nicht vollkommen umgangen werden [Westermark, 2007, S. 10].

Das es sich bei der geschilderten Problemstellung keineswegs um ein individuelles Problem respektive um eines mit geringer Tragweite handelt, lässt sich an einem Beispiel aus der freien Wirtschaft verdeutlichen. Bei der Umsetzung der Sprachsteuerung von *Apple* (*Siri* genannt [Inc., 2013]) spielten ähnliche Probleme eine erhebliche Rolle, da ein möglichst zuverlässiger Service angestrebt wurde, sodass auf das so genannte Multipath TCP (MTCP) zurückgegriffen werden musste. Da sich dieses Verfahren

„Avoid Reinventing TCP“

[Snader, 2000, S. 67].

[445] Für weiterführende Informationen wird beispielsweise [Schnabel, 2004] empfohlen.

[446] Im 2,4-Gigahertz-Bereich sind es lediglich drei.

noch in der Entwicklung befindet und nicht von allen involvierten Geräten ohne Weiteres unterstützt wird, bietet sich die Nutzung im Rahmen dieser Ausarbeitung nicht an.

24.3. Mixed Reality-Anwendung

Den wichtigsten Teil nimmt die Evaluation des Zielsystems im Hinblick auf die Erhaltung realer, sozialer Interaktionen und damit einhergehender Kommunikation ein. Die erarbeiteten Grundlagen sowie in den vorangegangenen Abschnitten erläuterten Entscheidungen münden daher in folgendem Postulat:

> *Die Vermischung von realer und virtueller Welt sowie die Kombination beider Genres führt zur Kreation eines gemeinsamen topischen Raums, in welchem die partizipierenden Spieler reale, soziale Interaktionen durchführen. Dies sorgt insbesondere dafür, dass direkte Face-2-Face-Kommunikation eingesetzt wird.*

24.3.1. Vorgehen

Mit der Durchführung einer vergleichenden Evaluation soll dieses Postulat auf Korrektheit überprüft werden. Insgesamt haben 16 Probanden (∅ 27 Jahre) teilgenommen, 11 männliche (∅ 27 Jahre) und 5 weibliche (∅ 27 Jahre). Diese testeten die Anwendung in Vierergruppen in zwei mal vier Spielsessions. Jede Gruppe spielte dabei das entwickelte Spiel in zwei unterschiedlichen Konfigurationen, welche sich in der Art der Interaktionsimplementierung und -durchführung unterscheiden. In einer Version wurden lediglich virtuelle Interaktionen verwendet und in der anderen spezielle MR-Interaktionen, welche der MR-Definition auf Seite 13.3 genügen[447]. Die Reihenfolge wurde dabei alternierend geändert, sodass abwechselnd mit einer anderen Version gestartet wurde. Es wurden sowohl rein männliche als auch rein weibliche und ebenfalls gemischte Teams zusammengestellt. Die Teilnehmer stammen dabei sowohl aus dem näheren als auch erweiterten Bekanntenkreis der Autoren. Jeder der Teilnehmer und Teilnehmerrinnen spielt privat Computer- oder Konsolenspiele oder hat diese schon einmal gespielt. 87,5% besitzen Erfahrungen im Bereich von kooperativen oder kollaborativen Konsolen- oder Computerspielen und in etwa 1/3 hat mindestens schon ein Mal ein echtes Pen-&-Paper-Rollenspiel gespielt. Die Überlegung hinter diesen Vorerfahrungsbefragungen war die, möglichst viele verfälschende Aspekte auszuschließen. Fehlende Erfahrung in Computer-/Konsolenspielen und speziell in Pen-&-Paper-Rollenspielen könnte im Allgemeinen zu mehr Kommunikation führen, welche von technischen

[447] Wie in Abschnitt 14.3 auf Seite 244 beschrieben, existieren einige Module, welche in zweifacher Ausführung vorliegen. Exakt wird dies anhand eines Beispiels in Abschnitt 14.3.1 auf Seite 248 erläutert.

oder spielerischen Gegebenheiten handelt[448]. Ferner haben 40% der Probanden bereits Erfahrungen mit virtuellen Umgebungen gemacht, was speziell im Hinblick auf das Engagement bzw. den Immersionsgrad Rückschlüsse ermöglicht.

Die Gruppenmitglieder in den einzelnen Gruppen kannten sich jeweils untereinander, sodass eine gewisse Homogenität der Kommunikationsbasis herrschen sollte. Unbekannte Gruppenmitglieder können zumeist zu einer zusätzlichen Kommunikationsbarriere führen[449], wodurch das Evaluationsergebnis verfälscht werden könnte, da nicht jeder Teilnehmer den gleichen Spaß beim Spielen hat und somit die Gefahr besteht, dass der Median der Gruppenmeinung verzerrt wird. Pen-&-Paper-Rollenspiele werden oftmals auch in unbekannten Gruppen gespielt, deren Teilnehmer sich durch das gemeinsame Spielen kennenlernen und im Idealfall wiedertreffen, um dieses oder ein weiteres Rollenspiel weiterzuführen. In dieser Evaluation geht es jedoch nicht darum, die kommunikationsfördernden Aspekte des Pen-&-Paper-Rollenspiels an sich nachzuweisen, sondern innerhalb mehrerer Vergleichsgruppen zu hinterfragen, ob die Einbringung von Mixed Reality-Aspekten die Kommunikation innerhalb einer Gruppe fördert. Aus diesem Grund sollte eine möglichst vertraute Basis geschaffen werden, in der Hoffnung, dass dadurch Kommunikationsbarrieren durch Fremdheit minimiert werden.

Um eine Auflockerung innerhalb der Gruppe zu erzielen und gezielt die Kooperation bzw. Kollaboration und Kommunikation sowie ein gemeinsames Bewusstsein und gleiche Anfangs- und Rahmenbedingungen zu schaffen, wurde in jeder Gruppe zunächst ein kurzes Kooperationsspiel aus [Hirling, 2014] gespielt[450]. Auf eine vorangestellte Warm-up-Phase, um zumindest die Steuerung kennenzulernen, wurde bewusst verzichtet, da für das Gesamtkonzept die Eingabegeräte eine wichtige Rollen spielen und ebenfalls evaluiert werden sollte, ob diese tatsächlich nicht als zusätzliche Barriere, sondern als Hilfsmittel wahrgenommen werden.

Für die Bewertung der Anwendung werden sowohl Fragebögen verwendet, welche den subjektiven Eindruck und die Erkenntnisse der Spieler berücksichtigen als auch ein unabhängiger Beobachter, welcher objektive Beurteilungen vornimmt und den spielenden Gruppen nicht als solcher vorgestellt wurde. Auf eine Videoaufzeichnung wurde bewusst verzichtet, da diese das Verhalten der beobachteten Personen ändert[451].

[448] Anders formuliert: Es wäre abzuraten Leute ein Computerspiel evaluieren zu lassen, welche nicht wissen was ein solches ist.

[449] Soziologisch beschreibt eine „Gruppe“ eine Sammlung mehrerer Personen, dessen Mitglieder alle in einer *unmittelbaren* sozialen Beziehung zueinander stehen und ein enges Beziehungsgefüge besteht [Bräuer, 2005, Kapitel 4].

[450] Konkret handelt es sich um das Spiel „Zweibeiniger Stuhl“.

[451] Obwohl beispielsweise viele öffentliche Verkehrsmittel über eine Videoüberwachung verfügen, ändern diese das Verhalten individueller Menschen nicht wesentlich, obwohl auf die Überwachung hingewiesen wird. Anders stellt sich die Situation dar, wenn eine Person sich ganz *persönlich* im Moment gefilmt fühlt, weil es ihr explizit ins Bewusstsein gerufen wird. Anhand der Datenbrille *Google Glass*, welche über eine eingebaute Kamera verfügt, wird dies besonders deutlich. Dort schlägt dem Träger oder der Trägerin sogar teils Feindseligkeit entgegen, weil die Menschen sich beobachtet fühlen [Tafeit, 2014].

Ein unabhängiger Beobachter stellt einen Kompromiss zwischen einer subjektiven Beobachtung durch die Autorenschaft und einer objektiven Beobachtung durch eine Video- oder Audioaufnahme dar. Zugleich ist es eine moralische Entscheidung, da sich eine Videoaufnahme ohne vorherige Kenntnis der Versuchsteilnehmer aus offensichtlichen Gründen verbietet.

Als Grundlage für den objektiven Beobachter sind *Cooperative Performance Metrics (CPM)* aus [Seif El-Nasr u. a., 2010] herangezogen und in Form von beobachtbaren Events zur Verfügung gestellt worden, sodass deren Auftreten erkannt und notiert werden können. Die CPM wurden innerhalb eines Forschungsprojekts eruiert und laut Seif El-Nasr u. a. von mehreren Experten sowie erfahrenen Spieldesignern von *Electronic Arts* und *Square Enix* bestätigt [Seif El-Nasr u. a., 2010]. Anhand derer soll die kooperative Natur und das Engagement eines Spiels evaluiert werden können[452] [Seif El-Nasr u. a., 2010]. Verwendet werden dabei die folgenden:

- Gemeinsames Gelächter während eines Spielevents;
- Verbale Aussagen, welche die Freude am Spiel erkennen lassen (u.a. zum Beispiel: „cool“, „macht Spaß“ und „wow“);
- Gemeinsame Erarbeitung von Strategien oder das Navigieren durch die Spielwelt während sich beraten wird;
- Gegenseitiges Helfen, beispielsweise durch Gespräche über das Interaktionsgerät und wie die Spielmechaniken genutzt werden können.

Für den Fragebogen der Spieler und Spielerinnen wurden Fragen erarbeitet, die möglichst keine Suggestion enthalten, um sowohl den empfundenen Spaß, die Kommunikationsstärke sowie das Gruppengefühl der einzelnen Versionen zu erfragen. Ebenso wurden Vorerfahrungen im Bereich von Computer-/Konsolenspiele im Allgemeinen sowie im Hinblick auf etwaige Mehrspielermodi in selbigen berücksichtigt. Hinzu kommt die Betrachtung des Immersionsgefühls in beiden Spielversionen, d.h. wie sehr sich die Spieler und Spielerinnen als Teil der Spielwelt gefühlt haben. Durch die durchgängige Verwendung eines Schulnotensystems wird die Beantwortung so intuitiv wie möglich gestaltet. Die Nutzung dieses Systems umgeht zudem weitere mögliche Problemstellungen bei der Berechnung von statistischen Kennzahlen, wie dem arithmetischen Mittel, da dieses zum Beispiel aufgrund der geringen Anzahl an Quantisierungsstufen weniger anfällig für Ausreißer ist.

[452] Obwohl die Metriken den Begriff *Kooperation* beinhalten, passen die verwendeten Spiele und überprüften Interaktionen auf die im Rahmen dieser Ausarbeitung gemachten Definition der *Kollaboration*. Siehe Kapitel 15 ab Seite 251.

24.3.2. Ergebnisse

Für eine Analyse der Ergebnisse wurden Einzelbewertungen der Gruppenteilnehmer kumuliert, jedoch für den weiteren Verlauf nicht als Durchschnittsnote betrachtet, sondern aus Gründen der Zugänglichkeit auf ein gängiges Schema normalisiert. Höhere Werte beschreiben dabei ein subjektiv besseres Ergebnis[453], was sich zudem in der Höhe eines Balkens oder dem Ausschlag eines Graphen in den entsprechenden Diagrammen niederschlägt. Für Detailbetrachtungen in einzelnen Kategorien wurde jeweils auf Einzelbewertungen aller oder spezifischer Gruppenteilnehmer zurückgegriffen. Die Zusammenfassung zu einer Gruppenmeinung ermöglicht jedoch, im Sinne des Spiels als soziale Gruppenaktivität, die aussagekräftigeren Ergebnisse.

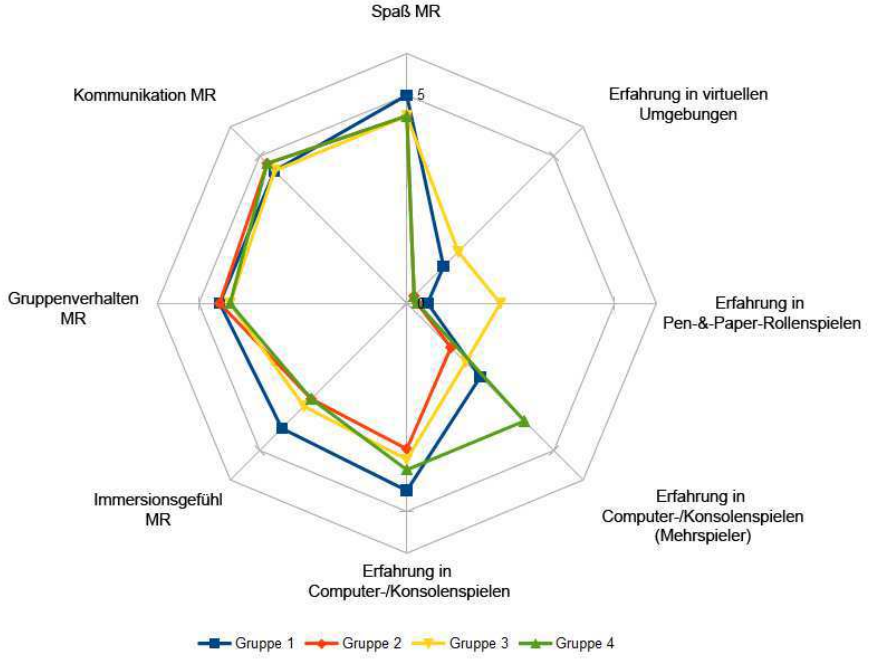


Abbildung 24.5.: Im Sterndiagramm lassen sich Unterschiede in den unterschiedlichen Gruppen direkt erkennen. Mögliche Aspekte, die miteinander in Beziehung stehen wurden gegenüber angeordnet um einen direkten Vergleich herstellen zu können.

Insgesamt lässt sich zunächst eine Korrelation zwischen Erfahrungen und vergebenen Noten in bestimmten Bereichen erkennen (siehe Abb. 24.5). Die Bewertungshöhe steht in direktem Zusammenhang mit der Vorerfahrung in dem jeweiligen Bereich; in eher unbekannten Bereichen wird häufig

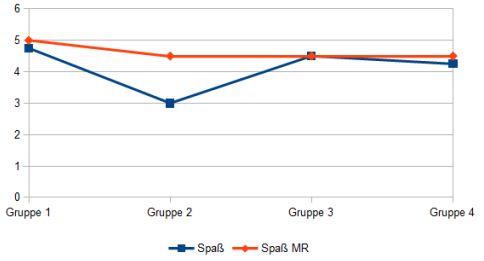
[453] mehr Spaß gegenüber weniger Spaß

neutral bewertet. In den hier vorliegenden Ergebnissen lässt sich in keinem der Fälle eine schlechtes Ergebnis aufgrund hoher Erfahrung in einer Kategorie erkennen. Das lässt darauf schließen, dass die Anwendung in keinem Bereich schlechter wahrgenommen wird und damit ausfällt, als herangezogene, weil bekannte, Vergleichswerke wie MMORPGs oder gewöhnliche Pen-&-Paper-Rollenspiele.

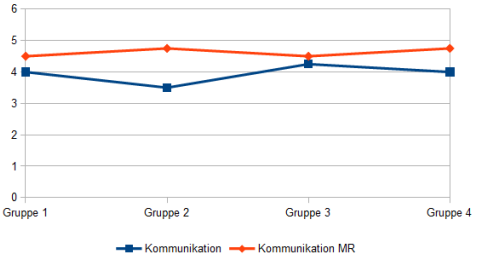
Der empfundene Spaß und die Kommunikation werden unabhängig der Erfahrungen in diesem Bereich durchweg positiv bewertet. Bis auf vier Probanden bewerteten alle den Spaßfaktor unabhängig von der Version, ob mit oder ohne MR-Interaktionen, mit „gut“ oder „sehr gut“. Jeweils zweimal wurde eine neutrale („befriedigend“) sowie ausreichende Bewertung vergeben. Die beiden zuletzt genannten Probanden gaben jedoch auch mangelnde Erfahrung im Bereich von Computer-/Konsolenspielen im Allgemeinen sowie Multiplayerspielen im Speziellen an sowie dass sie an so etwas generell keinen Spaß hätten. Bei der Bewertung der Kommunikation ergeben sich ähnliche Ergebnisse. Alle Probanden sprachen der MR-Version ebenfalls ein *hohes* bis *sehr hohes* Kommunikationsaufkommen zu, wohingegen die Bewertungen der nicht-MR Version lediglich zu ca. 70% in diesem Bereich lagen, was dennoch einem guten Ergebnis entspricht. Die durchweg hohen Werte in dieser Kategorie könnten auf die Interaktionsformen zurückzuführen sein, welche aufgrund des kollaborativen Charakters erhöhte Kommunikation erfordern und in beiden Versionen vorhanden sind.

Weitere Beziehungen zwischen der Vorerfahrung und korrespondierenden Bewertungen lassen sich ebenfalls im Sterndiagramm in Abb. 24.5 erkennen. Mögliche Aspekte, die miteinander in Beziehung stehen könnten, sind in der Abbildung gegenüber angeordnet worden. Erfahrungen im Bereich von Computer-/Konsolenspielen könnten sich beispielsweise stärker auf den Spaß der hier entwickelten Anwendung auswirken, da generell eine Vorliebe fürs Spielen besteht oder ein Vergleich hergestellt werden kann. Ebenfalls auf dieses Kriterium wirken sich Erfahrungen im Bereich von kooperativen bzw. kollaborativen Computer-/Konsolenspielen aus, genau so wie auf die Kommunikation und das Gruppenverhalten. Stärker beeinflusst werden sollte das Gruppenverhalten jedoch durch die Erfahrung in Pen-&-Paper-Rollenspielen, ebenso wie auf das Kommunikationsverhalten. Die Erfahrung in virtuellen Umgebungen bezieht sich hauptsächlich auf den empfundenen Immersionsgrad.

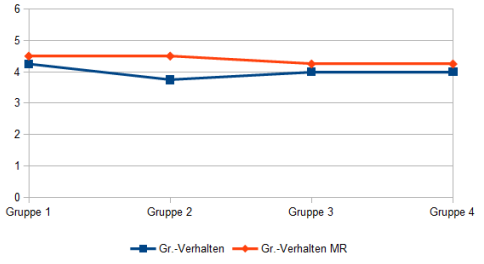
Die MR-Version hat durchgehend höhere Ergebnisse erzielen können als die „normale“ Version. Dies lässt sich in Abb. 24.6 erkennen. Die größte Abweichung besteht bei der Bewertung des Spaßfaktors. Ein Vergleich der Wahrnehmungen der einzelnen Gruppen in den unterschiedlichen Versionen ist in Abb. 24.6(a) dargestellt. Gruppe drei bewertet den Spaßfaktor identisch, ebenso wie die Gruppe eins und vier. Lediglich die zweite Gruppe sieht in diesem Bereich einen erheblichen Unterschied, welcher darüber hinaus den größten aller abgefragten Eigenschaften bildet. Begründet werden könnte dies mit der Vorerfahrung dieser Gruppe im Bereich von gewöhnlichen sowie Mehrspieler-Computer-/Konsolenspielen. Lediglich zwei der Gruppenmitglieder besitzen im Bereich von kollaborativen oder



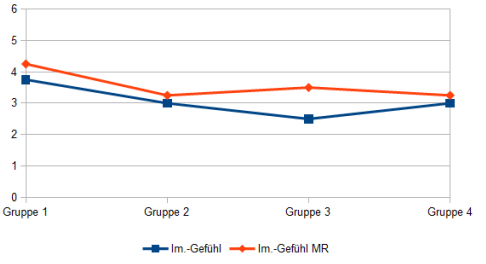
(a) Gruppenmeinungen zum Spaßfaktor



(b) Gruppenmeinungen zum Kommunikationsaufkommen



(c) Gruppenmeinung zum Gruppenverhalten



(d) Gruppenmeinung zum Immersionsgefühl

Abbildung 24.6.: In den Abbildungen werden die Gruppenmeinungen im Hinblick auf die verschiedenen Bewertungskategorien in beiden gespielten Versionen dargestellt. Die Y-Achse stellt die Gruppenbewertungen dar, welche wiederum entlang der X-Achse zu finden sind.

kooperativen Spielen Erfahrung, welche jedoch mit einer Note von „ungenügend“ bewertet wurde und damit nur minimal oder gar nicht vorhanden ist.

Die Reihenfolge der Spielversionen scheint im Allgemeinen auch keinen Unterschied in der Bewertung des Spaßfaktors auszumachen. Prinzipiell ist es denkbar gewesen, dass der zweite Durchlauf weniger Spaß bereitet, da alles bereits bekannt und somit weniger reizvoll wirkt. Insbesondere der Start mit der MR-Version besaß, aufgrund der Einbeziehung der realen Welt und dem neuartigen Charakter, größeres Potential für mehr Spaß und sozialer Interaktion in der Gruppe zu sorgen, welche im zweiten Durchlauf nur noch hätte geringer ausfallen können. Dies lässt sich jedoch widerlegen, da zwar vier Spieler den zweiten Durchlauf (nicht-MR) schlechter bewerten, teilweise sogar mit einer Abweichung von 2 Noten, umgekehrt jedoch kein einziges Mal.

Der Kommunikationsaspekt (Abb. 24.6(b)) wurde in allen Gruppen in der MR-Version höher bewertet, was dadurch begründet wurde, dass die Art der Interaktionen häufig nicht verstanden wurde oder sich

intensiver abgesprochen werden musste. Bei der Mikrophon-Interaktion[454] haben sich Spieler zudem oft den Spaß gemacht kurz vor Ende ein Geräusch zu erzeugen, sodass noch einmal von vorne begonnen werden musste. Das führte bei einigen Mitgliedern zu Gelächter, jedoch in wenigen Malen auch zu angeregten, teilweise negativ intonierten Diskussionen. Insofern scheint die Art der Interaktion hierbei eine tragende Rolle einzunehmen, ebenso wie die Verständlichkeit der Durchführung und die Art der Implementierung. Bei der Kommunikation wurde ebenfalls befürchtet, dass der jeweils zweite Durchgang zu weniger Kommunikation führen könnte, da beispielsweise sowohl die Steuerung, als auch das gesamte Spielkonzept und die Interaktionsdurchführung bereits bekannt war. Auch hier kann dies jedoch widerlegt werden, da fünf Probanden den zweiten Durchgang (nicht-MR) schlechter bewerteten, umgekehrt bei der MR-Version jedoch nur einer.

Das Gruppenverhalten (Abb. 24.6(c)) ist, möglicherweise aufgrund der erzwungenen Kollaboration, insgesamt als sehr hoch empfunden worden. Der Unterschied ist jedoch minimal. Die hohen Werte deuten insgesamt auf ein gutes Ergebnis und ermöglichen Rückschlüsse auf die Akzeptanz von durchgängig kollaborativen Spielen. Wird sowohl das Gruppengefühl, als auch der Spaßfaktor genauer betrachtet, lässt sich erkennen, dass 10% der Einzelbewertungen neutral ausgefallen sind und lediglich in ca. 5% eine schlechtere Note als „befriedigend“ vergeben wurde. Damit lässt sich die These, dass Spielern und Spielerinnen häufig der kollaborative Aspekt in Spielen fehlt bestätigen, ebenso, dass selbiger zu mehr Kommunikation und einem sozialen Miteinander führt. Deutlicher wird dies noch, wenn zusätzlich der Kommunikationsfaktor miteinbezogen wird, da hierbei in lediglich ca. 8% eine schlechtere Note als „gut“ vergeben worden ist.

Im Gegensatz zu den bereits betrachteten Diagrammen lässt sich über das Immersionsgefühl (Abb. 24.6(d)) kaum etwas aussagen. Die Benotung befindet sich zu nah an der Mitte und damit einer neutralen Bewertung im Sinne einer Enthaltung, was mit mangelnder Vorerfahrung in diesem Bereich in Korrelation zu stehen scheint. Ferner reichen, aufgrund der geringen Anzahl an Probanden, bereits einzelne positivere oder negativere Bewertungen von Teilnehmern, für eine signifikante Verzerrung der Ergebnisses in die eine oder andere Richtung.

Die Ergebnisse des unabhängigen Beobachters ergeben ein ähnliches Bild und entsprechen damit den subjektiven Bewertungen der Probanden in den einzelnen Kategorien. Gemeinsames Gelächter trat allgemein sehr häufig auf, insbesondere bei den Interaktionen. Die MR-Interaktionen stechen hierbei besonders heraus und führten zu viel Spaß. Häufig wurde die Übertragung der Spielgeräusche auf die Smartphones als Kommunikationsaufhänger genutzt, da ein ständiges herein- und herauslaufen aus dem Trigger-Radius unterschiedliche Resultate hervorbrachte und den Entdeckertrieb in vielen

[454] siehe Abschnitt 14.3.1 auf Seite 248

Spielern weckte. Am häufigsten führte die Mikrophon-Interaktion und damit die MR-Version der Anwendung zu Spaß innerhalb der Gruppe.

Verbale Aussagen über den Spielspaß wurden am häufigsten in der MR-Version gemacht. Insgesamt wurde viel positiv bewertet, besonders häufig jedoch bei den einzelnen MR-Interaktionen, da diese für die meisten Probanden vollkommen neu waren und „spektakulär“ wirkten. Selbst erfahrene Spieler neigten hier zur Begeisterung, während sehr unerfahrene teilweise sogar überfordert waren und nicht genau wussten, was zu tun ist.

Kollaborationen führten im Allgemeinen zu einer starken Absprache untereinander. Diese nahm allerdings in allen Gruppen im zweiten Durchlauf ab. Dies liegt möglicherweise an der Neuartigkeit der Durchführung, da die Steuerung im ersten Durchlauf noch erlernt und unter den einzelnen Spielern besprochen bzw. erläutert wurde. Häufig existierten Erkunder in den Gruppen, welche vorlaufen und die Karte erkunden wollten, von der Kamera und deren demokratischer Realisierung allerdings aufgehalten wurden. Die Kommentare hierzu waren sowohl positiv als auch negativ, dass dadurch jedoch eine Spaltung der Gruppe verhindert wurde, ist durchgehend positiv bewertet worden. Dies schien das Gruppengefühl und die Kommunikation während des Spiels zu stärken, da die Kameraimplementation dahin zielte, die ganze Gruppe dazu zu animieren zu einem bestimmten Punkt zu gehen. Die hohen subjektiven Bewertungen der Probanden bestätigen diese Beobachtungen zusätzlich. Da für viele Interaktionen mehrere Mitspieler benötigt wurden, bestand eine Vielzahl an Absprachen darin, die noch fehlenden Spieler und Spielerinnen dazu zu bewegen, ebenfalls den Trigger zu betreten[455], um die Interaktion durchführen zu können.

Im jeweils ersten Spieldurchlauf, sowohl bei MR als auch nicht-MR, konnte ein höheres Aufkommen von gegenseitigen Hilfsevents beobachtet werden. Zum einen fanden Gespräche über das Interaktionsgerät und dessen Bedienung statt und zum anderen wurde sich bei der Absprache von Interaktionsdurchführungen geholfen. MR-Interaktionen führten vermehrt zu sozialer Interaktion, da sich gegenseitig geholfen wurde Licht zu sammeln oder die Hand auf dem Gerät zu platzieren[456].

24.3.3. Fazit

Insgesamt kann das Postulat als erfüllt betrachtet werden, da das geplante Vorhaben, soziale Interaktion und damit Kommunikation zu fördern, funktionierte und die Anwendung von den Probanden auch als solche angenommen wurde. Jedoch lässt die Art der Durchführung viel Raum für Verbesserung,

[455] „Komm mal her, hier kann man was machen!“
[456] siehe Abschnitt 14.3.1 auf Seite 248

speziell im Hinblick auf die Spaßbewertungen, da keine ausreichend differenzierbare Unterscheidbarkeit zwischen den zwei Spielversionen attestiert werden kann. Interaktionen mit dem Smartphone durchführen zu können ist für viele bereits so innovativ und mit Freude empfunden worden, dass der Unterschied Interaktionen, dir MR-Aspekte aufweisen, marginal ausfällt. Dies scheint dazu zu führen, dass nicht mehr viel höhere Werte beim Spaßfaktor erreicht werden können. Um eine eindeutige Unterscheidbarkeit zu erreichen, müsste eine rein virtuelle Version erstellt werden, in welcher Spieler Interaktionen, wie gewohnt, rein virtuell durchführen und mit der hier konzeptionierten und implementierten MR-Version verglichen werden.

Ebenfalls hängt der Spaß innerhalb einer Gruppe sehr stark von der Harmonie und vorherrschenden Sympathie in selbiger ab. Sowohl die Zeitspanne die sich die Gruppenmitglieder untereinander bereits kennen, als auch die Tatsache, ob diese gemeinsam Computer- und Konsolenspiele gespielt haben, hat einen Einfluss auf den Spaßfaktor, die Kommunikation sowie das Gruppenverhalten.

Die empfundenen Unterschiede in Kommunikation und Gruppenverhalten sind ebenfalls sehr gering, was zusätzlich durch die Ähnlichkeit der beiden Versionen begründet werden kann. Es sind zwar durchgehend sehr hohe Werte erreicht worden, was als sehr positiv zu bewerten ist, ob Mixed Reality-Anwendungen grundsätzlich jedoch zu mehr sozialer Interaktion führen, muss kritisch hinterfragt und mit weiteren Evaluationen genauer untersucht werden.

Der Begriff der Immersion war für die meisten der Probanden vollkommen unbekannt. Einige dachten dabei zum Beispiel an das Gefühl des Versinkens in einen Film, weil dieser so spannend ist, wiederum andere verbanden direkte körperliche Auswirkungen damit, beispielsweise das der Geruch oder das Klima wahrgenommen werden. 60% der Probanden, welche den Immersionsgrad neutral bewerteten, gaben ebenfalls an, keine Erfahrungen in virtuellen Umgebungen zu besitzen. Die Abgabe der neutralen Bewertung wurde während des Auffüllens der Fragebögen häufig auch verbal mit fehlendem Verständnis begründet. Die Ausweitung der Evaluation um eine vorangestellte immersive Erfahrung in einem Immersion-Square könnte zu besseren Ergebnissen führen, was im Rahmen einer erweiterten Evaluation überprüft werden müsste.

Eine rein virtuelle Version des Spiels und eine ausgeweitete MR-Version mit noch mehr MR-Aspekten könnten ebenfalls besser geeignet sein, da der Unterschied zwischen den Versionen klarer erkennbar wird und somit ein ausgewogenerer Vergleich möglich wäre. Die Auswirkungen der Umgebungshelligkeit oder Tageszeit der Spielenden auf die virtuelle Welt, ebenso wie die Verwendung einer Dolby-Surround-Anlage zur Erzeugung von Atmosphäre in Verbindung mit Projektoren oder einer Nebelmaschine beispielsweise, könnten den umgekehrten Prozess ermöglichen[457].

[457] Durch die Verwendung eines digitalen Steuerungsprotokolls (z.B. DMX [Burghardt, 2009]) könnte all dies ermöglicht werden.

Über alle Ergebnisse sollte noch beachtet werden, dass sich die Gruppenmitglieder teils seit Jahren kannten und auch untereinander befreundet waren. Obwohl dies einerseits dafür sorgte, dass keine Barrieren durch fehlende Kommunikationsbereitschaft auftraten, wurde das generelle Spaß-Level dadurch sehr hoch angesetzt. Es darf vermutet werden, dass die Anwendung dadurch schon einen generell positiv besetztes Ergebnis einfuhr, dass die Gruppenmitglieder schon Spaß daran hatten, zusammen etwas zu unternehmen. Ebenfalls führt die Verwendung von 16 Probanden zu keiner statistischen Sicherheit.

„When my information changes, I alter my conclusions. What do you do, sir?“

(John Maynard Keynes [Keynes, 2014]*)*

KAPITEL 25

Zusammenfassung

In dieser Arbeit wurde ein System konzipiert, implementiert und evaluiert, dessen Komponenten nicht nur viele Aspekte der Informatik, sondern auch weitere Teilbereiche[458] berühren. Dabei wurde von der Fragestellung ausgegangen, wie Menschen in einer Welt der stetig steigenden Vernetzung und Digitalisierung erneut zu mehr persönlicher Interaktion angeleitet werden können, ohne die positiven Aspekte der Technologisierung aus den Augen zu verlieren oder ihnen gar entgegenzuwirken. Die Grundüberlegung hierbei war, dass die wachsende Globalisierung Menschen näher zusammenbringt, allerdings auch entfremdet: Wenn jederzeit alles mit einem Klick herbeigeholt werden kann, so sind auch umgekehrt rein digital entstandene Begegnungen von einer eher flüchtigen Beschaffenheit.

Insbesondere zeigt sich dies am Medium der Spiele: In ihrer ursprünglichen Form dienen diese als höchst soziale Aktivität, welche nicht nur das individuelle Vergnügen, sondern auch den sozialen Zusammenhalt einer Gruppe sowie Freundschaften und Bindungen stärken[459]. Reale Interaktion zwischen verschiedenen Menschen lässt sich nur teilweise durch digitale Elemente substituieren: Sämtliche Nuancen, die ein realer Mensch in seine Kommunikation einbringt, wie Gestik und Mimik sowie die Akustik und das Ambiente des gemeinsamen Kommunikationsraums zeigen der Informatik und speziell der HCI[460] Grenzen auf[461].

Im Rahmen dieser Ausarbeitung stellten sich die Autoren daher die Frage: Wie lässt sich moderne Technologie in einen realen Kontext einbinden, um Menschen wieder zu mehr direkter Interaktion anzuleiten? Die Antwort wurde in einer Verbindung mehrerer Elemente gesucht. Zunächst musste

[458] wie beispielsweise Storytelling und Soziologie

[459] Zur Lektüre sei an dieser Stelle [Fine, 2002] empfohlen.

[460] Genauer an dieser Stelle: der Human-Human-Interaction unter Zuhilfenahme von Computern, da in diesem Beispiel eine Kommunikation zwischen zwei Menschen unter Verwendung eines Computer stattfindet und nicht die Interaktion eines Menschen mit einem Computer.

[461] Wie ließe sich ein Theater- oder Kinobesuch mit den Methoden der Informatik vollständig simulieren? Nach derzeitigem Stand der Technik ist dies zumindest nicht möglich - und selbst wenn es möglich wäre, darf hinterfragt werden, ob dies auch gewünscht wäre.

hierbei eine Thematik gefunden werden, welche per se eine Zusammenarbeit zum Grundprinzip erhebt und damit gleichsam eine Anwesenheit der Spielenden sowie ein kollaboratives Verhalten erfordert. Pen-&-Paper-Rollenspiele sind meistens[462] als Gruppenaktivität ausgelegt, in der sich die Teilnehmenden ergänzen und gegenseitig helfen müssen, um gemeinsam einen Fortschritt zu erzielen. Die realen Aspekte eines Pen-&-Paper-Rollenspiels können, zumindest teilweise, durch digitale Elemente ersetzt werden. Diese Digitalisierung wurde entlang der Betrachtung der einzelnen Elemente eines Pen-&-Paper-Rollenspiels vorgenommen, in deren Verlauf die verschiedenen Komponenten auf ihre Eignung hin bewertet wurden, eine Digitalisierung derselben vorzunehmen[463]. Die Entscheidungen hierzu wurden teilweise in einem der vorliegenden Arbeit vorangegangenen Projekt getroffen und im Laufe dieser Ausarbeitung präzisiert. Dabei wurde ein besonderes Augenmerk auf die Realisierung von kollaborativen Interaktionen gelegt, welche letztlich als Entscheidungspunkte dienen, die der erlebten Geschichte eines Rollenspiels seine Individualität verleihen. Diese Entscheidungen werden von einem digitalisierten Game Master in Form einer regelbasierten künstlichen Intelligenz verarbeitet und führen gegebenenfalls zur direkten Änderungen an der Spielwelt[464]. Die Durchführung dieser Interaktionen sollte in einer Weise stattfinden, welche die Spieler und Spielerinnen zur gegenseitigen Interaktion und Kollaboration[465] anregt, um somit einen Raum für die Entfaltung eines realen, persönlichen Gruppengefühls zur Verfügung zu stellen. Dieser Raum wurde unter Berücksichtigung mehrerer Aspekte aufgespannt. Zunächst sollte sichergestellt werden, dass die Teilnehmenden keine Möglichkeit haben, in den Hintergrund zu treten und gleichsam sollte die Bildung einer Leitfigur nicht möglich sein. Aus diesem Grund wurde unter anderem die Entscheidung getroffen, den Spielenden nicht jeweils einen separaten Bildbereich auf einem Ausgabemedium zur Verfügung zu stellen, sondern alle Avatare gleichzeitig auf einem Bildschirm darzustellen. Um das entstehende soziale Gefüge auch technologisch zu unterstützen, wurde eine demokratische Kamera erdacht und implementiert, die der Mehrheit der Gruppenentscheidungen folgt und abweichendes, individuelles Verhalten minimiert[466].

Zu diesem Zeitpunkt waren zwei Voraussetzungen erfüllt: Spielerinnen und Spieler mussten physisch anwesend sein und konnten sich auch in der virtuellen Welt keinen Rückzugsraum erschaffen. Letztlich bestand noch eine weitere Gefahr der Separation, namentlich die der Spielenden vom Spiel selbst. Diese letzte Möglichkeit wurde adressiert, indem Smartphones als Eingabe- sowie Interaktionsgeräte konzipiert wurden, die nicht nur reale Eingaben in die virtuelle Welt übertragen, sondern gleichsam virtuelle Ereignisse in die reale Welt zurückführen. Unter Einsatz von Vibrationen, Ton, dem Licht-

[462] Eine Auflistung von Einzelspieler-Rollenspielen ist in [Schick, 1991] zu finden.
[463] siehe Abschnitt 2.1 auf Seite 14
[464] siehe dazu Abschnitt 19.3.1 auf Seite 285
[465] Im Rahmen dieser Ausarbeitung galt es die Begriffe der „Interaktion“, „Kooperation“ und „Kollaboration“ neu zu definieren, hierzu wird auf Kapitel 15 ab Seite 251 verwiesen.
[466] siehe Abschnitt 23 auf Seite 305

und Näherungssensor wurde der reale Raum zum Teil der Spielwelt: Die Smartphones bilden damit die Eckpunkte eines Mixed Reality-Volumens[467]. Im Folgenden werden zwei konkrete Beispiele der aus diesen Überlegungen hervorgegangenen Anwendung beschrieben, um entlang dieser Situationen zu erläutern, welche Komponenten auf welcher Art zusammenwirken und was die konkreten Zielsetzungen dahinter waren.

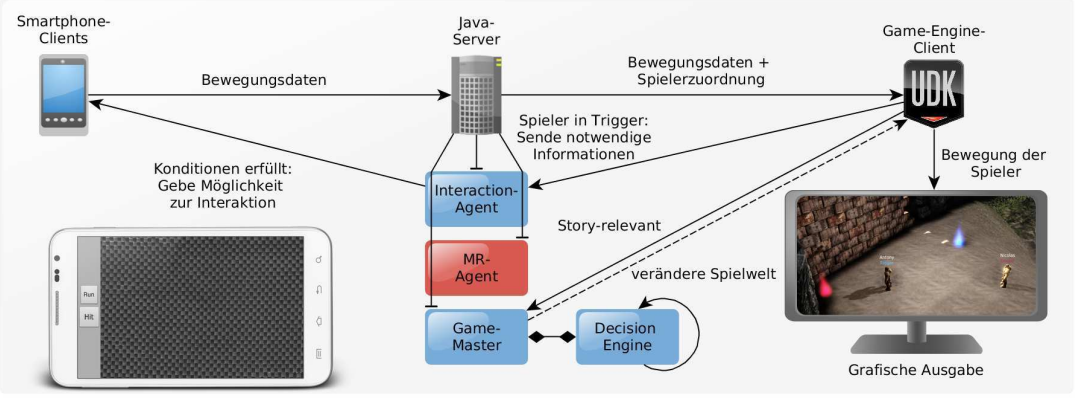


Abbildung 25.1.: In der Abbildung wird der Kreislauf zwischen den verschiedenen Komponenten dargestellt, welche bei der Steuerung der Avatare durch die Spielwelt und dem Betreten eines Triggers involviert sind. Die rote Färbung des MR-Agents symbolisiert dessen fehlende Beteiligung.

Abbildung 25.1 stellt einen Anwendungsfall dar, in dem die teilnehmenden Spielerinnen und Spieler ihre korrespondierenden Avatare in der Spielwelt bewegen und dafür ihre Smartphones benutzen[468]. Innerhalb der Spielwelt laufen sie in einen sogenannten Trigger[469], welcher eine Interaktion ermöglicht und relevant für den Fortgang der Story ist. Die Smartphones werden als Interaktionsgeräte verwendet und senden dem Server individuelle Bewegungsdaten, welche von diesem um die Information erweitert werden, welcher Avatar dem jeweiligen Smartphone zugeordnet ist, um die Gesamtinformation unmittelbar an den Game-Engine-Client weiterzuleiten. Diese Daten werden daraufhin umgerechnet und in das von der Game-Engine[470] benötigte Format gebracht[471].

Dem folgend werden die zugehörigen Avatare innerhalb der virtuellen Welt damit beauftragt, diese Bewegungsanweisung umzusetzen, was letztlich auf dem Ausgabebildschirm entsprechend in einer

[467] Diese Begriff wurde mit Rückbezug auf das Konzept der vierten Wand gewählt. Eine ausführliche Betrachtung dieser Gedankengänge ist in Abschnitt 13.1 auf Seite 229 zu finden.
[468] siehe Abschnitt 22.1 auf Seite 298
[469] siehe Abschnitt 18.2 auf Seite 273
[470] konkreter: dem UDK
[471] siehe Abschnitt 23.3 auf Seite 312

Animation und Positionsveränderung dargestellt wird. Dieser Zyklus wurde so latenzfrei wie möglich implementiert[472] und durch die Ergebnisse einer ausführlichen Evaluation bestätigt[473].

Sobald einer der Avatare einen Trigger-Bereich betritt, analysiert der Game-Engine-Client diesen Avatar und sendet eine Nachricht mit allen notwendigen Informationen an den Server respektive an den darauf befindlichen Interaction-Agent[474]. Da es sich bei dieser Interaktion zusätzlich um ein Story-relevantes Ereignis handelt, erhält der Game Master[475] ebenfalls eine Benachrichtigung. Dieser nimmt eine Extraktion der notwendigen Informationen vor und übergibt diese an die zugrundeliegende Decision Engine[476]. Damit diese möglichst nah an einen menschlichen Spielleiter heranreicht und zudem langfristige Entscheidungen sowie das gleichzeitige Eintreten mehrerer Konditionen ermöglicht, wurde für sie eine regelbasierte KI implementiert[477]. Durch das Hinzufügen von Wissen wird die Entscheidungsfindung angestoßen und hinterlegte Regeln überprüft. Sofern eine dieser Regeln eintritt, wird dies dem Game Master mitgeteilt, welcher daraufhin unter der Verwendung eines RPCs direkt die Spielwelt verändert, um beispielsweise eine Tür zu öffnen.

Der Interaction-Agent überprüft hierbei alle Bedingungen und übernimmt ab dem Zeitpunkt des Nachrichtenerhalts die Überwachung der durchführbaren Interaktion. Sobald alle Bedingungen erfüllt wurden, beispielsweise alle benötigten Spieler mit den richtigen Rollen ebenfalls den Trigger betreten haben[478], werden alle betroffenen Smartphone-Clients über die Möglichkeit zu einer Interaktion informiert. Dies forciert bereits eine direkte Kommunikation der Spieler untereinander, da diese sich absprechen müssen, um gleichzeitig den Interaktionsraum zu betreten oder eine Interaktion durchzuführen. Abbildung 25.2 stellt das weitere Vorgehen dar, sobald eine Interaktion tatsächlich angestoßen wurde.

Mit der Bestätigung des Interaktionshinweises auf den Smartphones wird in die korrespondierende Interaktionsansicht gewechselt und die eigentliche Durchführung kann beginnen. Interaktionen wurden unter anderem auf einen kollaborativen Charakter ausgelegt, indem die „gleichzeitige“ Durchführung auf den einzelnen Eingabegeräten erfordert wird[479]. Des weiteren wurden Interaktionen entwickelt, welche zu einer Vermischung der realen und virtuellen Welt führen, um das Mixed-Reality-Volumen aufzuspannen[480]. Der Interaktionsdurchführung folgt wiederum das Senden von Nachrichten an den Server bzw. den Interaction-Agent, welcher diese weiterverarbeitet und mit den Konditionen der Inter-

[472] siehe Kapitel 17 ab Seite 265
[473] siehe Abschnitt 24.1 auf Seite 319ff
[474] siehe Abschnitt 19.1 auf Seite 277
[475] siehe Abschnitt 19.3 auf Seite 284
[476] siehe Abschnitt 19.3.2 auf Seite 285
[477] siehe Abschnitt 16.1 auf Seite 257
[478] Was ebenfalls jedes Mal das Senden von Nachrichten durch den Game-Engine-Client auslöst.
[479] siehe Kapitel 15 auf Seite 251
[480] siehe Abschnitt 14.3.1 auf Seite 248

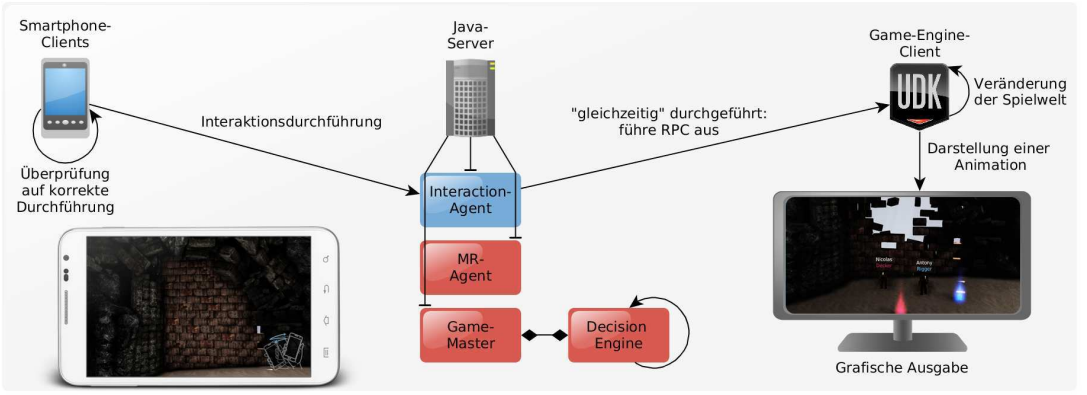


Abbildung 25.2.: In der Abbildung wird der Kreislauf zwischen den verschiedenen Komponenten dargestellt, welche bei der Durchführung einer Interaktion involviert sind. Die rote Färbung symbolisiert auch hier die fehlende Beteiligung.

aktion abgleicht. Die Registrierung einer „Gleichzeitigkeit“ in der Interaktionsdurchführung führt zur direkten Veränderung der Spielwelt durch die Verwendung eines RPCs, welcher mittels einer Nachricht an den Game-Engine-Client gesendet wird. Dieser nimmt die Veränderung vor und konfrontiert die Spieler mit der neuen Situation, welche auf dem Hauptbildschirm dargestellt wird.

Das Postulat der Arbeit, dass die Einbringung von Mixed Reality Aspekten tatsächlich zu einer erhöhten Interaktion zwischen den Teilnehmenden führt, konnte im Rahmen einer vergleichenden Evaluation weitestgehend bestätigt werden. Hierbei wurde jeweils eine Version der Anwendung mit und eine Version ohne Mixed Reality-Aspekte von mehreren Gruppen gespielt. Beide Versionen stießen dabei in verschiedenen Gruppen auf überwiegend positives Feedback, wobei eine leichte Tendenz zugunsten der Version mit Mixed Reality-Aspekten erkennbar ist. Ob eine eindeutigere Unterscheidbarkeit der beiden Versionen oder eine erhöhte Anzahl an Probanden zu aussagekräftigeren Ergebnisse führen könnte, gilt es in weiteren Forschungsprojekten in Erfahrung zu bringen.

Anhand der in diesem Kapitel beschriebenen Anwendungsfälle wurde der Kreislauf innerhalb des konzipierten und implementierten Systems mit all seinen Komponenten dargestellt und verdeutlicht. Das hier entwickelte Konzept darf im Vergleich zu ähnlichen Ansätzen insbesondere aus zwei Gründen als neuwertiges Konzept klassifiziert werden. Im Gegensatz zu stark ortsbezogenen Spielen ermöglicht das entwickelte System, durch das Einbeziehen der Umgebung, beim Durchspielen an verschiedenen Spielorten das Erleben unterschiedlicher Erfahrungen. Dies hängt damit zusammen, dass das Spielerlebnis und die Durchführung bestimmter Interaktionen teilweise in direkte Abhängigkeiten zu realweltlichen Ereignissen gesetzt wurde. Darüber hinaus ist zwar eine Ortsgebundenheit erwünscht,

damit der soziale Aspekt sowie die direkt Kommunikation gefördert werden, jedoch nicht zwingend erforderlich: Der Systemaufbau erlaubt auch eine verteilte Anwendung über mehrere Orte. Dabei ist nicht nur die Erweiterungsfähigkeit des Systems gemeint, mit der beispielsweise mehrere Gruppen zusammen in einem größeren Kontext an verschiedenen Orten zusammenspielen können, sondern auch die Portabilität des Systemaufbaus, sodass ein Laptop mit WLAN und ein paar Smartphones ausreichen, um ein Spiel zu beginnen - und zwar überall auf der Welt und zu jeder Zeit. Das weitere Potential des Systems sowie dessen Erweiterungsmöglichkeiten sollen im Rahmen des nun folgenden Ausblicks zusätzlich verdeutlicht werden. Im Anhang unter A befindet sich darüber hinaus eine kleine Bilderserie der entwickelten Anwendung zum Sammeln von Eindrücken.

„[…] we know what we are, but know not what we may be."

(William Shakespeares „Hamlet", Akt 4, Szene 5*)*

KAPITEL 26

Ausblick

Mit Abschluss der Evaluation wurde ein Projektstand bewertet, der auf vielerlei Wegen zur weiteren Forschung und für Folgeprojekte herangezogen werden kann. Einer dieser möglichen Wege ist die Präzisierung der Evaluationsergebnisse beziehungsweise die Durchführung von Folgeevaluationen unter Berücksichtigung weitergehender Fragestellungen. Eines der Kernthemen und somit auch der zentralen Fragestellungen dieser Ausarbeitung ist die nach dem Zusammenhang von Mixed Reality-Aspekten, zwischenmenschlicher Interaktion und Kommunikation. Dies öffnet eine interdisziplinäre Tür, hinter der sowohl ethnologische, psychologische, linguistische als auch allgemein humanwissenschaftliche Aspekte betrachtet werden können. Im Hinblick auf die Evaluation könnte beispielsweise auch ein Ansatz verfolgt werden, in welchem die Verwendung eines mehrdimensionalen Befindlichkeitsfragebogens (MDBF) eine zentrale Rolle einnimmt. Eine andere Idee wurde von den Autoren Sweetser u. Wyeth vorgestellt, bei der es sich um den vielfach zitierten[481] und weitestgehend anerkannten Ansatz des *GameFlow* [Sweetser u. Wyeth, 2005] handelt. Damit wird speziell das Engagement eines Spiels geprüft, dessen Erzeugung und Aufrechterhaltung unter anderem als Teilziel des hier entwickelten Systems vorhanden war. Da der Ansatz viele verschiedene Kriterien berücksichtigt und insgesamt sehr umfangreich ist, müsste kritisch überprüft werden, ob die Kriterien zur Abfrage im Status quo des Systems vorhanden sind oder ob Module und/oder Funktionen nachgerüstet werden müssen. Eine solche Analyse des bestehenden Systems auf diese Kriterien hin, gefolgt von einer solchen Evaluation, sollte im Rahmen einer Bachelorarbeit zu bewerkstelligen sein.

Da die Akzeptanz, das Engagement sowie der Spaß der Anwendung im Allgemeinen sehr von dem Anwendungsdesign und damit einhergehend der Benutzerfreundlichkeit sowie einer möglichst intuitiven Benutzeroberfläche abhängen, existieren auch an dieser Stelle diverse Evaluations- und Erweiterungsmöglichkeiten. Es ist ein schwieriges Problem, die Einflussnahme und Steuerung eines digitalen Systems an zentraler Stelle aufzugeben und die Kontrolle an die Nutzer zu übergeben. Die sich daraus ergebenden Konsequenzen lösen eine Schleife technischer und sozialer Wechselwirkungen aus, die

[481] vgl. [Scholar, 2014]

sich wiederum auf die Gruppendynamik und die Interaktionen auswirken. Das digitale, informationsverarbeitende System respektive die dahinter liegende Anwendung stellt zwangsläufig einen Flaschenhals dar, nach dessen Eigenschaften und Fähigkeiten sich die Nutzerinnen und Nutzer letztlich richten müssen. Bei ungeeigneten oder komplizierten Eingabegeräten ist die Gefahr gegeben, dass die Menschen bei kollaborativen Aufgaben eher damit beschäftigt sind, sich gegenseitig über die Steuerung abzusprechen, als mit der eigentlichen Interaktion beschäftigt zu sein. Dagegen kann belegt werden, dass gut designte und intuitive Benutzerschnittstellen eine ausgeglichenere Partizipation fördern können [Rogers u. a., 2009] [Israel u. a., 2012, S. 53][482]. Speziell der Einsatz eines mobilen Endgeräts als Interaktionskomponente stellt neue Anforderungen an die Human-Computer Interaction und, damit zusammenhängend, das HCI-Design. Die weltweit parallel voran getriebene Forschung sowohl von Hard- als auch Software im Bereich der Smartphone-Entwicklungen lässt die resultierenden Geräte immer stärker diversifizieren. Laut Lumsden leidet die Benutzerfreundlichkeit in direkter Folge unter diesem Umstand: Die Akzeptanz und Verbreitung von Smartphone-Anwendungen steht in direktem Zusammenhang mit der Usability, sodass diese Aspekte inzwischen immer mehr Aufmerksamkeit erfahren [Lumsden, 2011] [Lumsden, 2011, S. xvi] [483]. Demnach könnte eine weitere Abschlussarbeit dafür genutzt werden, Usability-Studien für die Smartphone-Anwendung durchzuführen, um damit auch generell die Forschung im Bereich der mobilen Interaktionsgeräte voranzutreiben. Das Design mobiler Anwendungen kann in vielerlei Aspekten mit bisher bekannten Heuristiken für Benutzeroberfläche überprüft werden, jedoch existieren hierbei speziellere Anforderungen oder Einschränkungen[484]. Hierfür könnten z.B. die in [Lumsden, 2011, Kap. 2] erläuterten Verfahren in einer gesonderten Evaluation zur Anwendung kommen.

Neben den Möglichkeiten weiterer Evaluationen existiert zudem Erweiterungspotential im Hinblick auf das Gameplay sowie die Interaktionsvielfalt und -durchführung. Die bisher punktuell eingesetzten MR-Aspekte könnten dabei sowohl innerhalb der virtuellen als auch in der realen Welt ausgeweitet werden. Dies könnte beispielsweise dazu führen, dass der Campus der Hochschule zeitweise zur Spielwelt wird, sodass einzelne Spieler oder auch ganze Gruppen von Spielern Aufgaben in örtlich getrennten Räumlichkeiten agieren müssen. Wahlweise kann hierbei das Smartphone direkt zur Navigation verwendet werden oder der beauftragte Spieler erhält zwar eine Karte, muss sich jedoch von den anderen Spielern, welche ebenfalls eine Karte und zusätzlich die Position des Spielers auf dem Hauptbildschirm angezeigt bekommen, durch die reale/virtuelle Welt „führen“ lassen. Auf dieser Karte könnten Icons dargestellt werden, welche in der realen Welt vorhandene Gegenstände oder Interaktionsmöglichkeiten repräsentieren, beispielsweise in Form von QR-Codes[485]. In diese Überlegung kann

[482] Als weiterführende Literatur werden hierbei sowohl [Jones u. Marsden, 2005] als auch [Rogers u. a., 2011] empfohlen.
[483] siehe auch *International Journal of Mobile Human Computer Interaction (IJMHCI)*[Global, 2013]
[484] kleineres Display, unterbrochene Aufmerksamkeit der Nutzer, …
[485] QR-Codes stellen eine modernere Variante der Barcodes dar [Hartz, 2013].

mit einfließen, dass die Anwendung auch den Betrieb mehrerer Game-Engine-Clients erlaubt, sodass das Spiel auch verteilt an mehreren Orten gleichzeitig gespielt werden kann. So wird die reale Kommunikation innerhalb jeder Gruppe weiterhin beibehalten, während die Gruppen untereinander digital vernetzt sind. Für weiterführende Informationen sei an dieser Stelle beispielsweise die Lektüre von [Reimann u. Paelke, 2005] empfohlen.

Darüber hinaus existieren im Anwendungsdesign sowie der Implementierung weitere Möglichkeiten zur Verbesserung. Ein erster Ansatz wäre eine weiterführende Entkopplung der Daten und Logik. Das Hinzufügen neuer Interaktionarten auf Seiten der Smartphone-Anwendung sowie die Platzierung und Einrichtung von Interaktions- und Story-Triggern könnte durch den Einsatz eines eigenen Protokolls erweitert und vereinfacht werden. Durch die Verwendung einer zusätzlichen Benutzeroberfläche könnten Trigger-Objekte ohne Zuhilfenahme des UDK in der Spielwelt verteilt werden. Derzeit ist eine funktionsfähige Installation und ein grundlegendes Verständnis der Game-Engine notwendig, um hier Änderungen vornehmen zu können. Selbiges gilt für die Modifikation der erzählten Story, sowohl im Hinblick auf die Textbausteine als auch auf Abhängigkeiten von Modulen und möglichen Vor- bzw. Nachbedingungen. Programmierer bieten Designern sowie Spielern häufig eine Reihe von Schnittstellen an, durch die ein Spiel verändert werden kann. Ein Scripting-basierter Ansatz könnte das Spiel für eine breitere Masse von Inhaltsanbietern zugänglich machen, damit Erweiterungen leichter vorgenommen werden können. Dafür müsste zusätzlich eine eigene Syntax sowie die korrespondierende Semantik konzeptioniert und implementiert werden. Dies ließe sich womöglich im Rahmen einer Masterarbeit bewerkstelligen.

Die realisierte Netzwerktopologie enthält ebenso noch Raum für Verbesserungen. Einer der signifikantesten Vorteile einer Client-Server-Architektur ist die Unabhängigkeit des Servers zu den verbundenen Clients. In der realisierten Anwendung wird derzeit jedoch nicht abgefangen, wenn ein Client die Verbindung verliert, sodass daraufhin das ganze System stoppt. Hier muss nicht nur eine technische, sondern auch eine logische Lösung erdacht werden. Kann das Spiel von drei Spielenden weitergespielt werden, wenn ein Teilnehmer beispielsweise das Spiel verlässt, absichtlich oder unabsichtlich? Wie wirkt sich das auf die Story aus? Welche Lösung kann dafür erdacht werden, dass zur Durchführung bestimmter Interaktionen teilweise besondere Spieler-Rollen vorhanden sein müssen? Soll oder *muss* dann eine Neuauswahl der Rollen möglich sein? Eventuell ist eine extra dafür vorgesehene KI besonders geeignet, um automatisch Anpassungen an die Spielwelt vornehmen zu können. Diese Fragen können Teil einer Bachelorarbeit oder eines Masterprojekts werden, wobei eine Implementierung als Proof-Of-Concept sicherlich in den Aufwandsbereich einer Masterarbeit hinein reicht.

Letztlich bestehen Möglichkeiten innerhalb der Intensivierung des Spielerlebnisses im Hinblick auf die Kombination von Story, Gameplay und Interaktionen. Es ist denkbar, dass die Anwendung beispiels-

weise um die Fähigkeit zur Kommunikation mit einer DMX-Schnittstelle erweitert wird, um Ereignisse der Virtualität mit Nebelmaschinen, Lasern, Suchschweinwerfern, Hebebühnen und weiteren Ereignissen in der Realität zu verbinden [Burghardt, 2009]. Umgekehrt wäre es ebenso möglich, Ereignisse in der virtuellen Welt durch Live-Eingaben, beispielsweise unter Verwendung eines Midi-Keyboards, auszulösen. Es wäre ein interessantes Forschungsfeld, wie der Gedanke der Modularität weiter gesponnen werden kann. Können Menschen beispielsweise in Echtzeit Module erzeugen und dem Spiel hinzufügen? Vielleicht auch über das Internet? Welche Grenzen weist das Konzept der „Nested Modules“ respektive der verschachtelten Module auf und wie können diese Grenzen durchbrochen werden? Wie können mehrere Gruppen verteilt über mehrere Räume mit mehreren Game-Engine-Clients an der gleichen Geschichte spielen? Schließlich: Lässt sich so ein ganzes Pen-&-Paper-Universum mit mehreren Spielinstanzen, wie das Universum von *Shadowrun* verwirklichen?

Sankt Augustin, 7. März 2014

- Antony Konstantinidis & Nicolas Kopp

Glossar

Das Glossar beinhaltet eine Auflistung von wichtigen Begriffen der Arbeit mit dazugehörigen Erklärungen und Seitenzahlen.

Avatar [68, 108, 109, 135, 192, 217, 262, 270, 282, 284, 297, 298, 307, 323, 340]

Als Avatar wird unter anderem in Computer- und Videospielen die virtuelle Repräsentation eines Spielers bezeichnet. Diese kann eine wie auch immer geartete animierte und interaktionsfähige Figur darstellen, allerdings können auch simple Bilder bereits als Avatar angesehen werden.

Backtracking [181]

Das Backtracking beschreibt die Möglichkeit innerhalb eines Problemfindungsansatzes einen Problempfad zurückzuverfolgen. Es wird ein beliebiger Pfad bis zum Ende respektive dem Zeitpunkt, an dem eine Weiterverfolgung ohne weiteren Gewinn verläuft, verfolgt, woraufhin der jeweils letzte Schritt / die jeweils letzten Schritte automatisch zurückgenommen wird/werden, um einen anderen, alternativen Pfad zu verfolgen.

Bot [299]

Im Bereich der Spielentwicklung sowie im Allgemeinen dem Spielsektor entspricht ein Bot einem NPC.

Charakterbogen [6]

Der Charakterbogen ist ein in Pen-&-Paper-Rollenspielen verwendetes Hilfsmittel, mit dem die Spielenden bestimmte Werte und Eigenschaften ihres Charakters verschriftlichen können. Da Charaktere auch über Monate und Jahre gespielt werden können und oftmals über eine Vielzahl an Gegenständen und Eingenschaften verfügen, ist es sinnvoll, diese schriftlich festzuhalten. Charakterbögen haben oft ein Regelwerk-Spezifisches Design und sind in Vordrucken vorhanden.

Crowdfunding [115]

Crowdfunding bezeichet eine Art der Kapitalbeschaffung, bei der viele Einzelpersonen durch geringe Beträge einen oftmals hohen Gesamtbetrag zusammentragen. Projekte, welche sich diese Finanzierungsart zu Nutze machen, bieten potentiellen Unterstützern besondere Gegenleistungen je nach Höhe des Spendenbeitrags an. Bei der Finanzierung von Spielen wird häufig das Eintragen als Beta-Tester angeboten sowie das Benennen von Kreaturen im Spiel mit dem Namen des Spenders.

Dalvik Virtual Machine [47]

Die Dalvik Virtual Machine ist die virtuelle Ausführungsumgebung des Betriebssystems von Google. Entwickelt wurde die DVM für mobile Betriebssysteme im Allgemeinen. Der in Java geschriebene Programmcode wird zu Bytecode kompiliert und in der Dalvik VM ausgeführt.

Deadlock [42, 43]

Ein Deadlock beschriebt den Zustand eines Systems, in dem mehrere Teilnehmer und/oder Komponenten auf Ressourcenfreigabe des jeweils anderen Teilnehmers warten. Da keine Komponente mit der Ausführung der Aufgabe fortführen kann, ohne dass ein anderer Teilnehmer diese Komponente freigibt, ergibt sich hierdurch ein "festfahren"des Systems, was letztlich als "Toter Verschluss", engl. "Deadlock"bezeichnet wird. Häufig wird dies durch das Beispiel der fünf speisenden Philosophen erläutert, siehe [Jobst, 2011]

Decision Engine [IX, 7, 8, 83, 112, 276, 284–286, 342]

Eine Decision Engine (dt. „Entscheidungsmaschine“) ist eine Black-Box, die anhand gegebener Informationen eine Entscheidung trifft. In Computer- und Videospielen werden diese oft als endlicher Automat realisiert, welcher in Abhängigkeit der Eingabeparameter in verschiedenen Zustände wechselt, welche dann die Entscheidung darstellen.

E-Book [123]

Ein E-Book bezeichnet ein Buch, welches in digitaler Form vorliegt und unter Verwendung spezieller E-Book-Reader gelesen werden kann. Häufig findet keine Unterscheidung im Datenformat statt, sodass ein einfaches PDF-Dokument ebenfalls als E-Book bezeichnet werden kann. Damit erweitert sich die Möglichkeit des Lesens auf jedes Gerät, welches PDF-Dateien verarbeiten kann, u.a. Smartphones, Tablets sowie gewöhnliche Computer.

Easter Egg [229]

Ein Easter Egg ist eine von den Herstellern einer Software, eines Spiels, eines Films und manchmal auch von Alltagsgegenständen platzierte Überraschung, die bei normaler Benutzung nicht auffällt. Meist sind Easter Eggs in irgendeiner Form humoristisch besetzt oder referenzieren andere popkulturistische Objekte oder Gegebenheiten.

Ego-Shooter [88]

Als Ego-Shooter wird eine Gattung von Computer- und Videospielen bezeichnet, welche als Perspektive des Geschehens eine Ich-Ansicht wählen. Oftmals ist diese Ansicht in solchen Spielen anzutreffen, in denen der Spieler gleichzeitig eine Waffe trägt, daher erklärt sich der Zusatz „Shooter“.

Engagement [137, 261, 265, 291, 321, 329, 345]

Beim Engagement handelt es sich um ein subjektives Gefühl der Versunkenheit in eine Aufgabe. Wird häufig mit dem Begriff der *Präsenz* gleichgesetzt [Heisenberg, 2012b, S. 13-14].

Enterprise JavaBeans [X, 47]

Enterprise JavaBeans beschreiben eine standardisierte Komponente innerhalb der Java Enterprise Edition (Java EE). Bei dieser speziellen Edition handelt es sich um ein Framework was insbesondere für die Entwicklung von Unternehmensapplikationen geeignet ist. Im Vordergrund stehen verteilte und web-basierte Anwendungen.

Erfahrungspunkt [6, 106]

Erfahrungspunkte sind eine Art Belohnung für durchlebte Situationen. Mit Hilfe des Konzeptes der Erfahrungspunkte kann ein virtueller Charakter individualisiert werden, indem beispielsweise aus einem vorgegebenem Kontingent an erhaltenen Erfahrungspunkten eine gewisse Anzahl in Attributwerte investiert wird. Vereinfacht ausgedrückt: Ein Charakter erhält beispielsweise 10 Erfahrungspunkte und man verteilt 5 davon auf „höher springen“ und 5 auf „schneller laufen“.

Finite State Machine [X, 180]

Ein endlicher Automat besteht aus einer Menge von endlichen, vorher festgelegten Zuständen und befindet sich zu einem Zeitpunkt immer nur in einem dieser Zustände. Die Zustände implementieren das

Verhalten des Automaten. Zustände können mit definierten Zustandsüberführungen gewechselt werden. Differenzierter betrachtet existiert in der theoretischen Informatik eine weitere Unterscheidung von endlichen Automaten in Akzeptoren und Transduktoren. Siehe Abschnitt 9.3.2

Flash [33]

Flash ist ein Browserplugin von Adobe, welches zur Anzeige multimedialer und interaktiver Elemente verwendet werden kann. Vor der Entwicklung von HTML5 und anderen Technologien war dies die einzige Möglichkeit und bis heute noch am weitesten verbreitete um z.B. Spiele im Browser anbieten zu können.

Game-Engine [19, 108, 111, 270, 273, 279, 287, 297–299, 320, 323]

Eine Game-Engine kann laut Amresh u. Okita folgendermaßen definiert werden:

> *„A game engine is a piece of software or code written by programmers to facilitate the game production pipeline. It can serve various needs specific to a particular game or in general to all games. [...] Some of the [...] common elements include camera systems, user interface, animation, physics and artificial intelligence.“*

[Amresh u. Okita, 2010, S. 353]

Durch Einsatz einer Game-Engine muss sich Spieldesigner respektive Programmierer muss sich nicht mehr tiefer Ebene darum kümmern, wie beispielsweise ein Bild Hardware-seitig auf einem Monitor dargestellt wird, sondern kann sich direkt auf hoher Abstraktionsebene mit der eigentlichen Implementierung der Spielidee auseinander setzen.

Goldener Pfad [78, 159]

Der „Goldene Pfad“ wird von Sheldon folgendermaßen erläutert:

> *„The Golden Path: In Computer Games, this is the optimum path a player may take through a game to experience the game design to its fullest.“*

[Sheldon, 2004, S. 169]

Pen-&-Paper-Rollenspiele zeichnen sich unter anderem dadurch aus, dass diese einen solchen Pfad nicht besitzen, da die Geschichte zur Laufzeit von den Spielern vorangetrieben und wird.

Google Play [44]

Google bietet auf der Plattform „Google Play“ Android-Applikationen, Spiele, Musik, Bücher sowie Filme an. Vor März 2012 existierten einzelne Angebote hierfür. Besonders bekannt war dabei der Android Market. Bei Google Play handelt es sich um Google's Pendant zu dem App Store von Apple und Windows Phone Store von Microsoft.

Immersion [140, 221, 248, 291, 329, 332]

Bei der Immersion geht es um das Eintauchen in die virtuelle Welt. Hierzu ist es notwendig, möglichst viele der menschlichen Sinne zu stimulieren, was unter anderem einen möglichst hohen Realitätsgrad und möglichst geringe Latenzen bei der Systemnutzung voraussetzt [Heisenberg, 2012b, S. 12].

Interaktion [103, 106, 131, 178, 197, 227, 242, 252, 277, 312, 334, 339, 347]

„[Interaktion, die]: aufeinander bezogenes Handeln zweier oder mehrerer Personen; Wechselbeziehung zwischen Handlungspartnern“ [Bibliographisches Institut GmbH, 2013]. Der Begriff kann insbesondere im heutigen, modernen Zeitalter des Computers, für mehr stehen als nur für einen Vorgang, welcher sich zwischen zwei oder mehr involvierten Menschen abspielt. Für mehr Informationen hierzu wird auf Kapitel 6 verwiesen. Außerdem galt es für die im Rahmen dieser Ausarbeitung implementierten Interaktionen eine eigene Definition zu finden, welche in Kapitel 15 zu finden ist.

Internet Protocol [X, 30, 31, 201, 202, 204]

Das Internet Protocol bildet die Grundlage für die Kommunikation von Netzwerkprogrammen. Es stellt die essentielle Funktionalität für Protokolle auf der Transportschicht zur Verfügung. IP ist ein verbindungsloses Protokoll, welches unzuverlässig ist, da ein Empfang von Paketen nicht gewährleistet wird.

Internet Service Provider [X, 38]

Ein Internetdienstanbieter stellt Kunden Dienste und Services zur Verfügung, welche für die Nutzung des Internets und darüber hinaus benötigt werden. Dies kann sowohl ein Internetanschluss, als auch ein Webspace oder kompletter Server sein.

JavaScript Object Notation [X, 91]

JSON ist ein Datenaustauschformat. Der Vorteil liegt darin, dass die Struktur auf einer Sammlung von Key-Value-Paaren und einer sortierten Liste beruht, welche beide in nahezu jeder Programmiersprache existieren. Damit kann dieses Format von fast allen Entwicklern zum Datenaustausch verwendet

werden. Hinzukommt, dass es sowohl für Menschen einfach zu lesen und schreiben ist, ebenso wie für Maschinen.

Kollaboration [116, 231, 329, 330, 340]

„When collaborating, people work together (co-labor) on a single shared goal." [Sellier, 2014]. Im Kontext von Spielen betrachtet definieren Zagal u. a. den Begriff wie folgt: *„In collaborative games, players share common goals and outcomes. All gamers either win or lose together."* [Voida u. a., 2010, S. 371]. Bei der Kooperation hingegen wird an unterschiedlichen Zielen gearbeitet, welche jedoch beispielsweise einen gemeinsamen Austausch in Form von Erfahrungen oder Wissen ermöglichen. Es handelt sich somit um eine weniger starre Abhängigkeit als bei der Kollaboration. [Kaul, 2012]
Im Rahmen dieser Ausarbeitung musste eine Definition auf anderer Ebene gefunden werden, hierzu wird auf Kapitel 15 verwiesen.

Kooperation [116, 329]

„When cooperating, people perform together (co-operate) while working on selfish yet common goals." [Sellier, 2014]. Im Bereich der Spielentwicklung existiert eine ähnliche Definition: *„In a cooperative game, players' goals are not necessarily in direct opposition, but neither are their goals completely aligned."* [Voida u. a., 2010, S. 371]. Insofern steht Kollaboration als Begriff für das Hinarbeiten auf ein gemeinsames Ziel. Es können dabei verschiedene Rollen eingenommen werden und unterschiedlich ausgeprägte Beiträge zum Erreichen des Ziels beigetragen werden. [Kaul, 2012]
Im Rahmen dieser Ausarbeitung musste eine Definition auf anderer Ebene gefunden werden, hierzu wird auf Kapitel 15 verwiesen.

Koplanarität [193]

In der Geometrie werden Punkte als koplanar bezeichnet, sofern dieses sich in einer Ebene befinden.

Künstliche Intelligenz [XI, 117, 186, 285, 286, 340]

Die Definitionen für Künstliche Intelligenz (KI) (engl. artificial intelligence) reichen von einfachem Pathfinding, über den Vergleich mit Individuen, welche über Intelligenz verfügen, bis hin zu sehr komplexen Definitionen, welche ebenfalls ein Lernen und eigenes Schlussfolgern der künstlichen Intelligenz erfordern. Dies liegt zum Teil daran, dass der Forschungszweig der künstlichen Intelligenz nicht ausschließlich im Bereich der Spieleentwicklung existiert, sondern viele verschiedene Anwendungsgebiete wie unter anderem das der Spracherkennung, Robotik sowie Mustererkennung umfasst [Dorn u. Gottlob, 2006, S. 975ff]. Siehe Kapitel 9.

Local Area Network [XI, 35]

Ein LAN beschreibt ein lokales Netzwerk, welches durch den Zusammenschluss mehrerer Computer oder anderer technischer Geräte zustande kommt. Eine spezielle Form eines solchen Netzwerkes ist das WLAN, bei welchem der Zusammenschluss durch ein lokales Funknetz erreicht wird.

Loopback-Device [72, 297]

In der Netzwerktechnik bezeichnet ein Loopback-Device ein Gerät, bei dem Sender um Empfänger quasi identisch sind. Jedes darüber gesendete Paket, wird unmittelbar als empfangenes Paket an den Netzwerkstack des Systems übergeben. Es wird somit keinerlei physikalische Verbindung verwendet.

Massive Multiplayer Online Role Playing Game [XI, 136, 158, 199, 332, 358]

Bei einem MMORPG handelt es sich um ein Online-Rollenspiel, welches über das Internet gleichzeitig von vielen hunderten Spielern gespielt wird. Eine virtuelle Welt bildet dabei einen unveränderlichen Raum in welchem sich die Spieler aufhalten. Typischerweise geht es bei MMORPGs um die Interaktion zwischen Spielern. Es ist möglich Aufgaben, so genannte Quests, in Gruppen oder auch alleine zu lösen. Die Erfüllung solcher Quests ist jedoch nicht zwangsläufig notwendig, da es sich um eine freie und offene Welt handelt, die nach Belieben durchlaufen werden kann.

MTU [XI, 33, 356]

Die MTU ist die maximal unterstützte, nichtfragmentierte Paketgröße eines Netzwerkpfades.

Multi-Touch [131]

Eingabegeräte welche die Bedienung mit Hilfe mehrerer Fingerberührungen ermöglichen, werden als Muli-Touch-fähig bezeichnet. Dadurch ist es beispielsweise möglich, dass mehrere Personen dasselbe Display zum Interagieren verwenden können.

Multipath TCP [XI, 327]

TCP-Verbindungen verwenden einen Pfad zur Datenübertragung, was bei gleichzeitiger Nutzung des UMTS- und Drahtlosnetzwerks zum Beispiel dazu führt, dass nur eine der beiden Verbindungen aktiv genutzt wird. Multipath TCP ermöglicht es jedoch, dass die bestehende TCP-Verbindung bei ausgelastetem WLAN auf das zu diesem Zeitpunkt effizientere UMTS-Netz ausgelagert wird und so zu einer besseren Performanz führt [Chirgwin, 2013].

Multiplexing [32, 201]

Multiplexing bedeutet in der Kommunikationstechnik, dass mehrere Datenströme zusammengefasst durch eine physikalische Verbindung übertragen werden. TCP zum Beispiel implementiert dieses Verfahren, da zwar mehrere Ports, welche in Datenströme münden, existieren, deren Daten jedoch über dieselbe Leitung gesendet und empfangen werden.

Nagle-Algorithmus [210, 265, 326]

Dadurch, dass jedes Paket einen festen 20 Byte IP- und 20 Byte TCP-Header besitzt, implementiert TCP den Nagle-Algorithmus, welcher kleine Pakete bis zu einer ausreichend großen Datenmenge zusammenfasst um die Headergröße zu rechtfertigen und den Datendurchsatz insbesondere bei geringen Bandbreiten zu optimieren. Typischerweise entspricht das anvisierte Ziel hier der MTU.

New I/O [XI, 40]

NIO beschreibt eine Sammlung von Java Schnittstellen, welche Funktionen für Eingabe-/Ausgabeoperationen anbietet. Diese neue Schnittstelle koexistiert neben der Standardschnittstelle in Java. Neu in dieser Sammlung ist u.a. die Möglichkeit für nichtblockierende Schreib- und Lesezugriffe im Netzwerk.

Non Player Character [XI, 88, 180, 185]

Non-Player-Charaktere, so genannte NPCs, sind Figuren oder Objekte, die vom Computer gesteuert werden und Gegenspieler oder Helfer, Gesprächspartner etc. eines echten, menschlichen Spielers darstellen.

Observer-Pattern [75, 273]

Das Observer-Pattern ist ein, aus der Softwareentwicklung bekanntes, Entwurfsmuster aus der Kategorie der Verhaltensmuster. Es wird eine Eins-zu-Viele-Abhängigkeit zwischen Objekten definiert. Ein Subjekt-Objekt kann mehrere Beobachter haben, welche bei Änderungen an ersterem benachrichtigt werden. In der Literatur wird häufig der Vergleich zu einem Zeitungsabo herangezogen, wobei der Herausgeber das Subjekt ist und die Abonnenten die Beobachter.

Oculus Rift [123]

Bei der *Oculus Rift* handelt es sich um ein Head-Mounted-Display, welches über zwei eingebaute Bildschirme dreidimensionale Inhalte direkt zu den Augen überträgt. Die Brille kann dabei durch eingebaute Sensoren ebenfalls den Bewegungen des Kopfes folgen und erlaubt somit die immersive Perzeption einer virtuellen Welt [Oculus VR, Inc., 2013].

Open Services Gateway initiative [XI, 47]

Innerhalb der OSGi Alliance wird ein offener Standard für die komponentenbasierte Software-Entwicklung auf Basis von Java-Technologien zu etablieren. Es handelt sich dabei um einen weltweiten Zusammenschluss namhafter Hersteller sowie Einzelpersonen und Institutionen.

Open Systems Interconnection Model [XI, 202]

Als Standard im Bereich der Netzwerkprotokolle gilt seit 1983 das Open Systems Interconnection Model (OSI). Dieses basiert auf sieben aufeinanderfolgenden Schichten, welche jeweils einen fest definierten Aufgabenbereich sowie Schnittstellen besitzen (siehe Abb. 11.2).

Pen-&-Paper-Rollenspiel [102, 197, 223, 228, 247, 257, 284, 328, 340, 352]

Das Genre der Pen-&-Paper-Rollenspiele lässt sich nicht ohne weiteres in einer kurzen Definition zusammenfassen, sodass auf Abschnitt 5.1.1.1 verwiesen wird. Für eine ausführlichere Charakterisierung der einzelnen Elemente eines solchen Spielgenres wird Abschnitt 2.1 empfohlen.

Post-Processing [306]

Unter Post-Processing (dt. Nachbearbeitung) wird das Aufbereiten einer Szene mittels verschiedener Techniken benannt. Namensgebend finden diese Operationen nach der gesamten Renderpipeline, jedoch vor der Anzeige statt.

Race Condition [42]

Bei einer Race Condition handelt es sich um ein Ereignis, welches auftritt, wenn ein Ergebnis mehrerer Operationen von deren Reihenfolge abhängt und diese nicht explizit garantiert werden kann.

Regelwerk [6, 14, 15, 52, 105, 106, 156, 247, 359]

Ein Regelwerk im Bereich der Pen-&-Paper-Rollenspiele ist eine Sammlung von Gesetzen und Weisungen, welcher sowohl Spieler, Spielleiter als auch Spieldesign unterworfen wird. Das Regelwerk zeigt

Grenzen auf, in denen sich Geschichte, Spieler und Spielleiter bewegen können und gilt gleichsam als „Naturgesetz“ des Spieluniversums.

Remote Procedure Call [XII, 83, 87, 92, 96, 109, 278]

RPC ist eine Technik zum Aufrufen von Funktionen aus anderen Adressräumen. Ein Aufruf kann sogar über das Netzwerk stattfinden. Damit muss bei einer Kommunikationsverbindung zwischen zwei Teilnehmern lediglich auf einer Seite eine entsprechende Programmlogik implementiert werden. Clients in einem Client-Server-Modell können Serverfunktionalität direkt über Remote Procedure Calls verwenden.

Renderpipeline [150, 357]

Die Renderpipeline, auch Grafikpipeline genannt, beschreibt den Prozess des Renderings. Hierbei handelt es sich um ein Modell, welches die einzelnen Schritte von den abstrakten Daten eines 3D-Objekts zu seiner fertigen, visuellen Repräsentation beschreibt. Dies finden sowohl software- als auch hardwareseitig statt und können zudem auf dem Hauptprozessor (CPU) sowie dem Grafikprozessor (GPU) stattfinden.

Retransmission Timeout [XII, 206, 326]

Der RTO übernimmt die Überprüfung, ob in einer bestimmten Zeit gesendeten Daten bestätigt werden. Sofern dies nicht der Fall ist, werden die Daten noch einmal gesendet. Siehe Abschnitt 11.3

Role-Playing Game [XII, 175, 209]

Wird als Bezeichnung eines Computerspielgenres verwendet, welches der digitalisierten Form klassischer Pen-&-Paper-Rollenspiele ähnelt. Prinzipiell ist der Begriff analog zu Massive Multiplayer Online Role Playing Game verwendbar, außer das es sich nicht um eine Online- oder Mehrspieler-Erfahrung handeln muss.

Round Trip Time [XII, 323]

Die Round Trip Time bezeichnet die Zeit die ein Paket vom Sender zum Empfänger und zurück benötigt. Diese zu berechnen ist bei TCP z.B. möglich, indem die Differenz der Sendezeit und die Zeit des Erhaltens der Bestätigung gemessen wird. Insofern lässt sich die RTT einfacher berechnen, da keine Synchronisierung der Zeiten auf Absender- und Empfängerseite benötigt wird. Schließlich wird lediglich auf einer der beiden Seiten die Zeit gemessen.

Sandbox-Spiel [152]

Sandbox-Spiele werden im Allgemeinen auch als Open-World-Spiele benannt. Es handelt sich um ein Genre von Spielen, bei denen Spieler nahezu unbegrenzte Freiheiten besitzen. Dies kann sich sowohl auf die Bewegungsfreiheit, als auch auf den Spielverlauf sowie die Spielwelt beziehen. In Minecraft kann zum Beispiel eine beliebige Spielwelt geschaffen und durchlaufen werden.

Spielleiter [X, 6, 16, 55, 56, 78, 156, 284, 340, 357, 359]

Der Spielleiter, auch „Game Master (GM)“ genannt, ist ein zentrales Element der Pen-&-Paper-Rollenspiele. Die Aufgabe des Spielleiters besteht darin, das Regelwerk zu kennen und anzuwenden, die Aktionen und Entscheidungen der Spieler auszuwerten und sowohl die Spielwelt als auch alle darin vorkommenden Charaktere zu kontrollieren, welche nicht von Spielern kontrolliert werden[486] [Schick, 1991, S. 10]. Der Spielleiter ist dabei nicht als Antagonist der Spieler zu verstehen, sondern als Schnittstelle und Feedbackmöglichkeit, mit dessen Beschreibungen und Ausführungen die Spieler die Spielwelt wahrnehmen

Splitscreen [10, 11, 88, 115, 261]

„Splitscreen“ steht als etablierter Begriff für die Aufteilung einer Wiedergabe- bzw. Projektionsfläche (oft ein Bildschirm) in mehrere Bereiche. Die Utilisierung eines Splitscreen-Verfahrens beim Computer- und Videospielen ist insbesondere bei Rennspielen anzutreffen. Prinzipbedingt können sich die Spieler bei diesem Genre innerhalb der Spielwelt sehr weit voneinander entfernen, sodass eine Bildaufteilung aus Praktikablitätsgründen sinnvoll ist, wie auch in Abbildung 26.1 zu sehen ist.

Abbildung 26.1.: Splitscreen-Ansicht von Split/Velocity
aus [TheKhoaNguyen, 2010].

Transmission Control Protocol [XII, 32, 210, 323, 325]

[486] So genannte NPCs

TCP ist ein verbindungsorientiertes Protokoll. Der TCP/IP Stack bietet eine automatische Fehlerkorrektur, Flusskontrolle sowie Übertragungszuverlässigkeit zwischen Ende-zu-Ende-Verbindungen. Es handelt sich hierbei um ein Protokoll auf der Transportschicht.

Trigger [92, 108, 273, 276, 277, 282, 335, 342]

Ein Trigger (dt. „Auslöser“) ist im Kontext dieses Projektes ein oft anzutreffendes Konzept der Computer- und Videospieltechnik. Trigger sind nichts anderes als bestimmte Areale oder auch Ereignisse, die von Spielern durch bloßes betreten oder generelles interagieren ein weiteres Ereignis anstoßen. Trigger dienen damit als Katalysator für weitere Berechnungen oder Situationsänderungen.

User Datagram Protocol [XII, 32, 210, 326]

UDP ist ein paketorientiertes und verbindungsloses Protokoll. Pakete können sowohl in falscher Reihenfolge als auch gar nicht bei deren Empfänger eintreffen. Der Vorteil von UDP ist eine erhöhte Geschwindigkeit in perfomanzkritischen Anwendungen, bedingt durch den verminderten Overhead.

View Frustum [306]

Das View Frustum bezeichnet die mathematische Abbildung einer 3D-Szene auf den Bildschirm. Sie bildet somit den Kegelstumpf, sofern der gesamte Abbildungsprozess als kompletter Kegel betrachtet wird.

Virtual Reality [123]

Mit dem Begriff der Virtual Reality ist eine vollständig synthetische Welt gemeint, welche die Partizipierenden umgibt und vollständig interaktionsfähig ist. Diese virtuelle Welt ist nicht notwendigerweise an eine Imitation der realen Welt gebunden; ebenfalls können physikalische Gesetze außer Kraft gesetzt werden, gleichfalls die Gesetze der Zeit und des Raumes im Allgemeinen [Milgram u. Kishino, 1994].

WAN [32]

Ein Wide Area Network ist - wie der Name schon sagt - ein Rechnernetz welches sich über einen sehr großen Bereich erstreckt. Im Gegensatz zu einem LAN, welches Geräte auf lokaler Eben miteinander verbindet, können sich Rechner innerhalb eines WANs in unterschiedlichen Ländern sowie Kontinenten befinden.

Whiteboxing [54]

Whiteboxing bezeichnet einen Begriff insbesondere aus dem Level-Design in der Computer-und Videospielproduktion. Dabei wird ein zunächst grobes, oftmals weisses (daher der Name) Modell der Spielwelt oder eines Objektes ohne Details modelliert, um die Wirkung oder Anwendbarkeit zu testen (Siehe Abbildung 26.2).

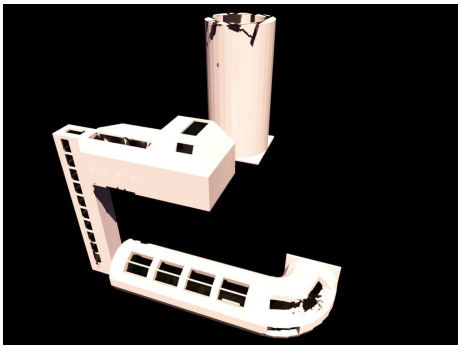


Abbildung 26.2.: Whiteboxing eines Computerspiel-Levels

Winsocket [41]

Winsocket bezeichnet eine Programmierschnittstelle, welche definiert wie Windows-Applikationen Netzwerkfunktionalitäten implementieren können.

Wireshark [322]

Wireshark ist ein freies Programm, mit welchem der Datenverkehr einer Netzwerkschnittstelle aufgezeichnet und im Nachhinein ausgewertet werden kann. Neben der einfachen Paketansicht kann nach speziellen Parametern gefiltert und somit Datensätze zu individuellen Zwecken aufbereitet werden. Weiterhin erstellt Wireshark auf Wunsch diverse Diagramme (Datendurchsatz, Datenfluss, Skalierung der TCP-Fenstergröße, …) auf Basis der gefilterten Daten. Hinzukommend zu den erfassten Informationen erweitert Wireshark die Daten mit zusätzlichen Angaben welche sich anhand der Datenkonstellation und des -verkehrs ergeben, was weitergehende Analysen ermöglicht.

Literaturverzeichnis

[Aarseth 2007] Aarseth, Espen: Allegories of space. In: Borries, Friedrich von (Hrsg.) ; Walz, Steffen P. (Hrsg.) ; Böttger, Matthias (Hrsg.): *Space Time Play*, Springer, 2007, S. 44–47 13.2

[Absolute Media 2011] Absolute Media: *The Internet of Things*. http://www.absolutemedia.co.uk/portfolio/item/23-intel. Version: 2011 2.5

[ACM 2013a] ACM, Inc.: *CSCW 2012*. http://cscw2012.org/program/accepted_workshops.php#W5. Version: Dezember 2013 139

[ACM 2013b] ACM, Inc.: *CSCW 2014 - Computer Supported Cooperative Work and Social Computing*. http://cscw.acm.org/. Version: Dezember 2013 139

[ADAC HEMS Academy GmbH 2014] ADAC HEMS Academy GmbH: *Ausbildung und Training - Flight Training beyond Limits-*. http://www.hems-academy.de/adac-full-flight-simulatoren-1-3/ausbildung-und-training/. Version: Januar 2014 6.1

[Aggarwal u. a. 2004] Aggarwal, Sudhir ; Banavar, Hemant ; Khandelwal, Amit ; Mukherjee, Sarit ; Rangarajan, Sampath: Accuracy in dead-reckoning based distributed multi-player games. In: *Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*. New York, NY, USA : ACM, 2004 (NetGames '04). – ISBN 1–58113–942–X, S. 161–165 288

[Alex 2014] Alex: *Client-Server Prinzip*. http://www.fachadmin.de/index.php/Client-Server_Prinzip. Version: jan 2014 20

[Alexander 2003] Alexander, Thor: *Massively multiplayer game development*. 1. Hingham, Mass : Charles River Media, 2003. – ISBN 978–1584502432 11.3.1, 286

[Amresh u. Okita 2010] Amresh, Ashish ; Okita, Alex: *Unreal game development*. Natick, Mass. : A K Peters, 2010. – ISBN 9781568814599 1568814593 2.2, 2.2.2.1, 26

[Andreas Floemer 2012] Andreas Floemer: *3.997 Smartphone-Modelle: So fragmentiert ist der Android-Markt*. http://t3n.de/news/3997-smartphone-modelle-388360/. Version: Mai 2012 2.5

[Andrew Davison 2005] Andrew Davison: *Killer Game Programming in Java*. 1. O'Reilly Media, 2005. – ISBN 978–0596007300 2.3.2, 2.3.2.1, 2.3.3, 2.7, 2.3.4, 4, 2.3.6.3, 11.1, 11.3.1

[Armitage 2001] Armitage, Grenville: Sensitivity of quake3 players to network latency. In: *Proceedings of the ACM SIGCOMM Internet MeasurementWorkshop*. Berkley, Calif, USA, nov 2001 11.3.1

[Arns 2007] Arns, Inke: *Interaktion, Partizipation, Vernetzung*. Version: Februar 2007. http://www.medienkunstnetz.de/themen/medienkunst_im_ueberblick/kommunikation/21/ 12.1

[Aronson 1997] Aronson, Jesse: *Dead Reckoning: Latency Hiding for Networked Games*. http://bit.ly/1dkzK8h. Version: September 1997 288

[Arthur C. Clarke 1968] Arthur C. Clarke: *2001: A Space Odyssey*. First Edition edition. Dutton Adult, 1968. – ISBN 1568653069 119

[Assmann 2007] Assmann, J.: *Das kulturelle Gedächtnis: Schrift, Erinnerung und politische Identität in frühen Hochkulturen*. Beck, 2007 (Beck'sche Reihe). http://books.google.de/books?id=MpGEsLAQPWEC. – ISBN 9783406568442 8

[Austermann 2012] Austermann, A.: *Algorithmisches Lernen mit der Methode der induktiven Inferenz*. Version: 2012. www2.cs.uni-paderborn.de/cs/ag-madh/vorl/Perlen/austermann.doc. – Seminar an der Universität Paderborn 9.3.5

[Austinat 2010] Austinat, Roland: *CryEngine 3: Inklusive 3D-Modus - Das kann Cryteks neueste Grafik-Engine*. http://www.pcgames.de/Crytek-Firma-20818/News/CryEngine-3-Inklusive-3D-Modus-Das-kann-Cryteks-neueste-Grafik-Engine-706535/. Version: März 2010 2.2.1

[Avedon u. Sutton-Smith 1971] Avedon, E.M. ; Sutton-Smith, B.: *The Study of Games*. New York : J. Wiley & Sons, 1971 http://books.google.de/books?id=-6u1AAAAIAAJ. – ISBN 9780471038399 1, 5

[Azuma u. a. 2001] Azuma, Ronald ; Baillot, Yohan ; Behringer, Reinhold ; Feiner, Steven ; Julier, Simon ; MacIntyre, Blair: Recent advances in augmented reality. In: *Computer Graphics and Applications, IEEE* 21 (2001), Nr. 6, S. 34–47 144

[Barra 2012] Barra, Hugo: *Google+ - 500 MILLION Today is a big day for Android... 500 million....* http://tinyurl.com/noqx2p8. Version: September 2012 2.5

[Bartlett u. Simpson 2005] Bartlett, K. ; Simpson, M.: Using Games As a Means for Collaboration. In: *Multimedia Modelling Conference, 2005. MMM 2005. Proceedings of the 11th International*, 2005. – ISSN 1550–5502, S. 346–351 12.1.1

[Batista u. a. 2001] Batista, Helder ; Costa, Vasco ; Pereira, João M.: Games of War and Peace: Large Scale Simulation Over the Internet. In: *Proceedings of the 7th International Conference on Virtual Systems and Multimedia*. Berkley, Calif, USA, 2001 (VSMM '01) 288

[Bearl 2012] Bearl, Dan: *How To Plan Your Kid's First D-and-D Adventure*. http://nerdywithchildren.com/how-to-plan-your-kids-first-dd-adventure. Version: Oktober 2012 1.3

[Becker u. Pant 2009] Becker, Arno ; Pant, Marcus: *Android: Grundlagen und Programmierung*. Heidelberg : Dpunkt-Verl., 2009. – ISBN 9783898645744 3898645746 2.5.1, 2.5.3

[Becker 2012] Becker, Peter: *Mobile Informationssysteme I*. http://tinyurl.com/q9x9g4c. Version: September 2012 2.3.5, 2.5.2

[Beigbeder u. a. 2004] Beigbeder, Tom ; Coughlan, Rory ; Lusher, Corey ; Plunkett, John ; Agu, Emmanuel ; Claypool, Mark: The Effects of Loss and Latency on User Performance in Unreal Tournament 2003. In: *Proceedings of 3rd ACM SIGCOMM Workshop on Network and System Support for Games*. New York, NY, USA : ACM, 2004 (NetGames '04). – ISBN 1–58113–942–X, 144–151 11.3.1

[Bell 2008] Bell, E.: *Theories of Performance*. SAGE Publications, 2008 http://books.google.de/books?id=Qb9wZsoJw3YC. – ISBN 9781412926386 185

[Benford u. a. 2006] Benford, Steve ; Crabtree, Andy ; Flintham, Martin ; Drozd, Adam ; Anastasi, Rob ; Paxton, Mark ; Tandavanitj, Nick ; Adams, Matt ; Row-Farr, Ju: Can you see me now? In: *ACM Trans. Comput.-Hum. Interact.* 13 (2006), März, Nr. 1, S. 100–133. http://dx.doi.org/10.1145/1143518.1143522. – DOI 10.1145/1143518.1143522. – ISSN 1073–0516 12.2

[Benford u. a. 2005] Benford, Steve ; Rowland, Duncan ; Flintham, Martin ; Drozd, Adam ; Hull, Richard ; Reid, Josephine ; Morrison, Jo ; Facer, Keri: Life on the edge: supporting collaboration in location-based experiences. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA : ACM, 2005 (CHI '05). – ISBN 1–58113–998–5, S. 721–730 12.2

[Bérard u. a. 2009] Bérard, François ; Ip, Jessica ; Benovoy, Mitchel ; El-Shimy, Dalia ; Blum, Jeffrey R. ; Cooperstock, Jeremy R.: Did M̈inority Reportg̈et it wrong? Superiority of the mouse over 3D input devices in a 3D placement task. In: *Human-Computer Interaction–INTERACT 2009*. Springer, 2009, S. 400–414 6.3

[Betable Tyler 2012] Betable Tyler: *A Comprehensive List of Game Developer Communities*. http://blog.betable.com/a-comprehensive-list-of-game-developer-communities/. Version: Februar 2012 2.3.2.1, 11.3.1

[Beyer 2010] Beyer, Susanne: Leben im Stand-by-Modus. In: *Der Spiegel* 29 (2010), Juli, S.56ff. http://www.spiegel.de/spiegel/print/d-71892495.html 5.1, 81

[Beznosyk u. a. 2012] Beznosyk, Anastasiia ; Quax, Peter ; Lamotte, Wim ; Coninx, Karin: The effect of closely-coupled interaction on player experience in casual games. In: *Proceedings of the 11th international conference on Entertainment Computing*. Berlin, Heidelberg : Springer-Verlag, 2012 (ICEC'12). – ISBN 978–3–642–33541–9, 243–255 6.4, 15.1, 337

[Bibliographisches Institut GmbH 2013] Bibliographisches Institut GmbH: *Interaktion*. http://www.duden.de/rechtschreibung/Interaktion. Version: Dezember 2013 6, 26

[Biermann 2013] Biermann, Ulrich: *Computerspiele als Kunst*. http://www.deutschlandfunk.de/computerspiele-als-kunst.807.de.html?dram:article_id=257798. Version: August 2013 110

[BITKOM Pressemitteilung 2012] BITKOM Pressemitteilung: *Jeder Dritte hat ein Smartphone (2012) - BITKOM*. http://www.bitkom.org/de/presse/74532_71854.aspx. Version: April 2012 3.1.4

[Björk u. a. 2001] Björk, Staffan ; Falk, Jennica ; Hansson, Rebecca ; Ljungstrand, Peter: Pirates! Using the physical world as a game board. In: *IN PROCEEDINGS OF INTERACT 2001*, 2001, S. 9–13 6.1, 6.2, 13

[Blackmoor 2005] Blackmoor, Brandon: *FAQ: Play by email (PBEM) role-playing games*. http://bit.ly/1gClc9j. Version: 2005 262

[Blow 2004] Blow, Jonathan: Game Development: Harder Than You Think. In: *Queue* 1 (2004), Februar, Nr. 10, 28–37. http://dx.doi.org/10.1145/971564.971590. – DOI 10.1145/971564.971590. – ISSN 1542–7730 2.2, 2.2, 2.1, 2.2, 2.2, 25

[Bogost 2004] Bogost, Ian: *Asynchronous Multiplay: Futures for Casual Multiplayer Experience*. http://www.bogost.com/writing/asynchronous_multiplay_futures.shtml. Version: 2004 11.1

[Boonsri Dickinson 2013] Boonsri Dickinson: *Internet Use Increases Worldwide, Connections Get Faster*. http://www.informationweek.com/infrastructure/management/internet-use-increases-worldwide-connect/240146912. Version: Januar 2013 2.3.1

[Bourg 2004] Bourg, Glenn Seemann David M.: *AI for Game Developers*. 1. O'Reilly Media, 2004. – ISBN 987–0–596–00555– 9, 9.3, 9.3.2, 227, 9.3, 9.3.4, 9.5, 9.3.5, 16.1

[Bourke 1988] Bourke, Paul: *Polygons and meshes*. Director iVEC at University of Western Australia - personal page. http://paulbourke.net/geometry/polygonmesh/. Version: Juli 1988 23.1, 23.4, 424

[Bowman 2010] Bowman, S.L.: *The Functions of Role-Playing Games: How Participants Create Community, Solve Problems and Explore Identity*. McFarland, Incorporated, Publishers, 2010 http://books.google.de/books?id=Pn3wR74wDCgC. – ISBN 9780786455553 6, 2, 8.3

[Braden 1989a] Braden, R.: *RFC-1122 - Requirements for Internet Hosts - Communication Layers*. http://tools.ietf.org/html/rfc1122. Version: Oktober 1989 2.3.2.1

[Braden 1989b] Braden, R.: *RFC 1122 Requirements for Internet Hosts - Communication Layers*. Internet Engineering Task Force, October 1989. http://tools.ietf.org/html/rfc1122 11.3

[BrainyQuote 2014] BrainyQuote: *Kenneth Starr Quotes at BrainyQuote.com*. http://www.brainyquote.com/quotes/quotes/k/kennethsta376738.html. Version: feb 2014 19

[Braun 2009] Braun, Herbert: *Amazon löscht gekaufte Kindle-eBooks*. http://www.heise.de/newsticker/meldung/Amazon-loescht-gekaufte-Kindle-eBooks-6887.html. Version: Juli 2009 115

[Brehm 2013] Brehm, Christian: *Deutscher Markt für Computer- und Videospiele: Prognose für 2013 bei 3,5 Prozent Umsatzwachstum - BIU e.V.* http://www.biu-online.de/de/presse/newsroom/newsroom-detail/datum/2013/08/13/deutscher-markt-fuer-computer-und-videospiele-prognose-fuer-2013-bei-35-prozent-umsatzwachstum.html. Version: August 2013 1, 5, B.9

[Briegleb 2013] Briegleb, Volker: *Ingress: Google öffnet seine virtuelle Welt für alle*. http://www.heise.de/newsticker/meldung/Ingress-Google-oeffnet-seine-virtuelle-Welt-fuer-alle-2037252.html. Version: Oktober 2013 12.2.1

[Bringezu 2001] Bringezu, Nancy: *Technik der Spieleprogrammierung: Game-Server und Mehrspieler-Szenarien*. Version: 2001. http://www.informatik.uni-ulm.de/ki/Edu/Seminare/Technik.der.Computerspiele/SS01/Bringezu_N_GameServerUndMultiPlayer.pdf. – Unterlagen der Universität Ulm 2.3.3

[Brinkbäumer u. Schulz 2010] Brinkbäumer, Klaus ; Schulz, Thomas: Der Philosoph des 21. Jahrhunderts. In: *Der Spiegel* 17 (2010), S. 67–78 6.2

[Brosh u. a. 2010] Brosh, Eli ; Baset, Salman A. ; Misra, Vishal ; Rubenstein, Dan ; Schulzrinne, Henning: The delay-friendliness of TCP for real-time traffic. In: *IEEE/ACM Trans. Netw.* 18 (2010), Oktober, Nr. 5, 1478–1491. http://dx.doi.org/10.1109/TNET.2010.2050780. – DOI 10.1109/TNET.2010.2050780. – ISSN 1063–6692 11.3.1, 290

[Brown 2007] Brown, Steve: *The Guide to Building Volume I – The Aurora Toolset Manual.* 1. The Builders Project, September 2007. http://nwvault.ign.com/View.php?view=other.detail&id=803 3.1.3

[Bruckman u. a. 1994] Bruckman, Amy ; Curtis, Pavel ; Figallo, Cliff ; Laurel, Brenda: Approaches to Managing Deviant Behaviour in Virtual Communities. In: *Proceedings of ACM CHI 1994*, ACM Press, 1994 (CHI '94), S. 183–184 6.4

[Bräuer 2005] Bräuer, Heidrun: *Interaktionsprozesse in Lerngruppen – Empirische Analyse des dynamischen Zusammenwirkens von Verhalten, sozialen Motiven und emotionalen Erleben.* Lüneburg, Niedersachsen, Universität Lüneburg, Diss., sep 2005 449

[Bulban u. Trotier 2012] Bulban, Franziska ; Trotier, Kilian: Game-Entwickler: Wir können auch Kunst. In: *Die Zeit* 50 (2012), Dezember, online. http://www.zeit.de/2012/50/Computerspiele-Entwickler-Kunst 110

[Burghardt 2009] Burghardt, F.: *Lichttechnik für Einsteiger: Die eigene Lightshow mit DMX professionell steuern.* Elektor-Verlag, 2009 http://books.google.de/books?id=0nQLQgAACAAJ. – ISBN 9783895761881 457, 26

[Busby u. a. 2010] Busby, Jason ; Parrish, Zak ; Wilson, Jeff: *Mastering Unreal Technology, Volume I: Introduction to Level Design with Unreal Engine 3.* Indianapolis : Sams, 2010. – ISBN 9780672329913 0672329913 9780672329920 0672329921 9780672330827 0672330822 22, 405, 406

[Cantwell u. a. 2012] Cantwell, Dale ; Broin, Daire O. ; Palmer, Ross ; Doyle, Greg: Motivating Elderly People to Exercise Using a Social Collaborative Exergame with Adaptive Difficulty. / Institute of Technology, Carlow, Ireland. 2012. – Forschungsbericht 1.3

[Capcom 2014] Capcom: *Resident Evil 5.* http://www.residentevil.com/5/. Version: feb 2014 336

[Casati u. a. 2007] Casati, Rebecca ; Matussek, Matthias ; Oehmke, Philipp ; Uslar, Moritz v.: Alles im Wunderland. In: *Der Spiegel - SPIEGEL SPECIAL* 3 (2007), Juni, 116-122. http://www.spiegel.de/spiegel/spiegelspecial/d-52058402.html 1, 190

[Casey 2006] Casey, Kevin P.: *Nintendo hopes Wii spells wiinner.* http://usatoday30.usatoday.com/tech/gaming/2006-08-14-nintendo-qa_x.htm. Version: August 2006 5.2.3

[Catalyst Game Labs 2013a] Catalyst Game Labs: *Shadowrun, Fifth Edition, Preview Omnibus - A collection of all six previews for Shadowrun, Fifth Edition.* http://www.shadowruntabletop.com/downloads/. Version: Juli 2013 14.3

[Catalyst Game Labs 2013b] Catalyst Game Labs: *Shadowrun, Fifth Edition Quick-Start Rules.* http://www.shadowruntabletop.com/downloads/. Version: Juli 2013 316, 14.2.1, 317

[CBS Interactive Inc. 2004] CBS Interactive Inc.: *Far Cry - Metacritic.com.* http://www.metacritic.com/game/pc/far-cry. Version: März 2004 2.2.1

[CerberusRCAF 2011] CerberusRCAF: *Minecraft - Enterprise-D Episode 1.1.* http://www.youtube.com/watch?v=TIh1fxlqIoY&feature=youtube_gdata_player. Version: April 2011 173

[Cerf u. Kahn 1974] Cerf, V. ; Kahn, R.E.: A Protocol for Packet Network Intercommunication. In: *Communications, IEEE Transactions on* 22 (1974), Nr. 5, S. 637–648. http://dx.doi.org/10.1109/TCOM.1974.1092259. – DOI 10.1109/TCOM.1974.1092259. – ISSN 0090–6778 2.3.2.1

[Chan 2013] Chan, Alan: *Unreal.* http://www.mobygames.com/game/unreal. Version: August 2013 2.2.2

[Chapanis u. a. 1951] Chapanis, A ; Frick, FC ; Garner, WR ; Gebhard, JW ; Grether, WF ; Henneman, RH ; Kappaif, WE ; Newman, EB ; Williams, AC: Human Engineering for an effective air navigation and traffic control system. In: *Washington, DC: National Research Council* xxii (1951), S. 84ff. 7.1.1

[Chen u. a. 2006] Chen, Kuan-Ta ; Huang, Chun-Ying ; Huang, Polly ; Lei, Chin-Laung: An empirical evaluation of TCP performance in online games. In: *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology.* New York, NY, USA : ACM, 2006 (ACE '06). – ISBN 1–59593–380–8 2.3.2.1, 6.4, 11.3.1, 284

[Cheung u. a. 2012] Cheung, Victor ; Chang, Y.-L. B. ; Scott, Stacey D.: Communication channels and awareness cues in collocated collaborative time-critical gaming. In: *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work.* New York, NY, USA : ACM, 2012 (CSCW '12). – ISBN 978–1–4503–1086–4, 569–578 6.4

[Chiang 2011] Chiang, Oliver: *The Master of Online Mayhem*. http://www.forbes.com/forbes/2011/0228/technology-gabe-newell-videogames-valve-online-mayhem.html. Version: September 2011 77

[China Daily 2011] China Daily: *Game Genres*. http://www.chinaculture.org/focus/focus/2011dmyx/2011-07/05/content_419145_8.htm. Version: Juli 2011 2.3.3, 11.1

[Chirgwin 2013] Chirgwin, Richard: *Multipath TCP: Siri's new toy isn't a game-changer*. http://bit.ly/1npJWjV. Version: oct 2013 26

[Chua u. a. 2010] Chua, C.K. ; Leong, K.F. ; Lim, C.S.: *Rapid Prototyping: Principles and Applications*. World Scientific, 2010 http://books.google.de/books?id=4OYcyiDUpsQC. – ISBN 9789812778987 3.1.2

[Chucklefish 2014] Chucklefish, Ltd.: *Starbound*. http://store.steampowered.com/app/211820/. Version: feb 2014 373

[Coffin 2012] Coffin, Ariane: *Hands On the New Nintendo Wii U at E3*. http://www.wired.com/geekmom/2012/06/nintendo-wii-u-e3/. Version: Dezember 2012 1.3

[Computer Bild 2011] Computer Bild: *Störungsfreien WLAN-Kanal wählen*. http://www.computerbild.de/artikel/cb-Ratgeber-Kurse-DSL-WLAN-Stoerungsfreien-WLAN-Kanal-waehlen-3059433.html. Version: feb 2011 24.2

[Cordes-Strehle 2013] Cordes-Strehle, Antraud: *Kunst statt Daddelspaß*. http://www.tagesschau.de/inland/notgamesfest100.html. Version: August 2013 110

[Cornett 2004] Cornett, Steve: The usability of massively multiplayer online roleplaying games: designing for new users. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA : ACM, 2004 (CHI '04). – ISBN 1–58113–702–8, 703–710 6.4

[Corporation 2014] Corporation, Valve: *Alien Swarm*. http://www.alienswarm.com/. Version: feb 2014 336

[Craig Villamor u. a. 2010] Craig Villamor ; Dan Willis ; Luke Wroblewski: *Touch Gesture REFERENCE GUIDE*. http://static.lukew.com/TouchGestureGuide.pdf. Version: April 2010 2.5

[Cross u. a. 2005] Cross, Brian ; Jouen, Nathanael ; Levine, Jason ; Szeto, Jon ; Taylor, Peter ; Toms, Malik ; weitere ; Rob Boyle, Jason Hardy Robyn King-Nitschke Michelle Lyons & Diane Piron-Gelman Mikael B. Mikael Brodu (Hrsg.): *Shadows of Asia*. FanPro LLC., 2005 14.2.1

[Crytek GmbH 2012] Crytek GmbH: *Layer Streaming - Doc 2. Sandbox Manual - CryENGINE 3 Free SDK*. http://freesdk.crydev.net/display/SDKDOC2/Layer+Streaming. Version: November 2012 2.2.1

[Crytek GmbH 2013a] Crytek GmbH: *Crytek - AI System*. http://mycryengine.com/index.php?conid=48. Version: August 2013 2.2.1

[Crytek GmbH 2013b] Crytek GmbH: *Crytek - Cry Engine Visuals*. http://mycryengine.com/index.php?conid=8. Version: August 2013 2.3

[Crytek GmbH 2013c] Crytek GmbH: *Crytek - Lincensing*. http://mycryengine.com/index.php?conid=3. Version: August 2013 2.2.1

[Crytek GmbH 2013d] Crytek GmbH: *Crytek - Sandbox*. http://mycryengine.com/index.php?conid=53. Version: August 2013 2.2.1

[Crytek GmbH 2013e] Crytek GmbH: *Crytek - Stereoscopic 3D*. http://mycryengine.com/index.php?conid=51. Version: August 2013 2.2.1

[Crytek UK Ltd. 2011] Crytek UK Ltd.: *Crytek Releases CryENGINE 3 SDK Free-of-Charge*. http://www.crydev.net/newspage.php?news=72255. Version: August 2011 2.2.1

[Cuaron 2013] Cuaron, Alfonso: *Gravity*. Oktober 2013 313

[Cusumano 2010] Cusumano, Michael A.: Platforms and services: Understanding the resurgence of Apple. In: *Communications of the ACM* 53 (2010), Nr. 10, S. 22–24 6.2

[Czerulla 2013] Czerulla, Hannes: *900 Millionen Android-Geräte aktiviert*. http://www.heise.de/newsticker/meldung/900-Millionen-Android-Geraete-aktiviert-1863837.html. Version: Mai 2013 2.5

[Dan Bornstein (Google) 2008] Dan Bornstein (Google): *2008 Google I/O Session Videos and Slides - Dalvik VM Internals*. Version: Mai 2008. http://bit.ly/13JUkK3 2.9, 2.5.1

[deGuzman u. Crawford 2013] deGuzman, Charlene ; Crawford, Miles: *I Forgot My Phone*. Youtube. http://youtu.be/OINa46HeWg8. Version: August 2013 13

[DeLoura 2002] DeLoura, Mark: *Spieleprogrammierung Gems 1*. Bd. 1. Mitp-Verlag, 2002. – 608 S. – ISBN 978–3826609237 9.3.2, 9.3.4, 9.3.4, 9.3.6, 18.1

[Dick u. a. 2005] Dick, Matthias ; Wellnitz, Oliver ; Wolf, Lars: Analysis of factors affecting players' performance and perception in multiplayer games. In: *Proceedings of 4th ACM SIGCOMM workshop*

on Network and system support for games. New York, NY, USA : ACM, 2005 (NetGames '05). – ISBN 1–59593–156–2, 1–7 11.3.1

[Diot u. Gautier 1999] Diot, Christophe ; Gautier, Laurent: A Distributed Architecture for Multiplayer Interactive Applications on the Internet. In: *IEEE Networks magazine* 13 (1999), S. 6–15 289

[Dorn u. Gottlob 2006] Dorn, Jürgen ; Gottlob, Georg: Künstliche Intelligenz. In: *Informatik-Handbuch*. Carl Hanser Verlag GmbH & Co. KG, 2006. – ISBN 978–3446401853, Kapitel 7, S. 975–996 9, 26

[Döring u. a. 2012] Döring, Tanja ; Sylvester, Axel ; Schmidt, Albrecht: Be-greifen Beyond the Surface - Eine Materialperspektive auf Tangible User Interfaces. In: Bernard Robben, Heidi S. (Hrsg.): *Be-greifbare Interaktionen: Der allgegenwärtige Computer: Touchscreens, Wearables, Tangibles und Ubiquitous Computing*. Transcript Verlag, 2012, Kapitel 5, S. 115–134 6.3

[Dringenberg u. a. 2002] Dringenberg, Rainer ; Gützlaff, Kathrin ; Hegele, Günter ; Henke, Ursula ; Rosenthal, Georg: INTERNET - vorgeführt und diskutiert. In: *Eindrücke vom Grenzverkehr zwischen realen und virtuellen Welten*. EFH RWL BOCHUM, 2002 (Schriftenreihe DENKEN und HANDELN (Bd. 46) der EFH RWL BOCHUM) 1, 5

[Ducheneaut u. Moore 2004] Ducheneaut, Nicolas ; Moore, Robert J.: The social side of gaming: a study of interaction patterns in a massively multiplayer online game. In: *Proceedings of the 2004 ACM conference on Computer supported cooperative work*. New York, NY, USA : ACM, 2004 (CSCW '04). – ISBN 1–58113–810–5, 360–369 11.1, 267

[Ducheneaut u. a. 2007] Ducheneaut, Nicolas ; Moore, Robert J. ; Nickell, Eric: Virtual "Third Places": A Case Study of Sociability in Massively Multiplayer Games. In: *Comput. Supported Coop. Work* 16 (2007), April, Nr. 1-2, 129–166. http://dx.doi.org/10.1007/s10606-007-9041-8. – DOI 10.1007/s10606–007–9041–8. – ISSN 0925–9724 11.1, 267

[DWTDmitri 2011] DWTDmitri: *English: This image illustrates the development history of Cry Engine game engines family*. http://commons.wikimedia.org/wiki/File:Cryengine_tree.svg. Version: Februar 2011 B.10

[Egenfeldt-Nielsen u. a. 2008] Egenfeldt-Nielsen, S. ; Smith, J.H. ; Tosca, S.P.: *Understanding Video Games: The Essential Introduction*. Taylor & Francis, 2008 http://books.google.de/books?id=Obxbis3vtK4C. – ISBN 9780203930748 10.3

[Epic Games, Inc. 2013a] Epic Games, Inc.: *Distributed Computing System | Unreal Swarm*. http://www.unrealengine.com/en/features/unreal_swarm/. Version: August 2013 2.2.2

[Epic Games, Inc. 2013b] Epic Games, Inc.: *UDN - Three - CameraTechnicalGuide*. http://udn.epicgames.com/Three/CameraTechnicalGuide.html. Version: November 2013 23, 421

[Epic Games, Inc. 2013c] Epic Games, Inc.: *UDN - Three - LevelStreamingHome*. http://udn.epicgames.com/Three/LevelStreamingHome.html. Version: August 2013 2.2.2, 3.1.2

[Epic Games, Inc. 2013d] Epic Games, Inc.: *UDN - Three - TcpLink*. http://udn.epicgames.com/Three/TcpLink.html. Version: August 2013 2.2.2.1, 3.1.5

[Epic Games, Inc. 2013e] Epic Games, Inc.: *UDN - Three - UnrealScriptReference*. http://udn.epicgames.com/Three/UnrealScriptReference.html. Version: August 2013 2.2.2.1, 3.1.8

[Epic Games, Inc. 2013f] Epic Games, Inc.: *Unreal Engine 4*. http://www.unrealengine.com/unreal_engine_4/. Version: August 2013 2.2.2

[Epic Games, Inc. 2013g] Epic Games, Inc.: *Unreal Engine Features*. http://www.unrealengine.com/en/features/. Version: August 2013 2.2.2

[Epic Games, Inc. 2013h] Epic Games, Inc.: *Unreal Engine Platforms*. http://www.unrealengine.com/platforms/. Version: August 2013 2.2.2

[Epic Games, Inc. 2013i] Epic Games, Inc.: *Unreal Networking Architecture*. http://udn.epicgames.com/Three/NetworkingOverview.html. Version: April 2013 2.3.3, 288

[Epic Games, Inc. 2013j] Epic Games, Inc.: *Unreal Tournament Series | Game Development Software*. http://www.unrealengine.com/en/showcase/unreal_tournament/. Version: September 2013 65

[Epic Games, Inc. 2014a] Epic Games, Inc.: *UDN - Three - Gameplay Debugging*. http://udn.epicgames.com/Three/NavigationMeshReference.html. Version: März 2014 23

[Epic Games, Inc. 2014b] Epic Games, Inc.: *UDN - Three - Navigation Mesh Reference*. http://udn.epicgames.com/Three/NavigationMeshReference.html. Version: März 2014 22.2, 416, 417

[Eric Schmidt 2013] Eric Schmidt: *Exclusive Interviews – Streamed Live – April 16th*. http://tinyurl.com/prxj9uu. Version: April 2013 2.5

[Erik Rodriguez 2012] Erik Rodriguez: *TCP vs. UDP*. http://www.skullbox.net/tcpudp.php. Version: November 2012 2.3.2.1, 2.3.2.2, 11.3

[Fairclough u. a. 2001] Fairclough, Chris ; Fagan, Michael ; Namee, Brian M. ; Cunningham, Pádraig: Research Directions for AI in Computer Games. In: *Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science*, 2001, S. 333–344 9, 9.3.2

[Fairservice 2001] Fairservice, Ddon: *Film Editing: History, Theory and Practice: Looking at the Invisible*. Manchester University Press, 2001 http://books.google.de/books?id=s-ng9Ez6KgEC. – ISBN 9780719057779 10

[Falk u. Davenport 2004] Falk, Jennica ; Davenport, Glorianna: Live role-playing games: Implications for pervasive gaming. In: *Entertainment Computing–ICEC 2004*. Springer, 2004, S. 127–138 12.2.1

[FanPro 2003] FanPro: *Dragons of the sixth world*. Chicago, IL : FASA Corp., 2003. – ISBN 3890646662 9783890646664 14.2.2

[Faust 2003] Faust, Martin: Mixed Reality Gaming Environment QuakeRunner. In: *Proceedings of the Second International Conference on Entertainment Computing*. Pittsburgh, PA, USA : Carnegie Mellon University, 2003 (ICEC '03), 1–4 12.1.3

[Faust u. ho Yoo 2006] Faust, Martin ; Yoo, Yong ho: Haptic Feedback in Pervasive games. In: *Third International Workshop on Pervasive gaming applications, PerGames*, 2006 15.1

[Fine 2002] Fine, G.A.: *Shared Fantasy: Role Playing Games as Social Worlds*. University of Chicago Press, 2002 http://books.google.de/books?id=rLlLbN0XuSEC. – ISBN 9780226249445 1, 2.1.3, 5, 68, 69, 73, 459

[Fishyyy 2011] Fishyyy: *Huge Minecraft Minas Tirith by Fishyyy*. http://www.youtube.com/watch?v=CjuN8sEfBxc&feature=youtube_gdata_player. Version: April 2011 173

[Flynt u. Caviness 2006] Flynt, John P. ; Caviness, Chris: *UnrealScript game programming all in one*. Boston, MA : Thomson Course Technology, 2006. – ISBN 1598631489 9781598631487 2.2.2.1

[Forlines u. a. 2007] Forlines, Clifton ; Wigdor, Daniel ; Shen, Chia ; Balakrishnan, Ravin: Direct-touch vs. mouse input for tabletop displays. In: *Proceedings of the SIGCHI conference on Human factors in computing systems* ACM, 2007, S. 647–656 5.2.1, 6.3

[Foster u. Mathews 2005] Foster, L.V. ; Mathews, P.: *Handbook to Life in the Ancient Maya World*. Oxford University Press, 2005 (World history). http://books.google.de/books?id=eHwLgl2dMg0C. – ISBN 9780195183634 5

[Freeman u. a. 2005] Freeman, Eric ; Freeman, Elisabeth ; Sierra, Kathy ; Bates, Bert: *Entwurfsmuster von Kopf bis Fuß*. 1. O'Reilly, 2005. – 672 S. – ISBN 978–3897214217 18.1, 384

[Friedl 2002] Friedl, Markus: *Online Game Interactivity Theory with Cdrom*. Rockland, MA, USA : Charles River Media, Inc., 2002. – ISBN 1584502150 6.4

[Fritsch u. a. 2005] Fritsch, Tobias ; Ritter, Hartmut ; Schiller, Jochen: The effect of latency and network limitations on MMORPGs: a field study of everquest2. In: *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*. New York, NY, USA : ACM, 2005 (NetGames '05). – ISBN 1–59593–156–2, 1–9 11.1, 17

[Galouye 2011] Galouye, Daniel F.: *Simulacron-3*. Phoenix Pick, 2011 http://books.google.de/books?id=1-QgzcsZPS8C. – ISBN 9781612420202 7

[GameDev.net 2004] GameDev.net: *Networking and Multiplayer FAQ*. http://www.gamedev.net/index.php?app=forums&module=forums&section=rules&f=15. Version: 2004 11.3.1

[Games Workshop Deutschland GmbH 2013] Games Workshop Deutschland GmbH: *Startseite Games Workshop*. http://www.games-workshop.com/gws/home.jsp. Version: August 2013 22

[Gameshouse gGmbH 2013] Gameshouse gGmbH: *Computerspielemuseum - Startseite*. https://www.computerspielemuseum.de/. Version: Dezember 2013 110

[Gaskill 2013] Gaskill, Andrew: *News - E3 2013 - Co-Optimus Game of Show Awards*. http://www.co-optimus.com/article/10390/e3-2013-co-optimus-game-of-show-awards.html. Version: Juni 2013 16.1

[Gautier u. Diot 1998] Gautier, Laurent ; Diot, Christophe: Design and Evaluation of MiMaze, a Multi-Player Game on the Internet. In: *IN PROCEEDINGS OF IEEE MULTIMEDIA SYSTEMS CONFERENCE*, 1998, S. 233–236 6.4

[Gearbox Software 2012] Gearbox Software: *Borderlands 2 Official Website*. http://www.borderlands2.com/us/#news. Version: September 2012 8.3

[Geddes 2008] Geddes, Ryan: *Beyond Gears of War 2*. http://uk.ign.com/articles/2008/09/30/beyond-gears-of-war-2?page=1. Version: September 2008 10.1

[Geelhaar 2006] Geelhaar, Jens: Interaktion mit verteilten digitalen Informationsräumen. In: *Knowledge Media Design - Theorie, Methodik, Praxis*. 2. Oldenbourg, 2006, S. 225–236 6

[Geelhaar 2012] Geelhaar, Jens: Zur Gestaltung be-greifbarer Mensch-Maschine-Schnittstellen. In: Bernard Robben, Heidi S. (Hrsg.): *Be-greifbare Interaktionen: Der allgegenwärtige Computer: Touchscreens, Wearables, Tangibles und Ubiquitous Computing*. Transcript Verlag, 2012, Kapitel 9, S. 191–214 6, 13.2, 13.1

[Gibson 1982] Gibson, J.J.: *Wahrnehmung und Umwelt: der ökologische Ansatz in der visuellen Wahrnehmung*. Urban & Schwarzenberg, 1982 (U-und-S-Psychologie). http://books.google.de/books?id=5sfDAAAACAAJ. – ISBN 9783541099313 7.3, 7.4

[Gilbert Dietrich 2011] Gilbert Dietrich: *Upload- und Download-Geschwindigkeiten im Vergleich*. http://irlandtipps.blogspot.de/2011/03/upload-und-download-geschwindigkeiten.html. Version: März 2011 2.3.4

[Gillespie 2013] Gillespie, Ed: *Sustainable storytelling is a powerful tool that communicates vision*. http://www.theguardian.com/sustainable-business/blog/sustainable-stories-powerful-tool-communicates-vision. Version: 2013 14

[Glenn Fiedler 2006] Glenn Fiedler: *Networked Physics - gafferongames.com*. http://gafferongames.com/game-physics/networked-physics/. Version: September 2006 38, 288

[Glenn Fiedler 2008] Glenn Fiedler: *UDP vs. TCP - gafferongames.com*. http://gafferongames.com/networking-for-game-programmers/udp-vs-tcp/. Version: Oktober 2008 2.3.2.2

[Glenn Fiedler 2010] Glenn Fiedler: *What every programmer needs to know about game networking - gafferongames.com*. http://gafferongames.com/networking-for-game-programmers/what-every-programmer-needs-to-know-about-game-networking/. Version: Januar 2010 2.3.3, 2.3.3, 11.1

[Global 2013] Global, IGI: *International Journal of Mobile Human Computer Interaction (IJMHCI)*. http://www.igi-global.com/journal/international-journal-mobile-human-computer/1126. Version: 2013 483

[Goad 2010] Goad, Libe: *Microsoft Kinect: What does it mean for social games?* http://blog.games.com/2010/06/14/microsoft-kinect-what-does-it-mean-for-social-games/. Version: Juni 2010 1.3

[Goh u. a. 2011] Goh, Wooi-Boon ; Fitriani ; Goh, Chun-Fan ; Tan, Jacquelyn ; Menon, Monica ; Cohen, Libby: Action role design and observations in a gestural interface-based collaborative game.

In: *CHI '11 Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA : ACM, 2011 (CHI EA '11). – ISBN 978–1–4503–0268–5, 763–772 15.1

[Goldt 2002] Goldt, M.: *Wenn man einen weissen Anzug anhat: ein Tagebuch-Buch*. Rowohlt Verlag GmbH, 2002 http://books.google.de/books?id=rz5cAAAAMAAJ. – ISBN 9783498024932 (document)

[Google Scholar 2014] Google Scholar: *Character Development and Storytelling for Games (Game Development Series) - Google Scholar, Zitationen*. http://scholar.google.de/scholar?cites=11490506615318944557&as_sdt=2005&sciodt=0,5&hl=en. Version: Februar 2014 177

[Goral u. a. 1984] Goral, Cindy M. ; Torrance, Kenneth E. ; Greenberg, Donald P. ; Battaile, Bennett: Modeling the interaction of light between diffuse surfaces. In: *ACM SIGGRAPH Computer Graphics* Bd. 18 ACM, 1984, S. 213–222 7.1

[Grassmann 2007] Grassmann, Hubert: *Lineare Algebra I*. http://www-irm.mathematik.hu-berlin.de/~hgrass/lat1.pdf. Version: September 2007 129

[Greif 2013] Greif, Björn: *Microsofts konzipiert Xbox One als Medienzentrale im Wohnzimmer*. http://www.zdnet.de/88155831/microsofts-konzipiert-xbox-one-als-medienzentrale-im-wohnzimmer/. Version: Mai 2013 101

[Griwodz u. Halvorsen 2006] Griwodz, Carsten ; Halvorsen, Pål: The fun of using TCP for an MMORPG. In: *Proceedings of the 2006 international workshop on Network and operating systems support for digital audio and video*. New York, NY, USA : ACM, 2006 (NOSSDAV '06). – ISBN 1–59593–285–2, 1:1–1:7 2.3.2.2

[Grønbæk u. a. 2007] Grønbæk, Kaj ; Iversen, Ole S. ; Kortbek, Karen J. ; Nielsen, Kaspar R. ; Aagaard, Louise: IGameFloor: A Platform for Co-located Collaborative Games. In: *Proceedings of the International Conference on Advances in Computer Entertainment Technology*. New York, NY, USA : ACM, 2007 (ACE '07). – ISBN 978–1–59593–640–0, 64–71 12.1, 12.1.2

[Grüter 2012] Grüter, Barbara: Interaktion in mobilen Spielwelten. In: Bernard Robben, Heidi S. (Hrsg.): *Be-greifbare Interaktionen: Der allgegenwärtige Computer: Touchscreens, Wearables, Tangibles und Ubiquitous Computing*. Transcript Verlag, 2012, Kapitel 16, S. 323–346 6.3

[Gruhn u. Thiel 2000] Gruhn, Volker ; Thiel, Andreas: *Komponentenmodelle: DCOM, JavaBeaans, Enterprise JavaBeans, CORBA ; [großartige Entscheidungshilfe ; kritischer Vergleich der wichtigsten Komponentenmodelle ; Beurteilung anhand praktischer Beispiele ; jedes Modell sorgfältig beschrieben]*. München [u.a.] : Addison-Wesley, 2000. – ISBN 382731724X 9783827317247 2.5.2

[Guild Wars 2006] Guild Wars: *Code=058*. http://bit.ly/13Ld1Co. Version: März 2006 2.3.2.2

[GuildWiki.de 2012] GuildWiki.de: *Emotes*. http://www.guildwiki.de/wiki/Emotes. Version: mar 2012 265

[Hageman 2005] Hageman, Rhonda K.: Senet, the Game of Passing. In: *JEFF STEPHENSON* 16 (2005), S. 3 5

[Hager 2005] Hager, Orm: *TCP Methoden*. http://doc-tcpip.org/Tcp/tcp-methoden.txt.html. Version: feb 2005 11.3

[Halper 2003] Halper, Nicolas: *Supportive Presentation for Computer Games*, Otto-von-Guericke-Universität Magdeburg, Diss., Oktober 2003 10

[Hao Chuah u. Lok 2012a] Hao Chuah, Joon ; Lok, Benjamin: Experiences in Using a Smartphone as a Virtual Reality Interaction Device / Virtual Experiences Research Group, University of Florida. 2012. – Forschungsbericht 3.1.4

[Hao Chuah u. Lok 2012b] Hao Chuah, Joon ; Lok, Benjamin: Experiences in Using a Smartphone as a Virtual Reality Interaction Device / Virtual Experiences Research Group, University of Florida. 2012. – Forschungsbericht 5.2.1

[Harebrained Schemes LLC 2013a] Harebrained Schemes LLC: *The Crunchy Bits*. http://golemarcana.com/the-crunchy-bits/. Version: November 2013 12.4

[Harebrained Schemes LLC 2013b] Harebrained Schemes LLC: *Game Overview*. http://golemarcana.com/game/. Version: November 2013 5.2.3, 5.2.3

[Harebrained Schemes LLC 2013c] Harebrained Schemes LLC: *Golem Arcana*. http://www.kickstarter.com/projects/1613260297/golem-arcana. Version: November 2013 5.2.3

[Harebrained Schemes LLC 2013d] Harebrained Schemes LLC: *Golem Arcana Grudge Match Highlights*. Youtube.com. http://www.youtube.com/watch?v=q_UCz6jbWJg&feature=youtube_gdata_player. Version: Oktober 2013 5.5

[Harold 2004] Harold, Elliotte R.: *Java network programming*. Sebastopol, Calif., Farnham : O'Reilly, 2004. – ISBN 0596007213 9780596007218 2.3.1, 2.3.2, 2.3.2, 2.6, 2.3.2.1, 39, 2.3.5, 2.4, 40, 41, 3.2.3.1, 3.2.7.1, 11.2, 277, 11.3.1

[Hartmann 2005] Hartmann, Erich: *Darstellende Geometrie für Architekten*. Fachbereich Mathematik Technische Universität Darmstadt, 2005 10.3

[Hartz 2013] Hartz, Wilko: *Basiswissen QR-Code*. http://qrcode.wilkohartz.de/. Version: mar 2013 485

[Hauteville 2011] Hauteville, Cédric: *Technical Game Design: UDK and CryEngine 3 Tool set comparison*. http://technicalgamedesign.blogspot.de/p/udkcryengine-3-tool-set-comparison.html. Version: Dezember 2011 3.1.2

[Havok.com Inc 2014] Havok.com Inc: *Havok Physics*. http://www.havok.com/products/physics. Version: Januar 2014 168

[Heiden u. a. 2001] Heiden, Wolfgang ; Frühling, Christian ; Deuer, Harald: Hypermedia Novel HYMN - A New Storytelling Paradigm. In: *proceedings casto1 - living in mixed realities*. Schloss Birlinghoven, Sankt Augustin, Germany, September 2001, S. 345–348 5.2.2.2, 8.2.3.1, 200, 8.5, 8.2.3.1, 201

[Heiden u. Ostovar 2006] Heiden, Wolfgang ; Ostovar, Arjang: Structuring Hypermedia Novels. Version: 2006. http://dx.doi.org/10.1007/11944577_10. In: Göbel, Stefan (Hrsg.) ; Malkewitz, Rainer (Hrsg.) ; Iurgel, Ido (Hrsg.): *Technologies for Interactive Digital Storytelling and Entertainment* Bd. 4326. Springer Berlin Heidelberg, 2006. – DOI 10.1007/11944577_10. – ISBN 978–3–540–49934–3, 98-103 8.2.3.1

[Heinemann 2006] Heinemann, Elisabeth: *Sprachlogische Aspekte rekonstruierten Denkens, Redens und Handelns*. DUV Deutscher Universitäts-Verlag, 2006. – 216 S. – ISBN 3–8350–9017–8 9.3.5

[Heisenberg 2012a] Heisenberg, Dr. G.: *Einführung in fortgeschrittene virtuelle Umgebungen*. 2012. – Vorlesungsunterlagen der Hochshcule Bonn-Rhein-Sieg 3.1.5

[Heisenberg 2012b] Heisenberg, Dr. G.: *Fortgeschrittene Virtuelle Umgebungen - Einführung*. 2012. – Vorlesungsunterlagen der Hochschule Bonn-Rhein-Sieg 6.2.1, 6.2.1, 6.2, 26

[Heisenberg 2012c] Heisenberg, Dr. G.: *Fortgeschrittene Virtuelle Umgebungen - Eingabegeräte*. 2012. – Vorlesungsunterlagen der Hochschule Bonn-Rhein-Sieg 6.2.1

[Heisenberg 2012d] Heisenberg, Dr. G.: *Fortgeschrittene Virtuelle Umgebungen - Szenengraph*. 2012. – Vorlesungsunterlagen der Hochschule Bonn-Rhein-Sieg 7.3

[Helman 1995] Helman, James: *Architecture and Performance of Entertainment Systems*, 5 1995. – Presentation, PDF 2.2

[von Henke 2009] Henke, Friedrich W.: *Einführung in die LISP-Programmierung*. http://www.informatik.uni-ulm.de/ki/Edu/Programmierkurse/Lisp/WS0910/. Version: okt 2009 359

[Herbst u. a. 2008] Herbst, Iris ; Braun, Anne-Kathrin ; McCall, Rod ; Broll, Wolfgang: TimeWarp: interactive time travel with a mobile mixed reality game. In: *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*. New York, NY, USA : ACM, 2008 (MobileHCI '08). – ISBN 978–1–59593–952–4, 235–244 12.2

[Herrlich u. a. 2012] Herrlich, Marc ; Walther-Franks, Benjamin ; Malaka, Rainer: Daten zum Anfassen: Be-greifen mit Interaktiven Bildschirmen. In: Bernard Robben, Heidi S. (Hrsg.): *Be-greifbare Interaktionen: Der allgegenwärtige Computer: Touchscreens, Wearables, Tangibles und Ubiquitous Computing*. Transcript Verlag, 2012, Kapitel 6, S. 135–154 6.3, 6.3, 6.4, 6.3

[Hewett u. a. 1992] Hewett, T. T. ; Baecker, R. ; Card, S. ; Carey, T. ; Gasen, J. ; Mantei, G. M. P. M. Perlman ; Strong, G. ; Verplank, W.: ACM SIGCHI Curricula for Human-Computer Interaction / ACM SIGCHI. Version: 1992. http://old.sigchi.org/cdg/cdg2.html. 1992. – Forschungsbericht 6

[Hüfner 2013] Hüfner, Daniel: *Microsoft will mit Kinect jede Oberfläche zum Touchscreen machen.* http://t3n.de/news/microsoft-kinect-touchscreen-487655/. Version: August 2013 1.3

[Hidenari Sawashima u. a. 1997] Hidenari Sawashima ; Yoshiaki Hori ; Hideki Sunahara: Characteristics of UDP Packet Loss: Effect of TCP Traffic. In: *INET '97*. Malaysia, Juni 1997 2.3.2.1, 3.1.5, 58

[Hirling 2014] Hirling, Hans: *Kommunikation-, Koordination, Kooperationsspiele.* http://www.praxis-jugendarbeit.de/spielesammlung/spiele-kommunikation.html. Version: feb 2014 24.3.1

[Hirsch 2009] Hirsch, Christian: *Unreal Engine 3 für Spieleentwickler kostenlos erhältlich.* http://www.heise.de/newsticker/meldung/Unreal-Engine-3-fuer-Spieleentwickler-kostenlos-erhaeltlich-853028.html. Version: November 2009 2.2.2

[Holler 2009] Holler, Manfred J.: *Springer-Lehrbuch: Einführung in die Spieltheorie*. 7. Springer, 2009. – 428 S. – ISBN 978–3540693727 229

[Hornecker u. Buur 2006] Hornecker, Eva ; Buur, Jacob: Getting a grip on tangible interaction: a framework on physical space and social interaction. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems* ACM, 2006, S. 437–446 7

[Huang u. Huang 2014] Huang, Cary ; Huang, Michael: *HTwins.net.* http://htwins.net/. Version: Januar 2014 7.4

[Huch 2010] Huch, Michael: *Test: AVM, Buffalo, D-Link, Linksys, Netgear & Co.: Die besten WLAN-Router*. http://www.computerbild.de/artikel/cb-Test-DSL-WLAN-Check_Lohnt_ein_neuer_Router-2562242.html. Version: apr 2010 24.2

[Huizinga 1955] Huizinga, J.: *Homo Ludens: A Study of the Play-element in Culture*. Beacon Press, 1955 (Beacon paperbacks ; 15 : Sociology). http://books.google.de/books?id=oZgA8UDf3_4C. – ISBN 9780807046814 72

[Humphries 2012] Humphries, Matthew: *Nintendo introduces Wii U as a social-focused console*. http://www.geek.com/games/nintendo-introduces-wii-u-as-a-social-focused-console-1493707/. Version: Juni 2012 1.3, 5.2.3

[Hurtienne u. a. 2008] Hurtienne, Jörn ; Israel, Johann H. ; Weber, Katharina: Cooking up real world business applications combining physicality, digitality, and image schemas. In: *Proceedings of the 2nd international conference on Tangible and embedded interaction* ACM, 2008, S. 239–246 7.1.1, 7.3

[IDC - Press Release 2013] IDC - Press Release: Apple Cedes Market Share in Smartphone Operating System Market as Android Surges and Windows Phone Gains, According to IDC / International Data Corporation (IDC). Version: August 2013. http://www.idc.com/getdoc.jsp?containerId=prUS24257413. Framingham Massachusetts, August 2013. – Forschungsbericht 2.5

[IDG Entertainment Media GmbH 2000] IDG Entertainment Media GmbH: *Test: Die Sims - Der Alltag wird zum (Sucht)-Spiel*. http://www.gamestar.de/spiele/die-sims/test/die_sims,32201,1330097.html. Version: März 2000 189

[IEEE 2012] IEEE: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. 3 Park Avenue New York, NY USA: Institute of Electrical and Electronics Engineers, Inc., mar 2012 24.2

[IMDb.com, Inc. 2014] IMDb.com, Inc.: *The Cheshire Cat (Character)*. http://www.imdb.com/character/ch0012248/quotes. Version: Februar 2014 (document)

[Immig u. Romer 1993] Immig, Michael ; Romer, Thomas: *Germany: sourcebook*. Chicago, IL : FASA Corp., 1993. – ISBN 1555601863 9781555601867 319

[Inc. 2013] Inc., Apple: *Apple - iOS - Siri*. https://www.apple.com/de/ios/siri/. Version: nov 2013 24.2

[Innocent u. Haines 2007] Innocent, Troy ; Haines, Stewart: Nonverbal Communication in Multiplayer Game Worlds. In: *Proceedings of the 4th Australasian Conference on Interactive Entertainment*. Melbourne, Australia, Australia : RMIT University, 2007 (IE '07). – ISBN 978–1–921166–87–7, 11:1–11:9 267

[Ishii 2008] Ishii, Hiroshi: The Tangible User Interface and Its Evolution. In: *Commun. ACM* 51 (2008), Juni, Nr. 6, 32–36. http://dx.doi.org/10.1145/1349026.1349034. – DOI 10.1145/1349026.1349034. – ISSN 0001–0782 12.3, 297, 12.3, 12.3

[Ishii u. Ullmer 1997] Ishii, Hiroshi ; Ullmer, Brygg: Tangible bits: towards seamless interfaces between people, bits and atoms. In: *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* ACM, 1997, S. 234–241 144

[Islas Sedano u. a. 2013] Islas Sedano, C. ; Carvalho, M.B. ; Secco, N. ; Longstreet, C.S.: Collaborative and cooperative games: Facts and assumptions. In: *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, 2013, S. 370–376 6.4, 131

[Israel u. a. 2012] Israel, Johann H. ; Hurtienne, Jörn ; Weber, Katharina: Be-greifbar oder Virtuell? - Der Designer entscheidet. In: Bernard Robben, Heidi S. (Hrsg.): *Be-greifbare Interaktionen: Der allgegenwärtige Computer: Touchscreens, Wearables, Tangibles und Ubiquitous Computing*. Transcript Verlag, 2012, Kapitel 2, S. 41–76 7, 7.1, 7.1.1, 7.2, 7.3, 7.4, 26

[Israel u. a. 2009] Israel, Johann H. ; Wiese, Eva ; Mateescu, Magdalena ; Zöllner, Christian ; Stark, Rainer: Investigating three-dimensional sketching for early conceptual design - Results from expert discussions and user studies. In: *Computers & Graphics* 33 (2009), Nr. 4, S. 462–473 6.3

[Jacob u. a. 2008] Jacob, Robert J. ; Girouard, Audrey ; Hirshfield, Leanne M. ; Horn, Michael S. ; Shaer, Orit ; Solovey, Erin T. ; Zigelbaum, Jamie: Reality-based interaction: a framework for post-WIMP interfaces. In: *Proceedings of the SIGCHI conference on Human factors in computing systems* ACM, 2008, S. 201–210 6.3

[Jacobs 2008] Jacobs, Scott: *Game Programming Gems 7*. Bd. 7. 1. Cengage Learning, 2008. – ISBN 987–1–58450–527–3 9.1, 222

[Jacobson 1988] Jacobson, V.: Congestion avoidance and control. In: *Symposium proceedings on Communications architectures and protocols*. New York, NY, USA : ACM, 1988 (SIGCOMM '88). – ISBN 0–89791–279–9, 314–329 2.3.2.1, 11.3

[Janssen 2013a] Janssen, Cory: *Artificial Intelligence (AI)*. http://www.techopedia.com/definition/190/artificial-intelligence-ai. Version: 2013 217

[Janssen 2013b] Janssen, Jan-Keno: *Oculus arbeitet an Virtual-Reality-Brille für Android-Geräte.* http://www.heise.de/newsticker/meldung/Oculus-arbeitet-an-Virtual-Reality-Brille-fuer-Android-Geraete-2036330.html. Version: Oktober 2013 6.2

[Jay Alabaster 2013] Jay Alabaster: *Android founder: We aimed to make a camera OS.* http://www.pcworld.com/article/2034723/android-founder-we-aimed-to-make-a-camera-os.html. Version: April 2013 2.5

[Jay Cotton u. Scott Coleman 1995] Jay Cotton ; Scott Coleman: *DOOM on the Internet via TCP/IP.* http://www.faqs.org/faqs/games/doom/howto-tcp/#ixzz0daOnpBYn. Version: Januar 1995 2.3.3

[Jeffrey Van Camp 2010] Jeffrey Van Camp: *Check out this stunning Facebook world map.* http://www.digitaltrends.com/computing/check-out-this-stunning-facebook-world-map/. Version: Dezember 2010 2.3.1

[Jensen u. a. 2002] Jensen, Carlos ; Davis, John ; Farnham, Shelly: Finding others online: reputation systems for social online spaces. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* New York, NY, USA : ACM, 2002 (CHI '02). – ISBN 1–58113–453–3, 447–454 6.4

[Jerz 2007] Jerz, Dennis G.: Somewhere Nearby is Colossal Cave: Examining Will Crowther's Original Adventure in Code and in Kentucky / Seton Hill University. Version: 2007. http://www.digitalhumanities.org/dhq/vol/001/2/000009/000009.html. 2007 (2). – Forschungsbericht 49

[Jüngst u. Link 2012] Jüngst, Johannes ; Link, Jasmin: Tangible Workplaces: Be-greifbare Interaktion in der Bürokommunikation. In: Bernard Robben, Heidi S. (Hrsg.): *Be-greifbare Interaktionen: Der allgegenwärtige Computer: Touchscreens, Wearables, Tangibles und Ubiquitous Computing.* Transcript Verlag, 2012, Kapitel 10, S. 215–234 13.2

[Jobst 2011] Jobst, Fritz: *Programmieren in Java.* München : Hanser Verlag, 2011. – ISBN 9783446417717 3446417710 2.4, 26

[Jon Watte 2008] Jon Watte: *Network bandwidth mathematics; peer-to-peer versus client/server.* http://www.enchantedage.com/node/20. Version: Mai 2008 2.3.4

[Jones u. Marsden 2005] Jones, Matt ; Marsden, Gary: *Mobile Interaction Design.* 1. John Wiley & Sons, 2005. – 398 S. – ISBN 978–0470090893 482

[Joyce 1987] Joyce, Michael: *afternoon, a story.* CD for Windows, 1987 8.1

[K 2008] K, Tobias: *Illustration of the Uncanny Valley with German text.* http://commons.wikimedia.org/wiki/File:Mori_Uncanny_Valley_de.svg. Version: Februar 2008 7.2

[Kajiya 1986] Kajiya, James T.: The Rendering Equation / California Institute of Technology. 1986. – PAPER, PDF 2.2

[Karmali 2013] Karmali, Luke: *Iwata Admits Lack of Software Hinders Wii U.* http://www.ign.com/articles/2013/08/12/iwata-admits-lack-of-software-hinders-wii-u. Version: August 2013 1.3

[Kastens 2011] Kastens, Uwe: *8. Logische Programmierung.* http://ag-kastens.uni-paderborn.de/lehre/material/gps/folien/gdp801-817.4.pdf. Version: 2011 358

[Kaul 2012] Kaul, Manfred: *Kollaboration und Kooperation.* https://kaul.inf.h-brs.de/wordpress/2012/12/kollaboration-und-kooperation/. Version: dez 2012 6.4, 26

[Keynes 2014] Keynes, John M.: *Wikiquote - John Maynard Keynes.* http://en.wikiquote.org. http://en.wikiquote.org/wiki/John_Maynard_Keynes. Version: Februar 2014 25

[Kim-Mai Cutler 2012] Kim-Mai Cutler: *This Is What Developing For Android Looks Like | TechCrunch.* http://techcrunch.com/2012/05/11/this-is-what-developing-for-android-looks-like/. Version: Mai 2012 2.5

[Kipp 2002] Kipp, Alexander: *Proseminar Rund ums Internet.* Version: Dezember 2002. http://goethe.ira.uka.de/seminare/internet/tcp+udp/. – Seminar der Universität Karlsruhe 2.3.2, 2.3.2.1, 31, 2.3.2.2, 11.2, 280

[Kleinert 2014] Kleinert, Ulrike: *Dream Addiction - Fantasy Art by Ulrike Kleinert.* http://dreamaddiction.de/. Version: feb 2014 (document), 5.4

[Klinge 2008] Klinge, Heiko: *Gründe für die KI-Stagnation in der Branche.* http://www.gamestar.de/specials/reports/1477118/kuenstliche_dummheit.html. Version: jan 2008 16, 19.1

[Knutsson u. a. 2004] Knutsson, Björn ; Lu, Honghui ; Xu, Wei ; Hopkins, Bryan: Peer-to-Peer Support for Massively Multiplayer Games. In: *Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies* Bd. 1. Hong Kong, China, mar 2004 (INFOCOM '04) 377

[Koivisto 2003] Koivisto, Elina M. I.: Supporting Communities in Massively Multiplayer Online Role-Playing Games by Game Design. In: *Proc. Level Up*, 2003 6.4

[Kolasinski 1995] Kolasinski, Eugenia M.: Simulator Sickness in Virtual Environments. / DTIC Document. 1995. – Forschungsbericht 123

[Konstantinidis u. Kopp 2013] Konstantinidis, Antony ; Kopp, Nicolas: *Induktive logische Programmierung mit Progol*. http://portfolio.thinklabs.de/Projekte/ILP-Mit-Progol. Version: feb 2013 235

[Kopp 2011] Kopp, Nicolas: *Ein System zur intuitiven Navigation in stereoskopischen interaktiven Umgebungen*, Hochschule Bonn-Rhein-Sieg, Bachelor Thesis, September 2011 22.2

[Korhonen u. Koivisto 2007] Korhonen, Hannu ; Koivisto, Elina M. I.: Playability heuristics for mobile multi-player games. In: *Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts*. New York, NY, USA : ACM, 2007 (DIMEA '07). – ISBN 978–1–59593–708–7, 28–35 6.4, 15.1

[Krishnamoorthy Dharmalingam 2009] Krishnamoorthy Dharmalingam: *Reading Bytes From Any InputStream*. http://www.javafundu.com/2009/03/reading-bytes-from-any-inputstream.html. Version: 2009 3.2.7.1

[Kuri 2013a] Kuri, Jürgen: *Google Glass: Erweitertes Testprogramm und Hardwaretausch für die Datenbrille*. http://www.heise.de/newsticker/meldung/Google-Glass-Erweitertes-Testprogramm-und-Hardwaretausch-fuer-die-Datenbrille-2035256.html. Version: Oktober 2013 6.2

[Kuri 2013b] Kuri, Jürgen: *GTA V: Rekordverkäufe am ersten Tag*. http://www.heise.de/newsticker/meldung/GTA-V-Rekordverkaeufe-am-ersten-Tag-1960962.html. Version: September 2013 74

[Kurosawa 1950] Kurosawa, Akira: *Rashomon*. Version: Dezember 1950. http://www.imdb.com/title/tt0042876/ 62, 206

[Kyle Kingsbury 2012] Kyle Kingsbury: *Know a Delay: Nagle's Algorithm and You*. http://boundary.com/blog/2012/05/02/know-a-delay-nagles-algorithm-and-you/. Version: Mai 2012 2.3.2.1, 11.3

[Labutov u. a. 2013] Labutov, Igor ; Yosinski, Jason ; Lipson, Hod: *AI vs. AI: Two chatbots talking to each other | Cornell Creative Machines Lab*. http://creativemachines.cornell.edu/AI-vs-AI. Version: Dezember 2013 109

[Lang 2012] Lang, Hans W.: *Theoretische Informatik: Komplexität deterministischer endlicher Automaten.* http://www.inf.fh-flensburg.de/lang/theor/automat-det.htm. Version: dez 2012 9.3

[Larsson u. Skårman 2002] Larsson, Johan ; Skårman, Johan: *Collaborative Games – Makes People Talk.* Göteborg, Sweden, IT University of Göteborg, Diplomarbeit, 2002 6.4

[Lau 2011] Lau, Nachi: *LucasArts: Knowledge-Based Behavior System A Decision Tree Finite State Machine Hybrid.* http://www.seas.upenn.edu/~cis568/presentations/knowledge-based-behavior-system.pdf. Version: dez 2011 9.3.5, 16.1

[Lüders u. Winkler 2006] Lüders, Christoph ; Winkler, Martin: Pingpong. In: *c't* 23 (2006), Nr. 23, S. 199 24.1.2

[Lea 1999] Lea, Douglas: *Concurrent programming in Java: design principles and patterns.* Reading, Mass. [u.a. : Addison-Wesley, 1999. – ISBN 0201310090 9780201310092 2.4

[Lecky-Thompson 2008] Lecky-Thompson, Guy W.: *Fundamentals of Network Game Development.* 1. Cengage Learning, 2008. – 336 S. – ISBN 978–1584505570 6.4, 11.1, 11.1.1, 269, 11.3.1, 15, 378, 24.2

[Leininger u. Amann 2007] Leininger, Wolfgang ; Amann, Erwin: *Einführung in die Spieltheorie.* http://bit.ly/1eNHfrt. Version: 2007 228

[Lengel u. Daft 1984] Lengel, Robert H. ; Daft, Richard L.: Information Richness: A New Approach to Managerial Behavior and Organization Design. In: *Research in Organizational Behavior* Bd. 6, 1984, S. 191–233 136

[LEO Dictionary Team 2013] LEO Dictionary Team: *pawn - Englisch - Deutsch Wörterbuch - leo.org.* http://dict.leo.org/#/search=pawn&searchLoc=0&resultOrder=basic&multiwordShowSingle=on. Version: November 2013 419

[Leschnikowski 2012] Leschnikowski, Dennis: *Steam-News: Sechs Millionen User gleichzeitig online: Valve verkündet beeindruckenden Rekord.* http://www.onlinewelten.com/games/steam/news/sechs-millionen-user-gleichzeitig-online-valve-verkuendet-beeindruckenden-rekord-117741/. Version: November 2012 6

[Lienemann u. Larisch 2010] Lienemann, Gerhard ; Larisch, Dirk: *TCP/IP - Grundlagen und Praxis: Protokolle, Routing, Dienste, Sicherheit.* Heise Medien, 2010 11.3

[Lin u. a. 2002] Lin, Yow-Jian ; Guo, Katherine ; Paul, Sanjoy: Sync-MS: Synchronized Messaging Service for Real-Time Multi-Player Distributed Games. In: *Proceedings of the 10th IEEE International Conference on Network Protocols*. Washington, DC, USA : IEEE Computer Society, 2002 (ICNP '02). – ISBN 0–7695–1856–7, 155–164 289

[Lindner 2012] Lindner, Manfred: *Data Communication Lectures of Manfred Lindner*. https://www.ict.tuwien.ac.at/lva/384.081/infobase/module.html. Version: September 2012. – Publikation der TU Wien 2.3.2

[Lion13 2011] Lion13: *US-Smartphone-Markt: Android vergrößert den Vorsprung gegenüber der Konkurrenz - Android-Hilfe.de*. Version: Juni 2011. http://www.android-hilfe.de/android-news/92860-us-smartphone-markt-android-vergroessert-den-vorsprung-gegenueber-der-konkurrenz.html 2.8

[Lippert 2011] Lippert, Robert: *Facebook küsst Google+: Circles als Hack*. http://www.heise.de/developer/meldung/Facebook-kuesst-Google-Circles-als-Hack-1272278.html. Version: Juni 2011 3, 80

[Lumsden 2011] Lumsden, Joanna: *Human-Computer Interaction and Innovation in Handheld, Mobile and Wearable Technologies*. 1. IGI Global, 2011. – 406 S. http://dx.doi.org/10.4018/978-1-60960-499-8. http://dx.doi.org/10.4018/978-1-60960-499-8 26

[Lutz u. a. 2012] Lutz, Rainer ; Schäfer, Sascha ; Diehl, Stephan: Using mobile devices for collaborative requirements engineering. In: *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering*. New York, NY, USA : ACM, 2012 (ASE 2012). – ISBN 978–1–4503–1204–2, 298–301 12.1.4, 12.2, 12.1.4.1

[MacDorman 2005] MacDorman, Karl F.: Androids as an experimental apparatus: Why is there an uncanny valley and can we exploit it. In: *CogSci-2005 workshop: toward social mechanisms of android science*, 2005, S. 106–118 7.2, 7.1, 7.1

[Magerkurth u. a. 2005] Magerkurth, Carsten ; Cheok, Adrian D. ; Mandryk, Regan L. ; Nilsen, Trond: Pervasive games: bringing computer entertainment back to the real world. In: *Comput. Entertain.* 3 (2005), Juli, Nr. 3, 4–4. http://dx.doi.org/10.1145/1077246.1077257. – DOI 10.1145/1077246.1077257. – ISSN 1544–3574 12.2, C.1, C.1, C.2

[Magerkurth u. a. 2004] Magerkurth, Carsten ; Engelke, Timo ; Memisoglu, Maral: Augmenting the virtual domain with physical and social elements: towards a paradigm shift in computer entertainment technology. In: *Comput. Entertain.* 2 (2004), Oktober, Nr. 4, 12–12. http:

//dx.doi.org/10.1145/1037851.1037870. – DOI 10.1145/1037851.1037870. – ISSN 1544–3574 6.4, 12.3, 12.3.2, 12.5, 12.3.2.1, 13

[Marchiafava 2012] Marchiafava, Jeff: *How Big Is A Minecraft World?* http://www.gameinformer.com/b/news/archive/2012/02/09/how-big-is-a-minecraft-world.aspx. Version: Februar 2012 7.4

[Mark Terrano u. Paul Bettner 2001a] Mark Terrano ; Paul Bettner: *1500 Archers on a 28.8: Network Programming in Age of Empires and Beyond.* http://www.gamasutra.com/view/feature/3094/1500_archers_on_a_288_network_.php. Version: März 2001 36

[Mark Terrano u. Paul Bettner 2001b] Mark Terrano ; Paul Bettner: *1500 Archers on a 28.8: Network Programming in Age of Empires and Beyond.* http://bit.ly/1emjOHY. Version: März 2001 258, 11.3.1

[Massachusetts Institute of Technology 2000] Massachusetts Institute of Technology: *Inventor of the Week Archive - Internet Protocols (TCP/IP).* http://web.mit.edu/invent/iow/cerf.html. Version: Juli 2000 2.3.2.1

[Matejka u. a. 2009] Matejka, Justin ; Grossman, Tovi ; Lo, Jessica ; Fitzmaurice, George: The design and evaluation of multi-finger mouse emulation techniques. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* ACM, 2009, S. 1073–1082 6.3

[Mathieu u. a. 2013] Mathieu ; Cry-Adam ; Grytek GmbH: *Home - Doc 5. Scripting Manual - CryENGINE 3 Free SDK.* http://freesdk.crydev.net/display/SDKDOC5/Home. Version: Februar 2013 2.2.1

[Matschijewsky 2009] Matschijewsky, Daniel: *Test: Dragon Age: Origins - Review: Das beste Rollenspiel des Jahres.* http://www.gamestar.de/spiele/dragon-age-origins/test/dragon_age_origins,43300,2309915.html. Version: November 2009 366

[McCarthy u. a. 2006] McCarthy, John ; Minsky, Marvin L. ; Rochester, Nathaniel ; Shannon, Claude E.: A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. In: *AI MAGAZINE* Bd. 27, 2006 216

[Michael Abrash 2012] Michael Abrash: *Latency – the sine qua non of AR and VR.* http://blogs.valvesoftware.com/abrash/latency-the-sine-qua-non-of-ar-and-vr/. Version: Dezember 2012 3.1.5, 24.1.1

[Microsoft 2006] Microsoft: *Description of Windows 2000 and Windows Server 2003 TCP Features.* http://bit.ly/Jsl0v0. Version: oct 2006 11.3

[Mikito Takada 2011] Mikito Takada: *Understanding the node.js event loop.* http://blog.mixu.net/2011/02/01/understanding-the-node-js-event-loop/. Version: Februar 2011 2.3.6

[Milgram u. Kishino 1994] Milgram, Paul ; Kishino, Fumio: A taxonomy of mixed reality visual displays. In: *IEICE TRANSACTIONS on Information and Systems* 77 (1994), Nr. 12, S. 1321–1329 7, 7.1, 7, 7.1, 144, 7.1, 7.1, 153, 26

[Millington u. Funge 2009] Millington, Ian ; Funge, John: *Artificial Intelligence For Games.* 2. Morgan Kaufmann, 2009. – 888 S. – ISBN 978–0123747310 9.1, 9.3.2, 9.3.3, 9.3.3, 9.4, 9.3.5, 9.3.5, 9.3.6, 9.3.7, 16.1

[Mitchell 2014] Mitchell, Bradley: *Online Games: Using computer networks to play games online.* http://compnetworking.about.com/od/homenetworkuses/a/network-online-games.htm. Version: jan 2014 11, 17

[MobyGames 2013] MobyGames: *D&D / AD&D licensees.* http://www.mobygames.com/game-group/dd-add-licensees. Version: September 2013 2.1.3, 3.1.3

[MobyGames.com 2013a] MobyGames.com: *MobyGames: Adventure (aka Collosal Cave).* http://www.mobygames.com/game/windows/adventure_. Version: November 2013 199

[MobyGames.com 2013b] MobyGames.com: *MobyGames: Borderlands 2.* http://www.mobygames.com/game/borderlands-2. Version: November 2013 213

[MobyGames.com 2013c] MobyGames.com: *MobyGames: Diablo.* http://www.mobygames.com/game/diablo. Version: November 2013 256

[MobyGames.com 2013d] MobyGames.com: *MobyGames: Dragon Age: Origins.* http://www.mobygames.com/game/dragon-age-origins. Version: November 2013 195, 257, 365, 366

[MobyGames.com 2013e] MobyGames.com: *MobyGames: Duke-Nukem 3D.* http://www.mobygames.com/game/dos/duke-nukem-3d. Version: November 2013 245

[MobyGames.com 2013f] MobyGames.com: *MobyGames: Game Browser: 1st-Person Perspective.* http://www.mobygames.com/browse/games/x,7/list-games/. Version: November 2013 247

[MobyGames.com 2013g] MobyGames.com: *MobyGames: Game Browser: 3rd-Person Perspective.* http://www.mobygames.com/browse/games/x,16/list-games/. Version: November 2013 242

[MobyGames.com 2013h] MobyGames.com: *MobyGames: Game Browser: Isometric.* http://www.mobygames.com/browse/games/x,25/list-games/. Version: November 2013 10.3

[MobyGames.com 2013i] MobyGames.com: *MobyGames: Game Browser: Top-Down.* http://www.mobygames.com/browse/games/x,13/list-games/. Version: November 2013 248

[MobyGames.com 2013j] MobyGames.com: *MobyGames: Game Browser: Top-Down 1993.* http://www.mobygames.com/browse/games/1993/x,13/list-games/. Version: November 2013 371

[MobyGames.com 2013k] MobyGames.com: *MobyGames: Game Browser: Top-Down 2003.* http://www.mobygames.com/browse/games/2003/x,13/list-games/. Version: November 2013 371

[MobyGames.com 2013l] MobyGames.com: *MobyGames: Game Browser: Top-Down 2013.* http://www.mobygames.com/browse/games/2013/x,13/list-games/. Version: November 2013 371

[MobyGames.com 2013m] MobyGames.com: *MobyGames: Game Browser: Top-Down Action.* http://www.mobygames.com/browse/games/x,17/action/list-games/. Version: November 2013 252

[MobyGames.com 2013n] MobyGames.com: *MobyGames: Game Browser: Top-Down RPG.* http://www.mobygames.com/browse/games/x,17/role-playing-rpg/list-games/. Version: November 2013 252

[MobyGames.com 2013o] MobyGames.com: *MobyGames: Gears of War.* http://www.mobygames.com/game/gears-of-war. Version: November 2013 244

[MobyGames.com 2013p] MobyGames.com: *MobyGames: Grand Theft Auto.* http://www.mobygames.com/game/windows/grand-theft-auto. Version: November 2013 250, 251

[MobyGames.com 2013q] MobyGames.com: *MobyGames: Half-Life.* http://www.mobygames.com/game/half-life. Version: November 2013 246

[MobyGames.com 2013r] MobyGames.com: *MobyGames: Mega Man.* http://www.mobygames.com/game/mega-man-. Version: November 2013 254

[MobyGames.com 2013s] MobyGames.com: *MobyGames: S.T.A.L.K.E.R.: Shadow of Chernobyl*. http://www.mobygames.com/game/windows/stalker-shadow-of-chernobyl. Version: November 2013 196

[MobyGames.com 2013t] MobyGames.com: *MobyGames: Stats*. http://www.mobygames.com/moby_stats. Version: November 2013 242

[MobyGames.com 2013u] MobyGames.com: *MobyGames: Super Mario Bros.* http://www.mobygames.com/game/super-mario-bros. Version: November 2013 253

[MobyGames.com 2013v] MobyGames.com: *MobyGames: The Legend of Zelda*. http://www.mobygames.com/game/legend-of-zelda. Version: November 2013 249

[MobyGames.com 2013w] MobyGames.com: *MobyGames: The Stanley Parable*. http://www.mobygames.com/game/windows/stanley-parable. Version: November 2013 304

[MobyGames.com 2013x] MobyGames.com: *MobyGames: Tomb Raider*. http://www.mobygames.com/game/tomb-raider. Version: November 2013 243

[MobyGames.com 2014a] MobyGames.com: *MobyGames: Diablo III*. http://www.mobygames.com/game/diablo-iii. Version: März 2014 106

[MobyGames.com 2014b] MobyGames.com: *MobyGames: Skyrim*. http://www.mobygames.com/game/elder-scrolls-v-skyrim. Version: März 2014 193

[Mohr 2013] Mohr, Joachim: *Affine Abbildungen*. http://kilchb.de/faqmath5.html. Version: Dezember 2013 129

[Montola 2005] Montola, Markus: Exploring the Edge of the Magic Circle: Defining Pervasive Games. In: *CD-ROM Proceedings of Digital Arts and Culture. Copenhagen*, 2005, S. 1–3 294

[Montola 2009] Montola, Markus: *Pervasive Games: Theory and Design (Morgan Kaufmann Game Design Books)*. Elsevier Science & Technology, 2009. – 336 S. – ISBN 978–0123748539 294

[Montola 2011] Montola, Markus: A ludological view on the pervasive mixed-reality game research paradigm. In: *Personal Ubiquitous Comput.* 15 (2011), Januar, Nr. 1, 3–12. http://dx.doi.org/10.1007/s00779-010-0307-7. – DOI 10.1007/s00779–010–0307–7. – ISSN 1617–4909 139, 294

[Mulligan u. Patrovsky 2003] Mulligan, Jessica ; Patrovsky, Bridgette: *Developing Online Games. An Insiders Guide*. New Riders, 2003. – 528 S. – ISBN 978–1592730001 6.4

[Mulvihill 2014] Mulvihill, Mike: *What is Shadowrun? - Shadowrun Primer*. PDF. https://s3.amazonaws.com/ch_shadowrun/shadowrun_primer.pdf. Version: Januar 2014 14.2.1, 14.2.1

[Nagle 1984] Nagle, J.: *RFC-896 - Congestion Control in IP/TCP Internetworks*. http://tools.ietf.org/html/rfc896. Version: Januar 1984 2.3.2.1, 11.3

[Nardi u. Harris 2006] Nardi, Bonnie ; Harris, Justin: Strangers and friends: collaborative play in world of warcraft. In: *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. New York, NY, USA : ACM, 2006 (CSCW '06). – ISBN 1–59593–249–6, 149–158 6.4

[Nareyek 2000] Nareyek, Alexander: Intelligent Agent for Computer Games. In: *In Proceedings of the Second International Conference on Computers and Games*, 2000 9.1, 9.3.1, 9.3.2, 226

[Nareyek 2004] Nareyek, Alexander: AI in Computer Games. In: *Queue* 1 (2004), Februar, Nr. 10, 58–65. http://dx.doi.org/10.1145/971564.971593. – DOI 10.1145/971564.971593. – ISSN 1542–7730 219, 9.1, 9.3.2, 9.3.3

[Nextragen 2012] Nextragen: *Zur Analyse : Was bedeutet Jitter*. http://www.nextragen.de/fileadmin/Documents/Whitepaper/WP_Jitter.pdf. Version: jan 2012 11.1.1

[Nolan 2000] Nolan, Christopher: *Memento*. Version: Oktober 2000. http://www.imdb.com/title/tt0209144/ 63, 207

[Norman 1999] Norman, Donald A.: Affordance, conventions, and design. In: *interactions* 6 (1999), Nr. 3, S. 38–43 7.4, 170

[Nova u. a. 2006] Nova, Nicolas ; Girardin, Fabien ; Molinari, Gaelle ; Dillenbourg, Pierre: The underwhelming effects of automatic location-awareness on collaboration in a pervasive game. In: *International Conference on the Design of Cooperative Systems*. Carry-le-Rouet, Provence, France, may 2006 (COOP '06) 12.2

[Oculus VR, Inc. 2013] Oculus VR, Inc.: *Oculus VR | Oculus Rift - Virtual Reality Headset for Immersive 3D Gaming*. http://www.oculusvr.com. Version: Dezember 2013 114, 26

[Oracle 2011] Oracle: *Oracle Java Reference - Class Inputstream*. http://docs.oracle.com/javase/6/docs/api/java/io/InputStream.html#available(). Version: Dezember 2011 3.2.7.1

[Pantel u. Wolf 2002] Pantel, Lothar ; Wolf, Lars C.: On the impact of delay on real-time multiplayer games. In: *Proceedings of the 12th international workshop on Network and operating systems support*

for digital audio and video. New York, NY, USA : ACM, 2002 (NOSSDAV '02). – ISBN 1–58113–512–2, 23–29 6.4

[Pappe 2011] Pappe, G.: *P&P-Rollenspiel: der kollektive Zugang zu utopischen Weltentwürfen und individuellen Phantasie-Konstrukten.* Logos-Verlag, 2011 http://books.google.de/books?id=4Lmy-wWmPJAC. – ISBN 9783832527778 3.1.1

[Patrick Wyatt 2012] Patrick Wyatt: *Choosing a game network library - Code Of Honor.* http://www.codeofhonor.com/blog/choosing-a-game-network-lib. Version: April 2012 3.1.5

[PC Games 2010] PC Games: *Screenshots zu Need for Speed 14: So macht Criterion Need for Speed: Hot Pursuit perfekt.* http://www.pcgames.de/Need-for-Speed-Hot-Pursuit-PC-234890/Specials/Need-for-Speed-14-So-macht-Criterion-Need-for-Speed-Hot-Pursuit-perfekt-754829/galerie/1254191/. Version: März 2010 1.3

[PCWelt 2013] PCWelt: *Provider im Bandbreitenvergleich.* http://www.speedmeter.de/ranglist. Version: August 2013 2.3.4

[Pearce 1994] Pearce, Celia: The ins & outs of non-linear storytelling. In: *ACM SIGGRAPH Computer Graphics* 28 (1994), Nr. 2, S. 100–101 8.2

[Peel 2013] Peel, Jeremy: *Just how big is a Minecraft world? Big, as it turns out.* http://www.pcgamesn.com/minecraft/just-how-big-minecraft-world-big-it-turns-out. Version: feb 2013 173

[Perron u. Wolf 2008] Perron, B. ; Wolf, M.J.P.: *The Video Game Theory Reader 2.* Taylor & Francis, 2008 http://books.google.co.uk/books?id=oe0zNalKkTgC. – ISBN 9780203887660 10.3, 255

[Petlund u. a. 2008] Petlund, Andreas ; Evensen, Kristian ; Griwodz, Carsten ; Halvorsen, Pål: TCP enhancements for interactive thin-stream applications. In: *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video.* New York, NY, USA : ACM, 2008 (NOSSDAV '08). – ISBN 978–1–60558–157–6, 127–128 11.3.1

[Philip Elmer-DeWitt 2013] Philip Elmer-DeWitt: *Chart of the day: U.S. smartphone market 2005-2012 - Apple 2.0 -Fortune Tech.* http://tech.fortune.cnn.com/2013/03/20/apple-smartphone-comscore-android/. Version: März 2013 B.3

[Piccione 2010] Piccione, Peter A.: *In Search of the Meaning of Senet.* http://www.gamesmuseum.uwaterloo.ca/Archives/Piccione/index.html. Version: Januar 2010 1

[Plüss 2004] Plüss, Ä.: *Java - exemplarisch: learning by doing.* Oldenbourg, 2004 http://books.google.de/books?id=b-HMaUG1ct0C. – ISBN 9783486200409 2.3.1, 2.3.2, 2.6, 11.2

[Pohl 2008] Pohl, Klaus: *Requirements Engineering: Grundlagen, Prinzipien,Techniken.* 2. dpunkt.Verlag GmbH, 2008. – 764 S. – ISBN 978–3898645508 292

[Pol u. a. 2002] Pol, Martin ; Koomen, Tim ; Spillner, Andreas: *Management und Optimierung des Testprozesses: ein praktischer Leitfaden für erfolgreiches Testen von Software mit TPI und TMap.* Heidelberg : Dpunkt-Verl., 2002. – ISBN 3898641562 9783898641562 2.5

[Postel 1981] Postel, J.: *RFC: 791 - Internet Protocol.* http://tools.ietf.org/html/rfc791. Version: September 1981 2.3.2

[Pritchard 2014] Pritchard, Martin: *What is Evaluation.* http://evaluationtoolbox.net.au/index.php?option=com_content&view=article&id=11&Itemid=17. Version: feb 2014 24

[Projekt Blinkenlights 2013] Projekt Blinkenlights: *Pong | Project Blinkenlights.* http://blinkenlights.net/blinkenlights/pong. Version: Dezember 2013 C.2, C.3

[Puls 2012] Puls, Margaret: *Did Arthur C Clarke invent the iPad?* http://margaretpuls.wordpress.com/2012/02/12/worlds-first-ipad/. Version: Februar 2012 119

[Puppe 1990] Puppe, Frank: *Einführung in Expertensysteme.* Berlin Heidelberg : Springer, 1990 9.3.5, 236

[Qusay H. Mahmoud 1996] Qusay H. Mahmoud: *Sockets programming in Java: A tutorial.* http://www.javaworld.com/javaworld/jw-12-1996/jw-12-sockets.html. Version: Dezember 1996 2.3.5

[Radoff 2010] Radoff, Jon: *History of Social Games.* http://radoff.com/blog/2010/05/24/history-social-games/. Version: Mai 2010 B.1, B.8

[Reif 2000] Reif, Gerald: *Expertensysteme.* http://www.dfki.uni-kl.de/~aabecker/Mosbach/Experten/Reif-node8.html. Version: feb 2000 236

[Reimann u. Paelke 2005] Reimann, Christian ; Paelke, Volker: Adaptive Mixed Reality Games. In: *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. New York, NY, USA : ACM, 2005 (ACE '05). – ISBN 1–59593–110–4, 302–305 26

[Rey u. Wender 2010] Rey, Günter D. ; Wender, Karl F.: *Neuronale Netze: Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung*. 2. Huber, 2010. – 205 S. – ISBN 978–3456848815 9.3.6

[Robben 2012] Robben, Bernard: Die Bedeutung der Körperlichkeit für be-greifbare Interaktion mit dem Computer. In: Bernard Robben, Heidi S. (Hrsg.): *Be-greifbare Interaktionen: Der allgegenwärtige Computer: Touchscreens, Wearables, Tangibles und Ubiquitous Computing*. Transcript Verlag, 2012, Kapitel 1, S. 19–39 6, 6.1

[Robben u. Schelhowe 2012] Robben, Bernard ; Schelhowe, Heidi: *Be-greifbare Interaktionen: Der allgegenwärtige Computer: Touchscreens, Wearables, Tangibles und Ubiquitous Computing*. 1. Transcript, 2012. – 398 S. – ISBN 978–3837620054 6.2

[Rocha u. a. 2008] Rocha, José B. ; Mascarenhas, Samuel ; Prada, Rui: Game mechanics for cooperative games. In: *ZON Digital Games 2008* 1 (2008), S. 72–80 6.4

[Rogers 2009] Rogers, Rick: *Android application development*. Sebastopol, Calif. : O'Reilly, 2009. – ISBN 9780596521479 0596521472 2.5.3

[Rogers u. a. 2009] Rogers, Yvonne ; Lim, Youn-kyung ; Hazlewood, William R. ; Marshall, Paul: Equal opportunities: Do shareable interfaces promote more group participation than single user displays? In: *Human–Computer Interaction* 24 (2009), Nr. 1-2, S. 79–116 7.3, 26

[Rogers u. a. 2011] Rogers, Yvonne ; Sharp, Helen ; Preece, Jenny: *Interaction Design: Beyond Human-Computer Interaction*. Bd. 3. John Wiley & Sons, 2011. – ISBN 978–0470665763 482

[Ryan u. Deci 2000] Ryan, Richard M. ; Deci, Edward L.: Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. In: *American psychologist* 55 (2000), Nr. 1, S. 68 13.2

[Sakaguchi u. Sakakibara 2001] Sakaguchi, Hironobu ; Sakakibara, Motonori: *Final Fantasy: The Spirits Within*. Juli 2001 150

[Sakarias411 2013] Sakarias411: *Steam Community - Complete Ending Chart*. http://steamcommunity.com/sharedfiles/filedetails/?id=186774783. Version: Oktober 2013 B.7

[Salen u. Zimmerman 2003] Salen, K. ; Zimmerman, E.: *Rules of Play: Game Design Fundamentals.* MIT Press, 2003 https://mitpress.mit.edu/books/rules-play. – ISBN 9780262240451 71, 72, 8.0.1, 179, 181, 182, B.1

[Sander 2011] Sander, Ralf: *Smartphones - so verändern sie unser Leben.* http://www.stern.de/digital/telefon/2-mobile-world-congress-smartphones-so-veraendern-sie-unser-leben-1655195.html. Version: feb 2011 21

[Sauter 2014] Sauter, Marc: *Upvoid Engine: Algorithmen für prozedurale Minecrafts.* http://www.golem.de/news/upvoid-engine-algorithmen-fuer-prozedurale-minecrafts-1402-104390.html. Version: Februar 2014 188

[Schick 1991] Schick, Lawrence: *Heroic Worlds: A History and Guide to Role-Playing Games.* Prometheus Books, Publishers, 1991. – ISBN 9780879756536 1.1.1, 9, 10, 2.1, 2.1.1, 2.1.2, 2.1.4, 5.1.1.1, 84, 86, 14.2, 314, 315, 14.2.1, 462, 26

[Schmitt 1999] Schmitt, Bernd: Experiential marketing. In: *Journal of marketing management* 15 (1999), Nr. 1-3, S. 53–67 6.2

[Schnabel 2004] Schnabel, Patrick: *Netzwerktechnik-Fibel: Grundlagen. Übertragungstechnik und Protokolle. Anwendungen und Dienste, Sicherheit.* 1. Books on Demand Gmbh, 2004 445

[Scholar 2014] Scholar, Google: *Penelope Sweetser.* http://scholar.google.com.au/citations?user=mrvBuvsAAAAJ&hl=de. Version: feb 2014 481

[Seay u. a. 2004] Seay, A. F. ; Jerome, William J. ; Lee, Kevin S. ; Kraut, Robert E.: Project massive: a study of online gaming communities. In: *CHI '04 Extended Abstracts on Human Factors in Computing Systems.* New York, NY, USA : ACM, 2004 (CHI EA '04). – ISBN 1–58113–703–6, 1421–1424 6.4

[Seif El-Nasr u. a. 2010] Seif El-Nasr, Magy ; Aghabeigi, Bardia ; Milam, David ; Erfani, Mona ; Lameman, Beth ; Maygoli, Hamid ; Mah, Sang: Understanding and evaluating cooperative games. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* New York, NY, USA : ACM, 2010 (CHI '10). – ISBN 978–1–60558–929–9, 253–262 6.4, 10, 338, 339, 360, 24.3.1

[Sellier 2014] Sellier, Alexis: *Cooperation vs Collaboration.* http://cloudhead.headmine.net/post/3279118157/cooperation-vs-collaboration. Version: jan 2014 6.4, 26

[Sharma u. a. 2011] Sharma, G. ; Shroff, G. ; Dewan, P.: Workplace collaboration in a 3D Virtual Office. In: *VR Innovation (ISVRI), 2011 IEEE International Symposium on*, 2011, S. 3–10 12.1.1

[Sheldon 2004] Sheldon, Lee ; Doell, Sandy (Hrsg.): *Character Development And Storytelling for Games*. 1. Auflage. Course Technology, 2004 (Game Development Series) 1, 3.1.1, 3.1.2, 3.1, 3.2.5, 3.2.5, 3.10, 3.2.5, 3.2.5, 3.11, 3.12, 4, 5.2.2.2, 8, 26

[Sheldon 2013] Sheldon, Lee ; Justak, Marta (Hrsg.): *Character Development And Storytelling for Games, Second Edition*. 2. Auflage. Course Technology, 2013 (Game Development Series) 5.2.3, 108, 8, 176, 178, 8.1, 8.1, 191, 8.1, 8.2.1, 8.2, 8.2.1, 8.2.2, 8.3, 8.2.2, 194, 8.2.3, 8.4, 8.3, 8.6, 8.3, 8.3, 203, 204, 8.7, 209, 210, 211, 13.1, 14, 14.1, 14.2, 312

[Sheldon u. a. 2003] Sheldon, Nathan ; Girard, Eric ; Borg, Seth ; Claypool, Mark ; Agu, Emmanuel: The effect of latency on user performance in Warcraft III. In: *Proceedings of the 2nd workshop on Network and system support for games*. New York, NY, USA : ACM, 2003 (NetGames '03). – ISBN 1–58113–734–6, 3–14 11.1, 285

[Shiffman 2011] Shiffman, Daniel: *Sorting the Vertices of a Polygon*. http://shiffman.net/2011/12/23/night-4-sorting-the-vertices-of-a-polygon/. Version: Dezember 2011 23.5

[Sick 2012] Sick, Andrea: Übersetzen und Begreifen – Medienkünstlerische Operationen im Labor. In: Bernard Robben, Heidi S. (Hrsg.): *Be-greifbare Interaktionen: Der allgegenwärtige Computer: Touchscreens, Wearables, Tangibles und Ubiquitous Computing*. Transcript Verlag, 2012, Kapitel 4, S. 101–112 6, 12.1, C.2

[Siegmund 2002] Siegmund, Gerd: *Technik der Netze*. Heidelberg : Hüthig, 2002. – ISBN 377853954X 9783778539545 3826650212 9783826650215 9783778540343 3778540343 2.3.2, 11.2

[Silva 2012] Silva, Vladimir: *Advanced Android 4 games*. Berkeley, Calif. : Apress, 2012. – ISBN 9781430240594 1430240598 2.5.1

[Smith 2010] Smith, John: *The evolution of Pokémon over the years*. http://www.examiner.com/article/the-evolution-of-pok-mon-over-the-years. Version: Juli 2010 370

[Snader 2000] Snader, Jon C.: *Effective TCP/IP Programming: 44 Tips to Improve Your Network Programs: 44 Tips to Improve Your Network Programs*. 1. Amsterdam : Addison-Wesley, 2000. – 320 S. – ISBN 978–0201615890 272, 11.2, 11.3, 11.3, 11.3, 24.1.2, 24.2, 443, 444

[Soderlund 2005] Soderlund, Staffan: *Method for Playing Games Using Brain Waves*. Juli 2005. – CIB: A63F13/10; A63F7/00; A63F7/04; A63F9/00; A63F9/24; (IPC1-7): A61B5/0476; A63F13/10; A63F9/14 C.1

[id Software 2014] Software id: *Quake III Arena*. www.idsoftware.com/games/quake/quake3-arena. Version: jan 2014 12.1.3

[SourceMaking 2013] SourceMaking: *Spaghetti Code*. http://sourcemaking.com/antipatterns/spaghetti-code. Version: dez 2013 349

[Soute u. a. 2010] Soute, Iris ; Markopoulos, Panos ; Magielse, Remco: Head Up Games: combining the best of both worlds by merging traditional and digital play. In: *Personal Ubiquitous Comput.* 14 (2010), Juli, Nr. 5, 435–444. http://dx.doi.org/10.1007/s00779-009-0265-0. – DOI 10.1007/s00779–009–0265–0. – ISSN 1617–4909 12.2

[Spielberg 2002] Spielberg, Steven: *Minority Report*. Juni 2002. –) 127

[Spronck u. a. 2006] Spronck, Pieter ; Ponsen, Marc ; Postma, Eric: Adaptive game AI with dynamic scripting. In: *Machine Learning*, Kluwer, 2006, S. 217–248 9.3.1, 225, 16.1, 16.1

[Spur u. Krause 1997] Spur, Günter ; Krause, Frank-Lothar: *Das virtuelle produkt*. Hanser Munich and Vienna, 1997 7.4

[Stallmeister 2012] Stallmeister, Ute: *Stiftung Weltbevölkerung: Weltbevölkerung zum Jahreswechsel 2012/2013*. http://www.weltbevoelkerung.de/oberes-menue/presse/presse/presseinformationen/news-ansicht/display/weltbevoelkerung-zum-jahreswechsel-20122013.html. Version: Dezember 2012 2

[Stary 1996] Stary, Christian: *Interaktive Systeme: Software-Entwicklung und Software-Ergonomie*. Braunschweig; Wiesbaden : Vieweg, 1996. – ISBN 3528153849 9783528153847 6.2.1, 6.2.1

[Stöcker 2010] Stöcker, Christian: *Videospiel "Heavy Rain": Der beste Digital-Thriller aller Zeiten*. http://www.spiegel.de/netzwelt/games/videospiel-heavy-rain-der-beste-digital-thriller-aller-zeiten-a-680010.html. Version: feb 2010 126

[Steinlechner 2013] Steinlechner, Peter: *Blizzard: Diablo 3 erscheint für die Playstation 4 - und die PS3 - Golem.de*. http://www.golem.de/news/blizzard-diablo-3-erscheint-fuer-die-playstation-4-und-die-ps3-1302-97731.html. Version: Februar 2013 1.3

[Stoecker 2013] Stoecker, Christian: *Gamescom-Trend: Wir spielen mal Gott mit dem Touchscreen*. http://www.spiegel.de/netzwelt/games/gamescom-trend-a-917914.html. Version: August 2013 3.1.4, 3.3, 6.3, 6.3

[Stoll 2013] Stoll, Johannes: *Osi 7 layer model*. https://de.wikipedia.org/w/index.php?title=Datei:Osi_7_layer_model.svg&filetimestamp=20130417132030&. Version: apr 2013 11.2

[Subramanian u. a. 2006] Subramanian, Vijay ; Ramakrishnan, K. K. ; Kalyanaraman, Shiv ; Ji, Lusheng: Impact of Interference and Capture Effects in 802.11 Wireless Networks on TCP. In: *Proceedings of the Second International Workshop on Wireless Traffic Measurements and Modeling*. New York, NY, USA : ACM, 2006 (WitMeMo '06). – ISBN 1–59593–511–8 24.2

[Surman 2003] Surman, David: *CGI Animation: Pseudorealism, Perception and Possible Worlds*. http://bit.ly/1dcroCP. Version: 2003 149, 151, 152

[Sweetser u. Wyeth 2005] Sweetser, Penelope ; Wyeth, Peta: GameFlow: a model for evaluating player enjoyment in games. In: *Comput. Entertain.* 3 (2005), Juli, Nr. 3, 3–3. http://dx.doi.org/10.1145/1077246.1077253. – DOI 10.1145/1077246.1077253. – ISSN 1544–3574 26

[SWTOR Kundendienst] SWTOR Kundendienst: *Wie konfiguriere ich meinen Router so, dass ich auf Star Wars: The Old Republic zugreifen kann?* http://www.swtor.com/de/support/helpcenter/2138 2.3.2.2

[Tafeit 2014] Tafeit, Christopher: *Google Glass: Die Datenbrille zum Absetzen - Golem.de*. http://www.golem.de/news/google-glass-die-datenbrille-zum-absetzen-1401-104219.html. Version: Januar 2014 451

[Tamura u. a. 2001] Tamura, H. ; Yamamoto, H. ; Katayama, A.: Mixed reality: future dreams seen at the border between real and virtual worlds. In: *Computer Graphics and Applications, IEEE* 21 (2001), Nov, Nr. 6, S. 64–70. http://dx.doi.org/10.1109/38.963462. – DOI 10.1109/38.963462. – ISSN 0272–1716 13

[Tanenbaum u. Wetherall 2010] Tanenbaum, Andrew S. ; Wetherall, David J.: *Computer Networks*. 5. Prentice Hall, 2010. – ISBN 978–0132126953 279

[Taylor 2009] Taylor, T. L.: *Play Between Worlds: Exploring Online Game Culture*. The MIT Press, 2009. – ISBN 978–0262512626 6.4

[Team 2013] Team, JBoss D.: *Drools Expert User Guide*, dez 2013. https://docs.jboss.org/drools/release/6.0.1.Final/drools-docs/pdf/drools-docs.pdf 232, 9.6

[The Arcade Flyer Archive 2014] The Arcade Flyer Archive: *The Arcade Flyer Archive - Video Game Flyers: Moto-Cross, Sega*. http://bit.ly/NOP4DS. Version: Februar 2014 307

[The Shakespeare Globe Trust 2014] The Shakespeare Globe Trust: *Globe Theatre - interior*. http://press.shakespearesglobe.com/media-gallery/on-site/globe-theatre-interior. Version: Februar 2014 301

[thedesigner u. Zephbane 2012] THEDESIGNER ; ZEPHBANE: *Cryengine 3 support level streaming with streaming volumes ?* http://www.crydev.net/viewtopic.php?f=321&t=86117. Version: März 2012 2.2.1

[TheKhoaNguyen 2010] THEKHOANGUYEN: *Geteilter Schirm, doppelter Spaß - Zehn der besten aktuellen Splitscreen-Games.* http://www.pcgames.de/Panorama-Thema-233992/Specials/Geteilter-Schirm-doppelter-Spass-Zehn-der-besten-aktuellen-Splitscreen-Games-757060/. Version: Juni 2010 361, 26.1

[Thorn 2012] THORN, Alan: *UDK game development.* Boston, Mass. : Course Technology PTR, 2012 http://site.ebrary.com/id/10521214. – ISBN 9781435460195 1435460197 1435460189 9781435460188 3.2.6.2, 10.1

[Thylmann u. a. 2013] THYLMANN, Marc ; POLS, Axel ; KLUSMANN, Bernd: *Smartphones sorgen für 96 Prozent des Handy-Umsatzes.* Februar 2013 13.2

[Tobler 1970] TOBLER, Waldo R.: A computer movie simulating urban growth in the Detroit region. In: *Economic geography* 46 (1970), S. 234–240 12

[tofuprod 2013] TOFUPROD: *The main console unit of the PlayStation 4, on display at E3 2013.* http://commons.wikimedia.org/wiki/File:PlayStation_Four.png?uselang=de. Version: Juni 2013 1.3

[Tokyoship 2013] TOKYOSHIP: *Vector layout illustration of the new PlayStation 4 controller prototype shown in the PlayStation Meeting 2013 as seen from the side.* http://commons.wikimedia.org/wiki/File:Dualshock_4_Layout_2.svg. Version: Februar 2013 1.3

[Tolstoy 2009] TOLSTOY, L.: *Childhood.* ReadHowYouWant.com, Limited, 2009 http://books.google.de/books?id=vR9wDj4A7dMC. – ISBN 9781427018519 5

[TR Jain 2006] TR JAIN, A.S.S.: *Quantitative Methods.* Vk Publications, 2006 http://books.google.de/books?id=aqtrjYFSiC8C. – ISBN 9788187140573 23.1, 23.2

[Tresca 2011] TRESCA, M.J.: *The Evolution of Fantasy Role-Playing Games.* McFarland & Company, 2011 (The Evolution of Fantasy Role-playing Games). http://books.google.de/books?id=8H8bzqj6S4sC. – ISBN 9780786460090 3.1.1, 52, 3.1.3

[Tychsen u. a. 2005] TYCHSEN, Anders ; HITCHENS, Michael ; BROLUND, Thea ; KAVAKLI, Manolya: The Game Master. In: *Proceedings of the second Australasian conference on Interactive entertainment.*

Sydney, Australia, Australia : Creativity & Cognition Studios Press, 2005 (IE '05). – ISBN 0–9751533–2–3, 215–222 2.1, 2.1.1, 2.1.2, 2.1.4, 3.1.1, 3.1.3, 53, 1, 2, 3, 3.1.3, 3.2

[Tychsen u. a. 2006] Tychsen, Anders ; Hitchens, Michael ; Brolund, Thea ; Kavakli, Manolya: Live Action Role-Playing Games Control, Communication, Storytelling, and MMORPG Similarities. In: *Games and Culture* 1 (2006), Nr. 3, S. 252–275 12.2.1

[Tychsen u. Smith 2008] Tychsen, Anders ; Smith, Jonas H.: Game Format Effects on Communication in Multi-player Games. In: *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*. New York, NY, USA : ACM, 2008 (Future Play '08). – ISBN 978–1–60558–218–4, 113–120 13

[Ullenboom 2012] Ullenboom, Christian: *Java 7 - mehr als eine Insel: das Handbuch zu den Java SE-Bibliotheken ; [Nebenläufigkeit, Datenstrukturen und Algorithmen, Ein/Ausgabe, XML, Swing, Grafik- und Netzwerkprogrammierung, RMI, JavaServer Pages und Servlets, JDBC, Reflection u.v.m.]*. Bonn : Galileo Press, 2012. – ISBN 3836215071 9783836215077 2.3.5

[Valve Corporation 2013a] Valve Corporation: *Steam: Download Stats, Satellite View*. http://store.steampowered.com/stats/content/. Version: November 2013 B.2

[Valve Corporation 2013b] Valve Corporation: *Steam: Game and Player Statistics*. http://store.steampowered.com/stats/. Version: August 2013 1, 1.1

[Valve Corporation 2013c] Valve Corporation: *Steam: Game and Player Statistics*. http://store.steampowered.com/stats/. Version: November 2013 5.1

[Valve Corporation 2013d] Valve Corporation: *Steam: Games List*. http://store.steampowered.com/search/#category1=998. Version: November 2013 78

[Virtually Better, Inc. 2014] Virtually Better, Inc.: *Virtually Better,Virtual Reality Exposure Therapy,Cognitive Behavioral Therapy Atlanta*. http://www.virtuallybetter.com/virtuallybetterclinic/VRtreatment.html. Version: Januar 2014 6.1

[Vogel u. Mauve 2001] Vogel, Jürgen ; Mauve, Martin: Consistency control for distributed interactive media. In: *Proceedings of the ninth ACM international conference on Multimedia*. New York, NY, USA : ACM, 2001 (MULTIMEDIA '01). – ISBN 1–58113–394–4, S. 221–230 289

[Voida u. a. 2010] Voida, Amy ; Carpendale, Sheelagh ; Greenberg, Saul: The individual and the group in console gaming. In: *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. New York, NY, USA : ACM, 2010 (CSCW '10). – ISBN 978–1–60558–795–0, S. 371–380 6.4, 26

[Volker Zota 2010] Volker Zota: *Microsoft Kinect ab 10. November in Deutschland zu haben.* http://www.heise.de/newsticker/meldung/Microsoft-Kinect-ab-10-November-in-Deutschland-zu-haben-1060185.html. Version: August 2010 14

[Voyles 2013] Voyles, Dave: *UnrealScript Game Programming Cookbook.* Birmingham : Packt Publishing, Limited, 2013. – ISBN 9781849695565 1849695563 10

[VRmagic Holding AG 2014] VRmagic Holding AG: *Eyesi Surgical.* http://www.vrmagic.com/simulators/eyesi-surgical/. Version: Januar 2014 6.1

[Wachowski u. Wachowski 1999a] Wachowski, Andy ; Wachowski, Lana: *The Matrix.* Movie. http://www.imdb.com/title/tt0133093/?ref_=nv_sr_1. Version: Juni 1999 138

[Wachowski u. Wachowski 1999b] Wachowski, Andy ; Wachowski, Lana: *Matrix Quotes.* http://www.imdb.com/title/tt0133093/quotes. Version: 1999 145

[Waibel 2012] Waibel, Barbara: *Das Zeppelin-Luftschiff LZ 129 Hindenburg.* Sutton Verlag GmbH, 2012 (Bilder der Luftfahrt). http://books.google.de/books?id=TPoJTm3FdfkC. – ISBN 9783954001057 14.2.1

[Walker 1999] Walker, Jill: Piecing Together and Tearing Apart: Finding the Story in Afternoon. In: *Proceedings of the Tenth ACM Conference on Hypertext and Hypermedia : Returning to Our Diverse Roots: Returning to Our Diverse Roots.* New York, NY, USA : ACM, 1999 (HYPERTEXT '99). – ISBN 1–58113–064–3, 111–117 8.1

[Wang u. a. 2009] Wang, Alf I. ; Jarrett, Martin ; Sorteberg, Eivind: Experiences from implementing a mobile multiplayer real-time game for wireless networks with high latency. In: *Int. J. Comput. Games Technol.* 2009 (2009), Januar, S. 6:1–6:14. http://dx.doi.org/10.1155/2009/530367. – DOI 10.1155/2009/530367. – ISSN 1687–7047 11.3.1, 286, 17, 434, 24.1.2

[Wang 2011] Wang, Xin: *Mobile Ad-Hoc Networks: Applications.* InTech, 2011. http://dx.doi.org/10.5772/882. http://dx.doi.org/10.5772/882 24.2

[Wasserbäch u. Holland 2013] Wasserbäch, Kai ; Holland, Matrin: *BattleTech-Erfinder will Tabletop-Spiel mit digitalen Elementen erweitern.* http://www.heise.de/newsticker/meldung/BattleTech-Erfinder-will-Tabletop-Spiel-mit-digitalen-Elementen-erweitern-1955124.html. Version: November 2013 5.2.3

[Watte u. a. 2005] Watte, John ; Woynillowicz, Joe ; Schultz, John: *MMORPG and the ol' UDP vs TCP - Page 3 - Multiplayer and NetworkProgramming.* http://www.gamedev.net/topic/

319003-mmorpg-and-the-ol-udp-vs-tcp. Version: Mai 2005 2.3.2.1, 2.3.2.2, 3.1.5, 11.3.1

[Weber 2013] Weber, Nina: *Borderlands 2 - Assault on Dragon Keep: Drachentöten im wilden Westen.* http://www.spiegel.de/netzwelt/games/borderlands-2-assault-on-dragon-keep-drachentoeten-leicht-gemacht-a-915130.html. Version: August 2013 8.3

[Weiser 1991] Weiser, Mark: The computer for the 21st century. In: *Scientific american* 265 (1991), Nr. 3, S. 94–104 6.1, 6.2, 6.3, 13.1

[Westermark 2007] Westermark, Christian: *Mobile Multiplayer Gaming*, KTH School of Information and Communication Technology, Diplomarbeit, jun 2007 11.3.1, 24.2

[Whitney Museum of American Art 2013] Whitney Museum of American Art: *Whitney Museum of American Art: Douglas Davis.* http://whitney.org/Exhibitions/Artport/DouglasDavis. Version: Dezember 2013 12.1

[Wikia.com 2013] Wikia.com: *Video game industry.* http://vgsales.wikia.com/wiki/Video_game_industry. Version: 2013 75

[Wikia.com 2014a] Wikia.com: *Shadowrun Incubus.* http://shadowrun.wikia.com/wiki/Incubus. Version: Januar 2014 320

[Wikia.com 2014b] Wikia.com: *Shadowrun timeline.* http://shadowrun.wikia.com/wiki/Shadowrun_timeline. Version: Januar 2014 14.2.1, 14.2.2

[wiki.beyondunreal.com 2009] wiki.beyondunreal.com: *UE3:GameViewportClient (UT3) - Unreal Wiki.* http://wiki.beyondunreal.com/UE3:GameViewportClient_(UT3). Version: August 2009 408

[Wikibooks 2013] Wikibooks: *Communication Networks/TCP and UDP Protocols.* http://en.wikibooks.org/wiki/Communication_Networks/TCP_and_UDP_Protocols. Version: Juli 2013 2.3.2.1, 30

[wikiHow 2014] wikiHow: *How to Play a 'Play By Post' RPG.* http://bit.ly/1hf5fDD. Version: mar 2014 261

[Wilkens 2012] Wilkens, Andreas: *Facebook hat eine Milliarde aktive Nutzer.* http://www.heise.de/newsticker/meldung/Facebook-hat-eine-Milliarde-aktive-Nutzer-1723387.html. Version: April 2012 2

[Wilkens 2013] Wilkens, Andreas: *Touchscreen-Bedienung entspricht frühkindlichen Bewegungsmustern.* http://heise.de/-2133968. Version: mar 2013 345

[wiseGEEK 2014] wiseGEEK: *What Is Application Architecture?* http://www.wisegeek.org/what-is-application-architecture.htm. Version: feb 2014 18

[Witt u. Vossen 2006] Witt, Kurt-Ulrich ; Vossen, Gottfried: *Grundkurs Theoretische Informatik.* 4. Vieweg+Teubner Verlag, 2006 (433). – ISBN 978–3834801531 9.3.2, 9.2, 9.3.2

[Wizards of the Coast, Inc. 2003] Wizards of the Coast, Inc.: *Dungeons & Dragons.* http://www.wizards.com/dnd/DnDArchives_History.asp. Version: 2003 11, 2.1.3, 87

[Wolf 2008] Wolf, M.J.P.: *The Video Game Explosion: A History from PONG to Playstation and Beyond.* Greenwood Press, 2008 http://books.google.de/books?id=XiM0ntMybNwC. – ISBN 9780313338687 307

[Yahn W. Bernier 2001] Yahn W. Bernier: *Latency Compensating Methods in Client/Server In-game Protocol Design and Optimization.* http://bit.ly/1iGpyuB. Version: 2001 2.3.3, 288

[Young 2009] Young, Warren: *Winsock Programmer's FAQ Articles: Which I/O Strategy Should I Use?* http://tangentsoft.net/wskfaq/articles/io-strategies.html. Version: Oktober 2009 2.3.6, 2.3.6.3, 2.5.3

[Zagal u. a. 2006] Zagal, José P. ; Rick, Jochen ; Hsi, Idris: Collaborative games: lessons learned from board games. In: *Simul. Gaming* 37 (2006), März, Nr. 1, S. 24–40. http://dx.doi.org/10.1177/1046878105282279. – DOI 10.1177/1046878105282279. – ISSN 1046–8781 6.4, 26

[Zagal u. a. 2000] Zagal, José P. ; Nussbaum, Miguel ; Rosas, Ricardo: A Model to Support the Design of Multiplayer Games. In: *Presence: Teleoper. Virtual Environ.* 9 (2000), Oktober, Nr. 5, S. 448–462. http://dx.doi.org/10.1162/105474600566943. – DOI 10.1162/105474600566943. – ISSN 1054–7460 6.4

[Zander u. a. 2005] Zander, Sebastian ; Leeder, Ian ; Armitage, Grenville: Achieving fairness in multiplayer network games through automated latency balancing. In: *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology.* New York, NY, USA : ACM, 2005 (ACE '05). – ISBN 1–59593–110–4, S. 117–124 289

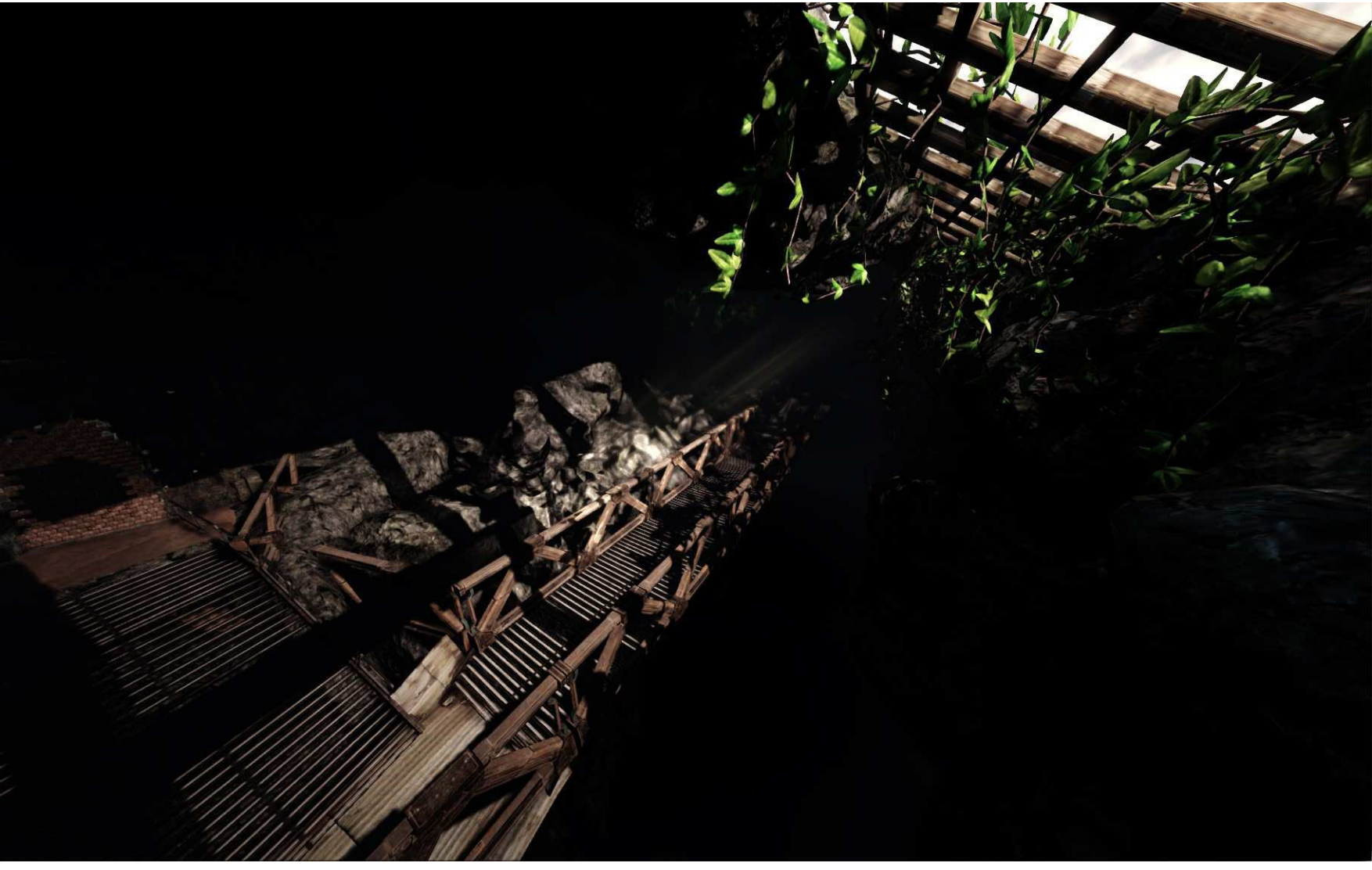
[Zentrum für Arbeitsbeziehungen und Arbeitsrecht 2014] Zentrum für Arbeitsbeziehungen und Arbeitsrecht: *Studenten statt Studierende - ZAAR - Zentrum für Arbeitsbeziehungen und Arbeitsrecht - LMU München.* http://www.zaar.uni-muenchen.de/studium/studenteninfo/student_prof/student/index.html. Version: Dezember 2014 (document)

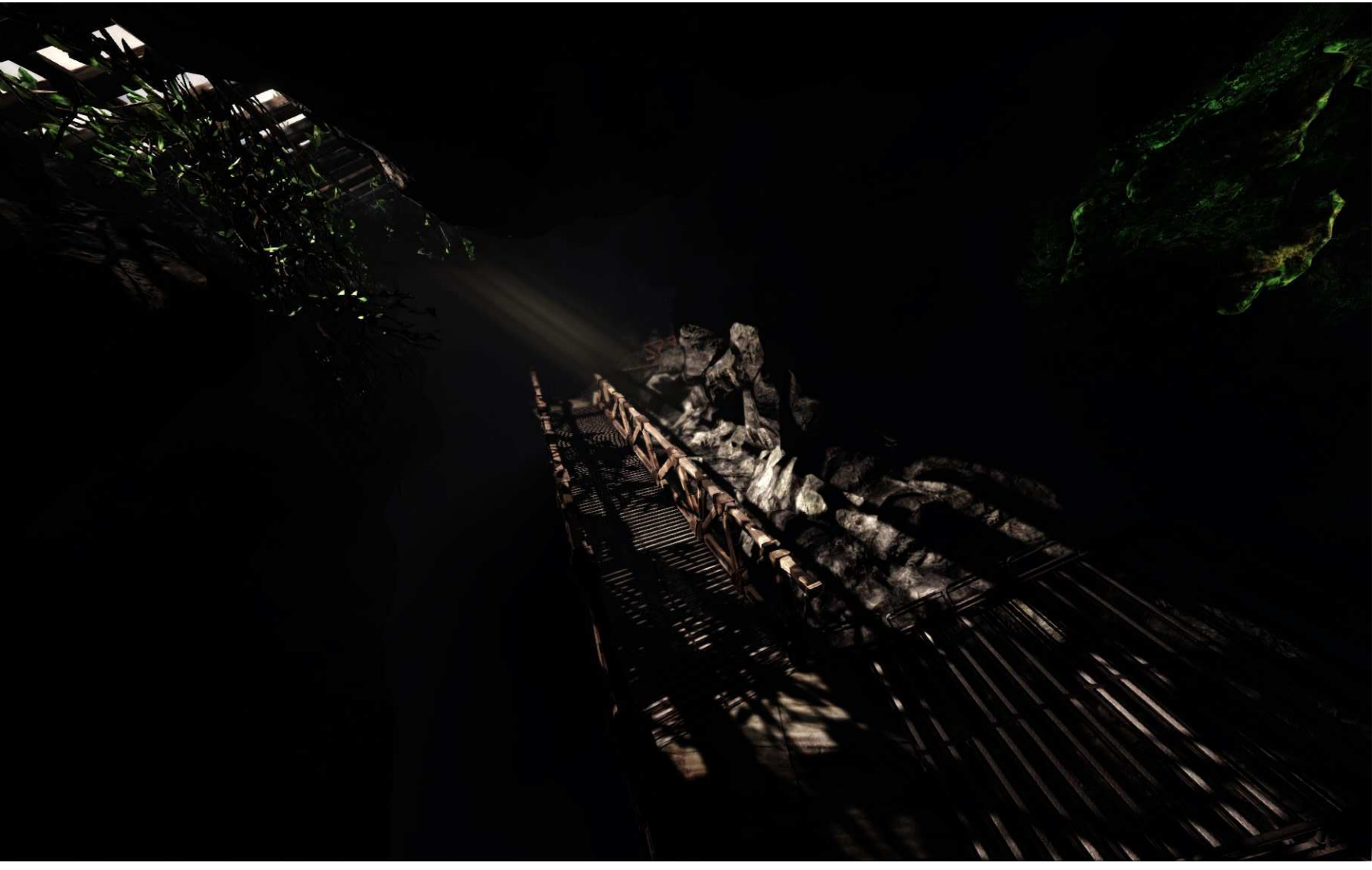
[Zimbardo u. Gerrig 2008] Zimbardo, Philip G. ; Gerrig, Richard J. ; Dietrich, Dr. S. (Hrsg.): *Psychologie*. 18. Pearson Studium, 2008 http://tinyurl.com/3j5tucr. – ISBN: 3827372755
6.2.1

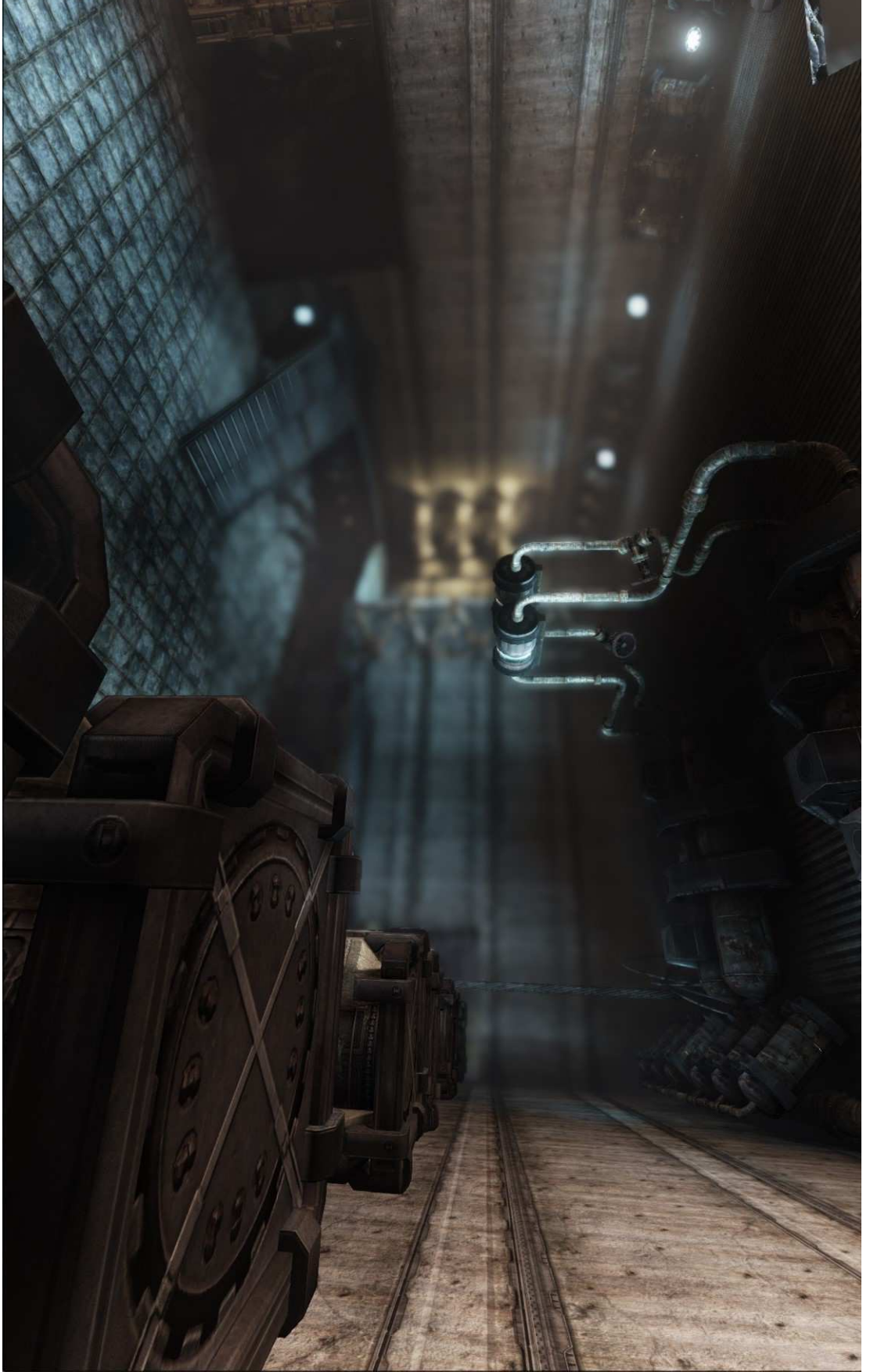
ANHANG A

Bilderserie der realisierten Anwendung

Die folgenden Grafiken zeigen Momentaufnahmen des Spiels und Illustrationen zum Look-&-Feel. Die dargestellten Spielwelten wurden vollständig unter Zuhilfenahme der bereitgestellten Elemente des UDK erstellt.







Antony
Rigger
Nicolas
Docker

Antony
Street Samurai
Nicolas
Decker

Antony
Rigger
Stefan
Street Samurai
Nicolas
Street Samurai

ANHANG B

Illustrationen

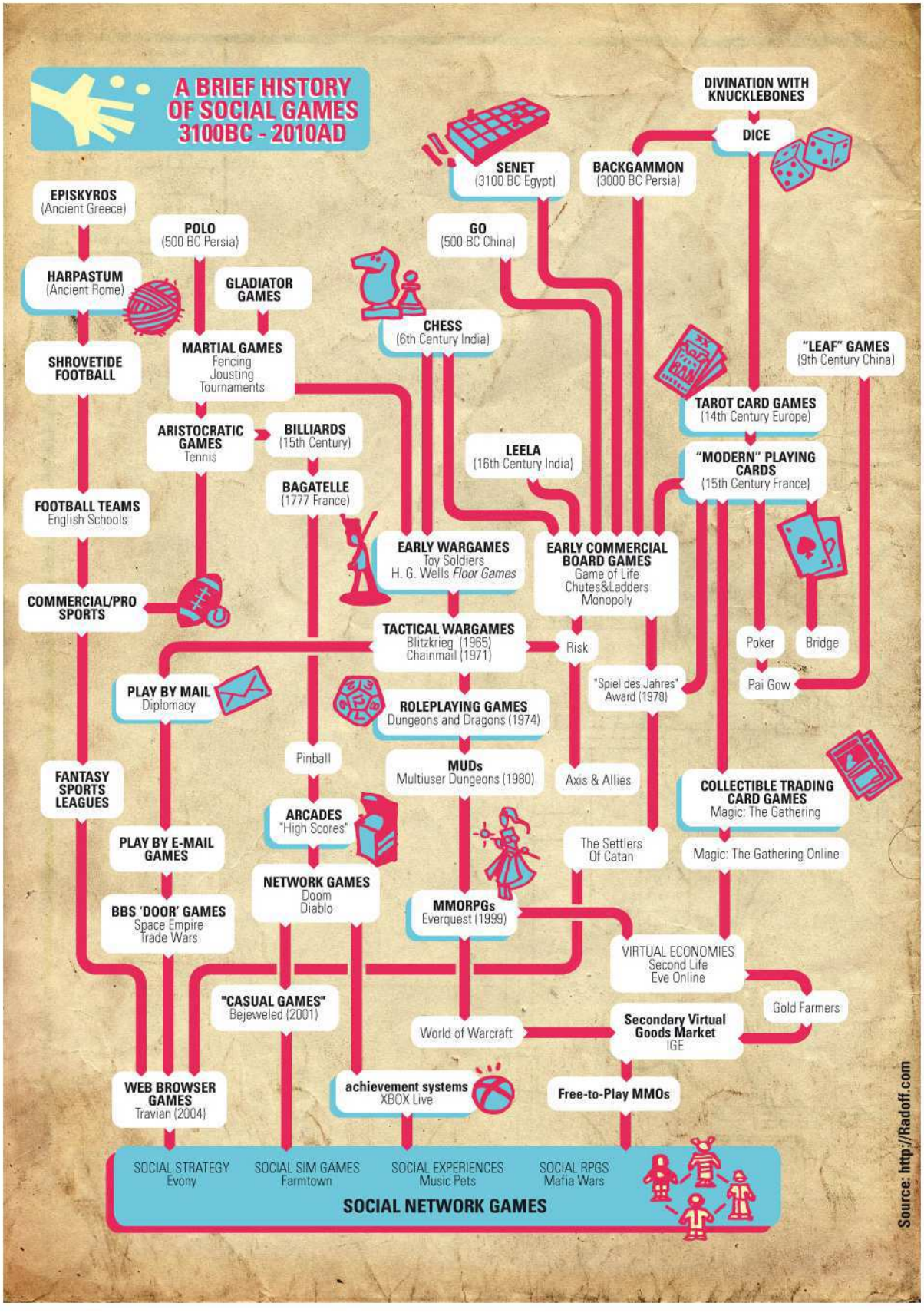

A BRIEF HISTORY OF SOCIAL GAMES 3100BC - 2010AD
DIVINATION WITH KNUCKLEBONES
DICE
SENET (3100 BC Egypt)
BACKGAMMON (3000 BC Persia)
EPISKYROS (Ancient Greece)
POLO (500 BC Persia)
GO (500 BC China)
HARPASTUM (Ancient Rome)
GLADIATOR GAMES
CHESS (6th Century India)
"LEAF" GAMES (9th Century China)
SHROVETIDE FOOTBALL
MARTIAL GAMES Fencing Jousting Tournaments
TAROT CARD GAMES (14th Century Europe)
ARISTOCRATIC GAMES Tennis
BILLIARDS (15th Century)
LEELA (16th Century India)
"MODERN" PLAYING CARDS (15th Century France)
BAGATELLE (1777 France)
FOOTBALL TEAMS English Schools
EARLY WARGAMES Toy Soldiers H. G. Wells Floor Games
EARLY COMMERCIAL BOARD GAMES Game of Life Chutes&Ladders Monopoly
COMMERCIAL/PRO SPORTS
TACTICAL WARGAMES Blitzkrieg (1965) Chainmail (1971)
Risk
Poker
Bridge
PLAY BY MAIL Diplomacy
"Spiel des Jahres" Award (1978)
Pai Gow
ROLEPLAYING GAMES Dungeons and Dragons (1974)
Pinball
MUDs Multiuser Dungeons (1980)
Axis & Allies
COLLECTIBLE TRADING CARD GAMES Magic: The Gathering
FANTASY SPORTS LEAGUES
ARCADES "High Scores"
The Settlers Of Catan
PLAY BY E-MAIL GAMES
Magic: The Gathering Online
NETWORK GAMES Doom Diablo
BBS 'DOOR' GAMES Space Empire Trade Wars
MMORPGs Everquest (1999)
VIRTUAL ECONOMIES Second Life Eve Online
"CASUAL GAMES" Bejeweled (2001)
Gold Farmers
World of Warcraft
Secondary Virtual Goods Market IGE
WEB BROWSER GAMES Travian (2004)
achievement systems XBOX Live
Free-to-Play MMOs
SOCIAL STRATEGY Evony
SOCIAL SIM GAMES Farmtown
SOCIAL EXPERIENCES Music Pets
SOCIAL RPGS Mafia Wars
SOCIAL NETWORK GAMES
Source: http://Radoff.com

Abbildung B.1.: Die Vollbildgrafik zeigt ein Schaubild von Jon Radoff zur historischen Entwicklung sozialer Spiele.

nach [Radoff, 2010]

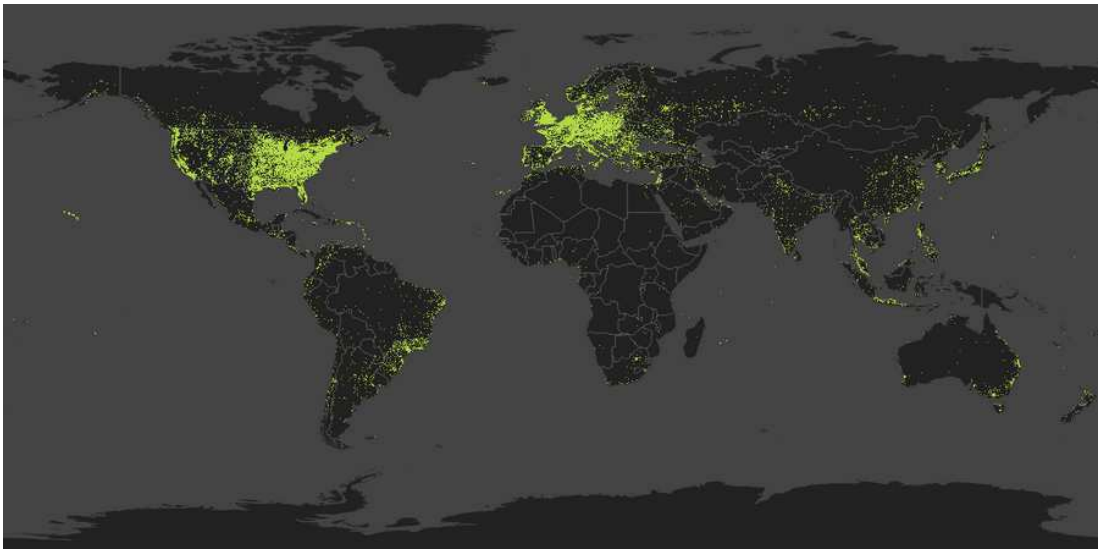


Abbildung B.2.: Anzahl der aktiven Spieler bei Steam am 01. November 2013 als Satellitenbild. Jeder Punkt repräsentiert dabei einen Nutzer oder eine Nutzerin, die in den letzten 24 Stunden online waren.

aus [Valve Corporation, 2013a]

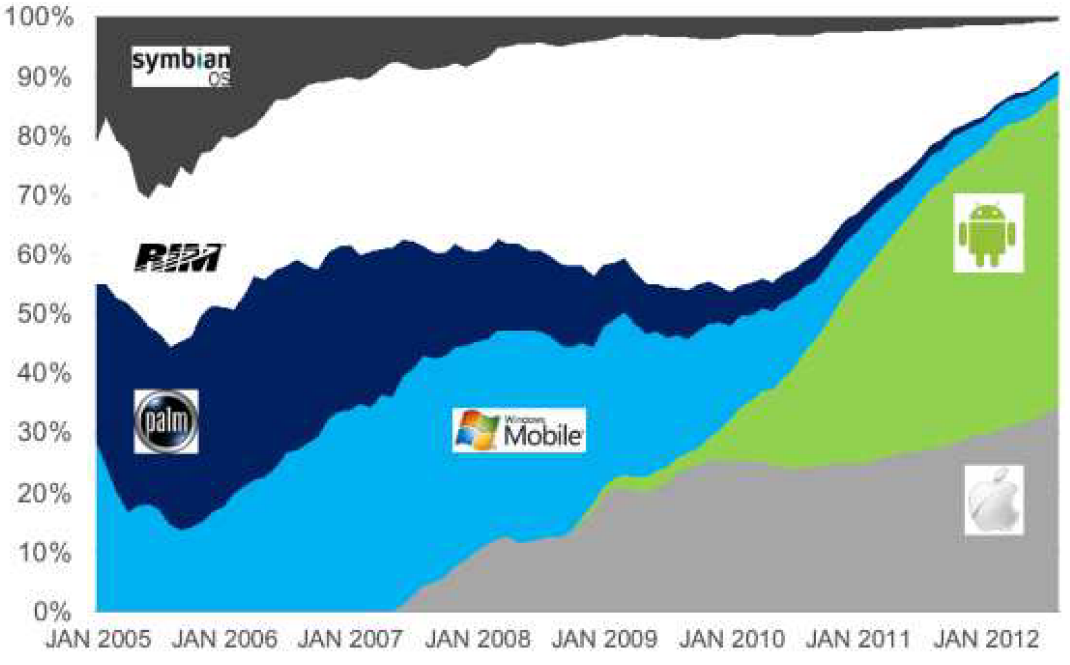


Abbildung B.3.: Eine Auswertung des Marktforschungsunternehmens comScore zeigt über einen Verlauf von sieben Jahren den stetigen Wachstum Androids.

aus [Philip Elmer-DeWitt, 2013]

Elements of a game definition	Parlett	Abt	Huizinga	Caillois	Suits	Crawford	Costikyan	Avedon u. SuttonSmith
Proceeds according to rules that limit players	X	X	X	X	X	X		X
Conflict or contest	X					X		X
Goal-oriented / outcome-oriented	X	X			X		X	X
Activity, process, or event		X			X			X
Involves decision-making		X				X	X	
Not serious and Absorbing			X					
Never associated with material gain			X	X				
Artificial / Safe / Outside ordinary life			X	X		X		
Creates special social groups			X					
Voluntary				X	X			X
Uncertain				X				
Make-believe / Representational				X		X		
Inefficient					X			
System of parts/Resources and Tokens						X	X	
A form of art							X	

Tabelle B.1.: Die Tabelle zeigt die Gemeinsamkeiten und Unterschiede verschiedener Definitionen von Spielen.

nach [Salen u. Zimmerman, 2003, S. 79]

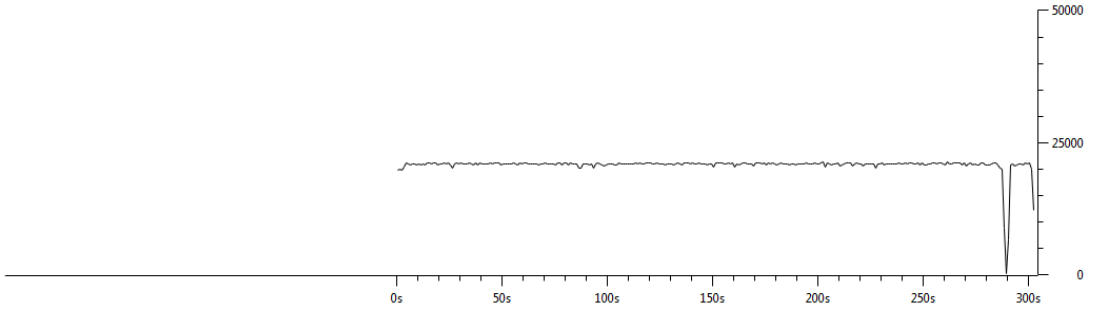


Abbildung B.4.: Kaum feststellbarer Jitter. Die Aussetzer am Ende sind nicht ausschließlich auf das Netzwerk zurückzuführen. Siehe Abschnitt 24.1.

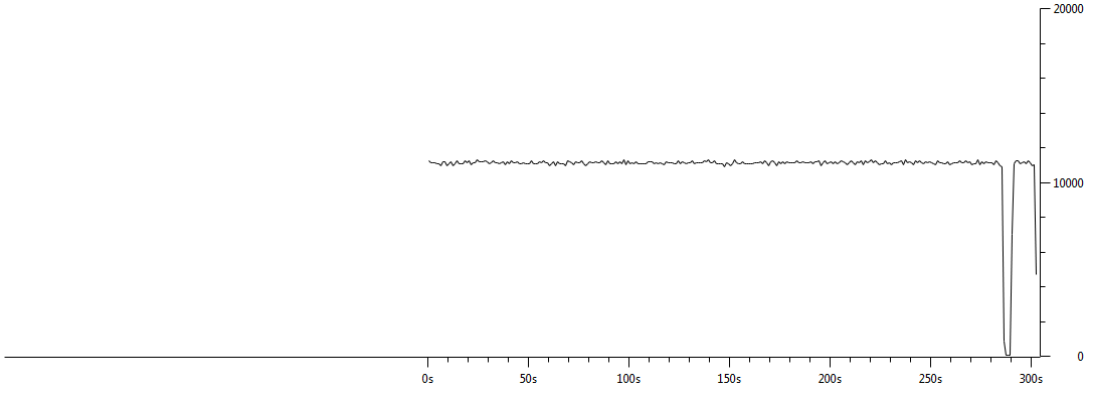


Abbildung B.5.: Kaum feststellbarer Jitter. Die Aussetzer am Ende sind nicht ausschließlich auf das Netzwerk zurückzuführen. Siehe Abschnitt 24.1.

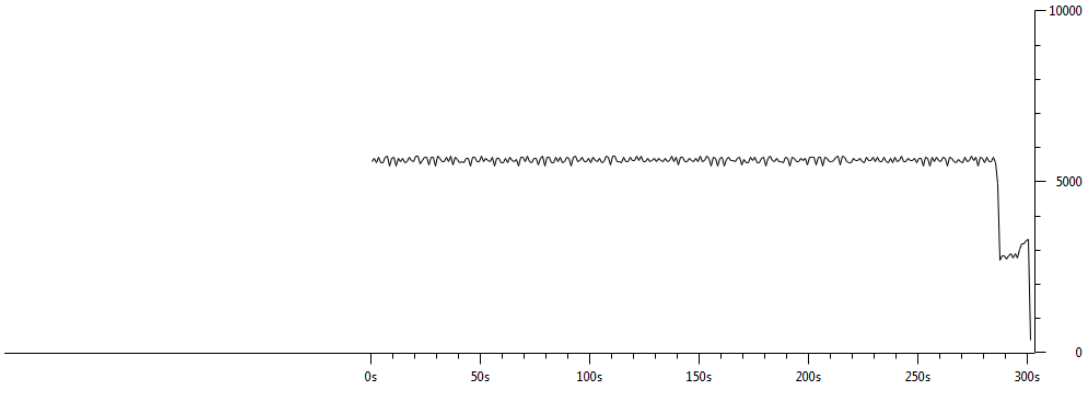
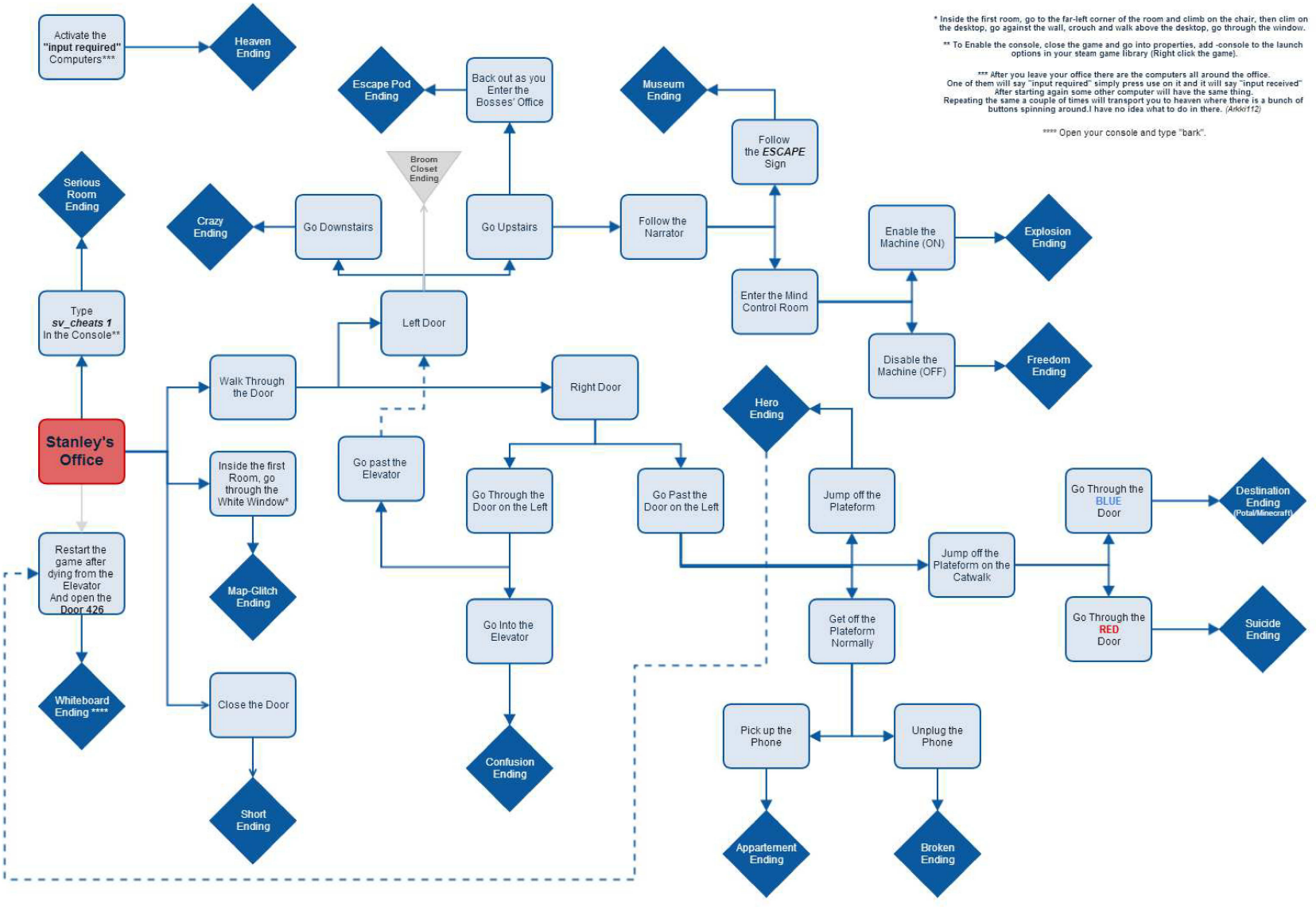


Abbildung B.6.: Kaum feststellbarer Jitter. Die Aussetzer am Ende sind nicht ausschließlich auf das Netzwerk zurückzuführen. Siehe Abschnitt 24.1.



* Inside the first room, go to the far-left corner of the room and climb on the chair, then clim on the desktop, go against the wall, crouch and walk above the desktop, go through the window.

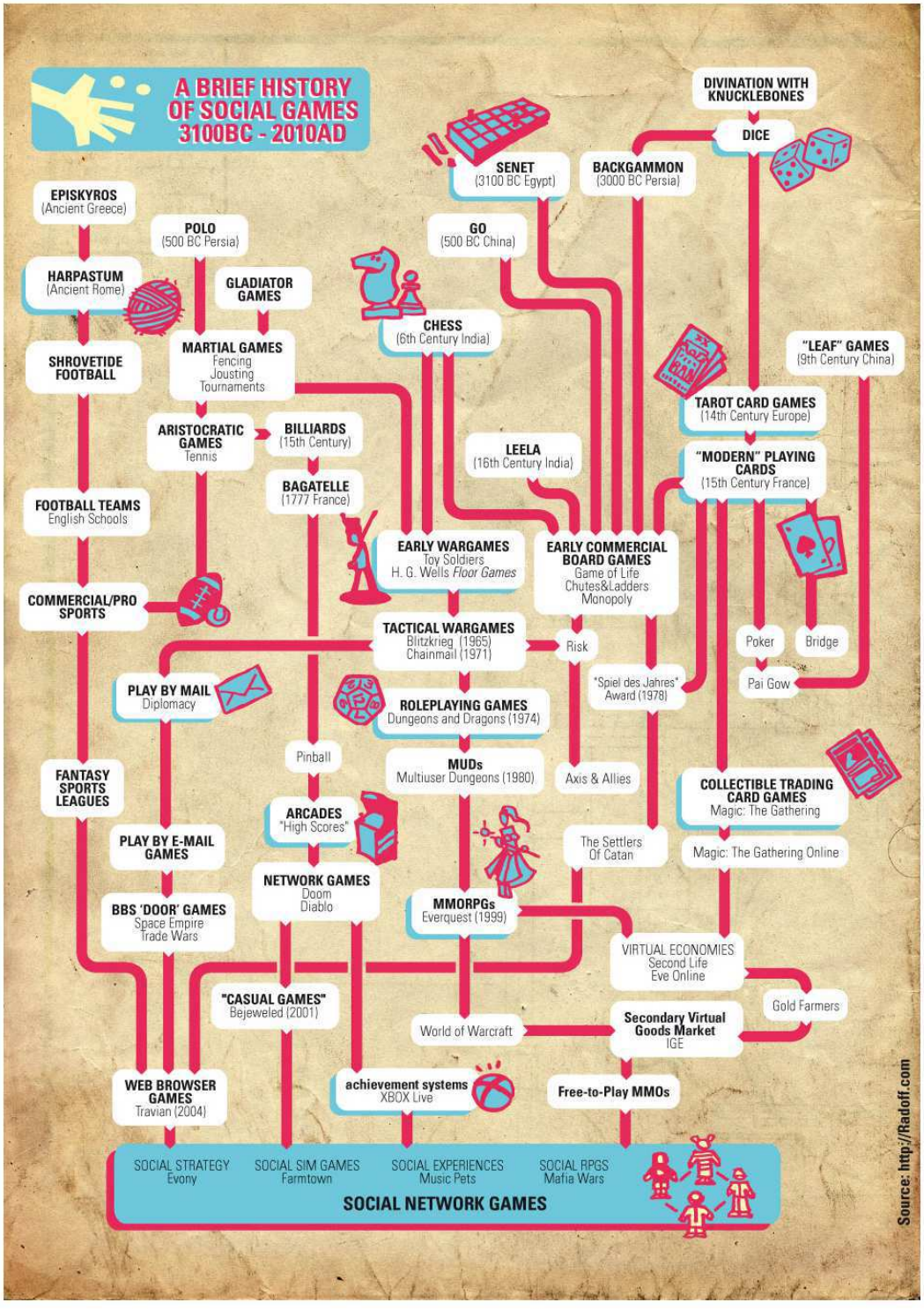
** To Enable the console, close the game and go into properties, add -console to the launch options in your steam game library (Right click the game).

*** After you leave your office there are the computers all around the office.
One of them will say "input required" simply press use on it and it will say "input received"
After starting again some other computer will have the same thing.
Repeating the same a couple of times will transport you to heaven where there is a bunch of buttons spinning around.I have no idea what to do in there. (Arkid112)

**** Open your console and type "bark".

Abbildung B.7.: Die Vollbildgrafik zeigt die Verzweigungen des Spiels *The Stanley Parable*. Es ist zu erkennen, dass das Spiel überwiegend aus simplen Verzweigungen besteht, obwohl beim Durchspielen der Eindruck erweckt wird, dass eine viel größere Entscheidungsfreiheit besteht.

aus [Sakarias411, 2013]



xxvi

Abbildung B.8.: Schaubild von Jon Radoff zur historischen Entwicklung sozialer Spiele.

Nach [Radoff, 2010].

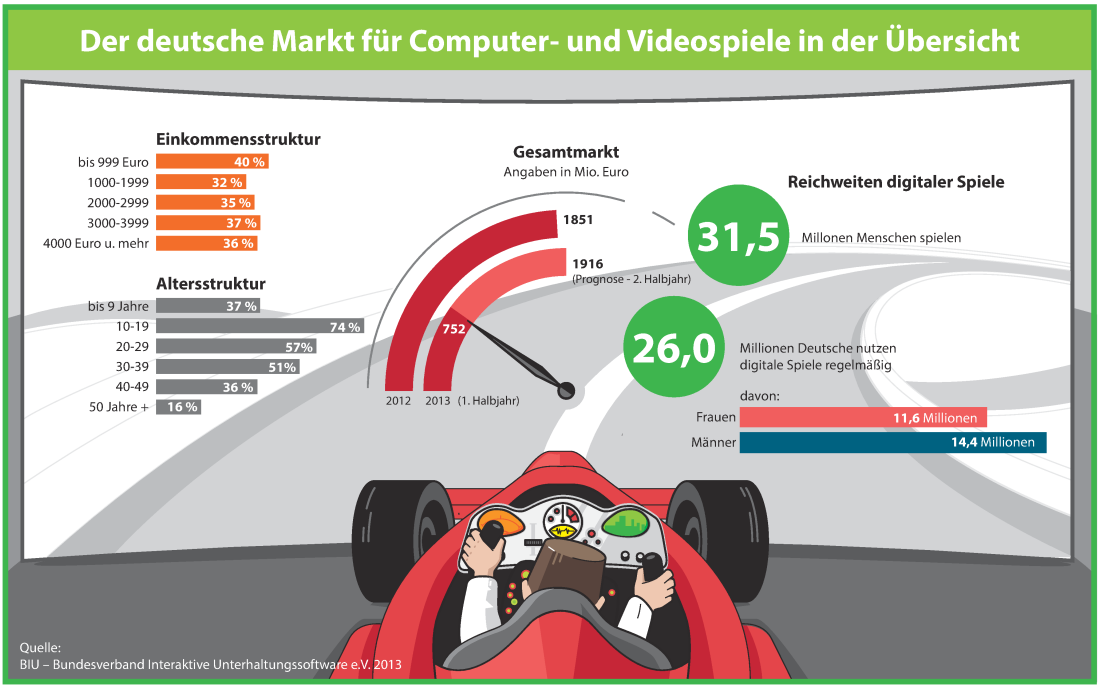


Abbildung B.9.: Der deutsche Markt für Computer- und Videospiele in der Übersicht.

Nach [Brehm, 2013].

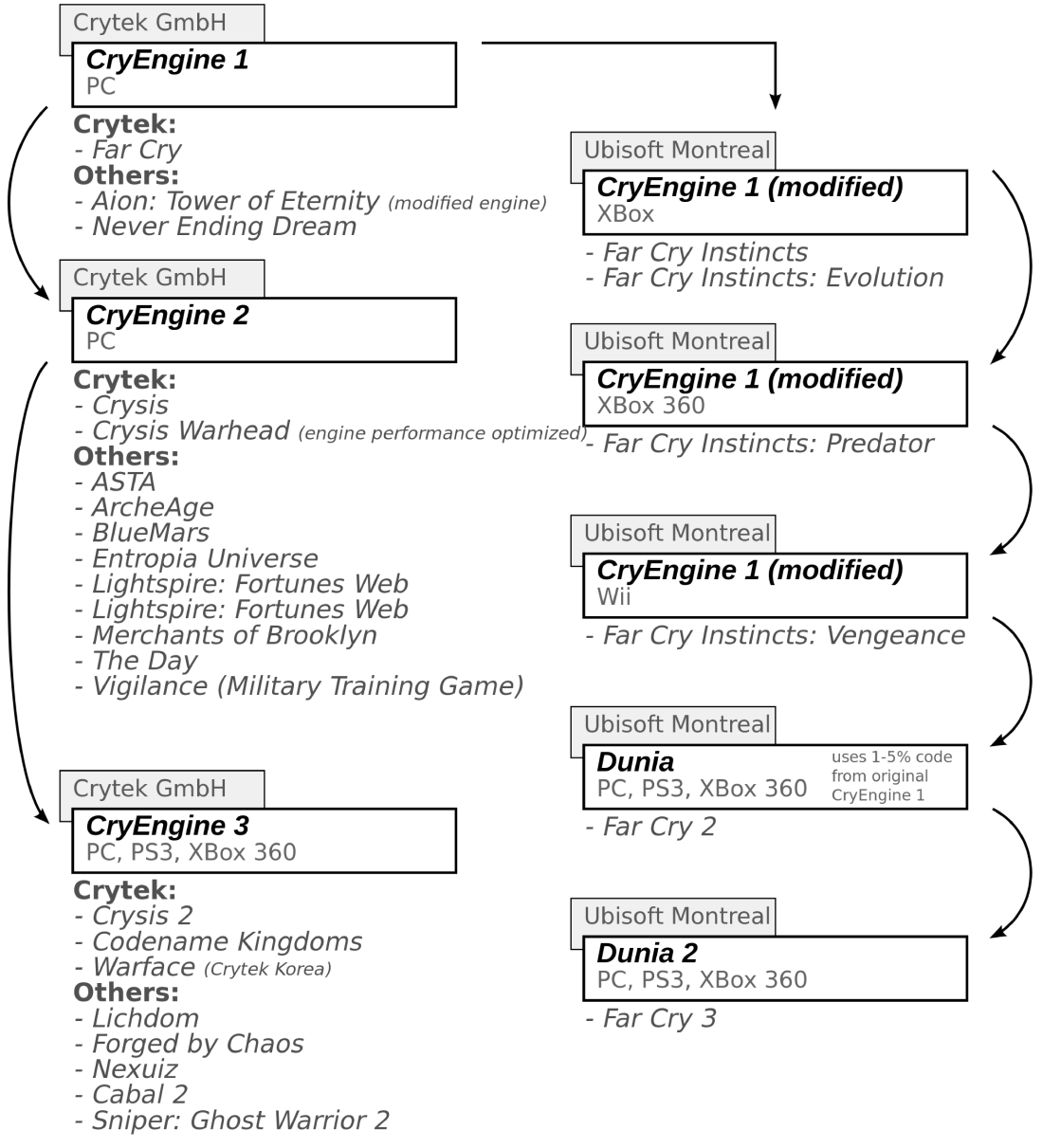


Abbildung B.10.: Diese Abbildung zeigt Spiele, welche mit der Cry Engine realisiert wurden oder mit dieser verwand sind.

Nach [DWTDmitri, 2011].

ANHANG C

Weiterführende Informationen

C.1. Arten von Pervasive Games

Smart Toys: Hierbei handelt es sich um ein physikalisch vorhandenes Spielzeug (vgl. Abbildung C.1), welches Beispielsweise auch an einen an einen PC angeschlossen werden kann. Eine dazugehörige Software berechnet dann anhand verschiedener Auswertungen Ereignisse, wie Bewegungen oder Geräusche.

Smart Toys können beispielsweise um ein Display erweitert werden, um zusätzliche Informationen anzuzeigen, oder Spielzeuge, die Geschichten erzählen und auf diese Weise mehr Interaktivität erzeugen.

Affective Games: Hier wird versucht, den emotionalen Zustand der Spielerin oder des Spielers in das Spiel mit einfließen zu lassen, um eine individuelle Spielerfahrung zu erschaffen. Die Schwierigkeit liegt hierbei in der verlässlichen Messung und Auswertung aussagekräftiger Signale, weswegen hauptsächlich einfach quantisierbare Informationen, wie Puls Hautleitfähigkeit gesetzt wird. Hauptsächlich vertreten sind hier derzeit Spiele im Bereich der Brain-Computer-Interface(s) (BCI), bei denen Spiele mit Hilfe von „Gedanken“ gesteuert werden können (vgl. [Soderlund, 2005] und Abbildung C.2).

Augmented Reality Games: Zu guter Letzt gibt es eben noch die reinen Augmented Reality Games. Hier kann unterschieden werden, ob die AR beispielsweise durch Head-Mounted Display(s) (HMD) hergestellt wird, durch visuelle Projektion auf reale Objekte oder durch einbezug mobile Geräte wie Smartphones [Magerkurth u. a., 2005, S. 11f.].

Abbildung C.1.: Ein „smartes“ Spielzeugschiff von *Zowie*

aus [Magerkurth u. a., 2005, S. 3]

Abbildung C.2.: Beim Spiel *Brainball* wird ein Ball mit Hilfe von EEG-Signalen kontrolliert.

aus [Magerkurth u. a., 2005, S. 5]

C.2. Das Projekt „Blinkenlights“

Das Projekt „Blinkenlights“ des Chaos Computer Club (CCC) ist ein Beispiel dafür, wie Partizipation und Kollaboration im Rahmen einer Mixed Reality-Anwendung auf eine moderne und öffentliche Stufe gehoben werden kann. Hierbei werden Interaktion, Virtualität, Partizipation und Kunst in einer neuen Art und Weise vermischt. Dazu wurde von September 2001 bis Februar 2002 eine Lichtanlage am „Haus des Lehrers“ in Berlin installiert, indem die oberen acht Stockwerke des Gebäudes mit insgesamt 144 Lampen hinter den Fenstern ausgestattet wurden. Auf diese Weise entstand ein interaktives Display mit einer Auflösung von 18 x 8 Pixeln. Neben der Möglichkeit der statischen Anzeige verschiedener Bilder und Animationen, die von jedem mittels der Software „Blinkenpaint“ erstellt und eingereicht werden konnten, bot das Projekt eine weitere Besonderheit: Menschen konnten zur Laufzeit des Projektes eine bestimme Rufnummer anrufen und mittels der Handytastaur eine Großprojektion des Spiels „Pong“ live an der Hauswand spielen (vgl. Abbildung C.3). [Sick, 2012, S. 106], [Projekt Blinkenlights, 2013]

(a) „Pong“ auf der Hauswand

(b) „Pong“ auf der Hauswand - Spielsituation mit Handy

Abbildung C.3.: Auf den Abbildungen C.3(a) und C.3(b) ist zu sehen, wie die Installation und die Adaption des Spiels „Pong“ auf der Hauswand realisiert wurde .

aus [Projekt Blinkenlights, 2013]

Nachwort zur Veröffentlichung

Dieses Werk entstand in den Jahres unseres Studiums und bis zur Veröffentlichung war es ein langer Weg auf dem uns viele Personen begleitet haben. An dieser Stelle danken wir nochmals ausdrücklich unseren Mentoren Herr Heiden und Herr Heisenberg, die uns bei der Veröffentlichung nicht nur unterstützt, sondern auch aktiv zu angeleitet- und begleitet haben. Wir danken allen Menschen, die uns in unserem Studium und während der Entstehungszeit dieser Arbeit (und darüber hinaus) Mut, Unterstützung und Wärme gaben.

Und wir danken Dir, liebe Leserin und lieber Leser.

Im Dezember 2014
Nicolas Kopp & Antony Konstantinidis

© 2013-2015
Nicolas Kopp & Antony Konstantinidis
www.thinklabs.de