

# Interaktivität und visuelles Feedback mittels Animationen in Computerspielen

JÜRGEN KOLLER

DIPLOMARBEIT

eingereicht am  
Fachhochschul-Masterstudiengang

DIGITAL ARTS

in Hagenberg

im September 2011

© Copyright 2011 Jürgen Koller

Diese Arbeit wird unter den Bedingungen der *Creative Commons Lizenz Namensnennung–NichtKommerziell–KeineBearbeitung Österreich* (CC BY-NC-ND) veröffentlicht – siehe <http://creativecommons.org/licenses/by-nc-nd/3.0/at/>.

# Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die aus anderen Quellen entnommenen Stellen als solche gekennzeichnet habe.

Hagenberg, am 22. September 2011

Jürgen Koller

# Inhaltsverzeichnis

|   |            |
|---|------------|
| <b>Erklärung</b>  | <b>iii</b> |
| <b>Kurzfassung</b>  | <b>vi</b>  |
| <b>Abstract</b>   | <b>vii</b> |
| <b>1 Einleitung</b>   | <b>1</b>   |
| 1.1 Motivation . . . . .  | 1          |
| 1.2 Aufbau der Arbeit . . . . .                                       | 2          |
| <b>2 Begriffe</b>   | <b>4</b>   |
| <b>3 Die Entwicklung von Charakter Animationen in Spielen</b>         | <b>12</b>  |
| 3.1 Character-Animationen in Film und Spiel . . . . .                 | 12         |
| 3.1.1 Unterschiede . . . . .  | 13         |
| 3.1.2 Animationsprinzipien für den Spielercharakter . . . . .         | 15         |
| 3.2 Kurze historische Übersicht . . . . .                             | 18         |
| 3.3 Verwendete Methoden und Technologien in der Branche . . . . .     | 24         |
| 3.4 Zusammenfassung . . . . .   | 31         |
| <b>4 Der Spielercharakter</b>   | <b>32</b>  |
| 4.1 Körpersprache als visuelle Feedbackform . . . . .                 | 32         |
| 4.1.1 Körpersprache als Form der Kommunikation . . . . .              | 33         |
| 4.1.2 Arten der Körpersprache . . . . .                               | 35         |
| 4.2 Feedback Animationen des Charakters im<br>Computerspiel . . . . . | 38         |
| 4.2.1 Anbringen der Körpersprache . . . . .                           | 38         |
| 4.2.2 Körpersprache in Computerspielen . . . . .                      | 40         |
| 4.3 Interaktion mit dem Spielercharakter . . . . .                    | 46         |
| 4.3.1 Das Mapping Modell . . . . .                                    | 47         |
| 4.3.2 Interface . . . . .   | 51         |
| 4.3.3 Der Spielercharakter als Interface . . . . .                    | 61         |
| 4.4 Zusammenfassung . . . . .   | 66         |

|  |            |
|--|------------|
| <b>5 Die Zustandsvermittlung des Spielercharakters</b> | <b>68</b>  |
| 5.1 Problemansatz . . . . .                            | 68         |
| 5.2 Fragestellung . . . . .                            | 74         |
| 5.3 Methode . . . . .                                  | 76         |
| <b>6 Projekt: Homunculus</b>                           | <b>79</b>  |
| 6.1 Concept . . . . .                                  | 79         |
| 6.2 Der Charakter . . . . .                            | 80         |
| 6.3 Die drei Zustände . . . . .                        | 82         |
| 6.4 Die Steuerung . . . . .                            | 87         |
| 6.5 Die virtuelle Welt . . . . .                       | 88         |
| <b>7 Schlussfolgerung</b>                              | <b>91</b>  |
| 7.1 Konsequenzen . . . . .                             | 91         |
| 7.2 Ausblick . . . . .                                 | 93         |
| <b>A Informationen und Hilfestellung zu UDK</b>        | <b>95</b>  |
| A.1 Dokumentationen und Tutorials . . . . .            | 95         |
| A.2 Animationen und Blendshapes in UDK . . . . .       | 96         |
| A.3 Cloth . . . . .                                    | 97         |
| A.4 Rigging . . . . .                                  | 97         |
| A.5 3rd Person Camera . . . . .                        | 98         |
| A.6 Input Customization . . . . .                      | 98         |
| A.7 Programming . . . . .                              | 99         |
| <b>B Input Prototyp</b>                                | <b>100</b> |
| <b>C Animation Middleware</b>                          | <b>101</b> |
| <b>D Inhalt der DVD</b>                                | <b>103</b> |
| D.1 Masterarbeit . . . . .                             | 103        |
| D.2 Literatur . . . . .                                | 103        |
| D.3 Projekt . . . . .                                  | 103        |
| D.4 Input Prototype . . . . .                          | 104        |
| <b>Quellenverzeichnis</b>                              | <b>105</b> |
| Literatur . . . . .                                    | 105        |
| Online-Quellen . . . . .                               | 107        |

# Kurzfassung

Die vorliegende Arbeit thematisiert die Anwendung von Character-Animationen in Computerspielen, um gewisse körperliche Zustände des Spielercharakters, ohne den Einsatz von virtuellen Interface Elementen, dem Spieler zu vermitteln.

Anfangs werden die Unterschiede zwischen Character-Animationen in Film und Spielen erläutert. Des Weiteren wird eine kurze Übersicht über die angewandten Methoden und Technologien für interaktive Animationen, sowie auch über Spiele, welche die Entwicklung der Character-Animationen geprägt haben, gegeben.

Der Hauptfokus liegt jedoch bei dem Spielercharakter. Die Körpersprache als visuelle Feedbackform wird auf deren Anwendung in Computerspielen, insbesondere am Spielercharakter, ermittelt. Der Prozess der Interaktion zwischen Spieler und Spielercharakter wird anhand eines Modells beschrieben. Die einzelnen Schichten dieses Modells, allen voran die physische und virtuelle Interface-Schicht, werden auf ihre Aufgabe bei der Informationsvermittlung durchleuchtet. Infolgedessen werden alternative Wege der Verlagerung dieser Informationen in Erwägung gezogen. Anschließend wird die Vermittlung körperlicher Zustände anhand des Spielercharakters konkret mit drei Beispielszuständen analysiert.

# Abstract

This thesis addresses the use of character animations in computer games as a way to convey to the player the different body states of the player character, without the usage of virtual interface elements.

To begin with, a short illustration of the differences between character animation in film and games will be presented. Subsequently, a brief overview of methods and technologies used for interactive animations, as well as the games which promoted the development of character animations, will be given.

The central focus of this thesis is aimed at the player character. The application of body language as a form of visual feedback in computer games, especially via the player character, will be investigated. The process of interaction between the player and the game will be described with the aid of a formal model. The different layers of this model, in particular the physical and virtual interface layer, will be examined in respect to their intended purpose and function in the information relaying process. Consequently, other ways of information placement and transmission will be considered. Following this, the conveyance of body states by means of the player character will be analysed with three specific body state examples.

# Kapitel 1

## Einleitung

### 1.1 Motivation

Der Prozess der Interaktion zwischen Spieler und Computerspiel ist das wichtigste Alleinstellungsmerkmal des Spielemediums. Die Geschehnisse im Spiel können nur durch die aktive Entscheidung und Handlung des Spielers, sowie dem Liefern von Feedback von Seiten des Spieles, fortgeführt werden. Zwar laufen im Hintergrund Prozesse wie Zeitmesser und Simulationen ab, welche dem Regelwerk oder der Lebendigkeit der Spielwelt dienen, der fortschreitende Erfahrungswert für den Spieler jedoch bleibt aus.

Seit der Entwicklung der ersten Computerspiele war der Spielercharakter ein wichtiger Bestandteil der Spiel-Erfahrung. Durch ihn ist es dem Spieler erlaubt, mit der Spielwelt zu interagieren. Diese Interaktion wird durch ein Eingabegerät ermöglicht, welches durch die Belegung von Aktionen auf verschiedene Tasten dem Spielercharakter diverse Angaben und Befehle vermittelt. Abhängig vom Mapping der Tasten (siehe Abschnitt 4.3.1) liefert der Spielercharakter Feedback in der Form von Zustandsänderungen seines Selbst. Solche Änderungen können u.a. den Positions- und Rotationswechsel beinhalten. Aus Gründen der Effizienz oder Einfachheit wird Feedback vom Spiel auch in der Form von Elementen am virtuellen Interface (siehe Abschnitt 4.3.2) dem Spieler geliefert. Hier bilden dann Zahlen oder 2-D Texturen den u.a. geänderten Zustand des Spielercharakters ab.

Ältere Spiele aus den Anfangszeiten der Spieleentwicklung hatten eine einfache, abstrakte und symbolische Darstellung ihres Spielercharakters. Dies ist infolge von Designentscheidungen auch heutzutage in diversen modernen Spielen der Fall. Aufgrund solcher Limitierung mussten ältere Spiele Informationen des Spielercharakters am HUD<sup>1</sup> darstellen lassen, damit Spieler über den Zustand ihres Spielercharakters klare Informationen erhalten. Erst durch die Entwicklung von modernen Technologien und erhöhter Rechenleistung ist eine komplexere Darstellung möglich geworden. Vor allem die Darstel-

---

<sup>1</sup>Heads-Up Display

lung von anthropomorphen Kreaturen hat von dieser Entwicklung profitiert. Körperbewegungen und Gesten aus dem echten Leben können mit einer höheren Genauigkeit und Detailtreue wiedergegeben werden. Komplexe und detailreiche Animationen des Spielercharakters erlauben es, dem Spieler ein besseres Feedback seiner Aktionen zu geben. Ab diesem Zeitpunkt stellt sich dann die Frage, inwiefern gewisse Zustandsänderungen, meist rein historisch bedingt am virtuellen Interface dargestellt, sich anhand der Animationen des Spielercharakters darstellen lassen würden. Die durch die Eingabebefehle des Spielers ausgelösten Zustände wie Laufen, Rennen oder Springen werden schon seit Anbeginn der Computerspiele dargestellt. Erst bei Zuständen wie die Gesundheit oder Geschicklichkeit, welche auf direkte oder indirekte Art durch die virtuelle Umgebung beeinflusst oder ausgelöst werden, gibt es noch wenige Umsetzungen. Je nach Genre oder Design wäre der Vorteil einer solchen Umsetzung die gesteigerte Immersion des Spielers, da dieser nicht durch virtuelle Interface Elemente außerhalb der Spielewelt abgelenkt wird. Da der Fokus des Spielers, zumindest bei Third-Person Spielen, auf dem Spielercharakter liegt, wäre dies auch ein geeigneter Ort, Informationen an den Spieler zu vermitteln.

Die Frage ist somit, wie effektiv solche Zustände vom Spielercharakter, ohne virtuelle Interface Elemente, dem Spieler anhand von Animationen vermittelt werden können.

## 1.2 Aufbau der Arbeit

Kapitel 2 erklärt die in dieser Arbeit wichtigsten Begriffe wie etwa den *Spielercharakter* oder die *Interaktivität*. Dieses Kapitel dient auch zugleich als eine Ansammlung an Schlagwörtern, welche eine Übersicht über die wichtigsten Themen in dieser Arbeit liefert.

In Kapitel 3 wird über die Entwicklung von Character-Animationen in Spielen berichtet. Hier werden die wichtigsten Unterschiede zwischen Animationen in Filmen und Spielen aufgezeigt, sowie auch eine mögliche Anpassung der Animationsprinzipien für Spiele dargestellt. Eine kurze, historische Übersicht zeigt die wichtigsten Spiele, welche die Character-Animation in diesem Medium vorangetrieben haben. Abschließend werden diverse, in der Branche verwendete Methoden und Technologien der interaktiven Animation vorgestellt.

Kapitel 4 widmet sich dem Spielercharakter. Hier wird die Körpersprache als eine visuelle Feedbackform der Kommunikation zwischen Spieler und Spielercharakter begutachtet. Diverse Arten der Körpersprache werden erwähnt und ihre Anwendung in Spielen betrachtet. Deren Anwendung unterscheidet sich primär zwischen NPCs<sup>2</sup> und dem Spielercharakter. Der Unterschied zwischen diesen beiden Typen an Charakteren wird beschrieben.

---

<sup>2</sup>Non-Player Characters

Grundsätzlich wird jedoch näher auf den Spielercharakter und weiter auf die Interaktion zwischen ihm und dem Spieler eingegangen darauf eingehen. Diesbezüglich wird das Mapping Modell von Jesse Schell in modifizierter Form herangezogen und erläutert. Allen voran wird die physische und virtuelle Interface-Schicht von diesem Modell auf ihren Beitrag zur Immersion des Spielers und der Unterstützung von Animationen des Spielercharakters analysiert. Des Weiteren wird auf die Formen der Verlagerung von Interface Elementen auf den Spielercharakter und die Spielewelt eingegangen.

Kapitel 5 setzt sich mit der Vermittlung von Zuständen des Spielercharakters, in der Form von Animationen, an den Spieler auseinander. Es werden drei Zustände analysiert und ihre Anwendung am Spielercharakter erläutert.

Kapitel 6 beschreibt die Entstehung des Spieleprojektes *Homunculus*, und wie die drei Zustände in diesem Spiel angewandt wurden.

Kapitel 7 fasst alle bisherigen Informationen zusammen und bietet einen Ausblick über die mögliche weitere Entwicklung von Animationen des Spielercharakters.

Anhang A besteht aus einer Liste, mitsamt Beschreibungen, von URL Adressen, welche auf Informationen hinweisen die bei der Entwicklung des Projektes sehr nützlich gewesen sind. Drei Input Prototypen, welche das Mapping der Controller Tasten für den Geschicklichkeitszustand des Spielercharakters getestet haben, werden in Anhang B kurz vorgestellt. Anhang C führt eine kleine Liste an bekannten Animation Middleware Lösungen auf. Zum Abschluss wird in Anhang D der Inhalt der beiliegenden DVD dargestellt.

## Kapitel 2

# Begriffe

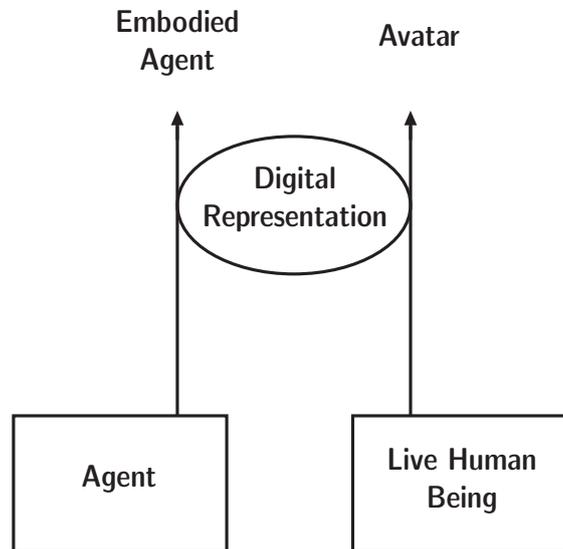
In Folge werden die wichtigsten in der Forschungsfrage vorkommenden Begriffe erläutert, um eine klare Ausgangsposition für den Rest dieser Arbeit zu schaffen.

**Character-Animation:** Die *Character-Animation* ist ein Spezialgebiet im Animationsbereich, welche sich mit der Bewegung, Gedankenwelt und emotionalen Darstellung von figürlichen Charakteren befasst [79]. Im Rahmen dieser Arbeit beschränkt sich dieses Gebiet auf den Bereich der Computerspiele (siehe Abschnitt 3.1) und umfasst den digitalen, anthropomorphen Spielercharakter, welcher in einem 3D Programm modelliert und mit einem Skelett versehen worden ist. Der Begriff der Animation stammt vom lateinischen<sup>1</sup> *animo* (von *anima* – „beleben, in etwas Lebendes verwandeln“) und *animus* („Leben, Geist, Seele“) und lässt erkennen, dass durch die Animation eines Charakters dieser erst „zum Leben erweckt wird“. Erst wenn ein digitaler Charakter menschenähnliche Verhaltensmuster darstellt, erfährt man etwas über dessen Persönlichkeit und Gefühlszustand, womit eine Verbindung oder Beziehung zwischen dem Charakter und dem Spieler aufgebaut werden kann.

**Spielercharakter:** Als *Spielercharakter* ist der vom Spieler steuerbare digitale, anthropomorphe Charakter gemeint. Meist wird dieser auch als *Avatar* bezeichnet und repräsentiert die digitale Erscheinungsform des Spielers in der virtuellen Welt, welche in Echtzeit durch den Spieler gesteuert werden kann [1, Seite 2]. Im Zusammenhang mit Avatar fällt auch manchmal der Begriff *Embodied Agent* (siehe Abb. 2.1). Ein *Embodied Agent* ist eine digitale Erscheinungsform in einer virtuellen Welt, welche die Befehle eines *Agenten* (eine mathematische Formel oder Funktion) ausführt [1, Seite 3]. Ein NPC wäre ein Beispiel eines *Embodied Agents*. Anhand des Spielercharakters wird dem Spieler erlaubt, mit der virtuellen Welt zu interagieren

---

<sup>1</sup><http://de.pons.eu/>



**Abbildung 2.1:** Eine schematische Darstellung von *Avatars* und *Embodied Agents*. Bildquelle: [1, Seite 3].

(siehe Abschnitt 4.3).

Katherine Isbister [8, Seite 212] beschreibt drei Bezeichnungen wie man Spielercharaktere im Spiel repräsentieren kann: *Tools*, *Puppets* und *Masks*. Die *Tools* Bezeichnung beschreibt Spiele, in welchen der Spieler keine virtuelle Bezugsperson braucht, um das Spiel genießen zu können (als Beispiele werden *Warcraft 3* und *Dance Dance Revolution* angegeben). Die Interaktion mit dem Spiel geschieht direkt mit dem Interface, ohne mit einem virtuellen Charakter kommunizieren zu müssen. Anders ist es bei den *Puppets* – dort wird der Spielercharakter anhand seiner Bewegung, visuellen Eigenschaft und Interaktion mit NPCs definiert [8, Seite 215]:

„These characters are a joy to manipulate and provide gratifying reactions to player success. Their social qualities take the form of personality as expressed in movement rather than in actual engagement with other characters during game play.“

*Link* von *The Legend of Zelda: Ocarina of Time* und *Mario* in *Super Mario Bros.* sind zwei Beispiele von *Puppet* Charakteren. Zuletzt gibt es die *Masks* Spielercharaktere, welche in Spielen mit einer bedeutenden sozialen Komponente vorkommen, wie es typischerweise bei MMOGs<sup>2</sup> der Fall

<sup>2</sup>Massively Multitplayer Online Game



**Abbildung 2.2:** First-Person Perspektive (*Mirrors Edge*), Third-Person Perspektive (*Super Mario 64*), Top-Down Perspektive (*Alien Swarm*), Side-Scrolling Perspektive (*Braid*), Isometrische Sicht (*Fallout 2*), Preset Kamera (*Resident Evil*).

ist. Hier überwiegt die soziale Interaktion mit anderen Spielern, wobei jeder Spieler mit seinem Spielercharakter alternative Persona erkunden und mit diesen auftreten kann [9, Kapitel: Masks and Trance].

**Kamera Perspektive:** Eine für den Körper des Spielercharakters wichtige Berücksichtigung ist dessen Sichtbarkeit für den Spieler. Der Spieler muss Einblick in die virtuelle Welt bekommen, damit dieser sich anhand des Spielercharakters orientieren kann. Gleichzeitig liefert der Spielercharakter dem Spieler Feedback zu dessen Lage und Zustand. Diese Information wird zu einem relativen Grad über den Körper des Spielercharakters kommuniziert, weshalb die Perspektive oder Sicht der Kamera ein entscheidender Faktor für die Immersion des Spielers ist.

Traditionell gibt es die folgenden Kameraansichten für Computerspiele: *First-Person Perspektive*, *Third-Person Perspektive*, *Top-Down Perspektive*, *Side-Scrolling Perspektive*, *Isometrische Sicht*, *Preset Kamera* und die *freie Kamera* (siehe Abb. 2.2). Sie bieten verschiedene Darstellungsformen und unterschiedliche Sichtbarkeitseinschränkungen für den Körper des Spielercharakters an. Dies kann zur Folge haben, dass nicht alle Bereiche des Körpers verwendet werden können, um dem Spieler Feedback zu geben. Die Top-Down Perspektive, in Bezug auf den Körper, ist die eingeschränkteste Sicht, da von oben nur ein limitierter Körperflächenbereich als Feedbackmittel zur Verfügung gestellt wird. In der First-Person Perspektive ist der Grad der Immersion am wenigsten ausgeprägt, obwohl im ersten Moment genau das Gegenteil behauptet werden kann.

Brandon Sheffield [19] argumentiert, dass Spieler einen Bezugsrahmen



Abbildung 2.3: *Mirrors Edge* und *Aliens vs. Predator (2010)*.

brauchen, um in ein Spiel eintauchen zu können. Ein Spieler oder Zuschauer, wie im Film, identifiziert sich mit einem Charakter den er sieht und nimmt selten die Rolle dieses Charakters ein. Zwar entwickeln sich immer mehr Ansätze, um den Immersiongrad in Spielen mit einer First-Person Perspektive zu steigern (z.B. durch stärkere Animation der Kamera oder durch Zeigen von animierten Körperflächen wie etwa Hände und Beine in *Mirrors Edge* oder *Aliens vs. Predator (2010)* (siehe Abb. 2.3)), jedoch bleibt bei all diesen Ansätzen der Großteil des Körpers versteckt. Von daher sind Kameraansichten wie die isometrische Sicht, die Side-Scrolling Perspektive und die Third-Person Perspektive die bessere Wahl bezüglich der Vermittlung von Charakter Feedback via dem Körper. Chaim Gingold [6, Seite 84] schreibt, dass erst mit *Super Mario 64* die erfolgreiche Abkoppelung der Erkenntnisperspektive von der First-Person in die Third-Person Perspektive gelungen ist (humorvoll metaphorisch in Form von *Lakito* mit einer Kamera auf einer Wolke fliegend) und somit die Entwicklung weiterer Spiele geprägt und beeinflusst hat.

Die Third-Person Perspektive erlaubt neben der Betrachtung des Kör-

pers auch die genaue Lage des Spielercharakters zu erkennen. Durch die Platzierung der Kamera und des Spielercharakters sieht man meist nur den Rücken des Charakters. Bei einer drehbaren Kamera oder einer Drehung des Charakters (unabhängig von der Kamera) kann man den ganzen Körper betrachten. Auf diese Weise ist es möglich, den Körper uneingeschränkt als Feedbackkanal zu verwenden.

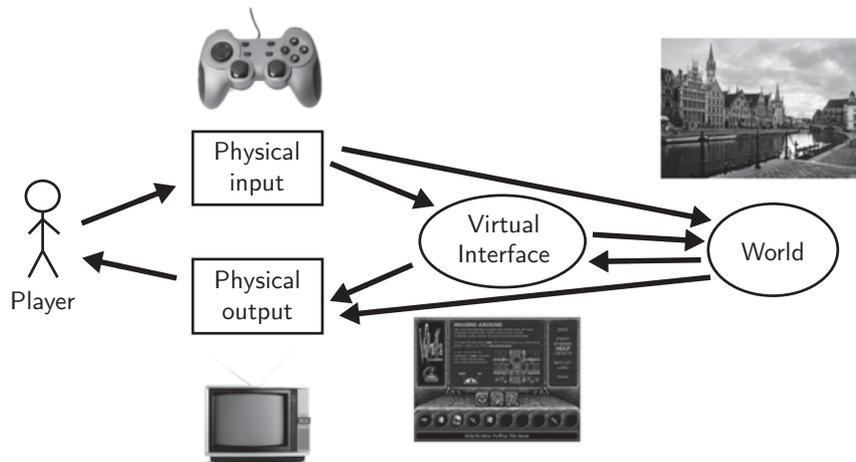
Im Rahmen dieser Arbeit ist der Spielercharakter ein *Puppet* und wird in der typischen Third-Person Perspektive dargestellt. Eine für den Spieler steuerbare Kamera erlaubt es, den Spielercharakter aus verschiedenen Perspektiven zu begutachten – der zentrale Rotationspunkt der Kamera liegt jedoch beim Spielercharakter. Auf diese Art und Weise bildet der Spielercharakter somit das Zentrum der Aufmerksamkeit für den Spieler und erlaubt es, mit diesem eine Beziehung aufzubauen [61]:

‘The more I know of a character, the more I am able to become him.’

**Interaktivität:** Der implizierte, in Echtzeit stattfindende, Informationsaustausch und die Wechselbeziehung zwischen dem Spieler und dem Spiel wird als *Interaktivität* bezeichnet (siehe Kapitel 4.3). Dieser Informationsaustausch ist das wichtigste Unterscheidungsmerkmal zum Film. Im Gegensatz zum linearen Handlungsablauf eines Filmes kann der Spieler, in seiner eigenen Zeit, die Entscheidung treffen, wann und wie eine Handlung oder ein Geschehen stattfinden soll. Um diese Entscheidungsmacht bewerkstelligen zu können, muss der Spieler mit dem Spiel kommunizieren können. Dies geschieht in der Form eines Eingabegerätes, welches versucht, menschliche Gedankengänge zu abstrahieren und metaphorisch in Form von Tasten und Analog-Sticks darzustellen. Anhand einer solchen Steuerung kann ein Spieler seine Gedankenvorgänge, in eingeschränkter Weise, in das Spiel auf seinen Spielercharakter übertragen, und diesen somit in der virtuellen Welt steuern und Befehle erteilen. In dieser Arbeit wird als Eingabegerät ein Gamecontroller (spezifischer, ein Xbox 360 Controller) in Betracht gezogen.

Unter dem Begriff *Mapping* (siehe Abb. 2.4) werden die Daten vom Gamecontroller, in modifizierter Form, auf das Spiel und dessen virtuelle Schnittstelle übertragen [18, Seite 224]. William Lidwell et al. [12] definiert Mapping als die Beziehung zwischen der Steuerung und dessen Ausführungen oder Auswirkungen. Ein gutes Mapping erfüllt die Erwartungen des Spielers, wenn die Auswirkungen durch die Interaktion mit dem Gamecontroller für den Spieler vorhersehbar sind (siehe Abschnitt 4.3).

**Feedback:** Damit der Spieler weiß, ob seine Steuerungseingaben etwas bewirkt haben, muss das Spiel im Gegenzug *Feedback* liefern (siehe Abschnitt 4.2). Feedback erlaubt dem Spieler Entscheidungen zu treffen, welche



**Abbildung 2.4:** Das *Mapping* vom Informationsfluss zwischen verschiedenen Schichten. Bildquelle: [18, Seite 225].

Aktionen er als nächstes ausführen soll. Dies geschieht entweder durch direktes Feedback von der Spielewelt selbst (z.B. durch Drücken der Taste A erkennt der Spieler, dass sein Spielercharakter gesprungen ist), oder durch Feedback vom virtuellen Interface (z.B. eine statisch-platzierte Lebensanzeige am rechten Bildschirmrand aktualisiert einen Zahlenwert, nachdem der Spielercharakter des Spielers Kampfschaden genommen hat).

**Körperbewegung, Gestik und Zustand:** Da der Mensch nur fünf Sinnesorgane hat, sind die Möglichkeiten an Feedback-Kanälen grundsätzlich eingeschränkt. Für Computerspiele sind derzeit nur der visuelle, auditive und haptische Kanal von Bedeutung, wobei der visuelle Kanal in dieser Arbeit einen höheren Stellenwert einnimmt. Es ist durch diesen Kanal, dass der Spieler die Körperbewegung und Gestik seines Spielercharakters wahrnimmt (siehe Abschnitt 4.1). Als Menschen sind wir naturgebunden darauf konditioniert, täglich mit Körpersprache umzugehen. Im Gegensatz zur mündlichen ist die Körpersprache grundsätzlich eine universellere Art der Kommunikation. Während die Sprache sich spezifischer ausdrücken lässt, stößt sie länderübergreifend schnell an ihre Grenzen. In Fällen von sprachlichen Verständigungsproblemen werden Gesten verwendet, um Angelegenheiten dem Gegenüber zu vermitteln. Kulturspezifische Gesten erschweren die Kommunikation, jedoch existieren Gesten und Körperbewegungen, welche kulturübergreifend funktionieren [14], wie zum Beispiel ein Lächeln oder eine Verteidigungshaltung. Von daher ist die Anwendung von Körperbewegung und Gestik auf Charaktere in Spielen eine optimale Möglichkeit und Gelegenheit, um mit dem Spieler kommunizieren zu können. Die technologische Entwicklung be-



**Abbildung 2.5:** Wander sein erschöpfter Zustand in *Shadow of the Colossus*, und der Prinz von *Prince of Persia (2008)* beim Klettern.

züglich der effektiven Darstellung von Character-Animationen in Spielen hat sich in den letzten Jahren beschleunigt, wodurch die Technologie immer weniger zum limitierenden Faktor geworden ist (siehe Abschnitt 3.3).

Ein wichtiger Aspekt der zwischenmenschlichen Kommunikation ist die Körperbewegung. Sie verstärkt und verdeutlicht das Gesagte, liefert aber auch Informationen über eine Person durch dessen Körperhaltung, Körperorientierung und räumliches Verhalten [10, Seite 4]. Die Gestik ist eine spezielle Form der Körperbewegung, in welcher Gedankenvorgänge und Emotionen primär durch Hand-, Kopf- und Haltungsausdruck vermittelt werden. Die Gestik wird im Allgemeinen auch als *observed action* bezeichnet [14, Seite 21], da die empfangenen Signale von größerer Bedeutung sind als jene, welche wir glauben ausgeschickt zu haben. Im Falle des Spielercharakters bedeutet dies Animationen zu erstellen, welche klare Körpersignale an den Spieler liefern. Dem Spieler muss klar dargestellt werden, in welchem Zustand sich der Spielercharakter befindet (siehe Kapitel 5) – sei es ein erschöpfter Zustand in welchem der Spielercharakter keuchend seinen Körper nach vorne beugt, oder ein Geschicklichkeitszustand in welchem der Spielercharakter mit flinken Griffen eine Wand hinauf klettert (siehe Abb. 2.5).

Im Rahmen dieser Arbeit wird die Körperbewegung und Gestik des Spielercharakters verwendet, um dem Spieler auf diese Art Feedback bezüglich des Geschicklichkeits-, Erfahrungs- und physischen Gesundheitszustandes zu vermitteln. Auf das Gesicht und die Mimik wird bewusst verzichtet, da dies eine eigene Ausarbeitung für sich beanspruchen würde. Aufgrund der Third-Person Perspektive ergibt sich die Einschränkung, dass der Spieler meist den Rücken des Spielercharakters sieht. Von daher wird gezielt darauf geachtet, dass Körperbewegung und Gestik aus dieser Sicht funktionieren und vermittelt werden können. Die Arme und Hände spielen dabei eine wichtige Rolle – zumal sie, seit „der Befreiung von ihrer antiken Fortbewegungsrolle“ [14, Seite 24] zu einem wichtigen Instrument für die Vermittlung von Körper-

signalen geworden sind. Im Kontext der Third-Person Perspektive werden diese „freien oberen Extremitäten“ größtenteils nicht vom Rumpf verdeckt und eignen sich daher sehr für die Feedback Vermittlung.

Zusammenfassend wurden die wichtigsten Begriffe erläutert und deren Definitionen im Rahmen dieser Arbeit klargestellt. Begriffe wie beispielsweise *Spielercharakter* oder die *Kamera Perspektive* haben verschiedene Definitionen in diversen Spiele Genres und müssen im Rahmen dieser Arbeit durch ihren Bedeutungsbereich spezifiziert werden. Auch können die aufgelisteten Begriffe als Schlagwörter für die vorkommenden Themen in dieser Arbeit gesehen werden und somit eine kontextuelle Übersicht bieten.

## Kapitel 3

# Die Entwicklung von Charakter Animationen in Spielen

Die moderne Character-Animation hat ihren Ursprung im Film. *Walt Disney Studios* haben durch ihre Zeichentrickfilme ein großes Publikum angesprochen und somit die Character-Animation zu einem hohen Bekanntheitsgrad verholfen. Im gleichen Studio wurden auch die ersten Schritte der Formulierung von Prinzipien der Character-Animation erstellt, welche auch in heutigen modernen Produktionen unentbehrlich sind.

Computerspiele sind, im Gegensatz zum Film und Animationsfilm, ein relativ junges Medium. Character-Animationen in Spielen müssen mit mehreren Einschränkungen arbeiten welche es im Film nicht gibt, allen voran der Interaktivität welches das Medium Computerspiele ausmacht. Die Animationsprinzipien gelten auch im interaktiven Medium, jedoch sind es die Übergangsanimationen zwischen verschiedenen Eingabebefehlen, welche die größte Herausforderung darstellt. Die Geschichte der Animation in Computerspielen ist eine Dokumentation der technischen Einschränkungen und Unerfahrenheit in diesem relativ jungen Medium. Erst mit der Entwicklung neuer Technologien, leistungsstarker Rechner und Aneignung an Erfahrungen haben sich die Character-Animationen immer mehr hervorgehoben. Moderne Spiele setzen verstärkt auf Character-Animationen, um Charaktere glaubwürdiger zu machen. Auch die Industrie entwickelt immer neue Animationstechnologien, um dem interaktiven Medium gerecht zu werden.

In Folge wird auf die Entwicklung von Character-Animationen in Spielen eingegangen. Dieses Kapitel erhebt nicht den Anspruch eine ausführliche Dokumentation zu sein. Eher wird auf die wichtigsten Aspekte hingewiesen, um ein grundlegendes Verständnis zu diesem Thema zu entwickeln.

### 3.1 Character-Animationen in Film und Spiel

,It's important to know that when we are animating characters

around, we are animating shapes. Therefore what we do with those shapes, how we position them on screen, what angle they have compositionally and how those angles play off each other, will affect whatever it is we are trying to say.’ [37]

Character-Animationen im Film wie auch in Spielen haben das Ziel, dem Zuschauer und Spieler eine Vorführung von menschlichen Körperbewegungen, Gestiken, Ausdrücken, Emotionen und Gedanken zu geben – anders ausgedrückt, die Persönlichkeit des Charakters zu vermitteln. Aufgrund der täglichen Interaktion und Kommunikation mit anderen Menschen und dem unbewussten Lesen derer Körpersprache ist der Mensch dazu fähig, die Motivation eines zum Leben erweckten und personifizierten, digitalen Körpers zu erkennen. Um überzeugend wirken zu können, muss dieser digitale Charakter die Feinheiten und Nuancen der Körpersprache darstellen und diese dem Zuschauer oder Spieler vermitteln können. Das „Show, don’t tell“ Prinzip [40] im Bereich des kreativen Schreibens findet hier Anwendung, da durch die Handlungen des Charakters die Identifikation mit diesem erleichtert wird.

### 3.1.1 Unterschiede

‘It’s hard to deny that the sensation of controlling a digital object is one of the most powerful — and overlooked — phenomena ever to emerge from the intersection of people and computers.’ [72]

Obwohl Character-Animationen im Film sowie auch in Spielen das Ziel der Informationsvermittlung verfolgen, gibt es wesentliche Unterschiede in ihrer Durchführung. Die zwei wichtigsten Merkmale sind die Interaktivität und das in Szene setzen des Spielercharakters durch die Kamera.

Im Film hat die Character-Animation den Vorteil der kontrollierten Inszenierung in der Szene. Die Bewegung der Kamera ist vorgegeben, detaillierte Charakter Modelle und komplexe Rigs erlauben eine feine Regulierung der Körperbewegungen und das Rendern der digitalen Szenen erstellt Schnappschüsse, welche für den Zuschauer nicht aktiv modifizierbar sind.

In Computerspielen muss auf Grund der Interaktivität das Geschehen in Echtzeit berechnet werden. Vorhandene Rechenleistung und technologische Einschränkungen bestimmen daher den Grad des Details von Charakter Modellen, die Komplexität verwendeter Rigs, die Vielzahl an Animation Clips und die Verwendung von Animationstechnologien in Echtzeit Umgebungen. Erst in den letzten Jahren hat die Character-Animation aufgrund von steigender Rechenleistung und Entwicklung neuer Technologien ein auffallend hohes, qualitatives Niveau erreicht. Der Spielercharakter und die Kamera lassen sich meist vom Spieler direkt oder indirekt steuern, eine gezielte Inszenierung des Charakters in der virtuellen Welt ist daher nur eingeschränkt möglich. Der Spielercharakter ist von allen Seiten aus betrachtbar, die Darstellung seines Körpers und dessen Animationen müssen von allen Kamera-

winkeln aus funktionieren, um dem Spieler Feedback liefern zu können. Demzufolge werden meist übertriebene Gesten oder herkömmliche Standardposen verwendet [22, Seite 35].

Für die Character-Animation in Computerspielen ist es wichtig im Rahmen der Interaktivität und dessen Gegebenheiten zu arbeiten (z.B. fertige Animation Clips in einer Echtzeit Umgebung mit verschiedenen Kamera Winkeln begutachten, im Gegensatz zur Prüfung von vorgerenderten Playblasts [22, Seite 35]). Die Interaktivität ist das ausschlaggebende Merkmal von Computerspielen – Elemente, welche diese ausmachen, sollen den Vorrang in der Entwicklung bekommen [47]. Noah Falstein [40] sieht im Allgemeinen den Grund des Scheiterns vieler interaktiven Produkte bei der Vernachlässigung und dem halbherzigen, nachträglichen Anhängen von interaktiven Merkmalen, statt diesen Priorität in der Entwicklungsphase zu gewähren. Mike Jungbluth [47] weist auf die Wichtigkeit der Interaktion vom Zuschauer oder Spieler mit dem Spielercharakter hin, und dass das Animieren im interaktiven Medium im Gegensatz zur traditionellen Film Animation eine andere Denkweise verlangt. Er erwähnt folgende zwei Bereiche, welche dem Spieler erlauben, mit dem Spielercharakter zu kommunizieren und welche für den Animator beeinflussbar sind: Die *Transitional Animations* und *Player Input*. Laut Jungbluth sind es diese beiden Momente, in welchen der Spieler mit dem Spielercharakter „redet“ und der Spieler dies geradezu „fühlen“ kann wenn die Umsetzung erfolgreich war.

**Transitional Animations:** *Transitional Animations* werden abgespielt, wenn der Charakter eine Entscheidung getroffen hat [47]. Beim Spielercharakter ist diese Entscheidung u.a. der Moment, in dem der Spieler sich für eine neue Aktion entschieden hat und dies anhand seiner Steuerungseingabe mitteilt. Der Spielercharakter muss auf diese Änderung reagieren, indem er einen Übergang von seiner derzeitigen Bewegung in eine neue Animation erstellt. Das Denken und Entscheidungen treffen treibt den Charakter voran und sind die Quintessenz der Character-Animation [47]. Diese Übergangsanimationen, welche durch die Interaktion entstehen, sind für Spiele eines der wichtigsten Elemente, um die Persönlichkeit, Ziele und den Zustand des Spielercharakters dem Spieler zu vermitteln. Sie bieten auch die Gelegenheit an, den Spielercharakter mit *Anticipation* zu versehen [47]:

„Transitions are pure anticipation of what is about to happen next, so relish in it!“

**Player Input:** Anhand des Gamecontrollers wird es dem Spieler erlaubt, mit dem Spiel bzw. seinem Spielercharakter zu kommunizieren. Der Spieler entscheidet sich für eine Aktion und drückt dabei den relevanten Knopf, um diese auszulösen. Der Entscheidungsvorgang bis zum Knopfdruck beschreibt

eine gewisse Anticipation, eine Erwartung, die der Spieler für sich selbst erfüllt. Wird diese Erwartung anhand der Animation vom Spielercharakter erfüllt, kann der Spieler das Resultat seiner Erwartungen genießen. Auch das Mapping, die Auslegung der Tasten und das haptische Feedback des Gamecontrollers beeinflussen die Erwartungen und den Gemütszustand des Spielers [47] (siehe Abschnitt 4.3).

**Empathie:** Im Rahmen dieser Arbeit soll der Unterschied der Empathie in Film und Spielen anhand von Expertenmeinungen im Bereich des Spielercharakters erwähnt werden. Zahlreiche Literatur ist vorhanden, welche sich mit dem Thema Empathie beschäftigt – eine ausführliche Auseinandersetzung mit dem Thema würde den Rahmen dieser Arbeit sprengen. Die *Encyclopedia Britannica* [25] definiert den Begriff Empathie folgendermaßen:

‚Empathy [is] the ability to imagine oneself in another’s place and understand the other’s feelings, desires, ideas, and actions. [...]The most obvious example, perhaps, is that of the actor or singer who genuinely feels the part he is performing.‘

In seinem Buch *The Art of Game Design* [18, Seite 124] sieht Jesse Schell in der Empathie ein Mittel zur interaktiven Problemlösung in Spielen, ergänzend zum Einsatz von Empathie im Filmbereich. Durch das Projizieren eigener Gefühle und dem eigenen Entscheidungsvermögen auf den Spielercharakter werden dem Spieler die Grenzen und Möglichkeiten seines Spielercharakters ersichtlich, und erlauben ihm eine bessere Entscheidung, wie er mit diesem Probleme lösen kann.

Ed Hooks [7, Seite 103] hingegen argumentiert, dass Empathie nur durch physische Distanz zustande kommt. Da der Spieler selbst im virtuellen Medium durch den Spielercharakter vertreten ist, ist es nicht möglich, die gleiche Art von Empathie zu erzeugen, wie es im Film oder Theater mit Schauspielern der Fall ist. Der Schauspieler oder digitale Charakter muss autonom in seiner Handlung bleiben. Da jedoch der Spieler emotional in das Spiel einbezogen werden soll, muss Empathie auf eine andere Art erstellt werden. Hooks [7, Seite 104] sieht hier die Entwicklung in Richtung *secondary characters* gehen – NPCs welche mit dem Spielercharakter interagieren und ihn begleiten. Sie erwecken im Spieler u.a. Sympathie und Zuneigung, wodurch langsam eine Beziehung aufgebaut wird. Ein oft zitiertes Beispiel ist der Charakter *Alyx Vance* in *Half-Life 2*, im späteren Spielverlauf ein konstanter Begleiter des Spielercharakters.

### 3.1.2 Animationsprinzipien für den Spielercharakter

‚The driving force behind the action is the mood, the personality, the attitude of the character — or all three. Therefore, the mind

is the pilot. We think of things before the body does them.’ —  
Walt Disney [64]

Das 1981 erschienene Buch *The Illusion of Life*, auch als die „Animator Bibel“ bezeichnet, führt eine Liste von zwölf Methoden und Prinzipien an, welche Disney für ihre Animationen verwenden, um ihre Charaktere lebendig wirken zu lassen [23, Seite 47]. Diese zwölf Prinzipien der Animation sind die Grundlage für die Erstellung von glaubwürdigen Bewegungen und sind auch heute noch in Zeiten von 3-D Animationen relevant. In Folge werden die zwölf Prinzipien auf spezielle Anforderungen für den Spielercharakter, im Rahmen einer virtuellen Echtzeit Umgebung, ergründet.

**Squash and Stretch:** Durch Quetschen und Strecken des Körpers wird dessen Gewicht und Volumen dargestellt. Wichtig dabei ist das Erhalten des Gesamtvolumens. Das visuelle Vermitteln von Gewicht ist eines der wichtigsten Aspekte der Animation, weshalb *Squash and Stretch* grundsätzlich in allen Bewegungen des Spielercharakters vorkommen soll. Zur Betonung von markanten Squash and Stretch Momenten, wie das Fallen und am Boden Aufkommen des Spielercharakters, können haptische Mittel wie das Vibrieren des Gamecontrollers eingesetzt werden, um das Gefühl des Aufkommens zu verstärken.

**Anticipation:** Das Ausholen einer Bewegung leitet die Aufmerksamkeit des Spielers und erlaubt ihm sich auf die kommende Aktion vorzubereiten. Neben Squash and Stretch ist es eines der wichtigsten Prinzipien, da *Anticipation* auch einer Vorbereitung zur Gewichtsverlagerung nahe kommt. Komplikationen treten für den steuerbaren Spielercharakter auf, da Anticipation meist in den Transitional Animations (siehe weiter oben) vorkommt und dort auch die Responsiveness des Spielercharakters eine Rolle spielt. Auf diesen Konflikt wird näher im Abschnitt 4.3.2 eingegangen, grundsätzlich ist es jedoch ein iteratives Verfahren, um den richtigen Ausgleich zwischen Anticipation und Responsiveness zu finden. Wie auch weiter oben beim Player Input erwähnt, findet eine gewisse Anticipation schon beim Spieler durch Drücken der Eingabetasten statt. Das Einbeziehen des Spielers macht Anticipation zu einem wichtigen Prinzip für Spiele. Anticipation muss nicht immer in Transitional Animations stattfinden – das Ausstrecken einer Hand des Spielercharakters beim Näherrücken an eine Wand wäre ein Beispiel dafür.

**Staging:** Bei der Inszenierung des Spielercharakters ist es wichtig, dessen Pose, Geste oder Körperbewegung unmissverständlich erkennbar darstellen zu lassen. Die Umgebung oder andere Bewegungen dürfen nicht davon ablenken. Anhand der Silhouette des Charakters muss klar sein was dieser

vermitteln möchte. Computerspiele mit einer Third-Person Perspektive haben hier Schwierigkeiten, da der Spieler die Kontrolle über die Kamera hat und somit den Spielercharakter und die Umgebung von allen Seiten aus betrachten kann. Im Gegensatz zum Animationsfilm muss hier verstärkt darauf geachtet werden, dass die Animationen von mehreren Perspektiven aus funktionieren. Auch der Detailgrad der Umgebung und die passende Animation des Spielercharakters müssen harmonieren, um sich die Aufmerksamkeit des Spielers zu teilen.

**Follow Through and Overlapping Action:** *Follow Through Actions* bezeichnen Teile eines Körpers (Arme, Haare) welche nach Vollendung der Hauptbewegung zeitversetzt ihre Bewegungen abschließen. Cloth, Hair und andere Physik-basierte Simulationen erlauben es in Echtzeit prozedural-generierte Follow Through Animationen zu erstellen. Handgefertigte Follow Through Animationen lassen sich in Transitional Animations realisieren, um mehr Kontrolle in der Zeitversetzung zu bekommen. *Overlapping Actions* sind Animationen von Körperteilen, welche sich zu unterschiedlichen Geschwindigkeiten bewegen. Dies kann dazu verwendet werden, am Ende einer Bewegung einen flüssigen Übergang in eine neue Bewegung (und deren Anticipation) einzuleiten. Zusätzlich zu den erwähnten Methoden bei den Follow Through Actions können bei Overlapping Actions Blending Methoden wie beispielsweise *Additive Blending* oder *IK*<sup>1</sup> Techniken verwendet werden, um im interaktiven Rahmen Übergänge zwischen den Animation Clips zu schaffen.

**Slow In and Slow Out:** Aufgrund der Schwerkraft und dem Gewicht eines Körpers muss dieser Energie aufwenden, um eine Beschleunigung zustande zu bringen. Das Gegenteil ist der Fall bei der Verlangsamung; dort muss die Geschwindigkeit des Körpers reduziert werden. Diese Beschleunigung und Verlangsamung widerspiegelt den realen Prozess des Transports der Masse eines natürlichen Körpers von einem Punkt zum anderen. Für die Animation des Spielercharakters bedeutet dies anhand von Blending verschiedene Übergänge für die Beschleunigung und Verlangsamung des Körpers zu gestalten. Die Responsiveness des Spielercharakters soll soweit angepasst werden, um das richtige Gefühl von Widerstand des Körpers zu erzeugen.

**Exaggeration:** Das *Exaggeration* Prinzip wird eingesetzt um Posen und Körperbewegungen zu verstärken und verdeutlichen, jedoch darf die Übertreibung dabei nicht zu oft vorkommen und theatralisch wirken. Da der Spieler meistens die Kontrolle über die Kamera im Spiel hat, werden Gesten und Körperbewegungen übertrieben, damit diese auch aus verschiedenen Kameraperspektiven erkennbar sind.

---

<sup>1</sup>Inverse Kinematic

**Solid Drawing:** In seinem Ursprung bezieht sich dieses Prinzip auf den Zeichentrickfilm, dessen grundlegende Bedeutung ist auch für digitale Charaktere relevant: den Spielercharakter kennen und verstehen lernen, um dessen Verhalten und Persönlichkeit durch Animationen darstellen zu können. Hier gilt es sich in den Charakter hineinzuversetzen, um ein besseres Verständnis für ihn aufzubauen.

**Appeal:** Das Design vom Charakter, sein Aussehen und seine Bewegung müssen beim Spieler Anklang finden und ihm gefallen. Die inneren Merkmale des Charakters müssen interessant sein, egal ob gut, böse oder abstoßend.

*Straight Ahead & Pose to Pose, Arcs, Secondary Action* und *Timing* sind wichtige Prinzipien für die Animation des Spielercharakters, haben jedoch keine spezielle für die Spielercharakter Animation herausstechende Eigenheiten. Eine ausführliche Erklärung zu diesen, wie auch den bereits erwähnten Prinzipien lassen sich im Buch *The Illusion of Life* [23] ab Seite 47 nachlesen.

## 3.2 Kurze historische Übersicht

Das Gebiet der Character-Animation hat eine verhältnismäßig kurze Entwicklungsgeschichte im Gegensatz zu anderen Entwicklungen der Menschheit. Ursprünglich entstammt sie dem Animationsfilm, welcher mit *Gertie the Dinosaur* [23, Seite 22] 1914 den ersten wahren Charakter mit einer ansprechenden Persönlichkeit, in einem Zeichentrickfilm, hervorgebracht hat. In späterer Folge gründete Walt Disney 1923 das *Disney Brothers Cartoon Studio* (heute *The Walt Disney Company*), mit speziellem Fokus auf das Gebiet der Character-Animation [23]. Eines der wichtigsten Erkenntnisse bezüglich Animationen ist aus dem Wissens- und Erfahrungsschatz der Animatoren bei Walt Disney entstanden: die zwölf Prinzipien der Animation. Diese Prinzipien sind nicht nur im Animationsfilm, sondern auch in leicht modifizierter Variante in Spielen anwendbar.

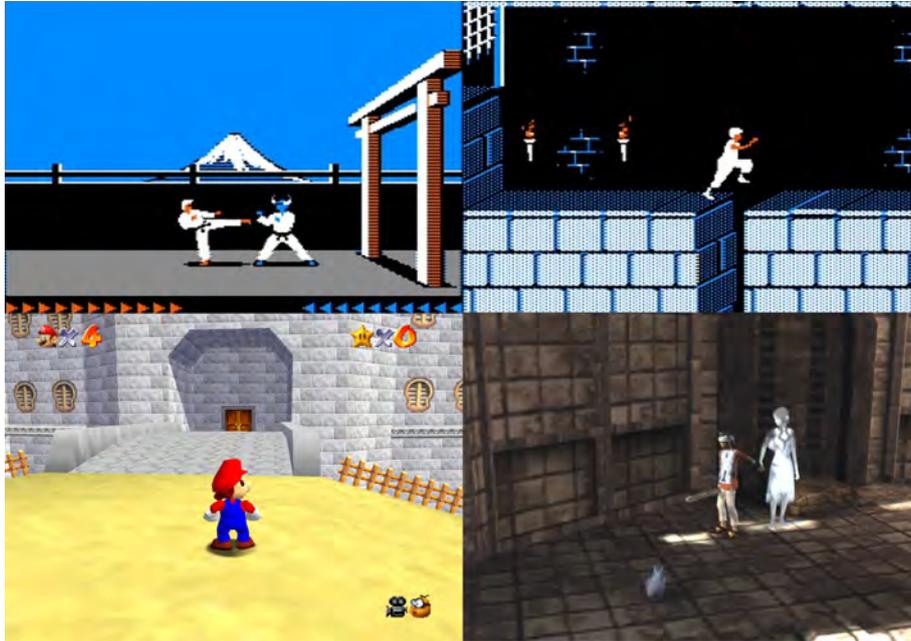
Im Gegensatz zum Animationsfilm ist die Geschichte von Character-Animationen in Spielen viel kürzer. Vermutlich ist der Bedarf an realistischen Character-Animationen in Spielen gewachsen nachdem ersichtlich wurde, welche Möglichkeiten des Ausdrucks digitale Charaktere im Film erreicht haben. Diese Möglichkeiten sollen auch in Spielen Einzug finden, ihre Wichtigkeit fasst Christiaan Moleman gut zusammen [58]:

„Animation can tell us about our characters, tell us who they are, tell us what they’re thinking, delivering what may be crucial gameplay information. It can do these things without breaking the illusion of being in the game world.“

Das Manipulieren eines Spielercharakters in Echtzeit, mit dem Ziel, eine überzeugende Körperbewegung zu produzieren, verlangt eine gewisse Technologiereife, genug Rechenleistung und ausreichend Arbeitsspeicher [48, 50, 83]. Zu Anfangszeiten waren die Rechenleistung und der Arbeitsspeicher die wesentlichste Einschränkung für Echtzeit-Animationen. Arbeitstechniken aus der Film Branche konnten nicht ohne weiteres in Spielen verwendet werden, es fehlte an Tools und Spiele-spezifischen Technologien.

**Karateka und Prince of Persia:** Erste bekannte Versuche, um Charakter-Animationen in Spielen zu realisieren, wurden von Jordan Mechner 1984 mit dem Spiel *Karateka* (siehe Abb. 3.1) vorangetrieben. Mechner wollte die von ihm gelernten Stummfilm Techniken verwenden um ein visuell anspruchsvolles, aber dennoch zugängliches Spiel zu entwickeln [51]. Das Resultat war laut Mechner ein Spielercharakter „mit flüssigeren und lebensechten Charakter Animationen wie es kein Spiel zuvor gehabt hat“ [51]. Animationsprinzipien wie *Anticipation* und *Follow through* sind vorhanden, selbst eine Verbeugungsgeste kann auf Knopfdruck initiiert werden, welche der Feind vor dem Kampf erwidert. Wie auch in *Prince of Persia* (1989) [52] (siehe Abb. 3.1), dem zweiten (und wohl bekanntesten) Spiel von Jordan Mechner, verwendet er das Rotoskopie-Verfahren, um dem Spielercharakter eine realistische Bewegung zu verschaffen [67, Seite 2]. Die Animationen des Prinzen sind um einiges komplexer und vielzähliger als bei *Karateka*, da dieser neben dem Schwertkampf u.a. auch springen und klettern kann. Anhand der Körperbewegungen und Steuerungen des Prinzen erkennt man seine Geschicklichkeit, auch wenn die Latenzzeit wegen der Anticipation um ein paar Millisekunden spürbar länger ausfällt.

**Super Mario 64:** Ein weiteres bekanntes und wichtiges Spiel, welches die Spielercharakter Animation vorangetrieben hat, ist *Super Mario 64* (siehe Abb. 3.1). Aufgrund dessen Bekanntheitsgrades gilt es vielerorts als Begründer des 3-D Jump 'n' Runs. Der Übergang von 2-D in eine 3-D Welt hat neue Herausforderungen mit sich gebracht, zumal der Spielercharakter mit einer Third-Person Kamera nun von allen Seiten aus anschaulich ist. Die Körperbewegung und Körpersprache von Mario ist jedoch immer klar erkennbar: er geht, schleicht, rennt, springt und haut, kann sich bücken und an gewissen Stellen festhalten. Ein abrupter Richtungswechsel wird mit einer klaren Umdrehungsanimation kommuniziert (siehe Abb. 3.2), sowie ein abruptes Losrennen vom Ruhezustand mit einem hektischen Losstrampeln klar angedeutet. Interessanterweise baut sich der Spieler seine eigene Anticipation Animation mit dem Weitsprung und Rückwärtssalto von Mario auf: beide fordern kurz vorm Sprung eine Duck-Aktion vom Spieler, was einer Anticipation nahe kommt (siehe Abb. 3.2). Ab einer gewissen Anzahl an gesammelten Schadenspunkten zeigt Mario dem Spieler im Ruhezustand sei-



**Abbildung 3.1:** *Karateka* (1984), *Prince of Persia* (1989), *Super Mario 64*, *Ico*.



**Abbildung 3.2:** Erste Reihe: Ein abrupter Richtungswechsel. Zweite Reihe: Ein Weitsprung mit Anticipation. Spiel: *Super Mario 64*.

nen angeschlagenen Gesundheitszustand. Verbringt Mario eine längere Zeit im Ruhezustand, setzt er sich hin und macht ein Nickerchen.

**Ico:** Die Spiele von *Team Ico*, unter der Leitung von Fumito Ueda, sind bekannt für ihren Schwerpunkt auf hochentwickelten Charakter-Animationen. *Ico* (2001, siehe Abb. 3.1) baut auf die Beziehung zwischen einem Jungen

(Ico) und einem Mädchen (Yorda) auf [62]. Aus diesem Grund wandte Ueda das Prinzip des *Subtractive Designs* [62] an, um allein diese Beziehung und deren unterstützende Elemente im Spiel hervorzuheben. Von daher stehen die Animationen beider Charaktere im Vordergrund, und Elemente wie ein HUD wurden vom Spiel gestrichen<sup>2</sup>. Was *Ico* auszeichnet ist die Interaktionsmechanik zwischen Ico und Yorda: Damit Yorda dem Spieler folgt, kann man ihre Hand nehmen und sie führen [56]:

„The tactile marriage of animation and interaction in this visually arresting mechanic creates a very real bond between our two protagonists, and you, the player.“

Um den delikaten Akt des Händehaltens zu betonen, vibriert der Gamecontroller jedes Mal wenn der Spieler seine Begleiterin zu ruckartig oder schnell mit sich zieht – als ob er ihr dabei wehtun würde [59]. Mit der Zeit wächst das Vertrauen von Yorda gegenüber Ico, und die Geschwindigkeit ihrer Reaktionen steigt an [59].

**Shadow of the Colossus:** In *Shadow of the Colossus* (siehe Abb. 3.3) ist es die Beziehung zwischen dem jungen Krieger Wanda und den Kolossen, welche den Hauptteil des Spieles ausmachen. Wie in *Ico* liegt auch hier das Augenmerk auf den hochwertigen Animationen vom Spielercharakter und dessen Interaktion und Berührung mit der Außenwelt [55], allen voran den beweglichen Kolossen, welche es zu erklimmen gilt. Aufgrund dieser unüblichen Art der Interaktion war es die Entscheidung von *Team Ico*, „eindrucksvolle, natürliche und schöne Animationen zu erstellen, welche aber nicht zwangsläufig realistisch sind“ [69, T=00:09:00]. Um diesem Ziel gerecht zu werden, mussten neue Animationstechniken (z.B. IK Systeme) angewendet werden. Diese sprechen u.a. das *sliding feet* Problem aller Charaktere und die dynamischen Bewegungen, die den Spielercharakter wie ein Pendulum beim Hinaufklettern der Kolosse hin und her schwingen lassen, an [69, T=00:11:50]. Ein weiteres markantes Animationsmerkmal des Spielercharakters ist das bewusste Einsetzen von Anticipation in Bewegungen um Animationen cinematischer zu gestalten [54].

**Assassin’s Creed und Prince of Persia:** Die Körperbewegungen der Spielercharaktere und deren Interaktion mit der Welt in *Assassin’s Creed* (siehe Abb. 3.3), dessen Nachfolger *Assassin’s Creed 2* und die *Prince of Persia* Spiele seit 2008, allesamt entwickelt von *Ubisoft Montreal*, gelten als weitere Meilensteine in der Entwicklung von Character-Animationen in Spielen [55]. Neben den athletischen Bewegungen der Spielercharaktere wurde vermehrt Wert auf deren Interaktion mit der Welt um sich gelegt: Anhand

---

<sup>2</sup>Seine Inspiration für *Ico* fand Ueda im Spiel *Another World*.



**Abbildung 3.3:** *Shadow of the Colossus*, *Assassin's Creed*, *Prince of Persia (2008)*, *The Last Guardian* [26].

der Haltung der Wache erkennt man ihren Wachsamkeitsstatus, man kann sich Gruppen von Menschen anschließen, um in der Menge unterzugehen und die NPCs der virtuellen Welt reagieren aufs Eindringen in ihren persönlichen Raum [55]. Im Verlauf des älteren *Prince of Persia: The Sands of Time* Spieles entwickelt sich der Prinz psychologisch weiter, was sich anhand seiner Animationen widerspiegelt [68, Seite 3]. Die Tendenz zu vermehrt realistischen Animationen hat andere Probleme mit sich gebracht [43, Seite 3]:

„Tory cites the uncanny valley as a hurdle in achieving realism with *Assassin's Creed* but distinguishes reality from believability, which is an equally if not more important goal.“

Diese Einstellung vertritt auch *Team Ico* [69, T=00:33:20] und zeigt, dass für den Spieler überzeugende Character-Animationen in modernen Spielen wichtiger sind als realistische. Auch Glenn Entis, EA chief visual technical officer, äußerte sich 2006 dazu [65].

**The Last Guardian:** Das dritte Spiel von *Team Ico*, *The Last Guardian* (noch nicht veröffentlicht, siehe Abb. 3.3), wird verstärkt auf eine Weiterentwicklung von Character-Animationen in Spielen setzen [42]. In diesem Spiel will Ueda mehr auf die Berührung eingehen, dem Spieler das „Gefühl



**Abbildung 3.4:** Beim Annähern einer Wand streckt Ico die Hand aus. Spiel: *Ico*.

der Wärme“ von Berührungen vermitteln [42]. Die nonverbale Interaktion eines Jungen mit dem Greifen-ähnlichen Riesen *Trico* soll dies darstellen. Während des Spieles baut sich eine Beziehung zwischen beiden Charakteren auf, wobei diese Beziehung sich nur durchs Beobachten und Interpretieren der Körpersprache und Gesten von *Trico* entwickeln kann [42]. Aber auch der Junge interagiert und nimmt die Welt um sich wahr. In der Nähe von *Trico* streckt er seine Hand automatisch aus um ihn zu berühren, während beim Annähern einer Wand sein Arm auf diese Annäherung eingeht und sich entsprechend ausrichtet. Letztere Mechanik hat schon *Ico* rudimentär implementiert gehabt (siehe Abb. 3.4).

**The Act:** *The Act* (siehe Abb. 3.5) wurde leider nie veröffentlicht, ist aber in der Hinsicht interessant zu erwähnen, da es ausschließlich die Körpersprache und Gestik als Spielmechanik verwendet hat, um mit Charakteren zu interagieren. Der Spieler musste die Körpersprache von NPCs deuten um zu wissen, wann er welche Aktion zu welchem Zeitpunkt machen konnte.

**Gothic:** Erwähnenswert ist noch *Gothic* (2001, siehe Abb. 3.5), welches anhand der Kampfanimationen und Responsiveness der Eingabetasten den Erfahrungszustand des Spielercharakters wiedergegeben hat. Max von Tettenborn ist einen Schritt weitergegangen und hat eine Modifikation geschrieben, welche klären sollte, „ob eine zahlenlose Werte- und Zustandsvermittlung den Spieler eher in die fiktive Umgebung eintauchen lässt und welche unerwünschte Konsequenzen dabei auftreten“ [73]. Gestiegene Körperkraft zeigte sich zum Beispiel in größer werdenden Muskeln, während Kampffähigkeiten die Geschwindigkeit der Animationen beeinflussten. Das Resultat war eine „gesteigerte Wahrnehmung der Umgebung und Geschichte“ und eine



Abbildung 3.5: *The Act* [60], *Gothic 2*.

„stärkere Identifikation mit dem Erlebten“ [73].

Es gibt noch mehrere Spiele, welche erkannt haben, wie wichtig Charakter-Animationen für den Spieler sind. *Uncharted*, *Infamous* oder die *Tomb Raider* Serie legen alle hohen Wert auf die Körperbewegung ihrer Spielercharaktere. Auch *Valve Corporation* ist bekannt für ihren Beitrag im Animationsbereich, mit Spielen wie *Half-Life* und dessen Nachfolger, *Half-Life 2*. Ersteres hat das skelett-basierte Animationsmodell eingeführt [81], während letzteres neue Maßstäbe in der Interaktion mit KI<sup>3</sup> gesteuerten NPCs gesetzt hat.

Mit steigender Rechenleistung und der Verbesserung und Entwicklung neuer Technologien werden die Spielercharaktere vermehrt eine stärkere Ausdruckskraft durch ihre Körpersprache bekommen. Aktuelle GDC<sup>4</sup> Präsentationen deuten an, dass Entwickler von modernen Spielen verstärkt versuchen, den Spielercharakter und dessen Dialog mit dem Spieler in den Vordergrund zu rücken (z.B. [41, 45, 48, 63, 75, 83]).

### 3.3 Verwendete Methoden und Technologien in der Branche

Die Interaktivität ist das wichtigste Alleinstellungsmerkmal von Spielen. Während Charaktere im Animationsfilm ihre Vorführung, ohne Eingriff des Zuschauers, auf vorgeordneten Bildern abhalten können, müssen Spiele diese Bilder in Echtzeit liefern. Die Bewegungen des Spielercharakters müssen sich spontan auf die Controller-Eingaben des Spielers anpassen und zugleich auch eine überzeugende Vorführung liefern [45].

<sup>3</sup>Künstliche Intelligenz

<sup>4</sup>Game Developer Conference

Diese Anforderungen verlangen ausgiebige Rechenleistung und Technologien, um eine Vorführung von Character-Animationen in Echtzeit implementieren und unterstützen zu können. Viele der Standardtechniken für die Echtzeitdarstellung von Character-Animationen wurden vom Animationsfilm übernommen, weisen jedoch gewisse Mängel auf. Das Prinzip vom Blending von Animation Clips wäre ein solches Beispiel, wo die Anzahl von bereitzustellenden Clips mit der Komplexität der Animation stetig anwächst. Dies kostet Zeit, Speicher und erlaubt dem Animator wenig Spielraum, sich mit interessanten Animationen und deren Nuancen zu beschäftigen. Verbesserte Technologien und Anwendungsmethoden würden dem Animator die eintönige, immer wiederkehrende Arbeit der Standard Bewegungsabläufe wie z.B. die Walk-Cycles und Turn-Cycles abnehmen. Dadurch hätte er mehr Zeit diese zu individualisieren und zu verfeinern. Zusätzlich kann er sich mehr mit Übergangsanimationen beschäftigen oder neue spezielle Animationen erstellen, welche z.B. der Zustandsvermittlung dienen. Letztere wären für den Spielercharakter wichtig, um ihm mehr Ausdruck, Lebendigkeit und Emotion zu verleihen. Um dies zu bewerkstelligen wird die Körperbewegung, Gestik und die Andeutung der Zugehörigkeit des Spielercharakters zur virtuellen Welt, durch dessen Interaktion und Reaktion, zum Ausdruck gebracht. Durch verbesserte Animationen und deren Nuancen würde auch die Beziehung zwischen Spielercharakter und Spieler gestärkt werden, da letzterer anhand seiner Erfahrungen im echten Leben den Spielercharakter besser und schneller deuten kann.

Viele der älteren Methoden und Technologien für die Echtzeit Character-Animation haben bewährte Arbeitsabläufe, welche den meisten Entwicklern zugänglich sind. Moderne Lösungen, welche einen vereinfachten, schnelleren Arbeitsablauf und mehr expressive Möglichkeiten versprechen, sind noch nicht Teil des zugänglichen Standard Arbeitsablaufes (siehe z.B. [50]). Firmen wie *Autodesk* etablieren sich zurzeit als Anlaufstelle für Echtzeit Animationslösungen und versuchen mit jeder neuen Version ihrer Produktpalette einen Standard Arbeitsablauf zu entwickeln und bestehende zu vereinfachen und optimieren. Die aktuellste Entwicklung ist *Autodesk's Projekt Skyline* (siehe Abb. 3.6), „an industry-standard platform for non-programming-savvy artists to view the impact of animation changes directly within the games engine“ [74].

Es ist wichtig zu beachten, dass *Middleware* Lösungen den Animator dabei unterstützen sollen, den Prozess des Importierens und Darstellens seiner Animationen im Spiel zu vereinfachen. Die Verwaltungsarbeit soll automatisiert und ihm abgenommen werden. Zeitintensive, immer wiederkehrende Arbeiten sollen reduziert werden, damit sich der Animator ganz auf die Essenz seines Feldes – einem Charakter Leben einzuhauchen – konzentrieren kann. Im Rahmen dieser Arbeit liegt der Schwerpunkt auf der Körperbewegung und Gestik des Spielercharakters und wie man Anhand dieser visuellen Feedbackformen gewisse Zustände dem Spieler vermitteln kann. *Middleware*

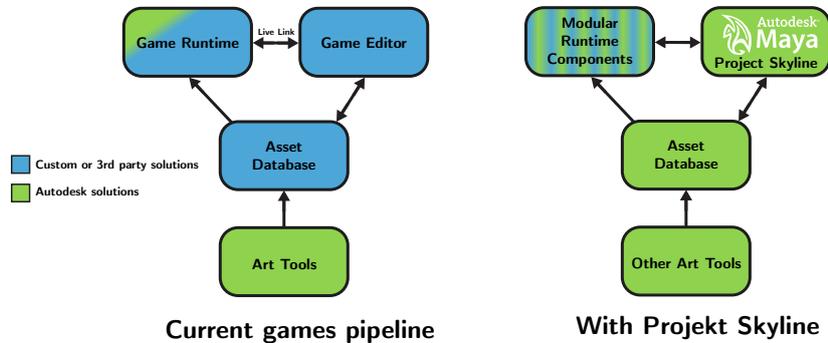


Abbildung 3.6: Autodesk Projekt Skyline. Bildquelle: [74].

Lösungen [19, Seite 12] welche u.a. diesen Prozess unterstützen sind aufgelistet im Anhang B zu finden.

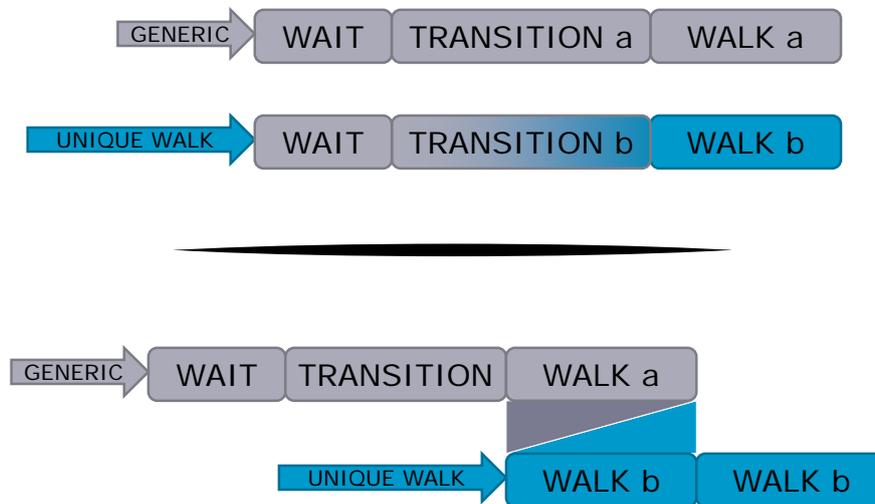
In Folge werden Methoden und Techniken beschrieben welche es erlauben, den oben erwähnten Schwerpunkt zu erreichen. Das Ziel, solche Methoden zu entwickeln und zu verbessern, beschreibt Jonathan Cooper in einem Zitat zu Animation Blending [39]:

‘This is the first step toward truly procedural movement in video-games, where the animator is providing the raw data to be manipulated by the player and AI to produce a more organic look and avoid the repetitiveness of seeing the same identical animations over and over again.’

Es ist diese prozedurale Generierung von Animationen, welche dem Spiel als interaktivem Medium gerecht werden. Die Vielfalt an verschiedenen Körperbewegungen, welche der Spieler zu jeder Zeit von seinem Spielercharakter und NPCs erwartet, können auf Grund des Arbeitsaufwands und der fast unendlichen Anzahl an verschiedenen Körperbewegungen nicht in der Form von statischen Clips geliefert werden. Animationsinhalte müssen daher dynamisch während der Laufzeit des Spieles generiert werden.

**Blending:** Das *Blending* von Animationen ist das zeitgleiche Kombinieren von zwei oder mehr Animation Clips um (in Echtzeit) eine neue Animation zu produzieren (siehe Abb. 3.7, Quelle: [48]). Diese Methode wird typischerweise bei *speed-based* Blends verwendet, worin z.B. der Übergang von einer *Geh-* zu einer *Renn-*Animation abhängig des Neigungsgrades des Analog-Sticks ist [39]. Die Cycles müssen gleicher Länge sein, und die Bewegungsaktionen dürfen nicht zu unterschiedlich sein.

*Assassin’s Creed: Brotherhood* verwendet eine Form des Blending welche die Entwickler *Transblend* genannt haben [48](siehe Abb. 3.7). Hier wird die

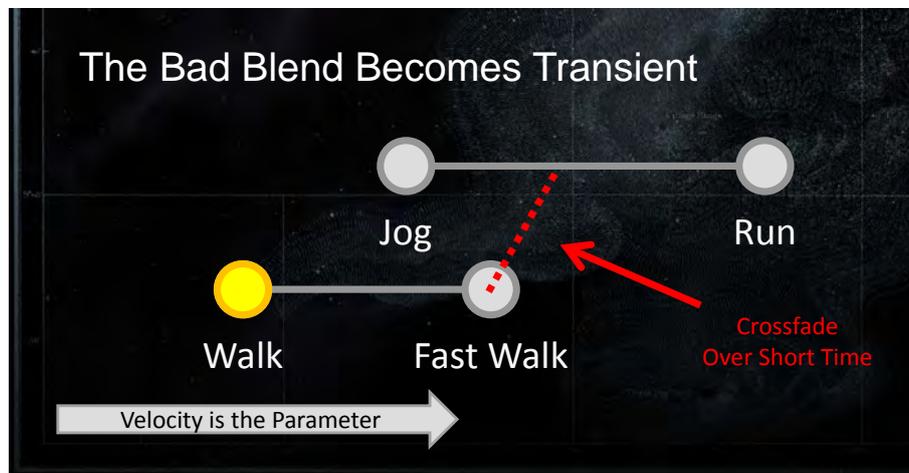


**Abbildung 3.7:** Oben: Einfaches Blending, Unten: *Transblend*. Bildquelle: [48].

Animation, welche der Transition Animation folgt, mit einem Blend durch eine ähnliche ersetzt, um Variation in der Bewegung einzubauen.

Ein einfaches Blending zwischen einer *Geh*- und einer *Renn*-Animation würde aufgrund der verschiedenen Foot Contacts nicht gut aussehen. Um dies zu lösen verwendet *Halo: Reach* zwei verschiedene Gangarten, jede mit ihrem eigenen Blend. Zwischen diesen beiden Gangarten gibt es über kürzere Zeit einen Crossfade, womit ein schlechter Blend relativiert wird [63](siehe Abb. 3.8).

In *Crysis 2* versuchen die Entwickler mit *Parametric Blending* [45] einen gewissen Grad an Vorhersehbarkeit ins Animationssystem einzubauen, um die Qualität der Animationen zu verbessern. Das Problem sind die Blend-Weights, welche keine Aussage über die eigentlichen Eigenschaften (natürliche Parameter wie Geschwindigkeit, Laufrichtung, Zielrichtung, Mood, etc.) eines Animation Clips treffen können. Neben der komplexen Berechnung von mehreren Animation Blends liefern diese keine vorhersehbaren Resultate. Beispielsweise liefert die Interpolation zwischen einer *Jog*-, *Lauf*- und *Renn*-Animation zwar eine gewisse Geschwindigkeitsanimation, jedoch kann im Vorhinein nicht die exakte Geschwindigkeit festgestellt werden. Einfacher wäre es, die natürlichen Parameter eines Animation Clips zu verwenden womit die Resultate vorberechenbar wären. Auch *Halo: Reach* verwendet Blending von Animation Clips durch Parameter für die prozedurale Anpassung der Charakter Sprünge anhand der Distanz und Höhen Parameter [63]. Anwendungsgebiete wären Blick- und Zielrichtung sowie KI Navigation. Siehe



**Abbildung 3.8:** Blending zwischen zweier Gangarten. Bildquelle: [63].

[45] für mehr Informationen.

**Layering:** Eine weitere Methode um Flexibilität in der Animationsdarstellung zu erhalten ist durch *Layering* gegeben [50]. Hier wird der Körper eines Charakters in verschiedene Layer unterteilt (z.B. Gesicht, Augen, Waffen, Atmen, Hände, etc.). Jeder Layer modifiziert eine gewisse Ansammlung an Joints und funktioniert unabhängig von anderen Layer (Technisch gesehen beinhaltet jeder Layer einen eigenen Animation Tree). Animationen innerhalb eines Layers sind synchron, Layer unter sich jedoch nicht. Dies hat den Vorteil, gewisse Animationen unabhängig von anderen abspielen zu lassen, womit neue Kombinationen an Körperbewegungen zur Laufzeit erstellt werden können. Beispielsweise kann eine Waffen-Ziel Animation während einer *Geh-*, *Lauf-* und *Duck-*Animation eingespielt werden, ohne eine Full-Body Animation für jede mögliche Zielausrichtung zu erstellen.

Das Layering Modell erlaubt verschiedene Arten von Layer Animationen. Neben dem *Full-Body Layer*, dessen Animationen den ganzen Körper des Charakters ansprechen, gibt es den *Overwrite* (bzw. *Partial*) und *Additive Animation Layer* [45]. *Overwrite Animation Layers* mit höherer Priorität überschreiben und ersetzen Animationen in Layers mit niedriger Priorität auf gleicher hierarchischer Ebene. Additive Animation Layers, wie der Name schon sagt, addieren ihre Animationen auf einen anderen Layer. Eine Additive Animation kann man sich folgendermaßen vorstellen: ein Charakter hat zwei Animationen, einmal eine Look-Around und einmal die Charakter T-Pose. Zieht man beide Animationen voneinander ab, ist das Ergebnis die Additive Animation. Diese sind nützlich, um sich wiederholende Animationen mit etwas Variation zu versehen oder die Übergänge zwischen Animationen zu verfeinern. Laut Ivo Herzeg [45] sind Additive Animations am besten

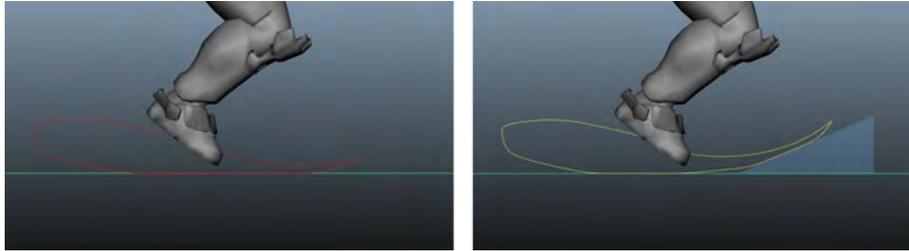


Abbildung 3.9: IK Fuß Platzierung. Bildquelle: [27].

für Becken und Rückgrat geeignet, die restlichen Bereiche eher für Overwrite Animations.

**Inverse Kinematik:** Die *Inverse Kinematik* (IK) ist mehrere Jahre schon fester Bestandteil von interaktiven Animationen und wird meist eingesetzt um Korrekturen in Echtzeit an Beinen, Armen und Rückgrat durchzuführen. Rigid Baked Animations verursachen meist Probleme, da sie sich nicht der Umgebung dynamisch anpassen können. Die Anpassung von Füßen und Beinen bei unebenem Boden (siehe Abb. 3.9) oder das Lösen des *sliding feet* Problems sind wichtige Einsatzgebiete von IKs. Sie werden auch oft für das Anpassen von Arm und Gewehr beim Zielen von Fernkampfaffen verwendet. *Crysis 2* [45] verwendet IKs am Rückgrat, um den Rückstoß von Waffen zu simulieren. In Kombination mit IKs für die Arme wird dies als eine „100% prozedurale Animation“ [45] angegeben. Weiteres gibt es noch das *Full-Body IK System* von *Autodesk*, welches immer mehr Einzug in Spielen findet. Durch das Bewegen von Gliedmaßen eines digitalen Charakters (z.b. das Ausstrecken der Arme nach oben) folgt der restliche Körper diesen Bewegungen und passt seine Haltung prozedural an.

Bezüglich der Anpassung von Beinen und Füßen bei unebenem Boden verwendet *Halo:Reach* folgendes interessante Verfahren: eine Kurve die den Bewegungspfad der Füße definiert passt sich dynamisch dem Boden an. Dadurch ändert und passt sich der Pfad der Fuß Animation dem Boden in



**Abbildung 3.10:** Eine Kurve definiert den Bewegungspfad der Füße. Bildquelle: [63].

Echtzeit an [63] (siehe Abb. 3.9).

**Prozedurale Animation:** Prozedurale Animationen im typischen Sinne sind Computer-generierte Simulationen, welche in Spielen meist für Particles, Cloth und Rigid Bodies Verwendung finden. Für Character-Animationen werden meist Ragdoll Physics verwendet, um das Bewegungsverhalten eines leblosen Körpers prozedural zu simulieren. In modernen Spielen wird dieses Verfahren meist in Zusammenhang mit Animation Blending verwendet [45] um den „billigen Puppen Effekt“ zu minimieren. Es gibt jedoch auch Spiele, welche gänzlich auf Prozedurale Animationen setzen, wie z.b. *Sumotori Dreams*<sup>5</sup>, in welchem die Sumo Charaktere rein durch Physik Simulationen animiert und gesteuert werden.

**Motion Capture:** Zuletzt gibt es noch das *Motion Capturing* Verfahren, welches durch das Erfassen und Übertragen von komplexen, menschlichen Bewegungen auf einen digitalen Charakter den zeitlichen Aufwand von Keyframe Animationen reduzieren, und zugleich auch die Animationen realitätsnah darstellen soll. Nicht alle Arten von Bewegungen können erfasst werden. Durch eine nicht Eins-zu-eins Übertragung (Retargeting) der Motion Capture Daten auf ein digitales Skelett können die Animationen realitätsfremd wirken. In Kombination mit realitätsnahen Charakter Modellen löst dies den *Uncanny Valley Effekt*<sup>6</sup> aus und lässt dem Charakter den gewissen „motion capture look“ haben [21, Seite 17]. Für *Prince of Persia: The Sands of Time* verwendeten die Animatoren Keyframe Animationen, um die Animationen und deren Übergänge besser kontrollieren zu können [68, Seite 2]. *Valve* verwendete Motion Capture um Referenzen zu erstellen, die meisten ihrer

<sup>5</sup> <http://www.gravitysensation.com/sumotori>

<sup>6</sup>Engl. „unheimliches Tal“, ein paradoxer Effekt der bei realitätsnahen, künstlichen, anthropomorphen Figuren auftritt. Während beim Zuschauer eine abstrakte und völlig künstliche Figur als solche akzeptiert wird, fällt bei zunehmend realistischen Figuren ab einem gewissen Punkt die Akzeptanz. Erst wenn die künstliche Figur nicht mehr von einer echten Figur zu unterscheiden ist, steigt die Akzeptanz [28].

Animationen für *Half-Life 2* sind jedoch mit Keyframe Animationen erstellt worden [66, Seite 1]. Auch *Team Ico* verwendete Keyframe Animationen für *Shadow of the Colossus*, um Charakteren wesenseigene Eigenschaften durch emotionale und überzeugende Animationen geben zu können, welche durch Motion Capture nicht überzeugend erfasst werden können [69, T=00:31:30].

All diese Methoden und Techniken sind dafür entwickelt worden, um Animationen in Spielen realistischer und glaubwürdiger darzustellen. Parallel dazu sollen sie dem Animator die Möglichkeit bieten, den Großteil seiner Zeit tatsächlich mit dem Animieren von Charakteren und anderen Objekten zu verbringen. Die Animation ist eine Detailarbeit – es sind die feinen Nuancen der zwischenmenschlichen, non-verbalen Kommunikation, die der Mensch unbewusst wahrnimmt. Fehler in der Animation von digitalen Charakteren fallen sofort auf und lassen sich schwer verbergen. Aus dieser Sicht ist es wichtig, dass die Entwicklung neuer Methoden und Techniken stetig vorangetrieben wird. Im Gegensatz zur Animation im Film ist die Animation in Spielen immer noch in ihren Kinderschuhen, weshalb dessen Entwicklung von Animatoren mitbestimmt werden kann. Dies muss auch wahrgenommen werden, da die Spiele Animation ein interdisziplinäres Feld ist und somit von vielen Fachrichtungen beeinflusst wird [22, Seite 35].

### 3.4 Zusammenfassung

In diesem Kapitel wurde die Entwicklung von Character-Animationen in Spielen kurz angerissen. Zu Beginn wurde auf die Gemeinsamkeiten und Unterschiede von Character-Animationen in Film und Spielen eingegangen. Anhand der bewussten Darstellung von Spielen als interaktive Medien wurden wichtige Bereiche und Merkmale von Animationen in Spielen, im Unterschied zum Medium Film, beschrieben. Anschließend wurde eine kurze historische Übersicht bezüglich der Entwicklung von Character-Animationen in Spielen erarbeitet. Erwähnt wurden wichtige Spielebeispiele, welche über die Jahre hinweg Character-Animationen vorangetrieben haben. Daraufhin wurde eine Übersicht von bekannten und aktuellen Technologien in der Spielebranche präsentiert, welche den Animator zur Realisierung von Animationen im Medium Spiele unterstützen soll.

## Kapitel 4

# Der Spielercharakter

Der anthropomorphe Spielercharakter ist die digitale Kreatur, durch welche der Spieler in der virtuellen Welt kommuniziert und interagiert. Mit dem Controller in der Hand und dem Bildschirm als Fenster in die Spielewelt, wird dem Spieler erlaubt, dem Spielercharakter Befehle zu erteilen und Handlungen ausführen zu lassen. Aufgrund sich ändernder Elemente am Bildschirm, in der Spielewelt oder am Spielercharakter, wird der Spieler von seinen Entscheidungen und Handlungen und deren Auswirkungen informiert.

Im Rahmen dieser Arbeit stellt sich jedoch die Frage, inwiefern man diese Feedback Informationen verstärkt durch die Animationen des Spielercharakters dem Spieler vermitteln kann. Da die Körpersprache die natürlichste Sprache des Menschen ist, würde es nahe liegen diese auch im Spiel zu verwenden. Da der Fokus des Spielers die meiste Zeit auf seinem Spielercharakter ist, wäre dies auch der beste Platz, um ihm den Großteil an Informationen zu vermitteln. Abstrakte Elemente am HUD können reduziert werden, um den Immersionsgrad des Spielers in der Welt beizubehalten. Damit sich der Spieler ganz auf das Spiel, seinen Charakter und dessen Nuancen in der Animation konzentrieren kann, müssen alle möglichen „Störquellen“ so gut es geht reduziert werden. Neben den erwähnten HUD-Elementen ist hier auch auf die Ergonomie und das Layout des Eingabegerätes zu achten, sowie auch auf die Qualität der visuellen und akustischen Ausgabesignale. Im Idealfall würden die Interface-Schichten dem Spieler transparent sein, und ihm eine ununterbrochene Immersion im Spiel erlauben.

### 4.1 Körpersprache als visuelle Feedbackform

In Folge wird auf die Wichtigkeit der Körpersprache als Ausdrucksform des Menschen eingegangen. Zusätzlich wird die Relevanz von Körpersprache in Spielen in der Form von Character-Animationen erläutert. Klassifizierungsansätze von Körpersprache und Gesten werden vorgestellt und deren Anwendbarkeit in Computerspielen beschrieben.

### 4.1.1 Körpersprache als Form der Kommunikation

Der biologische Körper des Menschen ist ein offenes System. Es steht im Austausch mit der Umwelt durch den Transfer von Energie und Materie in und aus dem Körper. Neben den ursprünglich-zwischenmenschlichen Kommunikationskanälen (z.B. olfaktorische und taktile Kommunikation) hat der Mensch komplexe Zeichensysteme entwickelt, um sich gegenseitig zu verständigen. Obwohl beispielsweise die Sprache ein Teil der komplexen Zeichensysteme zwischen Menschen ist, wird im Rahmen dieser Arbeit eher auf den einseitigen Informationsaustausch mittels der Körpersprache vom Spielercharakter zum Spieler eingegangen. Die Mimik wird auch außen vor gelassen, die Betonung liegt bei der Körperbewegung und Gestik.

Körpersprache ist das Vermitteln von nonverbalen Informationen durch die bewusste und unbewusste Bewegung von Muskeln und Körpergliedern. Ideen und Vorstellungen, welche sich sonst nicht anders ausdrücken lassen, werden auf die Muskeln und Glieder unseres Körpers abstrahiert und so zum Ausdruck gebracht. Das Ziel ist es, anderen Menschen diese Informationen zukommen zu lassen. Bekannte Begriffe, die zum Thema Körpersprache einfallen, sind die *Mimik* und *Gestik*. Als *Mimik* werden die kleinen Bewegungen auf der Gesichtsoberfläche bezeichnet, welche das Aussehen der Gesichtszüge verändern [5, Seite 8]. Als *Gestik* sind die „kommunikativen Bewegungen insbesondere der Arme, Hände und des Kopfes“ [80] gemeint. Manche dieser Bewegungen oder Zeichen verlangen eine Aufklärung über ihre verschlüsselte Bedeutung, bei anderen wiederum ist die Bedeutung dem Menschen seit Geburt an klar. Im Falle, dass die beispielsweise hochspezialisierte Verständigung durch Sprache aufgrund regionaler Unterschiede versagt, kann auf ein grundlegendes Repertoire an Körperbewegungen zurückgegriffen werden, um sich zu verständigen.

Diese Arbeit erhebt nicht den Anspruch, die Körperbewegung und Gestik von Menschen zu quantifizieren oder neue Studien im Bereich Körpersprache zu eröffnen. Vielmehr wird auf vorhandene Literatur und eigene Erkenntnis zurückgegriffen, um eine Basis und das Bewusstsein für die Körpersprache und deren Anwendung in der Character-Animation in Computerspielen zu entwickeln.

Das Quantifizieren von zwischenmenschlicher Körpersprache ist keine triviale Aufgabe, wie es Katherine Isbister in folgenden Sätzen schreibt [8, Seite 161]:

„[...] it is difficult to translate insights about holistic impressions of personality or social connection into quantifiable and testable predictions. [...] Body cues have a pervasive influence on social relationships and are therefore an important part of crafting truly engaging game characters that feel lifelike and that evoke social reactions from players.“

Desmond Morris [14] beschreibt den Menschen als handelndes Wesen, welches unwiderstehlich den Drang verspürt, sich durch *actions* ausdrücken zu wollen [14, Seite 1]:

,With his massive brain, man has increasingly internalized his behaviour through complex processes of abstract thought — through language, philosophy and mathematics. With his weak body, he has dramatically externalized his behaviour, scattering the surface of the globe with his artefacts — his implements, machines, weapons, vehicles, roads, works of art, buildings, villages and cities.’

Der Mensch hat sich also verstärkt zu einer Kreatur des Ausdrucks entwickelt. Damit seine körperlichen Ausdrücke Relevanz finden, müssen sie Informationen vermitteln und durch visuelle Signale von einem Zuschauer wahrgenommen werden. Diese *action* wird dann als Geste bezeichnet [14, Seite 21]. Im Allgemeinen wird der Begriff *observed action* verwendet, da es wichtiger ist, welche Signale beim Empfänger ankommen, im Gegensatz zu den Signalen, welche wir uns einbilden geschickt zu haben [14, Seite 21].

Im Bereich der Computerspiele ist es der Spieler, der den wahrnehmenden Zuschauer spielt. Anhand der Animation des Spielercharakters bekommt der Spieler Informationen in einem von ihm bekannten Feedback Kanal und ist somit über dessen Zustand informiert. Als Feedback Kanal dient die Körpersprache, die jedem Menschen in gewissem Ausmaß von Geburt an bekannt ist.

Der Zuschauer nimmt das Geschehen in der virtuellen Welt aus der Sicht einer virtuellen Kamera wahr. Im Kapitel 2 wird auf die Vorteile und Gründe der Third-Person Perspektive eingegangen – zusammenfassend eignet sie sich jedenfalls als flexibles Beobachtungsinstrument für den Spieler, da dieser dessen Spielercharakter von mehreren Winkeln aus betrachten kann. Hier liegt auch der Grund, warum die Mimik als visuelle Feedbackform nicht berücksichtigt wird (zumal die Mimik als eigenständiges Thema behandelt werden kann). Der Grund liegt in diesem Fall bei der Third-Person Perspektive von Spielen. Da der Spieler die meiste Zeit den Spielercharakter mit einem gewissen Abstand von hinten sieht, würde er den Großteil der Zeit das Gesicht des Spielercharakters nicht sehen. Auch die Nuancen des Gesichtsausdruckes würden ab einer gewissen Distanz nicht mehr auffallen. Die Mimik als Feedbackform eignet sich deshalb am meisten bei vorgegebenen Kamerawinkeln, Zwischensequenzen oder NPCs, die einen begleiten.

,Gestures, by definition, transmit signals, and these signals must come across clearly if we are to understand their messages. They cannot afford to be vague and woolly; they must be crisp and sharp and difficult to confuse with other signals.’ [14, Seite 45]



**Abbildung 4.1:** Der Charakter *Wander* aus verschiedenen Kamera Winkeln. Spiel: *Shadow of the Colossus*.

Die meisten modernen Spiele erlauben dem Spieler die Kontrolle über die Kamera. Unter der Annahme, dass der Spieler die Kontrolle über die Kamera hat und somit der Spielercharakter aus allen Winkeln und Distanzen zu jeder Zeit betrachtet werden kann, muss die Körpersprache des Spielercharakters entsprechend expressiv sein. Sie muss (zumindest so gut wie möglich) von allen Winkeln aus deutlich erkennbar sein (siehe Abb. 4.1), eine bewusste und gezielte Übertreibung der Bewegungen kann hier nachhelfen, um das Animierte erfolgreich dem Spieler zu vermitteln.

#### 4.1.2 Arten der Körpersprache

Um dem Spieler gezielt Informationen anhand der Körpersprache des Spielercharakters vermitteln zu können, muss auf theoretisches und praktisches Wissen zurückgegriffen werden. Als wichtige Quelle entpuppt sich der praktische Erfahrungsschatz, den man sich im Alltag durch das Beobachten anderer Menschen ansammelt. Auch das bewusste Interpretieren und Deuten des eigenen Gefühls und der eigenen Handlungen führt zu einem stärkeren Verständnis der körperlichen Ausdruckskraft. Samy Molcho [13, Seite 14] schreibt diesbezüglich: "Nachdrücklicher, überzeugender und genauer als durch Informationen und Beobachtungen der Außenwelt werde ich durch die Erfahrungen und Reaktionen meines eigenen Körpers über die Abhängigkeiten und Nuancen menschlichen Verhaltens belehrt.". Theoretisches, gesammeltes Wissen aus diversen Medien (Schrift, Bild, etc.) versuchen dem Thema Körpersprache eine geordnete Struktur zu verleihen und sind eine wichtige Quelle für formale Ansätze und Klassifizierungen.

Chuck Clanton, Co-Designer der virtuellen Welt *There* von *Makana Technologies*, teilt die Körpersprache in zwei Bereiche [8, Seite 178]: die *Autonomic Functions* und die *Intentional Expressions*. Die autonomen Körperfunktionen halten den Körper am Leben und drücken sich durch Bewegungen wie atmen und blinzeln aus. Sie wecken in einem Körper den Eindruck der Lebendigkeit und werden unbewusst wahrgenommen. Autonome Gesten kommen

auch im sozialen Kontext vor, wie z.B. das Anerkennen einer neuen Person in einer Gesprächsrunde durch kurzes Hin- und Anschauen der Person. Die beabsichtigten Ausdrücke beinhalten alle Arten von Bewegungen, die bewusst vom Körper gesteuert werden, wie beispielsweise winken oder lachen. Hier werden gezielt Informationen versendet, um von Empfängern interpretiert zu werden.

In Folge werden ein paar Auszüge aus dem gesammelten Repertoire an klassifizierten Gesten von Desmond Morris [14] vorgestellt. Das Augenmerk liegt auf Klassen von Gesten, welche als visuelles Feedback durch den Spielercharakter für den Spieler in Frage kommen könnten.

**Incidental Gestures:** Dies sind Gesten, welche aus mechanischer Notwendigkeit entstehen, jedoch eine untergeordnete Nachricht mit sich führen. *Actions* welche *Incidental Gestures* auslösen haben meist keine bewusst-sozialen Absichten hinter sich stehen, wie es bei *Primary Gestures* der Fall ist. Ein Beispiel wäre das Gehen von Punkt A zu Punkt B. Obwohl die Notwendigkeit uns treibt zu gehen, ist es die Gangart die verraten kann, welche Persönlichkeit oder Laune der gehende Mensch hat.

Im Gegensatz zu *Primary Gestures* sind *Incidental Gestures* eines der wichtigsten Gesten, welche ein Spielercharakter dem Spieler vermitteln kann. Dies hängt natürlich von Genre, Kontext, Design und Absicht der Entwickler ab. Desmond Morris stellt diesbezüglich folgende Frage [14, Seite 21]:

‘A convenient way to distinguish between Incidental and Primary Gestures is to ask the question: Would I do it if I were completely alone? If the answer is No, then it is a Primary Gesture. We do not wave, wink, or point when we are by ourselves; not, that is, unless we have reached the unusual condition of talking animatedly to ourselves.’

Da der Spielercharakter die meiste Zeit nicht auf direktem Wege mit dem Spieler kommuniziert (und somit die *fourth wall* erhalten bleibt, siehe Abschnitt 4.3.3), sind die Gesten des Spielercharakters, welche der Spieler zu interpretieren hat, meist *Incidental Gestures*. Dies ist nicht der Fall, wenn der Spielercharakter mit anderen NPCs im Spiel oder in seiner virtuellen Umgebung interagiert oder kommuniziert. In so einem Fall würde er zu *Primary Gestures* zurückgreifen. Dies wäre eine Gelegenheit für den Spieler, mehr über den Spielercharakter zu erfahren, da *Primary Gestures* mehr für soziale Interaktionen ausgelegt sind und somit ein breiteres Spektrum an Informationen vermitteln.

**Expressive Gestures:** *Expressive Gestures* sind primäre, biologische Gesten welche alle Menschen, mit Rücksicht auf gewisse kulturelle Unterschiede,

gemeinsam haben (z.B. lachen, weinen). Hier sind größtenteils die Gesichtsausdrücke gemeint, jedoch haben die Arme und Hände einen gleichwertigen Stellenwert bei der Informationsvermittlung - sie werden als "das sensibelste Werkzeug und die ausdrucksstärksten Glieder des Menschen" bezeichnet [13, Seite 156]. Das Achselzucken oder Winken wären zwei Beispiele von Expressive Gestures. Im Unterschied zu Incidental Gestures senden Expressive Gestures nur Signale ohne mechanische Notwendigkeit.

**Multi-Message Gestures, Regional Signals:** Dies sind Gesten und Zeichen, welche verschiedene Bedeutungen in anderen Ländern, Kulturen, Orten, Kontext oder Zeiträume haben. Ein Beispiel wäre das O.K. Zeichen, welches in Amerika für „alles in Ordnung“ steht, während es in Griechenland einer Beleidigung nahe kommt. Hier ist es wichtig verschiedene Gesten zu identifizieren und auf ihre Bedeutung im internationalen Raum einzugehen. Um klare Signale an den Spieler schicken zu können, müssen Gesten und Bewegungen verwendet werden, welche zumindest das breiteste Publikum, im Kontext des Spieles, verstehen kann.

**Auto-Contact Behaviour:** Das unbewusste Berühren des eigenen Körpers kann ungekünstelte Hinweise zu der inneren Stimmung einer Person liefern. Das Falten der Arme oder Umklammern der Hände sind zwei Beispiele von *Auto-Contact Behaviour*. Hier könnte ein Spielercharakter, der sich in einer gewissen Umgebung unwohl fühlt, durch Auto-Contact Behaviour seinen Gefühlszustand ausdrücken.

**Intention Movements:** *Intention Movements* sind *actions*, welche einer Handlung zuvorkommen, um diese einzuleiten. Sie sind vergleichbar mit der Anticipation von den Prinzipien der Animation (siehe Abschnitt 3.1.2) und werden gleichwertig angewendet.

**Locomotion:** Folgender Auszug beschreibt die sinnliche Erfahrung die das Gehen mit sich bringt [14, Seite 444]:

„By walking through a setting we are exposed to it in an ideal manner for imprinting its details on our minds. We take it in at a natural pace. We feel it and we respond to it. To walk through a landscape is to explore it.“

Desmond Morris beschreibt zwanzig grundlegende Arten wie der Mensch sich fortbewegen kann, als wichtigste jedoch wird das Gehen hervorgehoben. Das Gehen hat den Menschen erlaubt, sich auf der Erde zu verbreiten. Das aufrechte Gehen lässt heute immer noch Fragen im Zusammenspiel der Muskeln offen – wie der Mensch es schafft seinen Gang zu halten. Im Bereich der

Animation ist der Walk-Cycle immer noch einer der schwierigsten Bewegungen, um überzeugend darstellen zu können. Es ist daher eindeutig, dass ein großes Augenmerk auf die Fortbewegungsmechanismen des Körpers gelegt werden muss, um eine plausible und überzeugende Animation dem Spieler vermitteln zu können.

Die beschriebenen Klassen von Gesten sind für Computerspiele im Allgemeinen relevant. Sie zeigen auf, in welchen Situationen welche Arten von Gesten verwendet werden können mit dem Ziel, den Spielercharakter in der jeweiligen Situation natürlich wirken zu lassen. Für spezielle Fälle und Situationen gibt es noch mehrere Klassen von Gesten. Eine Behandlung aller speziellen Fälle und Arten würde den Rahmen dieser Arbeit sprengen. Für die Feedback Animationen eines Spielercharakters im Allgemeinen sind sie jedoch grundlegend.

## 4.2 Feedback Animationen des Charakters im Computerspiel

Einen digitalen, anthropomorphen Charakter mit der menschlichen Körpersprache zu versehen ist keine triviale Aufgabe. Der Animator muss sein Wissen und die Nuancen der Körpersprache in das digitale Medium transferieren, um den Charakter überzeugend, natürlich und lebendig wirken zu lassen. Dies gilt zugleich für den Spielercharakter als auch den NPCs. Je besser die Qualität und Ausdruckskraft der Animationen dieser Charaktere ist, desto besser wird das Feedback sein, welches der Spieler von ihnen erhält. Sein Verständnis für die digitale Welt und seine Handlungsentscheidungen hängen zum Teil davon ab.

Die Macht der erfolgreichen Anwendung von Körpersprache auf digitalen Charakteren hat Walt Disney schon früh erkannt [23, Seite 13]:

„Animation can explain whatever the mind of man can conceive.“

### 4.2.1 Anbringen der Körpersprache

Computerspiele, wie auch Film und Fernsehen, gehören zu den bewegten Bildern. Film und Fernsehen spielen eine vorgegebene bzw. lineare Anordnung von Bildern ab, um dem Zuschauer den Effekt von abgebildeter Bewegung darzustellen. Im Falle von Computerspielen geschieht das Generieren bewegter Bilder in Echtzeit, da der Spieler direkt mit dem Spiel interagieren kann (siehe Abschnitt 4.3). Das Resultat dieser Interaktion muss dem Spieler in Form von Bildern präsentiert werden, damit dieser *Feedback* erhält.

Der *Stroboskopeffekt* wird genutzt, um dem Zuschauer auf digitalem Wege eine bewegte Vorführung eines (unter anderem) anthropomorphen Charakters zu vermitteln. Dies war historisch gesehen in Zeiten ohne bewegter Bilder nur den Anwesenden einer Theatervorführung vorbehalten. Durch das bewegte Bild jedoch wird einem die Möglichkeit eröffnet, fiktionale Charaktere zu erschaffen, welche mit der Körpersprache des Menschen versehen sind. Die vom Zuschauer wahrgenommene Bewegung dieser Charaktere fällt unter den Begriff der Character-Animation und erlaubt es die Animation als ein Medium zu verstehen, welches die Körpersprache von Charakteren zu vermitteln versucht.

Das Anbringen der Körpersprache, mit anderen Worten das Animieren von digitalen, anthropomorphen Charakteren, ist keine einfache Aufgabe. Während im realen Leben Menschen anhand ihres eigenen Körpers die Körpersprache zur Schau stellen können (Schauspieler), muss ein Animator dieses Wissen auf einen digitalen Körper übertragen. Von daher unterscheidet sich die Arbeit eines Animators großteils von der eines Schauspielers. Ein guter Schauspieler ist sich seiner Körpersprache bewusst und weiß diese einzusetzen, um dem Zuschauer eine fiktive und überzeugende Person zu vermitteln. Er versetzt sich emotional in seinen Charakter und ist psychologisch vor Ort anwesend [46]. Er arbeitet für den Moment der Szene, geht auf andere Charaktere ein und versucht nicht den nächsten Moment vorherzusehen (Anticipation). Auch darf er dem Zuschauer die Gefühle seines Charakters nicht zeigen [7, röm. Seite x].

Ein Animator, wie auch ein Schauspieler, muss die Nuancen der Körpersprache kennen und wissen, wie diese eingesetzt werden können, um den Charakter für den Zuschauer überzeugend zu gestalten. Dieser Punkt verbindet beide Berufe. Beide wollen durch Gesten, Ausdrücke, Emotionen und Timing dem Zuschauer ein gewisses Bild von einem Charakter vermitteln [23, Seite 474]. Im Gegensatz zum Schauspieler muss jedoch ein Animator explizit die Gefühle und Emotionen seines Charakters dem Zuschauer vermitteln, damit dieser glaubhaft wirkt [23, Seite 505] und [7, Seite x]. Ein Animator arbeitet nicht vor Ort oder im Moment der Vorführung – seine Arbeit wird zu einem späteren Zeitpunkt vorgeführt. Im Vergleich zum Schauspieler ist Anticipation ein wichtiger Teil seiner Arbeit. Ohne diese und den restlichen Animationsprinzipien (siehe Abschnitt 3.1.2) würde sein Charakter starr und unnatürlich wirken. Frank Thomas und Ollie Johnstone fassen das Ziel eines Animators gut zusammen [23, Seite 473]:

‘In our animation we must not only show the actions or reactions of a character, we must picture also with the action the feelings of those characters.’

### 4.2.2 Körpersprache in Computerspielen

Moderne 3-D Computerspiele beinhalten mehrere verschiedene Animationssysteme, alle mit dem Ziel, dem Spieler eine glaubhafte und verständliche Welt zu vermitteln. Dies geschieht beispielsweise durch cloth, fluids, particles, Textur Animationen oder im Allgemeinen durch animieren von Attributen in der Spielwelt, wie das Flimmern von Licht oder das Öffnen einer Tür. In einer Studie im Bereich der Computer-gesteuerten Simulationen wurde gezeigt, dass Probanden mehr implizites Wissen durch animiertes grafisches Feedback erhalten haben, im Gegensatz zum textuellen Feedback [16, Seite 5]. Eine weitere Studie hat eine wichtige Eigenschaft von interaktiven Animationen hervorgebracht: „An important property of interactive animation is that it can shift a user’s task from cognitive to perceptual activity, freeing cognitive processing capacity for application tasks.“ [17, Seite 11]. Informationen können also meist besser und effizienter über Animationen vermittelt werden – allen voran, wenn Charaktere im Spiel sind und dies über deren Körpersprache erfolgt [58]. Prozesse in der virtuellen Spielwelt werden dem Spieler durch Beobachten näher gebracht – wenn nötig, ergänzend mit Text oder Sprache. Das visuelle, animierte Geschehen des Prozesses jedoch bleibt die primäre Vermittlung der Information. Prozesse welche nicht über Animationen vermittelt werden bekommen durch ihre Diskrepanz mit den animierten Prozessen eine höhere Aufmerksamkeit als zuvor.

Diese Arbeit beschränkt sich auf die Animationen des anthropomorphen Charakters in Computerspielen. Wie schon öfters erwähnt erlaubt die Körpersprache – die Primärsprache unseres Körpers [13, Seite 9] – das Vermitteln wichtiger Informationen an den Empfänger. Der Empfänger ist immer der Spieler. Eine Unterhaltung zwischen dem Spielercharakter und einem NPC könnte den Eindruck eines exklusiven Informationsaustausches beider Charaktere wecken. Einer der Charaktere könnte der Absender, der andere der Empfänger sein. Eine solche Unterhaltung dient dem Spieler und könnte als eine Inszenierung vom Spiel angesehen werden. Dadurch würde das Spiel dem Spieler (also dem Empfänger) mit Informationen für seine nächste Entscheidung und Handlung versorgen. Aufgrund des gesellschaftlichen Aspektes der Körpersprache wird zusätzlich auf die Interaktion zwischen Spielercharakter und NPCs eingegangen. Die Interaktion zwischen Spieler und Spielercharakter wird gesondert in Abschnitt 4.3 behandelt.

Interaktionen zwischen Charakteren sind im Idealfall vom sozialen Status des jeweiligen Charakters getrieben. Keith Johnstone prägte den Begriff „status transactions“ [9, Kapitel: Status] und beschreibt damit die Interaktion zwischen Charakteren und das Erhöhen oder Senken ihres Status, um das zu bekommen was sie wollen. Beim Status geht es jedoch nicht nur um den Dialog – es beinhaltet auch die Körperbewegung, den Augenkontakt, den persönlichen Bereich und die Handlungen eines Charakters [18, Seite 321]. “[. . .] every sound and posture implies a status [. . .]“ schreibt Keith

Johnstone [9, Seite 72]. Bei Interaktionen zwischen NPCs und Spielercharakter (oder nur zwischen NPCs) ist es somit wichtig die Absichten und den dadurch wechselnden Status von Charakteren zu berücksichtigen und dies in den Animationen zu reflektieren.

Abhängig vom Kontext des Spieles haben die Animationen von Spielercharakter und NPC mehreres gemeinsam. Beide laufen in Echtzeit ab und passen sich den ändernden Gegebenheiten im Spiel an. Der Spielercharakter wird vom Spieler kontrolliert - je nach Laune des Spielers ändert sich beispielsweise die Position des Spielercharakters. Der Spieler kann sich entscheiden, wohin er einen NPC leiten möchte, oder wann er seinen Tagesablauf unterbrechen möchte, um sich mit ihm zu unterhalten. Des Weiteren bestehen die Animationen beider Charaktere aus mehreren Animation Clips, welche je nach Anforderung miteinander vermischt werden. Es ist jedoch genau an diesem Punkt, wo sich die Animationen beider Charaktere im Wesentlichen voneinander unterscheiden. Während der Spielercharakter auf die spontanen und sofortigen Eingabeänderungen des Spielers reagieren muss (und somit die Transitional Animations (siehe Abschnitt 3.1.1) eine wichtige Rolle für das Feedback einer solchen Änderung spielen), ist dies beim NPC nicht der Fall – er wird nicht direkt vom Spieler kontrolliert. Der NPC kann deshalb großzügig auf Animationsprinzipien wie Anticipation zurückgreifen, um seine Bewegungen gut aussehen zu lassen. Auch die Klasse an Gesten, auf welche der Spielercharakter und NPC jeweils zugreifen kann, ist nicht in jeder Situation gleich (siehe Abschnitt 4.1.2).

### NPCs

Die Animationen eines NPC dienen der Informationsvermittlung für den Spieler. Selbst eine einfache Animation wie das Atmen vermittelt die Information eines lebendigen Wesens an den Spieler. Anhand der Körpersprache lässt sich ablesen in welchem Zustand sich ein NPC befindet und lässt auf die sozialen Interaktionsmöglichkeiten schließen (siehe Abb. 4.2). Ist ein NPC einem *freundlich* gesinnt, empfängt er den Spielercharakter vielleicht mit offenen Armen, oder er winkt einem zu, hat ein Lächeln im Gesicht oder er hält eine entspannte Pose ein. Bei einer *desinteressierten* Einstellung dreht der NPC diesem vielleicht den Rücken oder die „kalte Schulter“ zu, der Blickkontakt hält nicht lange an und das gehobene Kinn könnte eine herabschauende Einstellung vermitteln. Eine *feindliche* Einstellung dem Spielercharakter gegenüber könnte durch einen entsprechend bösen Blick signalisiert werden, eine konfrontierende, angespannte Körperhaltung mit scharfem Blick Richtung Spielercharakter oder eine angehobene, drohende Faust würde dem Spieler signalisieren, sich in Acht zu nehmen.

NPC Animationen sind ein wichtiges Mittel um dem Spieler Gameplay Informationen darzustellen, oder anders gesagt, dem Spieler Game Mechanics zu vermitteln (siehe Abb. 4.3). Anhand von Erfahrungen aus dem wirklichen



**Abbildung 4.2:** Malon begegnet Link mit einer freundlichen Begrüßung, der Teilzeitarbeiter wirkt desinteressiert, Darunia wirkt dem Spieler gegenüber im ersten Moment nicht freundlich gesinnt. Spiel: *The Legend of Zelda: Ocarina of Time*.

Leben können wir gewisse Bewegungen von NPCs deuten, neue Entscheidungen treffen und uns entsprechend auf unsere nächste Handlung vorbereiten. In *Shadow of the Colossus* würde beispielsweise das Ausholen einer der Colossi dem Spieler einen Angriff signalisieren (Anticipation), auf welchen er sich kurzfristig vorbereiten kann. Eine typische Reaktion wäre eine zeitlich passende Ausweichaktion. Genres wie *Beat 'Em Ups* (z.B. *Street Fighter*) leben von dieser Anticipation [20, Seite 20]. Ein weiteres Beispiel: Eine entspannte und gedanklich-abwesende Haltung einer Wache im Spiel *Thief* würde dem Spieler Unaufmerksamkeit signalisieren und ihn dazu einladen, die Wache außer Gefecht zu setzen oder sich an ihr vorbei zu schleichen. In vielen Spielen spielt die Silhouette des Charakters (siehe [11, Seite 16]) eine zur Animation ergänzende Rolle um Gameplay Funktionen zu vermitteln, wie es beispielsweise bei *Team Fortress 2* der Fall ist [57].

Christiaan Moleman prägt den Begriff „Emergent Character Expression“ [58] welcher NPCs mehr Intelligenz und Handlungsfähigkeit aufgrund von Animationen zuschreibt. „People like to project human qualities onto things. Perceived emotion suggests intelligence.“ [58]. Dies kann erreicht werden, indem man die Absichten von NPCs verstärkt durch Animationen darstellt. Als Beispiel beschreibt Moleman einen Feind welcher, durch Blickrichtung, einen geeigneten Ort für Deckung suchend, diesen erspät, den Spieler sieht und (vielleicht aus Angst oder Vernunft) zur Deckung flüchtet. Die Blickrichtung vermittelt viel über einen Charakter, deren Anwendung wird in Folge beim Spielercharakter erläutert.

### Der Spielercharakter

Der Spielercharakter ist ein zentraler Aufmerksamkeits- und Orientierungspunkt für den Spieler. Die Animationen des anthropomorphen Spielercharakters haben einen wesentlichen Stellenwert im Computerspiel. Zum einen müssen sie dem Spieler vermitteln, dass er eine lebendige Kreatur kontrol-



**Abbildung 4.3:** Das Ausholen des Feindes gibt dem Spieler Zeit um zu reagieren (*Shadow of the Colossus*), in Kampfspielen wie *Street Fighter 4* muss man Angriffe des Gegners frühzeitig erkennen um diese zu kontern, Silhouetten der Charaktere in *Team Fortress 2* [29], eine gähnende Wache in *Thief: Deadly Shadows*.

liert. Die Bewegungen des Spielercharakters müssen eine Lebendigkeit ausstrahlen (z.B. atmen) und überzeugend wirken (z.B. ein überzeugender Gang, mit richtigem Timing und Gewichtsverlagerung). Zum anderen muss sie dem Spieler eindeutiges und verständliches Feedback zu seinen Eingaben und Handlungen geben. Sieht der Spieler ein Objekt am Boden liegen und möchte es durch Tastendruck aufheben, ist es die Erwartung des Spielers, dass der Spielercharakter sich bücken wird, die Hand nach dem Objekt ausstreckt und es entweder haltet oder einsteckt. Eine Animation, die die Erwartungen des Spielers grundsätzlich entsprechen würde reichen, um diese zu erfüllen.

Die Animationen müssen auch „gut“ aussehen, damit der Spieler eine Freude hat, sich seinen Spielercharakter anzusehen. Zumal es der Spielercharakter ist, den der Spieler die meiste Zeit anschauen muss. Die Bezeichnung „gut“ ist relativ gehalten und meint, wie schon in Abschnitt 3.2 erwähnt, nicht unbedingt realistisch, aber auf jeden Fall überzeugend.

Wahrscheinlich das Wichtigste das der Spielercharakter haben muss, und was ihn animationstechnisch von den Character-Animationen im Film unterscheidet, sind die Übergangsanimationen, oder *Transitional Animations* (siehe Abschnitt 3.1.1). Es sind diese Animationen, welche den definierenden

Aspekt des Spiele Mediums, die Interaktivität, in einer visuellen Feedback Form dem Spieler übermittelt. Sie werden bei einer Änderung der Steuerungseingabe abgespielt und bilden den Übergang von der alten in die neue Aktion. Damit dieser Übergang für den Spieler überzeugend übermittelt werden kann, muss die Transitional Animation die alte und neue Animation fließend und natürlich-wirkend verbinden können. Sie muss visuell andeuten können, dass ein Übergang stattfindet, damit der Charakter nicht steif und unnatürlich wirkt. Hier werden meist Anticipations und Overlapping Actions verwendet, zwei von den Animationsprinzipien, welche in Abschnitt 3.1.2 beschrieben sind. Zur gleichen Zeit aber muss dieser Übergang in einem angenehmen Zeitraum passieren, da der Übergang sich sonst ungewollt träge anfühlen könnte. Hier kommen dann die Aspekte von *Game Feel* und *Responsiveness* ins Spiel, welche im kommenden Abschnitt behandelt werden.

Wie auch bei NPCs können die Animationen des Spielercharakters Gameplay Informationen liefern (siehe Abb. 4.4). Ein gutes Beispiel wäre der Akt des Schleichens. Eine geduckte, langsame und vorsichtige Bewegung des Spielercharakters vermittelt dem Spieler, dass er mit dieser Bewegung (in Kombination mit Sounds) höchstwahrscheinlich weniger auffällt als mit der normalen Gehbewegung. Ein weiteres Beispiel wäre die Kampfhaltung eines Kriegers in einem Mittelalter Spiel. Im Gegensatz zur entspannten Haltung vermittelt diese eine Bereitschaft für den Kampf und den Einsatz von offensiven und defensiven Bewegungen. Jedoch auch innere Zustände, welche das Gameplay beeinflussen, können vermittelt werden. Beispielsweise eine Verletzung, welche dem Spielercharakter Schmerzen bereitet, kann durch das Umfassen des verletzten Körperbereiches gezeigt werden. Eine verhinderte Bewegung oder eine langsamere Reaktionszeit im Kampf wäre die Folge. Das Betreten einer unheimlichen Umgebung könnte im Charakter Unbehagen auslösen. Sein Körper verengt sich und er kreuzt seine Hände vor der Brust. Beispielsweise steigt zwar durch die Anspannung seine Reaktionszeit, er wird dabei aber auch waghalsiger, was durch seine schnellen, aber auch gewagten Angriffe dargestellt werden kann.

Auch der Fortschritt und die Weiterentwicklung des Charakters kann anhand von Animationen, also der Körperbewegung und Gestik, dargestellt werden. Christiaan Moleman [58] schreibt, Kontraste in der Fortschrittsdarstellung dort zu setzen, wo sie Sinn machen und sowohl schrittweise als auch markante Entwicklungen dem Spieler darstellen. Ein Beispiel wäre der Erfahrungswert des Charakters. Während er sich am Anfang ungeschickt verhält (kann zusätzlich durch die Steuerung kommuniziert werden), werden seine Bewegungen mit der Zeit immer gezielter und eleganter. Moleman erwähnt auch den Statuswechsel zwischen Spielercharakter und NPC. Die Erfahrung und Macht des Spielercharakters lässt sich auch durch das Verhalten und die Körpersprache von NPCs darstellen. Ein NPC der auf den Spielercharakter erst hinabschauend wirkte, wird ihm nach einer erfolgreichen Quest mit mehr Respekt begegnen. Wie in Abschnitt 3.2 schon erwähnt, hat Max



**Abbildung 4.4:** Garrett beim Schleichen in *Thief: Deadly Shadows*, Piou Augustus kurz vorm Sterben in *Eternal Darkness: Sanity's Requiem*, die neutrale und die Kampfhaltung von Geralt in *The Witcher*.

von Tettenborn durch eine Modifikation von *Gothic* versucht, Statuswerte und Fortschritt des Spielercharakters u.a. anhand dessen Animationen darzustellen.

Die Lebendigkeit des Spielercharakters und dessen Zugehörigkeit in der virtuellen Welt lässt sich durch Animationen verfestigen. Autonome Körperbewegungen (siehe Abschnitt 4.1.2) geben dem Spieler den ersten Eindruck eines lebendigen Wesens. Weiterhin muss der Spielercharakter, welcher zwar in der virtuellen Welt existiert, aber von außen (vom Spieler) fremdgesteuert wird, eine feste Zugehörigkeit in der Welt vorweisen. Er darf in dieser nicht fremd wirken. Er muss also nicht nur auf die Steuerungsbefehle des Spielers reagieren, sondern auch auf Dinge in der virtuellen Welt eingehen. Wichtig dabei ist, dass der Spieler trotzdem das Gefühl der Kontrolle beibehält, da sonst der Eindruck erweckt wird, der Spielercharakter setzt sich dem Spieler entgegen. In einem Brief an Don Graham schrieb Walt Disney: „[. . .] the mind is the pilot. We think of things before the body does them.“ [64]. Ein effektives Mittel, um dem Spielercharakter etwas Autonomie zu verleihen und ihn mehr in Einklang mit der virtuellen Welt zu bringen, ist seine Blickrichtung verstärkt zu nutzen (siehe [8, Seite 167] und [41]). Indem der Blick seinen Aktionen vorausgeht, oder gewisse Dinge in der Welt seine Aufmerksamkeit

erhalten, lässt dies den Spielercharakter als intelligentes Wesen wirken, welches sich seiner Umgebung bewusst ist. Nicht nur die Blickrichtung, sondern auch die Justierung des gesamten Körpers trägt zur Raumwahrnehmung bei, wie in [66, Seite 1] beschrieben:

‚As you walk out a door and round a corner, your body is already adjusting ten steps before you get to the door. You’re accelerating and decelerating without even realizing it. Your footsteps match perfectly.‘

Physischer Kontakt mit der Welt ist ein weiterer wichtiger Punkt für den Spielercharakter. Ein fester Halt am Boden durch die Füße gibt dem Spielercharakter eine stärkere Präsenz in der Spielewelt. Die Füße müssen sich dem unebenen Boden anpassen und auch das *sliding feet* Problem (siehe Abschnitt 3.3) muss vermieden werden. Nähert man sich einer Wand oder einem Hindernis, können sich die Arme des Spielercharakters ausstrecken. Dies würde zeigen, dass er die Wand oder das Hindernis registriert und wahrgenommen hat. Christiaan Moleman beschreibt dieses Verhalten als „Contextual Anticipation“ [53].

Die Animationen von Spielercharakter und NPCs tragen viel zum Verständnis und zur Vermittlung von Informationen vom Zustand eines Charakters bei. Im speziellen kann der Spielercharakter dadurch dem Spieler seinen derzeitigen inneren Zustand vermitteln, was einen Einfluss auf seine Handlungsentscheidungen hat. Aufgrund der speziellen Beziehung zwischen Spieler und Spielercharakter kommt es auf die Situation, das Genre und die Designentscheidung des Spieles an, wann welche Arten von Animationen abgespielt werden sollen, um nicht im Kontext des Spieles deplatziert zu wirken. Im Endeffekt ist es das Ziel, durch die Interaktion von Spieler und Spielercharakter, beide näher zu bringen. Im Idealfall sollen so viele Spielelemente (z.B. die Character-Animation) wie möglich verwendet werden, welche dem Spieler ohne Erklärung Informationen vermitteln können. Dies würde den Immersionsgrad steigern und somit den Spieler leichter in die virtuelle Welt hineinziehen.

### 4.3 Interaktion mit dem Spielercharakter

Die Interaktion des Spielers mit dem Spielercharakter (oder im Allgemeinen, mit dem Spiel) ist das wichtigste Alleinstellungsmerkmal für Computerspiele im Gegensatz zu anderen bewegten Bildern wie Film oder Fernsehen. Der Akt der Interaktion mit dem Spielercharakter beinhaltet mehrere Schichten, durch welche Information fließen muss. Angefangen beim Spieler und seinem Controller, hinein in das digitale Medium und den dortigen Akteuren, und

wieder zurück zum Spieler in der Form von Sinnesreizen wie Bild und Ton. Um diesen Informationsfluss formal darzustellen, wird als Basis das Interaktionsmodell von Jesse Schell verwendet. Neben den verschiedenen Schichten des Modells wird auf das virtuelle und das physische Interface eingegangen und deren Merkmale beschrieben. Anschließend wird erläutert, wie die Animation des Spielercharakters dazu beitragen kann, die Distanz zwischen Spieler und Spiel zu verkleinern. Dies würde dem Spieler erlauben, stärker in die virtuelle Welt einzutauchen.

### 4.3.1 Das Mapping Modell

Eines der Vorteile vom digitalen Medium Spiel ist die Manipulation von bewegten Bildern. Während in Film und Fernsehen die Anordnung dieser Bilder als lineares Endprodukt dem Zuschauer präsentiert wird, ist dies bei Computerspielen anders. Der Zuschauer bzw. Spieler ist aktiv an der fortschreitenden Handlung im Spiel beteiligt. Neben Entscheidungen im Spiel selbst muss er sich regelmäßig entscheiden, welche Eingaben er am Eingabegerät ausführt. Er denkt daher in Aktionen, wodurch jede Handlung eine eigene Sammlung an Bildern produziert. Dem Spieler wird somit eine gewisse Entscheidungsmacht übertragen. Er kann den zeitlichen Ablauf und die Handlungen im Spiel für sich selbst entscheiden (im Rahmen des Machbaren) und somit seine eigene Erfahrungsgeschichte kreieren.

Damit der Spieler das Geschehen in einer virtuellen Welt beeinflussen kann, muss eine Verbindung zwischen Spieler und Welt aufgebaut werden. Die Absichten des Spielers müssen dem Spiel kommuniziert werden, und das Spiel muss im Gegenzug das Resultat dieser angewandten Absichten dem Spieler vermitteln. Um diesen Kommunikationsfluss darstellen zu können, wird im Rahmen dieser Arbeit das Interaktionsmodell von Jesse Schell verwendet [18, Seite 225], fortan als das *Mapping Modell* bezeichnet. Es gibt andere Interaktionsmodelle wie beispielsweise das *Execution-Evaluation Cycle* Modell von Donald Norman [15, Seite 45], diese sind jedoch eher der HCI<sup>1</sup> von klassischen Applikationen für den produktiven Einsatz zugeschnitten. Während solche Applikationen eher für die Vereinfachung von Aufgaben für den Benutzer entwickelt worden sind, fokussieren Spiele eher auf den Prozess der Interaktion mit der Welt anstatt dem Endresultat des Prozesses [2]. Auch setzen Spiele dem Spieler intern Aufgaben und Hindernisse die es zu bewältigen gibt, im Gegensatz zur produktiven Applikation, welche Aufgaben bearbeitet und dem Benutzer die Interaktion so einfach wie möglich gestaltet [2].

Das Modell von Schell ist ein allgemeiner Ansatz, um den Interaktionsfluss in Spielen zu beschreiben. In seiner derzeitigen Form beinhaltet das Modell den Spielercharakter ganz allgemein in der *virtuellen Welt* Schicht.

---

<sup>1</sup>Human-Computer Interaction

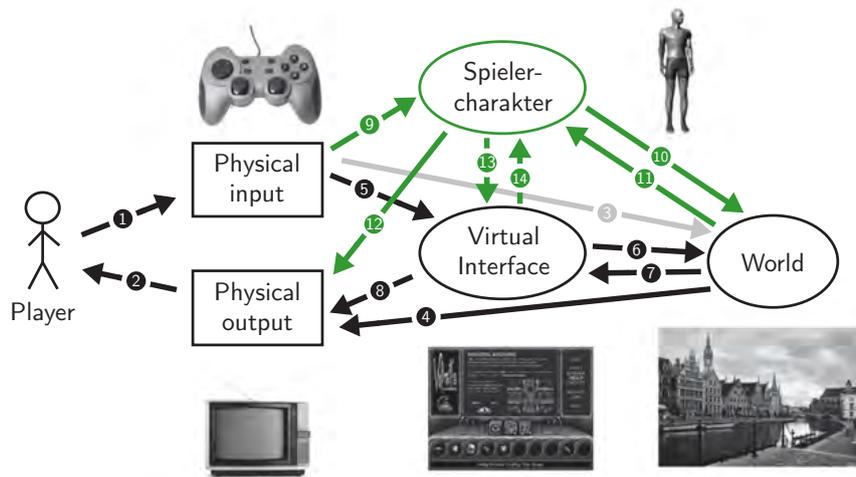


Abbildung 4.5: Das modifizierte *Mapping Modell*

Da sich diese Arbeit mit dem anthropomorphen Spielercharakter beschäftigt, wird dieses Modell um eine weitere Schicht, dem *Spielercharakter*, erweitert und spezifiziert (siehe Abb. 4.5). Dies ist nötig, um im späteren Verlauf jene Stellen identifizieren zu können, welche die Distanz zwischen dem Spieler und dem Spielercharakter reduzieren könnte und somit die Immersion des Spielers steigen lässt. Wichtig dabei ist, dass das modifizierte Mapping Modell für Spiele in der Third-Person Perspektive mit einer kontrollierbaren Kamera um den anthropomorphen Spielercharakter ausgelegt ist.

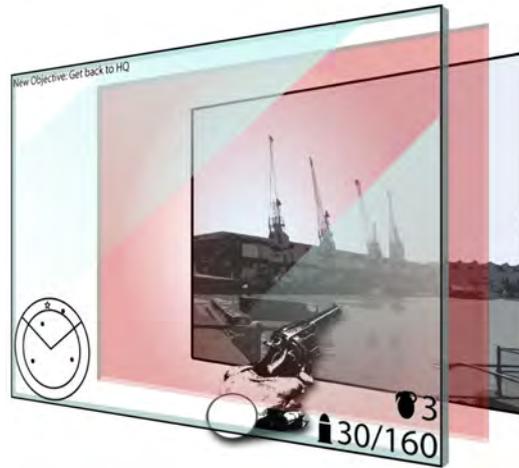
Es folgt eine kurze Erklärung zu dem Mapping Modell und die Zusammenhänge aller Schichten (siehe auch Abb. 4.6).

#### Schichten:

**Player:** Der Spieler beteiligt sich aktiv am Interaktionsprozess durch seine Entscheidungen und Handlungen, welche er durch das physische Interface zum Ausdruck bringt.

**Physisches Interface:** Das physische Interface besteht aus dem physischen Input (z.B. Controller) und dem physischen Output (z.B. der Bildschirm, Lautsprecher). Mit dem Controller kann der Spieler mit dem Spiel kommunizieren, anhand des Bildschirms sieht er das Resultat seiner Interaktion. Für eine genauere Ausarbeitung zum Thema Physisches Interface, siehe Abschnitt 4.3.2.

**Virtuelles Interface:** Das virtuelle Interface ist die Schicht zwischen dem physischen Interface und der virtuellen Spielewelt. Es beinhaltet Elemente,



**Abbildung 4.6:** Eine schematische Darstellung der Interface-Schichten.  
Bildquelle: [4].

welche nicht direkt Teil der Spielwelt sind, welche aber dem Spieler anhand von symbolischen Elementen gewisse, quantifizierte Informationen übermittelt. Ein Beispiel dafür wäre die Lebensanzeige in *Super Mario 64*. Auch bietet es dem Spieler gewisse Interaktionsmöglichkeiten an, wie beispielsweise das Drücken von virtuellen Knöpfen oder das Steuern der Kamera eines jeden Third-Person Spieles, mit welcher der Spieler die virtuelle Welt sieht [18, Seite 225]. Weitere Informationen zum virtuellen Interface sind im Abschnitt 4.3.2 zu finden.

**Spielercharakter:** Der Spielercharakter ist der Avatar des Spielers. Durch ihn interagiert der Spieler mit der Welt und erfährt nicht nur Informationen über diese, sondern auch über den Zustand seines Avatars.

**Virtuelle Welt:** Die virtuelle Welt beinhaltet alle Landschaften, Objekte, Charaktere und weitere Entitäten, welche dem Spieler eine virtuelle Realität simulieren. Der Spieler interagiert durch seinen Spielercharakter mit dieser Welt. Seine Entscheidungen und Handlungen beeinflussen und verändern sie auf Basis seiner Vorstellungen – natürlich unter Berücksichtigung von virtuellen und technischen Einschränkungen.

### Mapping:

Das *Mapping* beschreibt den Fluss von Information von einer Schicht zur anderen. Wichtig ist dabei zu beachten, dass diese Information während des Transfers einer Transformation unterlegen ist [18, Seite 224]. Ein Beispiel

soll dies erklären: Im Programmcode ist spezifiziert, ab welchem Neigungsgrad des Analog-Sticks eine *Geh-* oder *Renn-*Animation abgespielt werden soll. Je nach aktiver Neigung des Analog-Sticks wird der Neigungswert dem Programmcode zugeliefert. Abhängig vom definierten Schwellenwert führt das Programm einen Stück-Code aus, welcher die zutreffende Animation abspielt. Diese Animation wird auf dem Bildschirm dargestellt und beim Spieler im Auge abgebildet. Somit fließt die Information von der analogen Eingabe zur digitalen Konvertierung und Berechnung, von dort zum Bildschirm und via Lichtausstrahlung zurück ins Auge des Spielers.

William Lidwell et al. [12, Seite 128] beschreibt Mapping im Allgemeinen als die Beziehung zwischen der Steuerung und dessen Auswirkung oder resultierenden Bewegung. Folgendes Zitat von [12, Seite 128] fasst die Definition von Mapping gut zusammen:

„Good mapping is primarily a function of similarity of layout, behavior, or meaning. [...] When the effect corresponds to expectation, the mapping is considered to be good or natural. When the effect does not correspond to expectation, the mapping is considered to be poor.“

In Folge werden die Mapping Prozesse zwischen den Schichten des modifizierten Mapping Modells erläutert:

1. Hier findet der Informationstransfer seinen Ursprung. In Bezug auf den klassischen Gamecontroller wird eine Entscheidung des Spielers von den Nerven zu den Muskeln der Hand transportiert. Die Bewegungen der Finger schließen den Transfer der Information vom Körper hinaus zu dem Eingabegerät ab.
2. Die formulierte Informationsrückmeldung für den Spieler. Die Resultate der Aktionen des Spielers werden durch diverse Geräte (z.B. Bildschirm oder Lautsprecher) in der Form von Sinnesreizen dem Spieler vermittelt. Die drei wichtigsten Kanäle sind der visuelle, auditive und haptische Kanal. Anhand dieser Resultate formuliert der Spieler neue Entscheidungen für den weiteren Fortschritt im Spiel.
3. Dieser Informationstransfer ist durch das Hinzufügen der Spielercharakter Schicht obsolet geworden, da der Spieler nun direkt den Spielercharakter steuert.
4. Änderungen in der Spielwelt werden dem Spieler als Feedback vermittelt. Ein Beispiel wäre einem NPC zuzuschauen, wie er von Punkt A nach Punkt B geht.
5. Neben der Steuerung des Spielercharakters ist dem Spieler die Möglichkeit gegeben, direkt das virtuelle Interface anzusteuern. Ein Beispiel wäre das Aufrufen des Hauptmenüs durch die *Start* Taste.
6. Spezielle Stellen im Spiel können ein virtuelles Menü aufrufen, welches eine Interaktion mit diesem verlangt. Das Resultat wäre eine Änderung

in der Spielewelt. Zum Beispiel das Entriegeln eines Türschlosses durch die korrekte Bewegung von Dietrichen. Das Resultat dieser Aktion wäre das Öffnen einer Tür.

7. Hier werden Änderungen in der Spielerwelt auf dem virtuellen Interface dargestellt. Ein einfaches Beispiel wäre das Einsammeln von hundert Münzen. Jede eingesammelte Münze verschwindet vom Spiel, die Münzanzeige rechts oben im Bildschirm erhöht sich jedoch inkrementell.
8. Beim Vorhandensein des virtuellen Interfaces muss dies auch dem Spieler dargestellt werden. Hier werden üblicherweise audiovisuelle Elemente als Feedback vermittelt, wie z.B. das animierte Drücken eines Knopfes und dazu ein entsprechend passender Soundeffekt.
9. Der Spieler steuert den Spielercharakter direkt mit dem Controller. Hier spielt das Mapping der Tasten eine Rolle: Läuft der Spielercharakter langsamer oder schneller, abhängig davon wie stark der Analog-Stick geneigt wird [18, Seite 225]?
10. Die Handlungen des Spielercharakters ändern die Welt um ihn herum. Beispielsweise könnte ein Tritt gegen den Steinhaufen auf einem Berg der Auslöser einer Steinlawine gewesen sein.
11. Auch die Welt, in welcher der Spielercharakter lebt, kann ihn beeinflussen. Ein Monster kann beispielsweise den Spielercharakter angreifen und ihm Schaden zufügen.
12. Die Bewegungen und Handlungen des Spielercharakters werden auf die für den Spieler relevanten Feedbackkanäle (z.B. Bildschirm, Lautsprecher) gemapped.
13. Gewisse Attribute des Spielercharakters werden meist auf das virtuelle Interface für den Spieler gemapped. Ein klassisches Beispiel ist die Lebensanzeige, welche den Gesundheitszustand des Spielercharakters quantitativ darstellt.
14. Gewisse Aktionen für den Spielercharakter werden über das virtuelle Interface vermittelt. Quick Time Events, welche das Drücken einer visuell-vorgegebenen Reihenfolge von Tasten verlangen, sind ein gutes Beispiel für diese Art von Mapping.

### 4.3.2 Interface

Damit der Spieler effektiv mit dem Spielercharakter interagieren und kommunizieren kann, braucht es eine physische und virtuelle Interface-Schicht, um den Fluss an Informationen zu abstrahieren und für die jeweilige Schicht kompatibel zu machen. Da der Spieler mit jeder Aktion eine gewisse Erwartung verbindet, muss dieser Abstraktionsprozess auf einer für den Spieler verständlichen, nachvollziehbaren und gewissermaßen natürlichen Art passieren. Eine Analogie soll dies verdeutlichen:

Der Mensch selbst besteht aus mehreren Schichten, welche miteinander kommunizieren und im Informationsaustausch miteinander stehen. Der Gedanke, die Hand winken zu lassen, sendet eine Nachricht vom Gehirn zu den Nerven. Diese wiederum leiten die Muskeln dazu an, den Arm zu heben und die Hand winken zu lassen. Der Mensch ist sich dieses schrittweisen Prozesses nicht bewusst, es wirkt unverzögert, erfüllt aber die persönliche Erwartung, wie der Prozess des Winkens stattfinden soll. Kleine Details wie die zentimetergenaue Höhe des Armes werden abstrahiert und spielen keine Rolle in der Erwartungserfüllung.

Der Interaktionsprozess mit Spielen ist eine Erweiterung des Informationsflusses vom Körper in die Außenwelt. Diese äußeren Interface-Schichten sind dem Körper jedoch fremd in der Hinsicht, dass ihr Mapping zueinander nicht dem natürlichen Mapping zwischen Gehirn, Nerven und Muskeln entspricht. Aus Sicht der Science Fiction wäre eine Verbindung der Nerven direkt mit dem Spielercharakter der wohl effizienteste und schnellste Weg der direkten und effektivsten Interaktion.

Es ist somit das Ziel der physischen und virtuellen Interface-Schicht, dem Spieler so **effektiv** wie möglich bei dem Mapping des Informationsflusses entgegenzukommen. Dies basiert darauf, dass sich die Interaktion für den Spieler „gut anfühlen“ soll, etwas das schwer mess- und artikulierbar ist [71]. Steve Swink bezeichnet dieses Gefühl als *Virtual Sensation* oder *Game Feel* und beschäftigt sich in [70–72] genauer mit dem Thema.

### Physische Interface

Das *physische Interface* besteht aus zwei Hauptkomponenten: dem physischen Input und dem physischen Output. Um mehrere Sinne des Menschen in Anspruch zu nehmen, werden für den Output mehrere Ausgabegeräte verwendet. Die visuellen Sinnesreize sind eines der wichtigsten Feedback Informationen von Computerspielen – sie werden meist über einen Bildschirm geliefert. Akustisches Feedback wird über Lautsprecher vermittelt, während in manchen Fällen taktile Reize durch eine *Rumble* Funktion im Controller verschickt werden. Bei der Entwicklung der Ausgabegeräte erkennt man einen Trend, um den Zuschauer oder Spieler stärker in das digitale Medium eintauchen zu lassen. Die Auflösung von Bildschirmen steigt, sowie auch deren Größe. Um den Raumklang zu verbessern werden immer mehr Lautsprecher mit steigender Klangqualität eingesetzt. Aktuell wird auch versucht, mit 3-D Technologie den Spieler näher ins Spielgeschehen einbeziehen zu lassen. Das Hauptaugenmerk jedoch liegt im Rahmen dieser Arbeit bei dem Eingabegerät.

Der klassische Gamecontroller, sowie auch Tastatur und Maus, sind jahrelang die Standard Eingabegeräte für Spiele gewesen. Sie abstrahieren und mappen Eingabefunktionen auf binäre Knöpfe und Analog-Sticks ab, wodurch nach einer kurzen Lernphase die Bedienung der Knöpfe scheinbar au-

tomatisch erfolgt. Mit zunehmend komplexen Spielen stieg auch die Anzahl an Knöpfen und Tastenkombinationen, die es zu beherrschen galt. Für viele Menschen war dieser Lernaufwand zu hoch, weshalb komplexe Spiele für sie nicht zugänglich waren. Um neue und einfachere Interaktionsmöglichkeiten zu schaffen brachte *Nintendo* Ende 2006 mit der *Wii* Konsole eine neue Form des Eingabegerätes für den Massenmarkt heraus. *Microsoft* und *Sony* folgten jeweils mit *Kinect* und *Playstation Move* ein paar Jahre später.

Deren Prinzip baut darauf auf, dass mehr Menschen Computerspiele spielen, wenn das Eingabegerät einfacher und intuitiver zu bedienen ist. Da die Bewegungen des Körpers eine natürliche Kommunikationsmethode des Menschen ist, wäre es angebracht, die Interaktion mit dem Spiel durch die Körperbewegung voranzutreiben. Eines der Probleme von diesen neuen Interaktionsmöglichkeiten jedoch ist die Komplexität und Interpretation der Bewegungen im Spiel [76, T=00:06:00]. Im Gegensatz zu den klassischen Eingabegeräten erfassen die neuen *Motion Controller* Informationen nicht nur aus einem binären Knopfdruck oder einem richtungsweisenden Analog-Stick. Sie bekommen die Informationen zusätzlich aus einem dreidimensionalen Raum aufgrund von Orientierung, Position, Beschleunigung und Geschwindigkeit des Motion Controllers oder des Körpers vom Spieler. Zum einen wird dadurch die technische Umsetzung und das Mapping der Daten komplizierter. Zum anderen erschwert es den Lernprozess für den Spieler, da dieser gewisse Aktionen mit einer speziellen Bewegung vollbringen muss – für viele Menschen kein triviales Unterfangen [76, T=00:14:00]. Hier wird nicht nur das Drücken eines Knopfes verlangt, sondern die oben erwähnten Faktoren im dreidimensionalen Raum erfasst. Bei einer 1:1 Umsetzung der Bewegungen des Spielers im Spiel ist meist auch das Resultat der Bewegung enttäuschend für den Spieler, da die Bewegung in seiner Fantasie meist nicht der real-ausgeführten Bewegung entspricht [76, T=00:07:00]. Auch eine physische Gegenwirkung zu den Aktionen im Spiel bleibt aus. Die *Kinect* versucht zwar den Controller ganz aus dem Spiel zu nehmen und nur den Körper des Spielers zu erfassen, aber auch dieser Ansatz weist Mängel auf. Bei allen drei Motion Controls ist es die Latenzzeit der erfassten Bewegungen, welches ein wesentliches Hindernis für eine flüssige Darstellung am Bildschirm ist. Eine wichtige Berücksichtigung ist auch der Fakt, dass für viele Menschen eine Ganzkörperbewegung, um Spiele zu spielen, als anstrengend empfunden wird.

Was im ersten Moment als eine natürliche und einfache Weiterentwicklung des klassischen Controllers aussieht, bringt weitere Probleme mit sich, die den Immersiongrad des Spielers wesentlich stören. Solange die Verwendung eines klassischen Controllers für den Spieler gemüthlicher und weniger anstrengender ist, wird dieser nicht ersetzt. Aus diesem Grund wird im Rahmen dieser Arbeit weiterhin der klassische Gamecontroller verwendet und berücksichtigt.

Das physische Interface ist einer der wichtigsten Komponente im Inter-

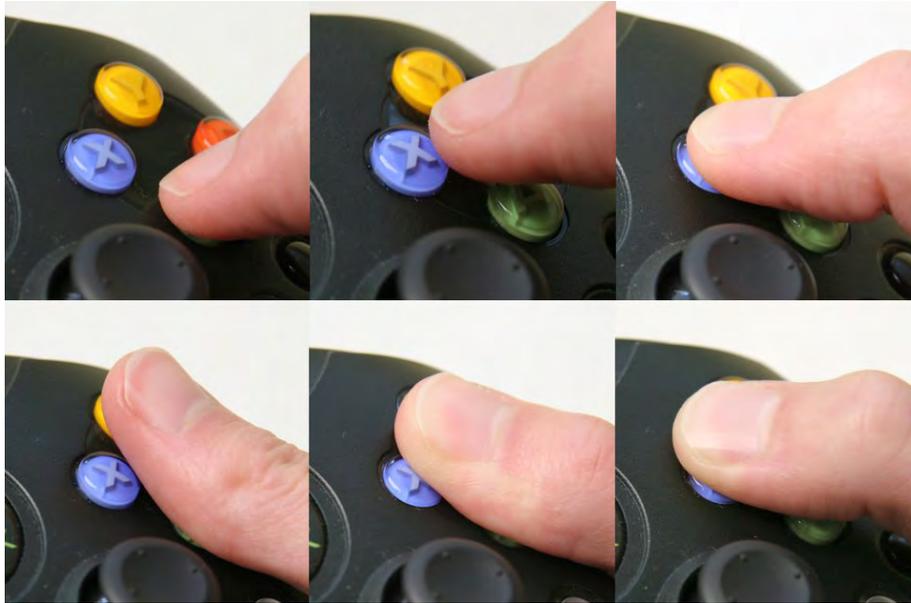


**Abbildung 4.7:** Es wird verstärkt auf die Ergonomie des Eingabegerätes geachtet: *NES* Controller (1986), *Gamecube* Controller (2002), *Xbox 360* Controller (2006). Bilderquelle: [35].

aktionsprozess zwischen dem Spieler und dem Spiel. Ohne die aktive Beteiligung des Spielers entsteht keine Interaktion mit dem Spiel, wodurch jeglicher Fortschritt verhindert wird. Diese Beteiligung kann nur durch ein Eingabegerät stattfinden. Aus diesem Grund muss das Gerät für dessen geplante Funktion maßgeschneidert sein. Das Eingabegerät muss angenehm in der Hand liegen und die Finger müssen ohne Mühe oder anstrengender Dehnung alle Knöpfe erreichen, also haptisch ansprechend sein. Das Material des Gerätes muss sich „gut“ anfühlen und das Drücken und Betätigen aller Knöpfe muss ein angenehmer Prozess mit haptischem und akustischem Feedback sein. Jesse Schell schreibt diesbezüglich: „The ideal interface becomes invisible to the player letting the player’s imagination be completely immersed in the game world.“ [18, Seite 227]. Das Ziel ist es also, das real existierende, erforderliche physische Gerät dem Spieler transparent werden zu lassen. Auch wenn anfangs eine Lernphase bezüglich des Umganges mit dem Gerät unvermeidlich ist, darf es in Folge dem Spieler nicht als Störfaktor während seiner Spielzeit auffallen. Dies gilt auch für das virtuelle Interface. Jegliche Hand- und Fingerbewegungen, welche Anstrengungen durch verkehrtes Mapping oder Layout verursachen, ziehen den Spieler aus dem Spiel hinaus und brechen seine Immersion ab. Feedback Prozesse, welche als Basis die Aufmerksamkeit und Immersion des Spielers beanspruchen, würden somit den Spieler nicht optimal, oder im schlimmsten Fall gar nicht erreichen. Es ist somit wichtig, potentielle Hürden in diesen ersten Schichten im Interaktionsmodell zu reduzieren oder zu vermeiden, damit der Spieler im Idealfall ununterbrochen im Spiel eingetaucht bleibt.

Ein wichtiger Aspekt eines guten Controllers ist, neben der ergonomischen Form (siehe Abb. 4.7), die Anordnung der Knöpfe. Etwas allgemeiner beschreibt es William Lidwell et al. [12, Seite 128]:

„Position controls so that their locations and behaviors correspond to the layout and behavior of the device. Simple control-effect relationships work best.“



**Abbildung 4.8:** Erste Reihe: *Precise Thumb*, Zweite Reihe: *Sloppy Thumb*. Bilderquelle: [24].

Die Knöpfe müssen so angeordnet sein, dass sie für den Spieler angenehm erreichbar sind. Auch die Form und Größe der Knöpfe spielen eine wichtige Rolle in der Art, wie ein Spieler den Controller hält und die Knöpfe betätigt. Mick West [24, Seite 20] erwähnt diesbezüglich den *Precise Thumb* und den *Sloppy Thumb* Player (siehe Abb. 4.8). Der *Precise Thumb* Player verwendet die Spitze seines Daumens, um die Tasten auf präzise Art zu betätigen. Der *Sloppy Thumb* Player lässt seinen Daumen, bzw. den Rücken seines Daumens, auf allen Tasten ruhen. Beim Betätigen einer Taste wird der Daumen geneigt. Mick West behandelt das Thema eher aus Sicht der technischen Mehrdeutigkeit, die größtenteils durch Sloppy Thumb Verhalten und Betätigen von mehreren Tasten gleichzeitig entsteht. Für den Spieler ist der Sloppy Thumb die weniger anstrengende und daher entspanntere Art mit dem Controller zu interagieren. Der *Gamecube* Controller von *Nintendo* baut auf diesem Prinzip auf, wie man anhand der Größe des A Knopfes, und der Lage der Y und X Knöpfe sehen kann (siehe Abb. 4.7) [24, Seite 20].

Um dem Mapping von Spielfunktionen auf Knöpfe entgegenzukommen, können metaphorische Elemente verwendet werden. Das virtuelle Resultat eines Eingabebefehls im Spiel kann dem Spieler durch das Gefühl der Betätigung sowie der Größe, Form und Lage eines Knopfes näher gebracht werden. Beim Betätigen eines Knopfes wird dem Spieler ein haptisches Gefühl vermittelt, das seinen Gemütszustand beeinflussen kann [47]. Dieses Gefühl kann man im Spiel anhand des Spielercharakters reflektieren. Als Beispiel



**Abbildung 4.9:** Der Z-Trigger des *N64* Controller. Bilderquelle: [35].

gibt Mike Jungbluth [47] den Analog-Stick vom Xbox 360 Controller an. Abgesehen von der richtungsweisenden Funktion hat der Analog-Stick auch eine digitale Knopf Funktion, indem man in der Ausgangsposition auf den Stick drückt. Dies fühlt sich sehr klobig und ungeschickt an. Diese Funktion könnte beispielsweise bei einem Balanceakt auf einem Seil verwendet werden:

Ein Spieler muss mit seinem Spielercharakter einen Abgrund überqueren, über welchem ein Drahtseil befestigt ist. Auf dem Seil stehend kippt der Spielercharakter langsam immer wieder zur rechten oder linken Seite. Der Spieler kann dem entgegenwirken, indem er den Analog-Stick in die entgegengesetzte Richtung drückt. Während mit der richtungsweisenden Funktion des Analog-Sticks die Balance gehalten wird, können gewisse Aktionen (z.B. das Greifen nach einem fliegenden Objekt) während des Balance Aktes nur durch Drücken des Analog-Sticks durchgeführt werden. Da im realen Leben der Akt der Balance den menschlichen Körper gänzlich beansprucht, würden zusätzliche Aktionen den Fokus und die Konzentration des Balancierenden reduzieren und ihn möglicherweise ungeschickt wirken lassen.

Ein weiteres Beispiel wäre der Z-Trigger vom *Nintendo 64* Controller, der an den Abzug einer Pistole erinnert (siehe Abb. 4.9). Spiele wie *Turok: Dinosaur Hunter* und *Perfect Dark* verwenden diesen Knopf als Auslöser ihrer Schusswaffen. In 3-D Spielen mit einem sichtbaren Spielercharakter in der Third-Person Perspektive wird oft der Analog-Stick verwendet, um die Bewegungsrichtung (nicht Position) des Spielercharakters zu bestimmen.

Ein weiterer wichtiger Punkt bei der Interaktion mit dem Controller ist die *Responsiveness* der Eingabebefehle: wie lange dauert es, bis das Resultat des Eingabebefehles visuell oder akustisch erkennbar ist? Vieles vom „Gefühl“ des Spieles hängt von dieser Latenzzeit ab [78]:

„A game with poor responsiveness will be described as being "sluggish", "unresponsive", "floaty" or "sloppy". A better game might be referred to as "tight" or "responsive".“

Die Responsiveness ist somit ein wichtiger Faktor für die Immersion des Spielers. Sie ergänzt aber auch das Gefühl, welches der Spieler beispielsweise durch die Animationen seines Spielercharakters vermittelt bekommt. Ein flinker Charakter würde gefühlsmäßig eine schnellere Reaktion vorweisen als ein verwundeter oder träger Charakter. Wichtig ist hier, die technische und die gezielt-ingesetzte Responsiveness zu unterscheiden. Die Responsiveness kann somit als Gestaltungsmittel verwendet werden, auch wenn sie in erster Linie bei einer zu hohen Latenzzeit als technische Hürde angesehen wird. Mick West hat sich mit den technischen Aspekten dieses Problems beschäftigt und wie man Responsiveness messen kann. Er schlägt als praktische Lösung vor, den Tastendruck und das darauf folgende Resultat am Bildschirm abzufilmen (60fps<sup>2</sup>) [77]. Eine Frame für Frame Analyse der Aufnahme würde die Latenzzeit verdeutlichen. Benjamin Heckendorn hat dieses Konzept ergänzt und für *Infinity Ward* einen eigenen Controller gebaut, der das Messen der Latenzzeit unterstützen soll [44]. Mick West unterscheidet in seinem Artikel [77] zwischen Spielen, welche mit 30fps oder 60fps laufen. Er klärt auf, warum Spiele mit 60fps eine mögliche, niedrigere Response time haben als Spiele mit 30fps. Ergänzend zu seinem Artikel bezüglich der technischen Aspekte von Responsiveness [78] weist er auf die praktische untere Grenze von 3/60 einer Sekunde, welche Spiele wie *Guitar Hero* haben. Richard Leadbetter [49, Seite 3] hat im Anschluss an die Artikel von Mick West mehrere Teste an diversen modernen Spielen durchgeführt mit unterschiedlichen Latenzzeit Resultaten (siehe Tabelle 4.1). Das Spielprinzip bei Spielen wie *Guitar Hero* oder *Street Fighter* basiert auf einer schnellen Response time, weshalb dort die Werte niedrig ausgefallen sind. Er ist zu dem Schluss gekommen, dass die Responsiveness ein wichtiger Aspekt für das Spielgefühl ist, weshalb Spieler gegenüber Spielen wie *Call of Duty* sehr empfänglich sind [49, Seite 3].

Mike Jungbluth beschreibt in einem ausführlichen Beispiel, wie durch das Mapping und Tasteneingaben am Controller die Persönlichkeiten der *Street Fighter* Charaktere (siehe Abb. 4.10) hervorgebracht wurden [47]:

„Ryu is meant to be a controlled personality, with his emotions kept in check leading a life of modest means. His moves and fight style reflect this with clean and simple sweeping motions that work best when the player is conscious of his opponent to calmly react and control the space. [...] Then there is Blanka, who is meant to be wild and has a move that actively requires the player to abandon everything except hammering violently

---

<sup>2</sup>Frames Per Second

**Tabelle 4.1:** Latenzzeit diverser Spiele. Auszug aus der Tabelle von [49, Seite 3].

| <i>Game</i>                    | <i>Latency Measurement</i> |
|--------------------------------|----------------------------|
| Call of Duty 4: Modern Warfare | 67ms–84ms                  |
| Guitar Hero: Aerosmith         | 67ms                       |
| Grand Theft Auto IV            | 133ms–200ms                |
| Street Fighter IV              | 67ms                       |
| Unreal Tournament 3            | 100ms–133ms                |



**Abbildung 4.10:** *Ryu, Blanka und Zangief* aus *Street Fighter 4*. Bilderquelle: [30].

on one button. That electric feeling comes through the kinetic press of the buttons, and matches perfectly with his appearance and character. [...] Zangief if a showy, brash wrestler and his input commands of large circular commands matches that. [...] All of these characters are instantly recognizable in personality, appearance and play style and it is no coincidence that players are so attached to their favorite. What the player is doing to the controller matches on multiple levels to what the character is doing on the screen.’

### Virtuelles Interface

Das virtuelle Interface ist eine virtuelle Schicht innerhalb des digitalen Raumes, welche anhand von mehreren Elementen die Kommunikation zwischen Spieler und Spielwelt vereinfacht. Das Interface besteht aus zwei wichtigen



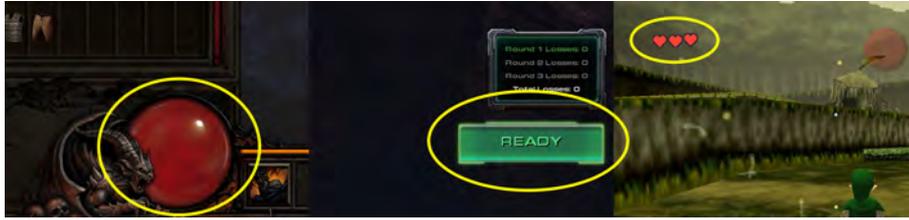
**Abbildung 4.11:** Der Grad der Kälte wird in *Cryostasis* mit einer Eisvignettierung dargestellt. In *Call of Cthulhu* wird der Geisteszustand durch verschwommene Effekte dargestellt [31].

Komponenten.

Die wichtigste Komponente dieser Schicht ist die virtuelle Kamera, durch die der Spieler Einblick in die digitale Welt und seinem Spielercharakter bekommt. Die Kamera ist in jedem Spiel vorhanden, je nach Genre ändert sich dessen Lage, Sicht und Interaktionsmöglichkeit (siehe Abschnitt 2). Das animierte Bewegen der Kamera und die Anwendung von Post-Processing Effekte sind weitere Mittel, um dem Spieler Feedback zu liefern (siehe Abb. 4.11).

Die zweite Komponente ist die Sammlung an bleibenden grafischen Symbolen, Texten, Bildern und deren Animationen, welche rein der Informationsvermittlung für den Spieler dienen. Zusätzlich werden auch akustische Elemente verwendet, um das visuelle Feedback zu ergänzen. Die visuellen Elemente bestehen traditionell meist aus 2-D Texturen, vermehrt werden auch 3-D Objekte in modernen Spielen verwendet. Diese Ansammlung an Elementen wird kollektiv auch oft als HUD bezeichnet und dient meist der Statusvermittlung des Spielercharakters.

Diese Elemente quantifizieren oder abstrahieren Informationen aus der Spielewelt, welche für den Spieler wichtig sind oder ihm eine Übersicht über gewisse Prozesse verschaffen soll. Oft werden symbolische Elemente verwendet um Zeichen darzustellen, welche den Vorstellungen des Spielers erwartungsgemäß entsprechen (siehe Abb. 4.12). Ein Knopf mit dem Schriftzug „Ready“ würde dem Spieler vermitteln, dass durch Drücken des virtuellen Knopfes etwas geschehen wird, was seine Zustimmung verlangt. Da das Konzept des Knopfdrucks dem Menschen schon seit dem Babyalter bekannt ist [76, T=00:03:00], ist dies eine Funktion oder Mechanik, auf welche er sich beziehen kann. Auch metaphorische Elemente werden verwendet, um gewisse Informationen darzustellen (siehe Abb. 4.12). Ein Beispiel wäre die Lebensanzeige von *Diablo 3*. Verwendet wird ein Gefäß aus Glas, welches eine rote Flüssigkeit beinhaltet. Dies könnte der „Lebenssaft“ oder, aufgrund der rötlichen Farbe, das Blut des Spielercharakters darstellen. Mit jedem Schadenspunkt, den der Spielercharakter erhält, wird die Flüssigkeit weniger, was



**Abbildung 4.12:** Die Lebensanzeige in *Diablo 3* [32], das Signalisieren der Bereitschaft per Knopfdruck in *Starcraft 2*, die Lebensanzeige in der Form von Herzen in *The Legend of Zelda: Ocarina of Time*.



**Abbildung 4.13:** *Starcraft 2*, *Amnesia: The Dark Descent*.

den Verlust von Blut entspricht. Bei *The Legend of Zelda: Ocarina of Time* werden kleine Herzsymbole als Lebensanzeige verwendet. Das Herz steht in diesem Zusammenhang als ein Zeichen von Leben. Wenn der Herzschlag aussetzt (also im Spiel keine Herzsymbole mehr vorhanden sind), bedeutet dies den Tod des Spielercharakters.

Die verwendeten Komponente und Elemente des virtuellen Interfaces hängen stark vom Genre des jeweiligen Spieles ab. Der Typ von Spiel ist ausschlaggebend für die Art der Information, welche dem Spieler präsentiert werden soll (siehe Abb. 4.13). Ein Strategiespiel wie *Starcraft 2*, welches im Grunde auf numerische Ressourcen und das Hantieren von dutzenden Einheiten gleichzeitig aufbaut, verlangt ein virtuelles Interface, welches eine entsprechende Übersicht aller komplexen Vorgänge gewährt. Wegen der exakten Hantierung und Verwaltung aller laufenden Prozesse im Strategiespiel werden viele Interface Elemente als Zahlen dargestellt. Anders ist es beim Spiel *Amnesia: The Dark Descent*, welches aus der First-Person Perspektive gespielt wird. Bis auf ein abrufbares Tagebuch und Inventar Fenster verzichtet das Spiel gänzlich auf grafische Symbole im Hauptfenster. Post-Processing Effekte, Atmen und Schnaufen des Spielercharakters und die animierte First-Person Kamera und Responsiveness des Eingabegerätes liefern den Großteil des Feedbacks an den Spieler.

Wie auch bei der Körpersprache muss beim virtuellen Interface darauf

geachtet werden, dass Symbole und Farben für unterschiedliche Zielgruppen und Kulturen in vielen Fällen unterschiedliche Bedeutungen haben [3, Seite 35].

### 4.3.3 Der Spielercharakter als Interface

Wie schon in Abschnitt 4.3.2 beschrieben ist es das Ziel, die physischen und virtuellen Interface-Schichten so transparent wie möglich für den Spieler zu gestalten. Der Spieler soll im Idealfall ununterbrochen im Spiel eingetaucht sein, ohne von physischen Störungen (z.B. anstrengende Tastenkombinationen am Controller) oder abstrakten Anomalien (z.B. farblich unpassende virtuelle Interface Elemente) abgelenkt zu werden. Dadurch wird es dem Spieler erlaubt, die Distanz zwischen seiner realen und die der virtuellen Welt zu verkleinern, um in letzterer effektiver eintauchen zu können. Eine weitere Motivation liegt auch bei der Zugänglichkeit von Spielen für die seit der Einführung der *Nintendo Wii* steigende Menge an „Casual Gamers“ [82, Seite 1] – Spieler, welche durch die Motion Kontrollen der *Wii* die Einstiegschürde des Interface gemeistert haben. Eine Vereinfachung und Reduzierung von Interface Elementen würde dieser Gruppe an Spielern entgegenkommen.

Bezüglich der Beziehung zwischen Spielercharakter und Spieler würde sich der Abstand zwischen beiden verkürzen. Um die Immersion aufrecht zu halten, bleibt die sogenannte *fourth wall*<sup>3</sup> erhalten. Der Spieler fühlt sich näher zum Spielercharakter hingezogen. „To break the fourth wall is normally to break the suspension of disbelief, to remind the audience it is just a film, just a television show, just a performance.“ [38]. Natürlich hängt es vom Design des Spieles ab, ob und wann die *fourth wall* gebrochen werden soll. Manche Entwickler wollen das Medium Computerspiel als solches zelebrieren, weshalb beispielsweise *Sonic* nach kurzer Zeit der Inaktivität ungeduldig mit seinem Fuß steppt und den Spieler dabei anschaut (siehe Abb. 4.14).

In den letzten Jahren hat sich in der Entwicklung von Computerspielen viel getan, um der Verschmelzung von Spieler und Spielercharakter näher zu kommen. Zum einen ist die Entwicklung auf die Verbesserung des physischen Interface zurückzuführen (siehe Abschnitt 4.3.2), zum anderen auf die verbesserte Darstellung des Spielercharakters. Letztere hat durch hochaufgelöste Texturen, Technologien wie beispielsweise *Normal Mapping* und ein höheres Polygonbudget zwecks verfeinern der Silhouette einen wichtigen Beitrag geleistet. Allen voran liefert die Animation des anthropomorphen Charakters den größten Beitrag zu einer überzeugenden Darstellung der virtuellen Welt (siehe Abschnitt 4.1 und 4.2). Eines der wichtigsten Entwicklungen jedoch ist die Verlagerung des virtuellen Interfaces in den Raum der Spieleswelt beziehungsweise in den Spielercharakter. Das Ziel dieser Verla-

---

<sup>3</sup>Der Begriff *fourth wall* kommt aus dem Theater und beschreibt die unsichtbare Wand zwischen der Vorführung auf der Bühne und dem Zuschauer. Im Moment, wo sich der Schauspieler dem Zuschauer zuwendet, wird die *fourth wall* als „gebrochen“ bezeichnet.



Abbildung 4.14: Sonic in *Sonic the Hedgehog*.

gerung ist es, den Spieler verstärkt durch das Vermitteln von weltinternen Informationen in die virtuelle Welt eintauchen zu lassen, im Gegensatz zur Vermittlung über eine abstrakte Schicht von außerhalb [82, Seite 1].

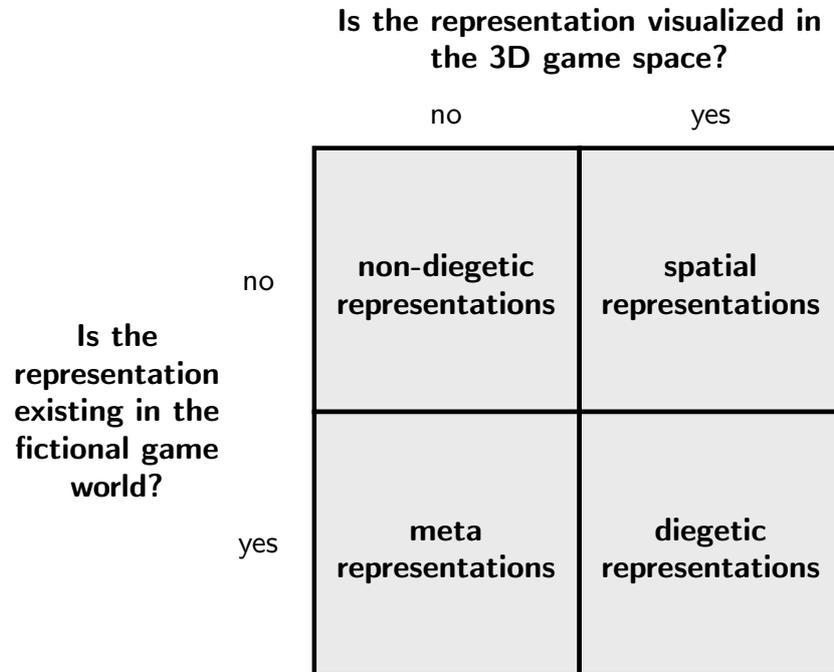
Erik Fagerholt und Magnus Lorentzon [4, Seite 46] führten eine Reihe von Begrifflichkeiten ein, um die verschiedenen möglichen Verlagerungspunkte der virtuellen Interface Elemente zu beschreiben (siehe auch Abb. 4.15 und die Beispiele in Abb. 4.16):

**Diegetic:** Dies sind die Interface Elemente, welche zu Gänze im Raum der Spielwelt integriert und somit ein Teil der Welt sind. *Dead Space* ist ein bekanntes Beispiel, welches alle Anzeigen und Menüs in der virtuellen Welt hat.

**Non-Diegetic:** Die Non-Diegetic Elemente sind genau das Gegenteil der Diegetic Elemente. Dies sind die typischen und klassischen Elemente im HUD außerhalb von Raum und Kontext der Spielwelt. Die meisten Spiele verwenden Non-Diegetic Elemente, ein gutes Beispiel ist *World of Warcraft*.

**Spatial:** Spatial Elemente werden zwar im Raum der Spielwelt dargestellt, sie sind jedoch nicht Teil vom Kontext der Spielwelt. Ein gutes Beispiel ist im Spiel *Left 4 Dead* zu finden: Charaktere, welche außerhalb des Sichtbereichs vom Spieler sind, werden mit einer farbigen Umrandung dargestellt.

**Meta:** Dies sind informationsliefernde Elemente, welche zwar im Kontext der Spielwelt existieren, jedoch nicht Teil des Raums der Spielwelt sind. Ein Beispiel wäre das *Pip-Boy* Overlay von *Fallout* oder die verschwommene Vignettierung in *Metro 2033* wenn der Spielercharakter keine Luft mehr bekommt.



**Abbildung 4.15:** Begrifflichkeiten des virtuellen Interface. Bildquelle: [36].

Zu beachten ist, dass der Ansatz der Verlagerung von Interface Elementen nicht pauschal für alle Computerspiele anwendbar ist. Je nach Genre, Setting und Bedürfnis des Spielers ist eine Verlagerung von virtuellen Interface Elementen sinnvoll oder nicht. Die Third-Person Perspektive beispielsweise bietet aufgrund der Sichtbarkeit des Spielercharakters andere Möglichkeiten der Verlagerung als die First-Person Perspektive.

In der Third-Person Perspektive kann der Körper des Spielercharakters sozusagen als Ablagerungsstelle von virtuellen Interface Elementen dienen. Ein bekanntes Beispiel von solchen *Diegetic* Darstellungen wäre die Lebensanzeige auf der Rüstung am Rücken des Spielercharakters bei *Dead Space*, was im Rahmen des Settings sinnvoll ist. *Dead Space* verlagert auch prinzipiell den Großteil seiner Menüs in die Spielwelt in der Form von holografischen Projektionen, welche der Spielercharakter selbst wahrnimmt (siehe Abb. 4.17). In einem weiteren Beispiel, in welchem der Geistes- und Gesundheitszustand vom Spielercharakter anhand von Character-Animation dargestellt wird, ist *Eternal Darkness: Sanity's Requiem*. Auch wenn ein sogenannter „Sanity Meter“ als HUD Element die geistige Gesundheit des Spielercharakters darstellt, erkennt man doch anhand der Character-Animation wie angeschlagen dieser ist (siehe Abb. 4.17). Auch der Spielercharakter *Leon*



**Abbildung 4.16:** Diegetic: *Dead Space*, Non-Diegetic: *World of Warcraft* [33], Spatial: *Left 4 Dead*, Meta: *Fallout*.

*Kennedy* in *Resident Evil 4* zeigt seinen Gesundheitszustand an, indem er seine Körperhaltung ändert und etwas gebückt mit einem Arm seinen Bauch festhält (siehe Abb. 4.17).

In der First-Person Perspektive hingegen muss primär mit den Händen und der virtuellen Kamera gearbeitet werden. Auch hier sind Diegetic Ansätze möglich: In *Metro 2033* hält der Spielercharakter ein Memo mit einer Liste von Aufgaben in der rechten Hand, während er mit einem Feuerzeug in der linken Hand dem Memo und seiner unmittelbaren Umgebung Licht spendet (siehe Abb. 4.18). In *Metroid Prime* wiederum wird das klassische (Non-Diegetic) HUD im Kontext vom Spiel auf die Innenseite vom Hightech Helm des Spielercharakters projiziert (siehe Abb. 4.18). Wie auch bei *Dead Space* nutzt das Spiel das Setting der Spielwelt, um sein virtuelles Interface in die virtuelle Außenwelt zu verlagern.

Das Setting eines Spieles bestimmt größtenteils auch den Grad der Freiheit, um virtuelle Interface Elemente auf überzeugende Art in die Spielwelt zu verlagern [36, Seite 2]. In Spielen mit einem Fantasie oder Science Fiction Setting ist dies meist einfacher darzustellen, als in Spielen mit einem hohen Grad an Realismus. Hier liegt auch der praktische Vorteil, den Character-Animationen als Informationsvermittlung mit sich bringen: grundsätzlich wird ein anthropomorpher Charakter immer, unabhängig vom Setting, menschenähnliche Körperbewegungen haben. Somit ist die Character-Animation, im Rahmen einer Third-Person Perspektive, grundsätzlich ein universaler In-



**Abbildung 4.17:** *Dead Space*: Lebensanzeige und virtuelles Interface in der Spielwelt, *Eternal Darkness: Sanity's Requiem*: gesunder und verwundeter Zustand, *Resident Evil 4*: gesunder und verwundeter Zustand.



**Abbildung 4.18:** *Metro 2033*, *Metroid Prime*



**Abbildung 4.19:** Minimal Design: *Half Life 2*, Stimmiges Design: *Diablo 3* [32].

formationsträger. Da der Spieler auch die meiste Zeit den Spielercharakter vor sich sieht, bietet sich dieser als idealer Informationsvermittler an.

Es mag jedoch nicht immer sinnvoll erscheinen, Elemente in die Spielwelt zu verlagern. Spiele wie *World of Warcraft* verlangen eine numerische und präzise Ausarbeitung von Prozessen, um wichtige Entscheidungen treffen zu können. Hier hat sich durch diverse Modifikationen des virtuellen Interface gezeigt, dass Spieler sich sogar zusätzliche HUD Elemente erstellt haben, um bessere und informierte Entscheidungen treffen zu können<sup>4</sup>. In Fällen wo ein Non-Diegetic Interface eingesetzt wird, kann dessen Design einerseits so minimal wie möglich ausfallen, um den Spieler nicht durch überflüssige Elemente abzulenken (siehe Abb. 4.19). Andererseits können die Elemente verstärkt auf das Design, das Setting und die Stimmung des Spieles abgestimmt werden, wie es z. B. bei *Diablo 3* der Fall ist (siehe Abb. 4.19).

#### 4.4 Zusammenfassung

In diesem Kapitel wurde auf den Spielercharakter als wichtigstes virtuelles Verbindungsglied zwischen dem Spieler und der Spielwelt eingegangen. Die Körpersprache wurde als visuelle Feedbackform herangezogen und auf verschiedene Arten der Gesten und Körperbewegungen untersucht. Die Anwendung der Körpersprache in Form von Animationen bei anthropomorphen Charakteren in Computerspielen wurde verdeutlicht sowie auch deren Verwendung, um Informationen an den Spieler zu vermitteln. Auch wenn die Spielwelt und NPCs durch Animationen bereichert werden, lag das Hauptaugenmerk beim Spielercharakter und dessen Informationsvermittlung an den Spieler.

Es existieren mehrere Interaktionsmodelle, welche allgemeine Ansätze für die Interaktion mit digitalen Programmen vorgeben. Das hier verwendete,

<sup>4</sup><http://www.wowace.com/addons/xperl/#w-introduction>

und um der Spielercharakter Schicht erweiterte *Mapping Modell* von Jesse Schell ist ein geeignetes Modell für die Interaktion mit Spielen. Betont wurden hier die physische und virtuelle Interface-Schicht. Ihre Rolle bei der Spieler-Spiel Interaktion wurde erläutert und welche Ansätze es gibt, um diese beiden Schichten transparenter werden zu lassen. Das Ziel ist, der Verschmelzung von Spieler und Spielercharakter näher zu kommen.

## Kapitel 5

# Die Zustandsvermittlung des Spielercharakters

Die Weiterentwicklung von Technologien und Rechenleistung hat in den letzten Jahren auch die Entwicklung von überzeugenden Animationen für den Spielercharakter gefördert. Spiele mit einem steuerbaren Spielercharakter haben verdeutlicht, dass neben einer überzeugenden und flüssigen Körperbewegung des Spielercharakters auch die Wichtigkeit dessen Animation als Informationslieferant zugenommen hat. Hiermit ist gemeint, dass der Spieler anhand der Character-Animation seines Spielercharakters dessen Zustand ablesen kann. Die menschliche Körpersprache ist das erst-gelernte, zwischenmenschliche Kommunikationsmittel, mit welchem man sich im Zweifelsfall auch ohne die gesprochene Sprache verständigen kann. Von daher bietet es sich an, diese auch in Spielen zu verwenden, um dem Spieler auf eine für ihn bekannte Art Informationen zu vermitteln.

Wie auch der Mensch kann der Spielercharakter zu jedem Zeitpunkt einen gewissen inneren und körperlichen Zustand durch seine Körpersprache zum Ausdruck bringen. In Folge werden drei Zustände ausgewählt und auf ihre Anwendbarkeit als Animationen behandelt. Wie schon des Öfteren erwähnt, wird im Rahmen dieser Arbeit der anthropomorphe Spielercharakter aus der Third-Person Perspektive mit kontrollierbarer Kamera berücksichtigt.

### 5.1 Problemansatz

Obwohl die Animationen und deren Übergänge sich weiterhin verbessern, gibt es noch wenige Ansätze, um gewisse spezielle Zustände des Spielercharakters effektiv durch dessen Körperanimationen an den Spieler zu vermitteln. Die hier gemeinten speziellen Zustände sind jene, welche typischerweise durch Zahlenwerte oder HUD Indikatoren dem Spieler vermittelt werden (wie es beispielsweise in Rollenspielen üblich ist). Zwar erfüllen diese Informationen durch ihre Abstrahierung einen praktischen Zweck, die Frage ist jedoch,

ob sich diese Informationen auch durch die Körperbewegung und Gestik des Spielercharakters vermitteln lassen, ohne dabei die für den Spieler relevanten Informationen ihm vorzuenthalten. Christiaan Moleman schreibt bezüglich der Platzierung von Feedback Informationen am Spielercharakter [53]:

„We spend much of our time staring at our avatars in 3rd person games. Not a bad place to put your gameplay feedback...“

Wie schon im Abschnitt 4.3.3 erwähnt, ist eine Verlagerung von Informationen vom virtuellen Interface auf die Animation des Spielercharakters nicht immer sinnvoll. Es ist trotzdem erstrebenswert, Änderungen am Zustand des Spielercharakters so gut wie möglich an dessen Animationen darzustellen. Auch wenn die Änderungen auf virtuellen Interface Elementen erscheinen und eine klare Auskunft für den Spieler geben, würde eine ergänzende Animation des Spielercharakters die Zustandsänderung in der Spielewelt zumindest natürlich wirken lassen. In Fällen wo der Spieler rasch eine ungefähre Einschätzung des Zustands seines Spielercharakters braucht oder verlangt, ist eine visuelle Darstellung der Körpersprache des Charakters ein schnellerer Weg als verschachtelte Menüeinträge ausfindig zu machen. Als gutes Beispiel dient hier der Gesundheitszustand des Spielercharakters, welcher schnell vermitteln kann ob der Charakter bei voller Gesundheit oder in einer Abstufung von dieser ist. Da der primäre Fokuspunkt des Spielers auf seinem Spielercharakter liegt, wäre dies auch ein geeigneter Ort um Informationen zu vermitteln. Auch ist die Körpersprache, wie schon des Öfteren erwähnt, ein Kommunikationsmittel, welches von jedem Menschen verstanden wird und somit eine verständlichere Kommunikation anbieten kann. Das Ziel ist eine schnellere, deutlichere und natürlichere Art, Informationen an den Spieler zu vermitteln. Somit ergibt sich die Motivation, so gut wie möglich Informationen am Spielercharakter darzustellen, auch wenn dies nur ergänzend zu Informationen im virtuellen Interface ist.

Seit es Animationen für Charaktere in Spielen gibt, werden auch Zustände des Charakters vermittelt. Wäre dies nicht der Fall, gäbe es keine Körperbewegungen am Spielercharakter. In *Karateka* beispielsweise erfährt der Spieler aufgrund der Animation seines Spielercharakters, ob dieser sich im Geh-Zustand befindet, oder ob dieser in einer Kampfhaltung und bereit für den Angriff ist (siehe Abschnitt 3.2). Es gibt jedoch verschiedene Arten von Zuständen, wobei nicht alle gleichwertig bei Spielen berücksichtigt werden. Während von der Steuerung aus angeregte Zustände in modernen Third-Person Spielen grundsätzlich dargestellt werden (z.B. Gehen, Rennen, Springen), ist dies weniger der Fall bei Zuständen, welche von der virtuellen Welt aus auf den Charakter einwirken (z.B. davongetragene Verletzungen nach einem Kampf). Der Grund der fehlenden Berücksichtigung Letzterer könnte darin liegen, dass diese Zustände schneller und einfacher am virtuellen Interface angezeigt werden können. Die knapp vorhandenen finanziellen und

zeitlichen Ressourcen bei der Entwicklung von Spielen dürften diesen Grund unterstreichen. Moderne Technologien und vorhandene Rechenleistung, im Gegensatz zum letzten Jahrzehnt, sind hier weniger die Einschränkung.

Um sich die erwähnten Zustände genauer betrachten zu können, muss eine Einteilung zwecks der Übersicht erstellt werden. Diese Einteilung soll auf der einen Seite diejenigen Zustände beinhalten, welche als direktes Resultat einer Steuerungseingabe dargestellt werden. Auf der anderen Seite sind solche Zustände vorhanden, welche das Resultat der Einwirkung der virtuellen Welt auf den Spielercharakter beinhalten. Je nach Kontext, Genre und Mapping können in beiden Einteilungen alle Arten der Körpersprache (siehe Abschnitt 4.1.2) vorhanden sein. In Folge wird auf die erstellte Einteilung genauer eingegangen.

Grundlegende Zustände, welche ein direktes Feedback anhand der aktiven Entscheidung des Spielers visuell durch Animationen darstellen, werden schon seit Anbeginn der 3-D Third-Person Spiele vermittelt. Mit der „aktiven Entscheidung“ ist das Interagieren mit den Tasten des Controllers gemeint. Hierzu zählt z.B. der Wechsel von einer neutralen in eine Angriffshaltung oder von einer *Geh-* in eine *Renn-*Animation. Mit einem Tastendruck sieht der Spieler somit unmittelbar das Resultat seiner Aktion durch die Animation seines Spielercharakters dargestellt. Auch das nicht-interagieren mit dem Controller löst einen Zustand des Spielercharakters aus (der Idle-Zustand), welcher durch die aktive Entscheidung des Spielers entstanden ist. Diese Zustände werden hier als *Immediate States* bezeichnet.

Im Gegensatz zu den *Immediate States* gibt es die anhaltenden Zustände, hier auch die *Persistent States* genannt. Diese Zustände werden nicht direkt durch Tasteneingaben vom Spieler veranlasst, sondern sind das Resultat der direkten oder indirekten Interaktion des Spielercharakters mit seiner Umgebung. Eine *direkte* Interaktion mit der Umgebung ist die Auswirkung einer seh- oder spürbaren Kraft auf den Körper mit einem unmittelbar ersichtlichen Resultat. Ein Beispiel einer direkten Interaktion wäre ein Monster aus der Umgebung, welches dem Spielercharakter Schaden zufügt und dieser daher etwas gekrümmt geht und seine Wunde zuhält. Das Eintreten dieses Schadenszustandes kann der Spieler nicht direkt durch Tastendruck beeinflussen, da dieser Zustand nicht auf einer Taste (sinnvollerweise) gemapped ist. Er hätte diesen Zustand vermeiden können durch das Ausweichen des Monsters – war dies nicht möglich, bleibt ihm nur die Möglichkeit der Heilung, um in den ursprünglichen gesunden Zustand zu wechseln. Eine *indirekte* Interaktion mit der Umgebung ist das Resultat einer langfristigen Auswirkung auf den Körper durch eine Ansammlung an subtilen Reizen oder Kräften, welche sich erst nach längerer Zeit der Auswirkung auf die Körpersprache des Charakters ausdrückt. Eine indirekte Interaktion wäre beispielsweise das ständige Laufen des Charakters in der virtuellen Welt. Obwohl das Laufen ein *Immediate State* ist, ist es die daraus resultierende körperliche Entwicklung (relative Zunahme an Muskelmasse, Verbesserung des Stoffwechsels und die

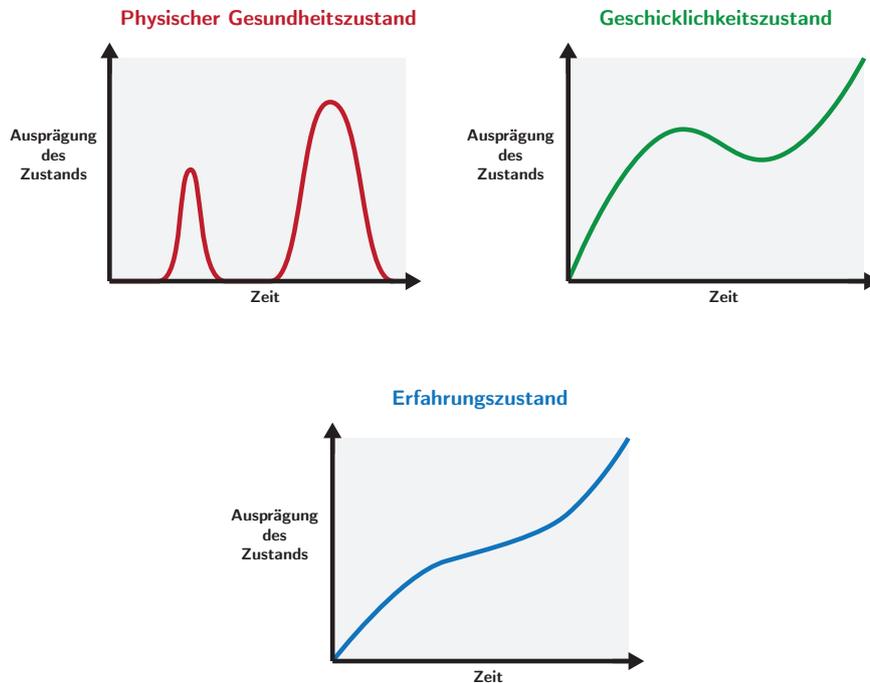
daraus resultierende körperliche Haltung) aufgrund von physikalischen Gesetzen und Hindernissen in der Spielewelt, welche eine Änderung an diesem Zustand hervorbringt. Sein Schritt wird sicherer, gezielter und kräftiger, was sich in der Lauf-Animation widerspiegeln lässt.

Das vorherige Beispiel zeigt auf, wie Persistent States am Spielercharakter auftreten. Die Immediate States sind die Basiszustände, welche natürlicherweise aufgrund der Interaktion mit dem Spiel auftreten. Sie sind das unmittelbare, visuelle Resultat der direkten Interaktion des Spielers mit dem Controller. Persistent States ergänzen oder modifizieren die Immediate States. Sie treten als Resultat der Interaktion mit der Spielewelt am Spielercharakter auf und verändern dessen (Immediate State) Animationen, um einen gewissen Zustand darzustellen. Diese Veränderung soll dem Spieler als Feedback dienen und ihn auf eine Änderung seines Spielercharakters durch äußere Einflüsse hinweisen.

Der Begriff der „Persistence“ deutet auf eine zeitlich-anhaltende Komponente hin. Persistent States reflektieren diejenigen Zustände im richtigen Leben, welche den Menschen mit fortlaufender Zeit ändern und dies durch seine Körpersprache reflektiert wird. Auf der einen Seite gibt es Persistent States, welche prinzipiell nur für kurze Zeit, aber dafür markant, die Körpersprache des Menschen ändert. Ein im Kampf verwickelter Mensch birgt als Resultat Wunden, welche die Haltung und Bewegung des Menschen ändern. Mit der Zeit jedoch heilen die Wunden (unter der Annahme, dass keine bleibenden Schäden davongetragen worden sind) und seine natürliche Bewegungsart stellt sich wieder her. Auf der anderen Seite gibt es Persistent States, welche sich ein Leben lang entwickeln und sich nur durch äußerst subtile, relativ schwer erfassbare Formen ausdrücken. Die Lebenserfahrung oder Weisheit im Allgemeinen und die sich dadurch ändernde Sichtweise und Einstellung gegenüber dem Leben und der Welt wäre ein solches Beispiel. Die Lebenserfahrung kann sich, im Gegensatz zum Gesundheitszustand, nicht mehr zurück zum ursprünglichen Zustand entwickeln – sie kann nur höchstens stagnieren.

In Folge werden drei Beispiele erwähnt, welche jeweils gleichmäßige Bereiche im Spektrum der Persistent States abdecken. Eine visuelle Darstellung zwecks der Verdeutlichung dieser drei Persistent States ist in Abbildung 5.1 zu sehen.

**Physischer Gesundheitszustand:** Dieser Zustand stellt die Änderung des äußerlichen Körpers nach einer Verletzung dar. Verletzungen des Körpers bringen meist eine Änderung der Körperbewegung mit sich oder vermitteln durch die angewandte Gestik die Art oder den Grad der Verletzung. Gewisse Körperfunktionen können dadurch beeinträchtigt sein, was sich durch die Animation darstellen und anhand der Responsiveness der Kontrolleingaben ergänzen und hervorheben lässt. Letzteres würde den Schmerz, die Qual und



**Abbildung 5.1:** Das Verhältnis zwischen Zeit und Zustandsausprägung von jeweils drei Persistent States.

die Aufteilung der Aufmerksamkeit des Opfers zwischen der Verletzung und dem Fokus auf das Erfüllen seiner Aufgaben darstellen. Ein Beispiel wäre das Brechen eines Beines nach einem Fall von einer mehreren Meter hohen Erhöhung. Das Humpeln oder Schleifen des Beines, ergänzend durch eine niedrigere Responsiveness der Kontrolleingaben, würde den Zustand überzeugend vermitteln.

Der verwundete Zustand des Körpers würde sich mit der Zeit heilen. Spezielle Mittel würden die Heilungsrate beschleunigen, um den Körper wieder in seinen ursprünglichen gesunden Zustand zu versetzen.

**Geschicklichkeitszustand:** Das Erlernen von neuen Fertigkeiten würde den Geschicklichkeitszustand hervorbringen. Gewisse physische Aktivitäten verlangen das Ausprobieren, Wiederholen und Imitieren von Aktionen, um den Körper neue Herausforderungen und Hindernisse meistern zu lassen. Der Körper gewöhnt sich an diese Aktionen – mit der Zeit werden diese verfeinert und die Geschicklichkeit der Aktivität nimmt zu. Bei physischen Aktivitäten würden sich der Metabolismus und die Muskelkraft entwickeln. Durch die zugenommene Körpermasse und dem gesteigerten Feingefühl werden die

Körperbewegungen flinker, gewisse Gesten entwickeln sich, welche den Geschicklichkeitsgrad der Person darstellen. Ein gutes Beispiel wäre das Hinaufklettern einer Bergwand: am Anfang agiert der Charakter noch unsicher, bei den Griffen verhält er sich ungeschickt. Neben der Animation reflektiert der Controller das unsichere Verhalten durch eine längere Responsiveness und einem hohen zeitlichen Schwellenwert beim schnellen Wechsel zwischen zwei Tasteneingaben. Durch ständiges Üben und Wiederholen der Kletteraktivität werden die Animationen flinker und geschicklicher. Der Charakter bekommt einen festeren Griff und seine Bewegungen werden zielstrebig. Da er sich seiner Bewegungen mehr bewusst ist, plant er seine Folgebewegungen schon im Voraus. Die Responsiveness erhöht sich, um dies zu reflektieren. Auch reduziert sich der zeitliche Schwellenwert zwischen dem Wechsel von zwei Tasteneingaben, da er sich länger mit einem Griff halten kann, bevor er mit der anderen Hand einen festen Griff bekommt.

Wie im richtigen Leben muss man sich weiterhin mit der ausgesuchten Aktivität beschäftigen, um in ihr besser zu werden. Eine Vernachlässigung der Beschäftigung über einen längeren Zeitraum würde als Folge eine Zurückentwicklung der erlernten Geschicklichkeit mit sich bringen.

**Erfahrungszustand:** Der Erfahrungszustand ist ein pauschaler Begriff, welcher als Resultat für eine Sammlung mehrerer Faktoren steht. Er soll alle möglichen Erlebnisse, Erfahrungen und Handlungen aus dem Leben in der Körpersprache zusammenfassend darstellen. Dies ist ein Zustand, der mit sehr subtilen Ausdrücken arbeitet – Erfahrungen prägen einen Menschen im Inneren und werden nur ganz dezent über längere Zeit außerhalb am Körper durch gewisse Verhaltensmuster und Haltungen zum Ausdruck gebracht. Um diesen Zustand in der Animation darstellen zu können, muss auf Körperbewegungen und Gestik zurückgegriffen werden, welche verstärkt an Erfahrungsfaktoren erinnern und diese auch implizieren. Das Gewinnen an Selbstvertrauen wäre ein solches Beispiel, wo eine aufrechte Körperhaltung, feste Blickrichtung und zurückgezogene Schultern den Eindruck eines zuversichtlichen, aufgrund von Erfahrungen geprägten Charakters erweckt.

Die Gesamtheit der Erfahrung wird sich im Leben eines Menschen nie reduzieren. Die Geschehnisse jeden Tages prägt den Menschen auf ein Weiteres, über die Jahre hinweg sammeln sich somit viele Eindrücke, welche im Endeffekt als Erfahrungen bezeichnet werden.

Das Spektrum der Persistent States stellt sich aus Zuständen zusammen welche, wie schon erwähnt, durch den Grad ihrer jeweiligen zeitlichen Beständigkeit definiert ist. Der veränderte physische Gesundheitszustand beispielsweise ist nur von kurzer Beständigkeit, bis sich der normale Gesundheitszustand wieder einsetzt – während der Erfahrungszustand im Laufe der Zeit immer stärker zum Ausdruck kommt, und zwar ohne Rückentwicklung.

Zustände mit einer längeren Beständigkeit wären also langlebiger als welche mit einer kurzen Beständigkeit. Im Rahmen der drei erwähnten Beispielzustände und weiteren Überlegungen scheinen auch folgende Zusammenhänge ersichtlich zu sein:

- Der Grad der Rückentwicklung zum Ausgangszustand (dem Immediate State) würde sich mit einer länger werdenden Beständigkeit soweit reduzieren, bis es keine Rückentwicklung mehr gibt.
- Die Zustände werden immer subtiler in ihrer Ausdrucksform, je langlebiger sie sind.
- Direkte Einflüsse aus der Umgebung oder Spielewelt scheinen kurzlebige Zustände zu produzieren, während indirekte Einflüsse eher langlebige Zustände erzeugen.
- Das Erfassen und Formulieren eines Zustandes durch eine Animation scheint komplexer zu werden, je langlebiger ein Zustand wird. Dies könnte damit zusammenhängen, dass unbedeutende kleine Änderungen über einen längeren Zeitraum schwer erfassbar sind.

Meist sind es zeitliche oder finanzielle Gründe von Seiten der Spieleentwickler, weshalb Persistent States nicht verstärkt Einzug in die Animationen von Spielercharakteren gefunden haben. Viele der Zustände lassen sich durch virtuelle Interface Elemente darstellen, was Kosten und Zeit sparen würde. Wie jedoch schon öfters erwähnt, würde eine Vermittlung dieser Informationen durch den Körper des Spielercharakters dem Spieler eine viel natürlichere und überzeugendere Welt darstellen (siehe Kapitel 4). Die Entwicklung neuer Technologien und schnellerer Arbeitsmethoden würde die Anwendung von Persistent States in Animationen hoffentlich begünstigen.

## 5.2 Fragestellung

Die vorherigen Kapitel haben u.a. auf die Wichtigkeit der Körpersprache als Informationslieferant hingewiesen. Neben deren Anwendung in Form von Animationen wurde, im Rahmen des Spielercharakters, auf die enge Verbindung zwischen der Spieler-Spiel Interaktion und der Character-Animation eingegangen. Der letzte Abschnitt hat auf die Zustandsarten eines Charakters aufmerksam gemacht, mit dem Ziel, Persistent States verstärkt in Spielen am Spielercharakter darzustellen. Somit ergibt sich folgende Fragestellung:

Wie kann die Darstellung von Körperbewegungen und Gestik als visuelle Feedbackform im Kontext der Spieler-Spiel Interaktion effektiv eingesetzt werden, um dem Spieler den Geschicklichkeits- und Erfahrungs- wie auch den physischen Gesundheitszustand des spielbaren anthropomorphen Hauptcharakters zu vermitteln?

Prinzipiell stellt sich also die Frage, wie man Animationen am Spielercharakter so gut wie möglich anbringen kann, um den Informationsgehalt der drei oben erwähnten Persistent States dem Spieler, durch den Spielercharakter, vermitteln zu können. Dabei ist die Rahmenbedingung zu beachten, dass der spielbare Hauptcharakter aus einer Third-Person Perspektive betrachtet wird, wobei sich die Kamera frei um den Charakter herum drehen kann.

Ein wichtiger Aspekt des Haupt- bzw. Spielercharakters ist sein anthropomorphes Merkmal. Diese Eigenschaft erlaubt dem Spieler, sich einfacher mit dem Charakter zu identifizieren. Aufgrund der Erfahrungen des Spielers beim Beobachten, Interagieren und Imitieren von anderen Menschen um sich, wird ihm das Verhalten und Aussehen einer weiteren menschen-ähnlichen Kreatur nicht fremd sein. Zumindest kann er der anderen Kreatur aufgrund ihrer Kommunikationssignale eine gewisse Bedeutung zuordnen. Eines dieser Kommunikationsmittel wäre die Körpersprache, eine grundlegende Art, mit der sich alle Menschen verständigen können. Es besteht daher die Annahme, dass auch eine digitale, anthropomorphe Kreatur die Möglichkeit hat, sich durch ihre Körpersprache so auszudrücken, dass der Spieler diese Signale verstehen und interpretieren kann. Je nach technologischer Entwicklung, rechnerischer Leistung und finanziellem und zeitlichem Budget des Entwicklers sind feinere Grade des Ausdrucks möglich.

Im Rahmen dieser Arbeit beschränkt sich die Körpersprache auf den Bereich der Körperbewegung und Gestik. Die Mimik wird außen vor gelassen, da sie erstens Thema einer eigenständigen Arbeit wäre, und zweitens das Gesicht des Spielercharakters in einem Spiel mit einer Third-Person Perspektive selten gesehen wird. Meist sieht man den Rücken des Charakters, da der Spieler die Kamera typischerweise so platziert, um nach vorne zu schauen. Erst durch ein bewusstes Drehen der Kamera, um das Gesicht zu sehen, oder in Cut-Scenes wo die Kontrolle der Kamera vom Spieler genommen wird, tritt das Gesicht des Charakters in den Vordergrund. Aus diesen Gründen werden nur die körperliche Motorik und die Signale des Körpers, des Gesichtes ausgeschlossen, als Informationsvermittler betrachtet.

Die Körpersprache ist eine visuelle Form der Kommunikation. Sie vermittelt Informationen über ein Lebewesen und dient somit als Feedbackform für einen Betrachter oder Zuschauer dieses Lebewesens. Im realen Leben wird diese Information durch den Raum über den visuellen Sinneskanal vermittelt. Letzteres ist auch bei Spielen der Fall. Die Information muss jedoch zusätzlich durch mehrere Schichten in den digitalen Raum übertragen und mit Änderungen unterzogen werden. Damit der Spieler mit dem Spiel kommunizieren und interagieren kann, bedarf es mehrerer Schichten der Informationslieferung und -Darstellung, wie es im Mapping Modell von Jesse Shell dargestellt wurde. Diese Schichten abstrahieren die für den Spieler wichtigen Informationen und liefern sie in einer für ihn verständlichen Form. Dadurch kann der Spieler konsequent Entscheidungen treffen und den Zyklus der Interaktion weiterführen.

Im digitalen Raum werden viele Informationen über den Zustand des Spielercharakters auf der virtuellen Interface-Schicht dargestellt, typischerweise in der Form eines HUD. Da die Körpersprache eine natürliche Form der Kommunikation des Menschen ist, stellt sich hier also die zentrale Frage, ob die Elemente aus der virtuellen Interface-Schicht nicht auch über den Spielercharakter in der Form von Animationen dem Spieler vermittelt werden können. Somit würde die virtuelle Interface-Schicht transparenter werden und den Spieler näher in die virtuelle Spielewelt eintauchen lassen. Im Speziellen sind es die Persistent States, deren Informationen üblicherweise als virtuelle Interface Elemente vorkommen. Unter Berücksichtigung der Langlebigkeit diverser Zustände im Spektrum der Persistent States, wurden folgende drei Zustände gewählt, um die sinnvolle und vollständige Übertragung ihres Informationsgehaltes auf die Animation des Spielercharakters zu überprüfen: der Geschicklichkeits-, Erfahrungs- und physische Gesundheitszustand. Dies soll als Proof-of-Concept dienen, um in Zukunft die Möglichkeit zu ebnet, Persistent State Informationen vom virtuellen Interface auf den Charakter zu übertragen.

Auch der Controller als physisches Interface spielt bei der Spieler-Spiel Interaktion eine wichtige Rolle, da es das einzige Mittel ist, um Entscheidungen des Spielers in die virtuelle Spielewelt zu transferieren. Erst der Controller erlaubt dem Spieler, seinen Spielercharakter kontrollieren zu können. Da erst durch die Eingaben des Spielers die Animationen des Charakters zum Ausdruck gebracht werden, muss auch der Controller bei der Vermittlung von Informationen berücksichtigt werden. Dazu zählt ein geeignetes Layout der Tasten und Knöpfe, ein haptisch ansprechendes Gewicht und Material, ein sinnvolles Mapping der Tasten auf Funktionen im Spiel und eine passende Responsiveness der jeweiligen Eingabebefehle. Aufgrund des fehlenden physischen Feedbacks der Ereignisse im Spiel sind es diese Controller Eigenschaften, welche dem Spieler ein „gutes Gefühl“ beim Spielen vermitteln. Diese Gefühle, wie beispielsweise eine lange Responsiveness, welche das Gefühl von Schwere und Trägheit vermittelt, können kontrolliert verwendet werden, um die darstellenden Animationen zu ergänzen und zu unterstützen.

### 5.3 Methode

Um die Fragestellung beantworten zu können liegt es nahe, einen Prototyp zu entwickeln, welcher sich mit der praktischen Umsetzung der Fragestellung befasst. Ziel des Prototyps ist die Umsetzung der gestellten Anforderung, im Kontext der Spieler-Spiel Interaktion, die drei erwähnten Zustände so effektiv wie möglich dem Spieler, anhand von Animationen des Spielercharakters, vermitteln zu können. Um dieses Ziel erreichen zu können, muss ein geeignetes Realisierungsmittel gefunden werden.

Die Vermittlung von Feedback Informationen an den Spieler basiert auf

visuelles und haptisches Feedback durch den Prototypen. Im Prototyp ist es der Spielercharakter, welcher die Quelle dieses Feedbacks ist. Die visuelle Information wird in der Form von Körperbewegungen und Gestik an den Spieler vermittelt. Im Kontext der Spieler-Spiel Interaktion liegt die Anforderung am Prototyp, diese visuelle Information in Form einer interaktiven Schnittstelle in Echtzeit steuern zu lassen. Ein Eingabegerät würde diese Steuerung zwischen dem Spieler und dem Prototyp gewährleisten. Haptisches Feedback würde somit zusätzlich durch diese Schnittstelle vermittelt werden.

Obwohl die visuelle Darstellung eines Körperzustandes auf Papier möglich ist, sind dessen Bewegungsabläufe in der Form von Animationen sinnvoller in einem Bewegtbild-Medium realisiert. Da zusätzlich der Prozess der Interaktion zwischen dem Spieler und dem animierten Hauptcharakter bzw. Spielercharakter im Vordergrund steht, müsste das Bewegtbild-Medium über Eingabemöglichkeiten verfügen. Das Abspielen von Animationen des Spielercharakters ist abhängig von der Interaktion mit dem Eingabegerät. Ohne diese Interaktion wird der Spielercharakter nicht vorangetrieben. Damit diese Beziehung zwischen Eingabegerät und Animation des Spielercharakters erhalten bleibt, sollten beide Aspekte im Prototyp berücksichtigt werden. Es ist der Prozess der Interaktion mit dem Spielercharakter, anhand eines Eingabegerätes, welcher wichtige, ergänzende Informationen zur Animation liefert, um die Zustände auf deutliche Art dem Spieler vermitteln zu können.

Um die Körperbewegung und Gestik des Spielercharakters so gut wie möglich vermitteln zu können, sollte im Prototyp die Möglichkeit bestehen, den Körper des Spielercharakters als Ganzes dem Spieler darstellen zu lassen. Die Darstellung des Spielercharakters als anthropomorphe Kreatur würde es dem Spieler vereinfachen, die Animation des Spielercharakters zu interpretieren und zu verstehen.

Unter Berücksichtigung der erwähnten Anforderungen würde eine Spiel-Engine als Realisierungsmittel des Prototyps in Frage kommen. Das Abspielen und Blending von Animationen des Spielercharakters muss gegeben sein. Letzteres ist wichtig, um einen flüssigen Übergang zwischen die sich in Echtzeit ändernden Zustände zu gewähren. Um eine wirksame Darstellung und Auslösung der erwähnten drei Zustände zu erlauben, muss die Engine das Erstellen einer virtuellen Welt zulassen. Die Engine muss auch die Möglichkeiten anbieten, das Mapping der Eingabebefehle modifizieren zu können, um somit gewisse Zustände und Interaktionsprozesse dem Spieler gezielt vermitteln zu können.

Der Vorteil der Verwendung einer Spiel-Engine ist der Funktionsumfang und der Verbleib im Medium der Computerspiele. Spiel-Engines erlauben auf eine flexible Art die gewünschten Anforderungen eines Spieles oder spezielle Elemente eines Spieles zu testen. Der Nachteil dieser Methode ist der größere Produktionsumfang. Im Falle von Animationen des Spielercharakters, welche zusätzlich von Eingabebefehlen abhängig sind, liefert eine Spiel-Engine

zwar brauchbare Resultate, jedoch muss dafür mehr Zeit in die Entwicklung investiert werden. Insbesondere die Auseinandersetzung mit Character-Animationen in Spielen verlangt eine gewisse Multidisziplinarität bei der Entwicklung eines Prototyps. Allen voran die Bereiche Gestaltung und Programmierung sind hier sehr stark und voneinander abhängig vertreten.

## Kapitel 6

# Projekt: Homunculus

### 6.1 Concept

*Homunculus* (6.2) ist ein Third-Person Jump 'n Run, worin eine puppenähnliche, menschliche Körperhülle zum Leben erweckt wird und man mit dieser das Ende der Talenge erreichen muss. Durch Erkunden der Umgebung erlernt man neue Arten und Ausmaße von körperlichen Bewegungen, welche eingesetzt werden müssen, um verschiedene Hindernisse zu bewältigen.

Die zwei Hauptaugenmerke bei der Realisierung des Spieles lagen bei der Vermeidung eines klassischen virtuellen 2D Interface (HUD) und der Vermittlung von Informationen über drei verschiedene Zustände des Spielercharakters durch dessen Animationen. Letztere wird durch den Prozess der



Abbildung 6.1: *Homunculus* Startbildschirm.

Spieler-Spiel Interaktion mit dem Eingabegerät (in diesem Fall einem *Xbox 360* Controller) in der Form von Responsiveness, Tastenlayout und Mapping ergänzt.

Die Third-Person Perspektive wurde gewählt, um dem Spieler die Möglichkeit zu geben seinen Spielercharakter im Ganzen zu sehen. Die frei drehbare Kamera erlaubt es auch, den Spielercharakter aus verschiedenen Winkeln zu betrachten.

Damit die drei erwähnten Zustände sinnvoll im Spiel zum Einsatz kommen können, wurde eine einfache, parkourähnliche Anlage gebaut. Diese Anlage besteht aus einfachen Hindernissen, welche der Spieler mit seinem Spielercharakter bewältigen muss, um an das andere Ende der Talenge zu kommen. Ein einfacher, sich teilender Pfad erlaubt es dem Spieler sich zu entscheiden, welchen Weg er als nächstes bestreiten möchte. Unabhängig dieser Entscheidung existiert ein sich leicht steigender Schwierigkeitsgrad, um den Weg zum Ziel interessant zu halten. Dem Spieler wird jedoch die Gelegenheit gegeben sich auf die Hindernisse, in der Form einfacher Variationen, vorzubereiten. Eine detaillierte Ausarbeitung des Game Designs ist auf der beiliegenden DVD dieser Arbeit in der Form eines Game Design Dokumentes zu finden.

Als Engine für die Umsetzung des Projektes dient das *Unreal Development Kit (UDK)*. Im Gegensatz zu anderen frei verfügbaren Engines bietet UDK eine Auswahl an fortgeschrittenen Animationstechniken und ein dazugehöriges, visuelles Knotensystem, mit welchem Attribute von verschiedenen Systemen und Animationen miteinander verbunden werden können. Zusätzlich bietet UDK Features an, welche eine einfache und schnelle Erstellung von 3-D Welten mit einer hohen optischen Qualität erlaubt. Informationen und Links zu allen Aspekten von UDK bezüglich der Entwicklung dieses Projektes lassen sich in Anhang A nachschlagen.

## 6.2 Der Charakter

Der Spielercharakter ist eine anthropomorphe Kreatur und wirkt im Äußeren wie eine Schaufensterpuppe (siehe Abb. 6.2). Entsprechend ist auch die Textur der Haut, welche von Farbe und Form her künstlich wirkt. Neben einem baumwollartigen Textil verzieren mehrere Tattoos den Körper des Charakters. Dem Spielercharakter fehlen die ausgeprägten Merkmale eines Gesichtes. Zum einen dient dies dem puppenähnlichen Aussehen, da auch Schaufensterpuppen ein ausdrucksneutrales Gesicht haben. Zum anderen dient dies der Zielsetzung für das Projekt, und zwar zur Betonung der Körperbewegung und Gestik, ohne den Einsatz von Mimik bei der Zustandsvermittlung des Spielercharakters an den Spieler.

Das puppenähnliche Aussehen dient dazu, den Charakter aus seiner Umgebung hervorzuheben. Der Kontrast wurde verstärkt, indem die Umgebung



**Abbildung 6.2:** Konzept, Maya Mesh und In-Game Modell des Spielercharakters.

einer natürlichen Landschaft aus Pflanzen, Steinen, Himmel und Boden entspricht. Auch wenn der Spielercharakter menschenähnliche Bewegungen beherrscht, wirkt sein Aussehen in solch einer Umgebung etwas fremd. Die helle Haut des Charakters verstärkt diesen Eindruck. Die Aufmerksamkeit des Spielers würde dadurch verstärkt auf den Spielercharakter fallen. Die Absicht dahinter erklärt sich dadurch, dass es die Animationen des Spielercharakters sind, welche die Informationen über seinen Zustand liefern. Indem der Charakter eine anthropomorphe Kreatur ist und diese verstärkt vom Spieler beachtet wird, lassen sich seine Bewegungen und Gesten schneller



**Abbildung 6.3:** Konzept und In-Game Darstellung des physischen Gesundheitszustandes.

und einfacher deuten.

Um die Optik der Bewegungen des Charakters zu verbessern, wurde der Lendenschurz des Spielercharakters mit einer Cloth Funktion versehen. Der Lendenschurz reagiert mit *Physics Colliders*, welche am restlichen Körper des Spielercharakters in UDK angebracht worden sind.

### 6.3 Die drei Zustände

Obwohl sich das Projekt größtenteils auf die Realisierung von Animationen für den Spielercharakter konzentriert, liegt das Hauptaugenmerk auf der Vermittlung durch Animationen von drei Zuständen, in welchen sich der Spielercharakter befinden kann (nähere Details dazu in Abschnitt 5). In Folge wird anhand von Stichpunkten beschrieben, wie sich diese drei Zustände im Spiel ausdrücken:

#### Physischer Gesundheitszustand

Als Beispiel wird ein gebrochenes/verstauchtes Bein verwendet. Dieser Zustand tritt ein, wenn der Charakter von einem höheren Abhang hinunter fällt (siehe Abb. 6.3).

- Der Charakter bleibt kurz liegen, kann nur schwer unter Schmerzen aufstehen. Bis zum Ende dieser Animation nimmt das Spiel die Kontrolle vom Spieler weg.
- Das rechte Bein wird verletzt. Durch Spiegeln der Animationen kann auch das linke Bein verletzt sein.
- Es gibt zwei Verletzungszustände: Beim ersten kann der Charakter sich

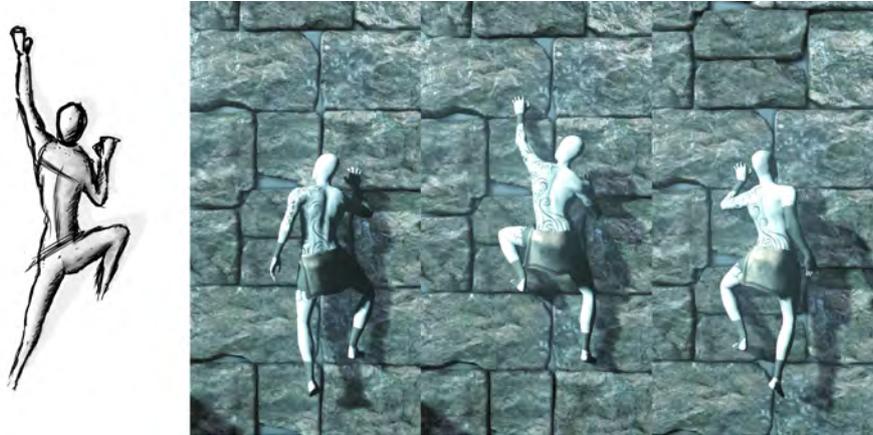
kaum auf dem verletzten Bein stützen. Beim zweiten fällt das Gehen schon leichter, zumindest humpelt der Charakter vor sich hin.

- Die Verletzung heilt sich mit der Zeit. Hier wird ein Blend zwischen den Animationen verwendet. Der Heilungsprozess ist schneller im Idle als im Gehzustand.
- Der Charakter im Idle-Zustand hält das schmerzende Bein mit der Hand.
- Der Charakter kann nicht mehr rennen oder springen, er kann nur langsam gehen.
- Beim Gehen: Die Dauer der Responsiveness ist länger als normal, das Gehen soll als mühsame Arbeit vermittelt werden.
- Mit der Zeit wird die Bewegung immer schneller. Auch das Humpeln verschwindet langsam. Die Responsiveness der Bewegung kehrt zum ursprünglichen Wert zurück.
- Durch das Blending der Animationen ändert sich auch das visuelle Bild des Spielercharakters. Erst hält er das schmerzende Bein und verweilt in einer gebückten Haltung. Seine Bewegung ist mit Mühe und Schmerz verbunden. Mit der Zeit wird seine gebückte Haltung minimiert, das Gehen fällt ihm schon leichter. Er hält zwar immer noch sein Bein, die Schmerzen jedoch haben sich reduziert. Seine Gangart hat sich zum leichten Humpeln verändert.
- Mit steigender Erfahrung und Geschicklichkeit wird beim Charakter nur mehr der zweite Gesundheitszustand ausgelöst, bzw. mit höchster Erfahrung verletzt er sich nicht mehr. Dadurch erhöht sich auch die Höhe, aus welcher der Charakter hinunterfallen kann, ohne sich stark verletzen zu müssen.

### **Geschicklichkeitszustand**

Hier wird als Beispiel das Klettern verwendet. Nähert sich der Charakter einer Wand, geht er durch Drücken beider Schultertasten in die Kletterpose und ist bereit für die Eingaben des Spielers. Die Kletterphase kann durch Drücken der B-Taste oder durch Runterdrücken des Analog-Sticks abgebrochen werden (siehe Abb. 6.4).

- Um die Wand hinaufzuklettern, müssen abwechselnd die linke und rechte Hand einen Griff erreichen und diesen festhalten, um den restlichen Körper hochzuheben.
- Die Schultertasten kontrollieren jeweils den linken und rechten Arm des Spielercharakters. Durch Drücken und Festhalten der jeweiligen Schultertaste streckt der Spielercharakter den Arm zum nächsten Griff aus und hält ihn mit der Hand fest. Anschließend erst kann der andere Arm zum nächsten Griff ausgestreckt werden. Lässt der Spieler die erstgedrückte Schultertaste los, bevor der andere Arm einen festen Griff



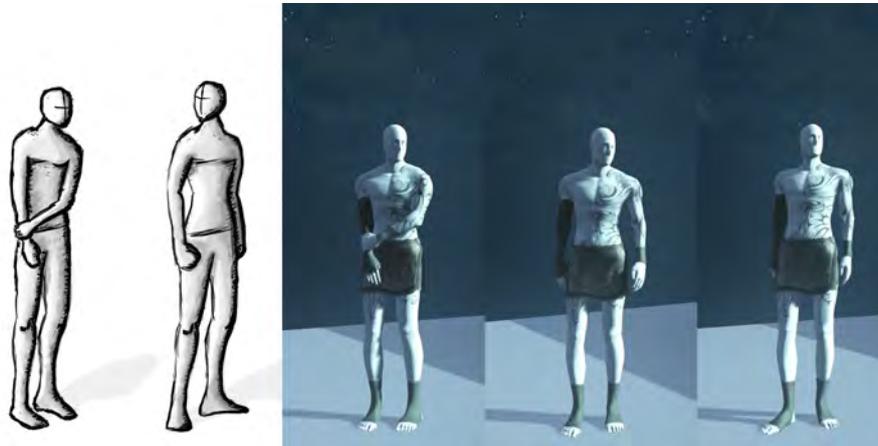
**Abbildung 6.4:** Konzept und In-Game Darstellung des Geschicklichkeitszustandes.

hat, fällt der Charakter hinunter, da ihn keine Hand mehr an der Wand hält.

- Die Animationen des Spielercharakters zeigen dessen Geschicklichkeit an. Im ungeschickten Zustand stellt die Animation ein etwas unsicheres und zögerndes Handeln des Spielercharakters dar. Im geschickten Zustand sind die Bewegungen flink und gezielt.
- Durch Blending wird kontinuierlich zwischen dem ungeschickten und geschickten Zustand verschmolzen.
- Die Responsiveness der Schultertasten zeigt in diesem Fall die Geschicklichkeit des Spielercharakters an. Es dauert eine Weile bis der Charakter seinen Arm zum nächsten Griff hebt. Beim Loslassen eines Griffes um fortzuschreiten muss genau geachtet werden, dass die andere Hand schon einen festen Griff hat. Mit fortgeschrittener Geschicklichkeit kann der Spielercharakter den Griff noch eine kurze Weile lang festhalten, auch wenn der Spieler die Schultertaste schon losgelassen hat. Dies erlaubt einen schnellen Aufstieg durch rhythmisches Hin- und Herschalten der Schultertasten, da zusätzlich auch das Ausstrecken der Arme schneller vorangeht.
- Nach einem längeren Zeitraum verliert der Spielercharakter langsam an Geschicklichkeit. Dies geschieht nur, wenn er länger nicht mehr geklettert hat. Die Responsiveness geht zurück, sowie auch das Blending vom geschickten in den ungeschickten Zustand.

### **Erfahrungszustand**

Der Erfahrungszustand wird von Anfang bis Ende vom Spiel dargestellt. Im praktischen Kontext ist der Spielercharakter in einer fremden Welt und muss



**Abbildung 6.5:** Konzept und In-Game Darstellung des Erfahrungszustandes.

diese erkunden und Hindernisse bewältigen. Je mehr er von der Welt kennt und je mehr Hindernisse er bewältigt hat, je erfahrener bzw. zuversichtlicher ist er (siehe Abb. 6.5).

- Die Erfahrung wird anhand der Haltung des Spielercharakters dargestellt. Die neue Welt ist ihm fremd, er wirkt leicht angespannt und defensiv. Seine Schultern neigen nach vorne, die Füße zeigen beide tendenziell nach innen.
- Je länger sich der Spielercharakter in der Welt aufhält und diese kennenlernt, je erfahrener wird er. Seine Haltung öffnet sich, er wirkt bereit, die Herausforderungen dieser Welt auf sich zu nehmen.
- Prinzipiell wirkt sich die Responsiveness auf alle Facetten der Spielercharakter Animationen aus. Mit steigender Erfahrung wird auch die Responsiveness in allen Animationsbereichen leicht erhöht.
- Die Erfahrung wird sich nie reduzieren. Das langsame Blending vom unerfahrenen in den erfahrenen Zustand ist eine permanente Aktion.
- Derzeit gibt es nur drei Erfahrungsstufen, welche sich auf diskrete Weise visuell äußern.

Um diese Zustände (sowie auch alle anderen Animationen) in UDK realisieren zu können, musste ein sogenannter *AnimTree* erstellt werden (siehe Abb. 6.6). Hier werden aus dem *AnimSet* die verwalteten Animationssequenzen in der Form von Knoten eingefügt. Diverse Verwaltungsknoten erlauben das Blenden und ein gezielt-kontrolliertes Abspielen von einer oder mehreren Animationssequenzen. Auch werden im *AnimTree* die sogenannten *MorphTargets*, bzw. *Blendshapes*, sowie auch die *Skeletal Control Chains*

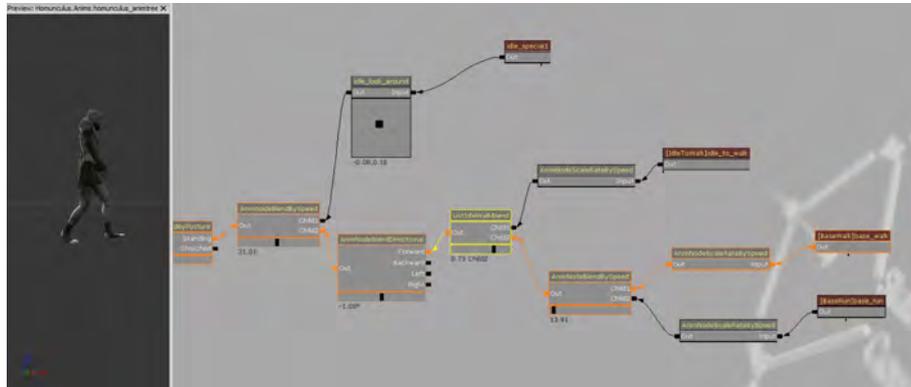


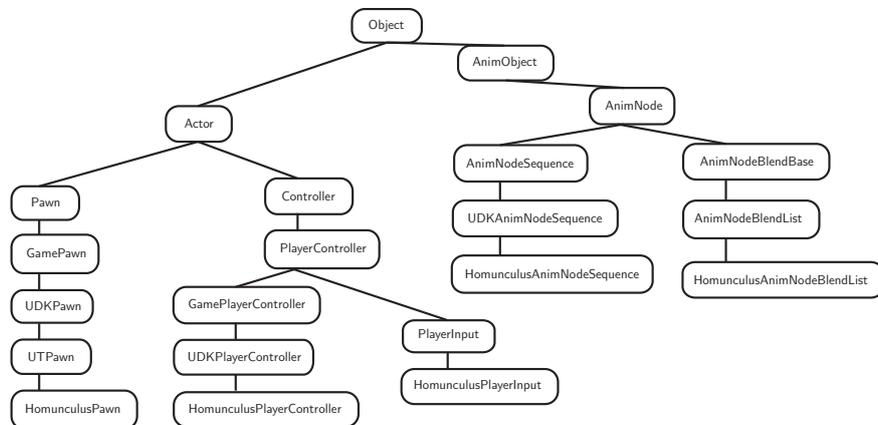
Abbildung 6.6: Ein Ausschnitt eines AnimTree.

(beispielsweise für IK) verwaltet und berücksichtigt. Alle diese Knoten werden durch Ein- und Ausgänge miteinander verbunden und in den AnimTree eingespeist.

Um das Blenden von Animationen zu vereinfachen bietet AnimTree mehrere vordefinierte Knoten an. Ein wichtiger solcher Knoten heißt *AnimNodeBlendByPhysics*. Dieser Knoten hat mehrere Eingänge, welche für typische Bewegungsabläufe wie Gehen oder Springen ausgelegt sind. Animationssequenzen können somit einfach in diese Eingänge eingespeist und bei entsprechenden, in UDK vordefinierten Eingabefunktionen abgespielt werden. Die typischen Bewegungsabläufe werden mit physikalischen Funktionen und Variablen wie Beschleunigung, Geschwindigkeit und Anziehungskraft getrieben. Es liegt somit nahe, dass Änderungen im Code im Idealfall diese Variablen betreffen, da Verwaltungsknoten wie *AnimNodeBlendBySpeed* auf diese physikalische Funktionalität zugreifen. Nebenbei ist *AnimNodeBlendBySpeed* eines der praktischsten Verwaltungsknoten für Fortbewegungsanimationen. Ausgehend aus beispielsweise der Grundgeschwindigkeit eines Walk-Cycles, erlaubt dieser je nach der Geschwindigkeitsänderung durch den Analog-Stick eine Anpassung der Abspielrate dieser Animation. Ab einer vordefinierten Geschwindigkeitsgrenze würde der Walk-Cycle beispielsweise in einen *Jogg*- oder *Run*-Cycle überblenden.

Mit Morphtargets im AnimTree sind nicht die Corrective Blendshapes gemeint, sondern jene, welche üblicherweise für Gesichtsanimationen verwendet werden. Hier können verschiedene Blendshape Animationen miteinander gemischt werden, um passend zur Situation im Spiel einen gewünschten Gesichtsausdruck zu kreieren.

*Skeletal Control Chains* sind speziell definierte Ketten von Joints, welche dynamisch in Echtzeit ihre Position und Rotation ändern können. Diese Änderungen überschreiben die Keyframe Animationen dieser Joints, welche zum Zeitpunkt der Ausführung aufgrund der vorgegebenen Animationsse-



**Abbildung 6.7:** Die Abhängigkeiten und Ableitungen wichtiger Klassen im Rahmen dieses Projektes.

quenz abgespielt werden. Skeletal Control Chains werden üblicherweise für *Foot Placement* und *Look At* eingesetzt, welche in UDK durch Angabe der gewünschten Joint Kette und erwähnten Verwaltungsknoten auf einfache Art möglich ist.

Um der Zielsetzung vom Projekt gerecht zu werden, musste neben dem Animieren des Spielercharakters auch die gewünschte Funktionalität in UDK einprogrammiert werden. Als sicherste Vorgehensweise hat sich das Ableiten von existierenden Klassen herausgestellt (siehe Abb. 6.7). In diesen neuen Klassen wurde somit die gewollte Funktionalität eingefügt, bzw. ungewollte Programmcode deaktiviert. Auf diesem Weg wurde die grundlegende, frei bewegliche Third-Person Kamera eingebaut, sowie auch die Abtastung von analogen Werten des Analog-Sticks, um eine kontinuierlich steigende oder fallende Geschwindigkeit zu erhalten. Auch wurde die Möglichkeit der Anticipation und Follow Through Actions eingebaut, sowie angepasste Verwaltungsknoten im AnimTree erstellt.

## 6.4 Die Steuerung

Als Eingabegerät wird der *Xbox 360* Controller verwendet. Zum einen wird der Controller systemintern von UDK unterstützt, und zum anderen liegt er sehr angenehm in der Hand. Ein wichtiges Kriterium bei der Auswahl des Eingabegerätes war die analoge Funktionalität, welche in der Form des Analog-Sticks gegeben ist. Damit lässt sich die Geschwindigkeit des Spielercharakters kontrollieren.

Damit der Spieler die Tastenbelegungen des Controllers nicht während des Spiels im virtuellen Interface präsentiert bekommt, wird nach dem Start-



**Abbildung 6.8:** Springen: eine der Tastenbelegungen des *Xbox 360* Controllers in *Homunculus* [34].

bildschirm ein Video mit einer visuellen Vorführung einer jeden Aktion und deren Tastenbelegung gezeigt (siehe Abb. 6.8). Dies würde dem Spieler auf einfache Art zeigen, welche Funktion die jeweilige Taste hat.

## 6.5 Die virtuelle Welt

Die Umgebung vom Spielercharakter ist, bis auf eine verlassene Ruine, ein naturbelassener Ort. Wie schon in diesem Kapitel erwähnt, dient dies der Kontrastsetzung zwischen dem Spielercharakter und seiner Umgebung. Die Platzierung der Gegenstände und Objekte in der Umgebung leiten den Spieler und erlauben ihm, sich zu orientieren (siehe Abb. 6.9).

Das Leveldesign ermöglicht dem Spieler sich zu Beginn mit den Bewegungseingaben des Spielercharakters vertraut zu machen. Eine große, offene Fläche dient diesem Zweck. Als erstes Hindernis muss der Spielercharakter über einer Mauer klettern und von dieser dann abspringen. Beim Aufkommen am Boden verletzt er sich leicht am Bein. Das kurze Humpeln beim Gehen soll dem Spieler zeigen, dass das Abspringen von bestimmten Höhen Konsequenzen in der Form von eingeschränkter Geschwindigkeit und träger Steuerungseingabe mit sich bringt.

Anschließend kann er sich für zwei Wege entscheiden. Der eine Weg führt in ein kleines Tal mit dichter Vegetation, während der andere Weg über die Ruine und einen kurzen Gebirgspass führt. Bei beiden Wegen muss der Charakter wiederum über Hindernisse klettern. Am Ende erreicht er die Lichtung

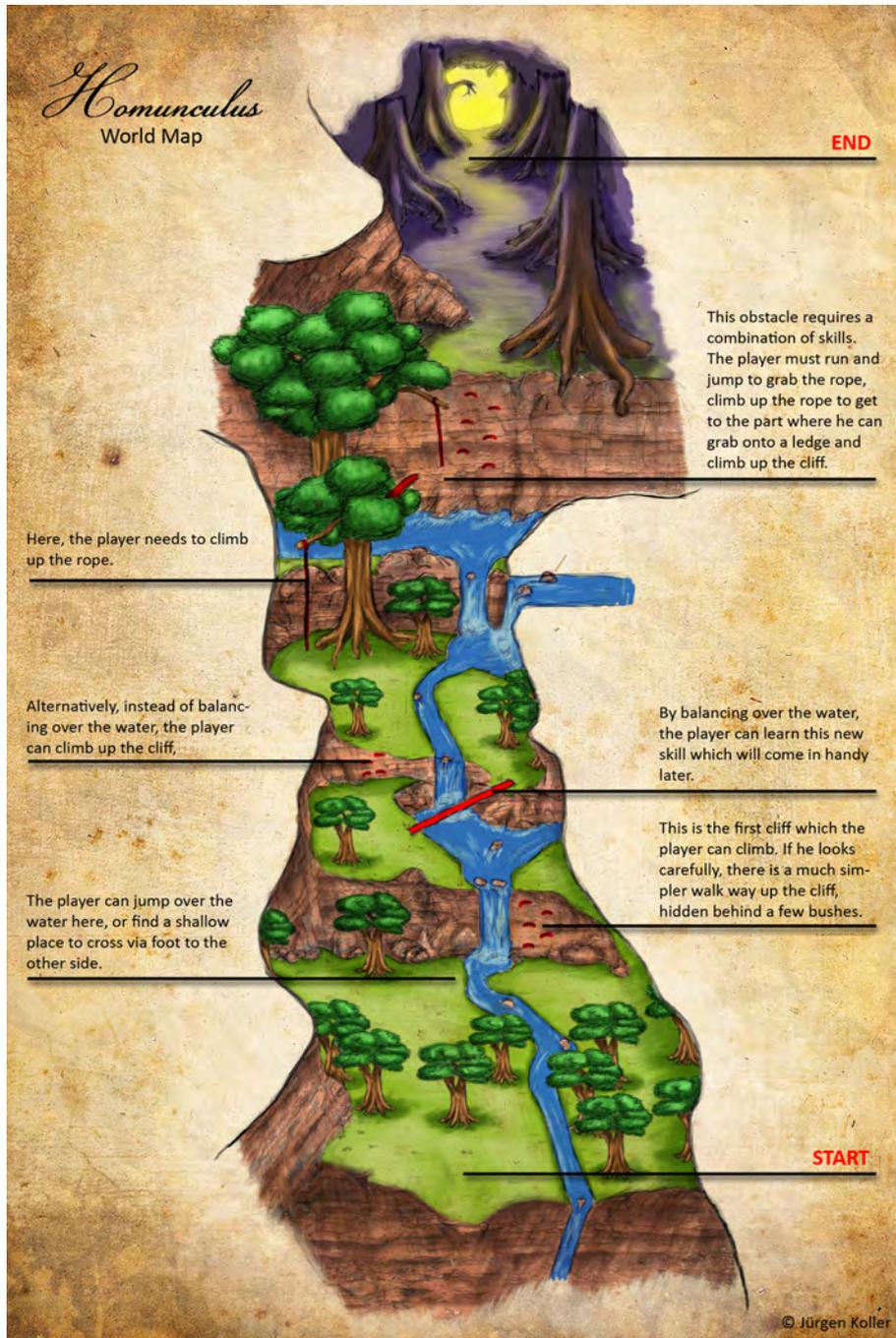


Abbildung 6.9: Erstes, ursprüngliches Levelkonzept.



**Abbildung 6.10:** Der Test-Level und die verlassene Ruine in *Homunculus*.

und somit das Ziel dieses Levels.

Das Erstellen der virtuellen Welt hat auch den praktischen Grund gehabt, sich mit diversen Tools in UDK auseinandersetzen zu können. Beispiele hier sind das *Landscape Tool*, welches das Formen von Bodenlandschaften erlaubt, oder das *Mesh Painting Tool*, welches durch den Prozess der Maskierung andere Materialien und somit auch Texturen zum Vorschein bringen kann. Auch die Physik und Partikel Systeme in der Form von Wind, Blättern und Nebel haben sich als überzeugende Gestaltungsmittel hervorgehoben. Neben dem eigentlichen Level wurde auch ein Test-Level (siehe Abb. 6.10) erstellt, um gewisse Funktionen, wie die Responsiveness der Eingabebefehle, gesondert zu testen.

# Kapitel 7

## Schlussfolgerung

### 7.1 Konsequenzen

Die vorliegende Arbeit beschreibt die Rolle der Animation des Spielercharakters in Computerspielen als eine wichtige Form der Informationsvermittlung für den Spieler. Der Interaktionsprozess zwischen Spieler und Spiel trägt einen wesentlichen Beitrag zu dieser Informationsvermittlung bei. Ohne Eingabegerät und dessen Mapping kann der Spielercharakter keine Aktionen ausführen. Es ist somit die Interaktivität und das visuelle Feedback mittels Animationen welche, im Rahmen des Fokus auf Körperbewegung und Gestik, dem Spieler gewisse Zustände des Spielercharakters darstellen sollen. Dabei ist wichtig zu beachten, dass sich diese Arbeit nicht mit der Mimik, sondern nur mit der Körperbewegung und Gestik beschäftigt.

Die Rolle der Animation als Informationsvermittler hat sich in den letzten Jahren verstärkt etabliert. Neben höherer Rechenleistung und der Entwicklung moderner Technologie hat auch der Interaktionsprozess maßgeblich zu dieser Entwicklung beigetragen. Die verstärkte Verwendung des Analog-Sticks hat gegenüber dem D-Pad eine feinere und subtilere Form der Bewegung und Kontrolle des Spielercharakters erlaubt. Entsprechend neue Möglichkeiten von feiner Darstellung an Animationen sind dadurch entstanden. Die Entwicklung ergonomischer Controller und verbesserter Tastenlayouts (z. B. die Schultertasten und der Z-Trigger) haben dem Mapping des Controllers neue Möglichkeiten ausführbarer Aktionen gegeben. Durch die neuen Tastenkombinationen und die daraus resultierenden Aktionen sind entsprechend neue Animationen des Spielercharakters entstanden. Der Trend zur verstärkten Immersion des Spielers in die Spielwelt hat neben der Verbesserung der Eingabegeräte auch die virtuellen Interface Elemente stärker in die Spielwelt integriert. Neben der Integration dieser Elemente in Raum und/oder Kontext der Spielwelt (siehe Abschnitt 4.3.3) ist auch der Spielercharakter und dessen Animation verstärkt zum Informationsvermittler ehemaliger virtueller Interface Elemente geworden.

Während im klassischen Animationsfilm ein Charakter ohne Einfluss des Zuschauers und unter vorgegebenem Kamerawinkel seine Vorführung vorrichten kann, ist dies genau das Gegenteil beim Computerspiel. Im Spiel ist es der Spieler, welcher den Spielercharakter steuert und seinen Blickwinkel in der virtuellen Welt bestimmt. Anhand des modifizierten Mapping Modells von Jesse Schell (siehe Abschnitt 4.3.1) lassen sich verschiedene Schichten erkennen, welche der Prozess der Interaktion durchlaufen muss, um den Informationsfluss zwischen Spieler und Spiel zu gewährleisten. Zwischen dem Spieler und dem Spielercharakter liegt die physische und virtuelle Interface-Schicht. Es liegt somit nahe, dass eine erhöhte Transparenz dieser beiden Schichten den Spieler näher an den Spielercharakter bringen würde, um dadurch seine Immersion in die Spielewelt zu erhöhen. Da der Spieler die meiste Zeit den Fokus auf dem Spielercharakter hat, wäre dieser ein geeigneter Ort um Feedback Informationen an den Spieler zu schicken.

Unabhängig vom Medium haben Character-Animationen in Film und Spiel etwas gemeinsam: beide versuchen einen künstlich-erstellten Charakter mit der vom Mensch bekannten Körpersprache zum Leben zu erwecken. Die Körpersprache, ein Kommunikationsmittel, welches jeder Mensch versteht, drückt sich u.a. in verschiedenen Arten von Körperbewegungen und Gesten aus, welche von anderen Menschen, die sie sehen, wahrgenommen werden. Ein solches für den Menschen grundlegendes Kommunikationsmittel wäre daher eine geeignete Art, Informationen von einem anthropomorphen Spielercharakter und dessen Zustand, in der Form von Animationen, an den Spieler zu vermitteln. Informationselemente aus dem virtuellen Interface ließen sich somit am Charakter darstellen und wären ihrer ursprünglichen Darstellungsart überflüssig geworden.

Die in Kapitel 5 erwähnten *Immediate States* zeigen auf, dass diese Art der Zustände grundsätzlich schon länger in Spielen mit Spielercharakteren zu finden ist. Die *Persistent States* jedoch finden nur gelegentlich, meist aus zeitlichen und finanziellen Gründen von Seiten der Entwickler, Einzug in Spielen mit einem Spielercharakter. Der Grund liegt teilweise in der Art, wie sich diese Zustände ausdrücken. Gewisse Zustände wie der Erfahrungszustand haben einen sehr subtilen Ausdruck, welcher sich erst mit einem Vor- und Nachher Vergleich nach einer längeren Zeitspanne bemerkbar machen. Ein anderer Grund wäre die nicht immer vorhandene Eindeutigkeit einer Animation und deren Zustand, den sie vermitteln möchte. Die Information solcher Zustände lassen sich meist einfacher auf andere Arten (z.B. im HUD) vermitteln, weshalb Zeit und Ressourcen eher für andere Entwicklungstätigkeiten verwendet werden. Jedoch abhängig von Genre und der Designentscheidung würde die Umsetzung solcher Zustände in der Form von Animationen viel zur Immersion des Spielers in die Spielewelt beitragen.

Zu diesem Zweck wurde die Umsetzung von drei verschiedenen *Persistent States* in der Form eines kleinen Third-Person Spieleprojektes getestet. Das Resultat zeigt, dass eine solche Umsetzung möglich ist, jedoch manche

subtilere Zustände (in diesem Fall der Erfahrungszustand) nicht eindeutig genug über Animationen vermittelt werden können. Das Problem könnte hier an der Eigenheit der Third-Person Kamera und dem fehlenden Einsatz an Gesichtsanimationen liegen. Die Mimik wurde ursprünglich ausgelassen, da der Spieler im Spiel meist den Rücken des Spielercharakters sieht. Durch diese Kamerasicht entgehen dem Spieler gewisse Körperbewegungen, allen voran die Mimik des Spielercharakters. Änderungen bei der Responsiveness des Controllers unterstützen zwar die Vermittlung, es fehlt jedoch beim Erfahrungszustand die eigene persönliche Sinneserfahrung, welche nur schwer primär alleine über den visuellen Sinneskanal vermittelt werden kann.

Ein Zustand kann auf eine übertriebene bzw. theatralische Art vermittelt werden. Dies hat den Vorteil, dass kleine Nuancen in der Körperbewegung und Gestik stärker und offensichtlicher erkannt werden können. Eine andere Möglichkeit wäre der Einsatz von wiedererkennbaren Körperbewegungen und Gesten, welche zwar nur annähernd den gewollten Zustand beschreiben, im Kontext des Spieles jedoch zur gewünschten Zustandsvermittlung führen.

In Fällen, wo eine Animation alleine den Zustand eines Spielercharakters nicht überzeugend vermitteln kann, können andere visuelle Kennzeichen (z.B. Farb- oder Texturänderungen) verwendet werden. Eine Unterstützung solcher Änderungen durch Animationen, welche annähernd versuchen, den gewollten Zustand durch Körperbewegung zu vermitteln, würden, trotz ihrer ergänzenden Funktion, den Spieler auf einer ihm bekannten Kommunikationsart Informationen liefern. Im Vergleich zum Einsatz von Animationen als primäres Mittel würden Animationen in einer ergänzenden Rolle somit eine ebenfalls wichtige Rolle in der Informationsvermittlung spielen.

## 7.2 Ausblick

Die Entwicklung von neuen Technologien, Methoden, Eingabegeräten und schneller Rechenleistung erlaubt immer komplexere Animationen am Spielercharakter. Ein Resultat dieser Entwicklung ist jedoch das *Uncanny Valley* Problem (siehe 3.3). Um dieses Problem zu vermeiden sollte darauf geachtet werden, dass es wichtiger ist, Animationen nicht realistisch, sondern überzeugend darstellen zu lassen. Eine Animation, welche dem Spieler gefällt und ihm nicht „falsch“ vorkommt ist wichtiger als eine, welche annähernd realistisch ist, jedoch durch diese Annäherung dem Spieler fremd wirkt.

Die Wichtigkeit der Körpersprache des Spielercharakter, insbesondere der Körperbewegung und Gestik, findet immer mehr Anklang in der Spieleentwicklung, wie es beispielsweise *Assassin's Creed*, *Prince of Persia* und die *Team Ico* Spiele gezeigt haben. Der Mensch genießt es, die Vorführung von Körperbewegungen und Gestik anderer Menschen zu beobachten. Dies lässt sich klar bei Schauspielern, Pantomimen, Athleten und sogar Politikern und Nachrichtensprechern erkennen. Auch die überzeugende Vorführung eines

Spielercharakters würde zum Genuss des Spielers führen, wenn die Steuerung und Animation die Erwartungen des Spielers erfüllt. Aus diesem Grund wird in Spielen immer öfters darauf geachtet, überzeugende Animationen mitsamt einer überzeugenden Steuerung dem Spieler zu liefern.

Der Trend der Entwicklung zu immer komplexeren Character-Animationen wird auch in Zukunft verstärkt weitergeführt, wie es Spiele wie *The Last Guardian* versprechen. *Persistent States* jedoch werden derzeit noch in wenigen Fällen als Animationen realisiert. Meistens ist es der Gesundheitszustand, welcher u.a. durch das Festhalten des eigenen Körpers dargestellt wird. *Shadow of the Colossus* und *Gears of War 2* sind gute Beispiele für den Einsatz solcher Animationen. Letzteres Beispiel geht einen Schritt weiter und ändert das Gameplay bei einer fast tödlichen Wunde, dargestellt durch den langsamen, am Boden kriechenden Spielercharakter. Die Entwicklung scheint in Richtung der Darstellung von *Gears of War 2* zu gehen, sodass der Gesundheitszustand sich durch eine vermehrt ändernde Animation und einschränkende Bewegung ausdrückt. Hier bestimmt jedoch auch das Game Design die Anwendung einer solchen Darstellung.

Die ersten beiden *Gothic* Spiele sind gute Vorzeigebeispiele für die Anwendung eines Geschicklichkeitszustandes. Mit der steigenden Waffenfertigkeit ändert sich die Animation und Responsiveness beim Angriff. Diese Vermittlung eines fortschreitenden Geschicklichkeitszustandes ist ideal für das Genre der Rollenspiele. Leider gab es noch wenig bekannte Entwicklungen in dieser Richtung.

Das Abbilden mehrerer Charakter Zustände in Form von Animationen und dem Mapping der Eingabebefehle, allen voran die Kategorie der *Persistent States*, wären jedenfalls ein erster Schritt, dem Spieler eine überzeugende Vorführung des Spielercharakters zu liefern.

# Anhang A

## Informationen und Hilfestellung zu UDK

Die folgenden Informationen und Links waren in der Entwicklung des Projektes sehr nützlich und hilfreich. Die Hoffnung besteht, dass für den interessierten Leser und Entwickler die Einarbeitung in UDK mit diesen Informationen etwas leichter ausfallen wird. Achtung: jeden Monat wird eine neue UDK Version veröffentlicht. Es kann passieren, dass diverse Informationen in den Links ab einer gewissen Build-Version veraltet sind.

### A.1 Dokumentationen und Tutorials

- <http://udn.epicgames.com/Three/WebHome.html>  
UDN: Die offizielle Dokumentation zu UDK. Die Seite wurde im August 2011 aktualisiert und ist dadurch viel übersichtlicher und hilfreicher geworden. Hier sollte man mit der Recherche oder Einarbeitung beginnen.
- <http://wiki.beyondunreal.com/>  
Unreal Wiki: Technische Dokumentation zu diversen Aspekten von UDK (u.a. UnrealScript). Sehr nützlich und hilfreich während der Programmierung.
- <http://udn.epicgames.com/Three/DevelopmentKitBuildUpgradeNotes.html>  
Eine Auflistung aller monatlichen UDK Builds. Aus Gründen der Feature Übersicht sehr hilfreich.
- <http://www.moug-portfolio.info/index.php?page=tutorials>  
Pflichtlektüre: Sehr empfehlenswerte Tutorials zu diversen UDK Themen. Allen voran der „Input commands“ Abschnitt zählt zur Pflichtlektüre.
- <http://forecourse.com/unreal-tutorials/>

Pflichtlektüre: Diverse Tutorials, insbesondere die UnrealScript und Scaleform Tutorials sind erwähnenswert.

- <http://udkc.info/index.php?title=Category:Tutorials>  
Sehr ausführliche Tutorial Liste zu diversen UDK Themen. Sehr empfehlenswert!
- <http://www.moddb.com/members/raven67854/tutorials>  
<http://www.hourences.com/tutorials/>  
<http://www.showmethatagain.com/?p=270>  
Diverse Tutorials zu UDK.
- <http://www.hourences.com/an-entire-simple-udk-game/>  
Ein grundlegendes Tutorial für ein einfaches Setup um UDK für das eigene Spiel vorzubereiten. Hier wird auch auf das Packaging vom Spiel eingegangen.

## A.2 Animationen und Blendshapes in UDK

- <http://www.3dbuzz.com/vbforum/showthread.php?181630-Animation-into-UDK/page2&s=54b0b370cfc85d64eed7586ffa7e99ab>  
Notizen zum Importieren von Animationen in UDK.
- <http://romanoachsenfahrt.blogspot.com/p/udk-tutorial.html>  
<http://forums.epicgames.com/threads/717339-Guide-Custom-Character-with-Custom-Animation>  
Tutorial: Importieren eines Charakters mitsamt Animationen in UDK und deren Einrichtung. Notizen zur Vorbereitung eines Charakters für den Importprozess sind gegeben. Der Forum Thread beinhaltet noch diverse weitere aufklärende Informationen.
- [http://udkc.info/index.php?title=Tutorials:Wraith's\\_\\_AnimTree\\_\\_crash\\_\\_course](http://udkc.info/index.php?title=Tutorials:Wraith's__AnimTree__crash__course)  
Tutorial: Einführung in den AnimTree von UDK. Pflichtlektüre!
- <http://www.youtube.com/watch?v=KprHpS0ZFv0>  
Tutorial: Animation Blending in Matinee.
- <http://forums.epicgames.com/threads/705696-Tips-regarding-new-things-in-UDK-compared-to-UT3>  
Informationen zu Additive Animationen und animierte Morph/Material Tracks.
- <http://forums.epicgames.com/threads/741324-Morph-Targets-Blend-Shapes?p=27578050>  
Informationen zu Morph Targets.
- [http://forums.epicgames.com/threads/762041-\(yet-another\)-quot-hey-how-do-i-control-my-own-character-quot-thread!](http://forums.epicgames.com/threads/762041-(yet-another)-quot-hey-how-do-i-control-my-own-character-quot-thread!)  
Informationen bzg. dem Kontrollieren eines eigenen Charakters in UDK.

- <http://forums.epicgames.com/archive/index.php/t-760858.html>  
Zugriff auf AnimSet via UnrealScript.
- <http://forums.epicgames.com/threads/767288-Matinee-question-How-to-manipulate-the-players-camera-via-Matinee>  
Spielerkamera in Matinee kontrollieren und animieren.
- <http://forums.epicgames.com/threads/719148-playcustomanim-function-not-finding-my-custom-animset>  
Abspielen eigener Animationen in UnrealScript.
- <http://forums.epicgames.com/threads/714149-Tutorial-Creating-custom-AnimNodes>  
Informationen zum Erstellen eigener AnimNodes.
- <http://forums.epicgames.com/threads/738008-How-does-Unreal-Engine-identify-which-animations-to-play-on-a-pawn>  
Hantieren von Animationen in UnrealScript.
- <http://forums.epicgames.com/archive/index.php/t-716039.html>  
OnAnimEnd Event.

### A.3 Cloth

- [http://www.nvidia.de/object/ue3\\_physx.html](http://www.nvidia.de/object/ue3_physx.html)  
nVidia PhysX Training Videos für UDK.
- [http://www.youtube.com/watch?v=NUpvSFrZKFQ&feature=player\\_embedded](http://www.youtube.com/watch?v=NUpvSFrZKFQ&feature=player_embedded)  
Physics Assets Tutorial.
- <http://forums.epicgames.com/threads/740462-UDK-cloth-from-Maya>  
Diverse Informationen bzgl. Cloth in UDK.
- <http://forums.epicgames.com/threads/709921-Cloth-simulation-on-player-character/page2?highlight=cloth%20collision>  
Cloth simulation am Spielercharakter. Sehr empfehlenswert!
- [http://forums.epicgames.com/threads/753734-multiple-meshes-and-clothing-\(cloth-sim\)](http://forums.epicgames.com/threads/753734-multiple-meshes-and-clothing-(cloth-sim))  
<http://forums.epicgames.com/threads/724998-Character-with-hanging-clothes>  
<http://forums.epicgames.com/threads/713941-Skeletal-meshe-s-cloth-colliding-with-itself>  
<http://forums.epicgames.com/threads/740462-UDK-cloth-from-Maya>  
Informationen zu Cloth am Spielercharakter.

### A.4 Rigging

- <http://forums.cgsociety.org/archive/index.php/t-889108.html>  
Hilfestellung zum Riggingprozess für UDK.

- <http://forums.epicgames.com/archive/index.php/t-773258.html>  
<http://forums.epicgames.com/threads/720751-Non-character-animation>  
Verschiedene Informationen bzg. Rigging für UDK.

## A.5 3rd Person Camera

- <http://forums.epicgames.com/threads/750868-UDK-action-game-3rd-person-camera-code>  
Pflichtlektüre: Erstellen einer Third-Person Kamera.
- <http://forecourse.com/2010/03/creating-a-third-person-camera/>  
Tutorial: Infos zum Erstellen einer 3rd Person Kamera. Empfehlenswert!
- <http://forums.epicgames.com/threads/723032-KISMET-CameraActor-No-Handler-for-CameraLookAt>  
Tutorial: Kamera-Setup ähnlich einer Überwachungskamera.
- <http://forums.epicgames.com/threads/756352-3rd-person-action-camera-highlight=xbox%20controller>  
Gutes 3rd-Person Kamera Tutorial.
- <http://forums.epicgames.com/threads/712633-Accessing-camera-class-from-player-controller>  
Zugriff auf Kamera von PlayerController Klasse.

## A.6 Input Customization

- <http://okita.com/alex/?p=671>  
<http://okita.com/alex/?p=740>  
<http://forums.epicgames.com/threads/767538-Custom-Input?highlight=xbox%20controller>  
Notizen zum Zugriff auf Input Daten.
- <http://www.youtube.com/user/ContagionTutorials#p/u/5/G-c9KPNdISk>  
Erstellung einer neuen PlayerInput Klasse.
- <http://www.moug-portfolio.info/index.php?page=input-commands>  
<http://forums.epicgames.com/archive/index.php/t-720608.html>  
Pflichtlektüre bzg. Input Handling! Auch der Zugriff auf die Daten eines XBox360 Controllers wird hier erklärt.
- <http://unrealmindset.wordpress.com/2010/08/09/wall-climbing-in-the-creature/>  
<http://forums.epicgames.com/threads/736304-Ledge-Grabbing>  
Tutorial und Infos zu Wall Climbing.
- <http://forums.epicgames.com/threads/726686-Simple-keypress-detection-in-UDK-using-Kismet>

Einfache Tastendruck-Erkennung in UDK.

- <http://forums.epicgames.com/threads/760905-Animation-with-keybindings>  
Animationen durch Tasten abspielen lassen (siehe auch Seite 3 vom Thread).
- <http://forums.epicgames.com/threads/711990-A-Quick-Custom-UserInput-Guide?highlight=quick%20input%20guide>  
Custom UserInput Guide und Infos.
- <http://forums.epicgames.com/archive/index.php/t-727045.html>  
XBox 360 Controller Tasten Mapping.
- <http://forums.epicgames.com/threads/772578-How-do-you-change-the-walk-speed-of-your-character>  
<http://forums.epicgames.com/threads/723871-About-Pawn-s-speed>  
<http://forums.epicgames.com/threads/737122-Changing-Backward-Speed>  
Empfehlenswert: Geschwindigkeit des Charakters ändern.

## A.7 Programming

- <http://forums.epicgames.com/threads/762910-Solved-Trigger-Volume-Touch-Event-detectable-in-code-this-way-or-other-Read-pls>  
Empfehlenswert: Trigger Volume Event in UnrealScript bearbeiten.
- <http://forums.epicgames.com/threads/585068-Handling-Kismet-events-in-UnrealScript>  
Kismet Events in UnrealScript behandeln.
- <http://forums.epicgames.com/threads/748876-Trigger-Kismet-Remote-Events-From-Script>  
Trigger Kismet Remote Events From Script.
- <http://forums.epicgames.com/threads/749787-Console-Event-From-UnrealScript-to-Kismet>  
Console Event From UnrealScript to Kismet.
- <http://forums.epicgames.com/threads/726873-Adding-footstep-sounds>  
Hinzufügen von Schritte und Sounds.
- <http://forums.epicgames.com/archive/index.php/t-729989.html>  
<http://forums.epicgames.com/threads/727721-Forcing-Tick-s-DeltaTime-to-be-a-constant-possible>  
<http://forums.epicgames.com/threads/717076-DesiredRotation-not-working-in-UDK>  
Informationen zur tick() Funktion.
- <http://forums.epicgames.com/threads/745191-Typecasting-tutorial>  
Typecasting in UnrealScript.

## Anhang B

# Input Prototyp

Um das Input Mapping vom Geschicklichkeitszustand zu testen, wurden drei Prototypen in *Construct* erstellt. Die drei Prototypen beinhalten verschiedene Geschicklichkeitszustände: *Balancieren*, *Klettern* und *Seilklettern*. Der *XBox 360* Controller diente in allen drei Fällen als das Eingabegerät. Um eine interessante Tastenbelegung zu erhalten, wurde das Feedback von erfahrenen und nicht-erfahrenen Spielern eingeholt.

Von diesen drei Prototypen wurde der *Klettern* Prototyp (siehe Abb. B.1) im Projekt realisiert.

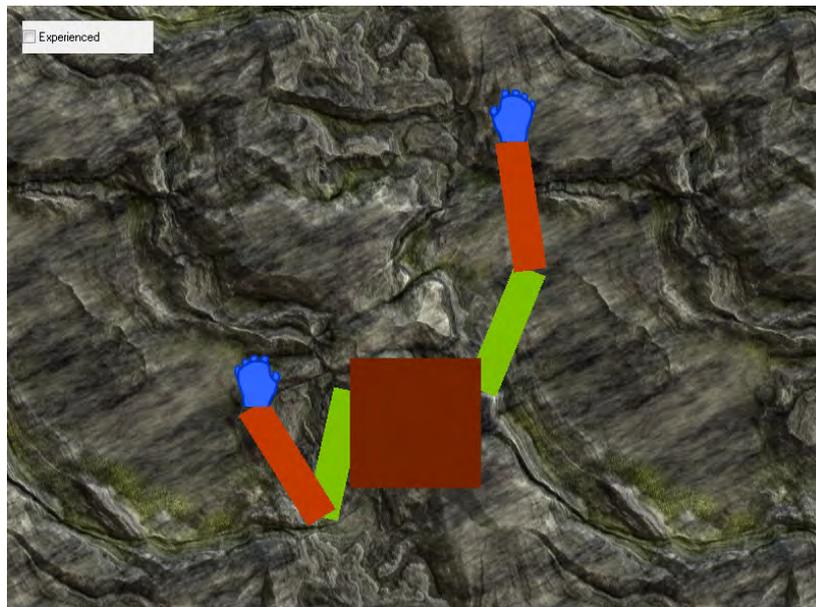


Abbildung B.1: Beispielbild vom *Klettern* Prototyp

## Anhang C

# Animation Middleware

### **OC3 FaceFX:**

,Solution for creating realistic facial animation from audio files.'  
<http://www.facefx.com>

### **Annosoft Lipsync:**

,Produces accurately timed mouth positions and phonemes from a wave file.'  
<http://www.annosoft.com/lipsync-sdks>

### **Havok Animation:**

,Animation SDK and tool chain that provides optimized playback and real-time blending.'  
[http://www.havok.com/index.php?page=havok-animation&hl=en\\_US](http://www.havok.com/index.php?page=havok-animation&hl=en_US)

### **Natural Motion Morpheme:**

Graphical authoring environment.  
<http://www.naturalmotion.com/morpheme>

### **Natural Motion Euphoria:**

,Creates animation on-the-fly, combining artificial intelligence, biomechanics and physics.'  
<http://www.naturalmotion.com/euphoria>

### **Rad Game Tools Granny 3D:**

3D toolkit that includes runtime animation, a character creator and 3D package exporting.  
<http://www.radgametools.com/granny.html>

**Havok Behaviour:**

, Cross-platform development system for creating dynamic event-driven character behaviours in a game.‘

<http://www.havok.com/index.php?page=havok-behavior>

**Autodesk HumanIK:**

, The software’s full body inverse kinematics (FBIK) system enables characters to interact realistically with their environment and other characters.‘

<http://usa.autodesk.com/adsk/servlet/pc/index?id=9491249&siteID=123112>

**Emotion FX Extended:**

, Allows artists to setup hierarchical state machines, logic and advanced hierarchical blend trees, all in a visual way with direct feedback using either the interface components or directly from the game controller.‘

<http://www.mysticgd.com/>

# Anhang D

## Inhalt der DVD

**Format:** DVD-R, Single Layer, ISO9660/Joliet/UDF Format

### D.1 Masterarbeit

**Pfad:** /

Koller\_Juergen\_2011.pdf Masterarbeit.

### D.2 Literatur

**Pfad:** /Literatur/

Online/ . . . . . Kopien der Internetseiten.

Praesentationen/ . . . . . Diverse Präsentationen in Form von Folien  
und Audio Aufnahmen.

### D.3 Projekt

**Pfad:** /Projekt/

Dokumentation/ . . . . . Das Game Treatment und Game Design  
Dokument.

Media/ . . . . . Concept Art, Screenshots, Video Capture.

Projekt/ . . . . . Die Source Files des Projektes. Die  
relevanten Pfade sind  
/Development/Src/HomunculusMod,  
/UDKGame/Config und  
/UDKGame/Content.

Bin/ . . . . . Zwei ausführbare Dateien: einmal das  
komplette Spiel, und einmal eine Version mit  
einfachem Level (aus Performance Gründen).

## D.4 Input Prototype

**Pfad:** /Input Prototype/

Prototype - Balancing/ Prototyp für Balancieren.

Prototype - Rope Climbing/ Prototyp für Seilklettern.

Prototype - Wall Climbing/ Prototyp für Klettern.

# Quellenverzeichnis

## Literatur

- [1] Jeremy Bailenson. *Avatars*. 2008. URL: <http://www.stanford.edu/~bailenso/papers/avatars.pdf>.
- [2] Pippin Barr, James Noble und Robert Biddle. „Video game values: Human-computer interaction and games“. In: *Interacting with Computers* 19.2 (2007). HCI Issues in Computer Games, S. 180–195. URL: <http://www.sciencedirect.com/science/article/B6V0D-4M1CYXM-1/2/43d5bafa6433a2e706f850ba26eff783>.
- [3] Severin Brettmeister und Tom Wendel. „Der Weg zum optimalen Interface – Schnittstelle von Mensch und Computer“. In: *Making Games Magazin* 19.1 (2010), S. 34–35.
- [4] Erik Fagerholt und Magnus Lorentzon. „Beyond the HUD — User Interfaces for Increased Player Immersion in FPS Games“. Magisterarb. Gothenburg, Sweden: Chalmers University of Technology, Sep. 2009. URL: <http://publications.lib.chalmers.se/cpl/record/index.xsql?pubid=111921>.
- [5] Gary Faigin. *The Artist's Complete Guide to Facial Expressions*. New York: Watson-Guption Publications, 1992.
- [6] Chaim Gingold. „Miniature Gardens & Magic Crayons: Games, Spaces, & Worlds“. Magisterarb. Atlanta, USA: Georgia Institute of Technology, School of Literature, Communication and Culture, Apr. 2003. URL: <http://levitylab.com/cog/writing/thesis>.
- [7] Ed Hooks. *Acting for Animators*. 2. Aufl. Portsmouth: Heinemann, 2003.
- [8] Katherine Isbister. *Better Game Characters by Design — A Psychological Approach*. San Francisco: Elsevier, 2006.
- [9] Keith Johnstone. *Impro — Improvisation and the Theatre*. 3. Aufl. London: Methuen Drama, 2007.

- [10] Renate Jonke. *Nonverbale Kommunikation / Körpersprache (Theorie)*. 2006. URL: <http://homepage.univie.ac.at/margarete.halmetschlager/LVX/erleben.html>.
- [11] Khang Le u. a. *The Skillful Huntsman*. Culver City: Design Studio Press, 2005.
- [12] William Lidwell, Kritina Holden und Jill Butler. *Universal Principles of Design*. Massachusetts: Rockport Publishers, 2003.
- [13] Samy Molcho. *Körpersprache*. München: Goldmann Verlag, 1998.
- [14] Desmond Morris. *People Watching — The Desmond Morris Guide to Body Language*. 2. Aufl. London: Vintage Books, 2002.
- [15] Donald A. Norman. *The design of everyday things*. 2. Aufl. New York: Doubleday, 1990.
- [16] Lloyd Rieber. „Animation as feedback in a computer-based simulation: Representation matters“. In: *Educational Technology Research and Development* 44.1 (1996), S. 5–22. URL: <http://dx.doi.org/10.1007/BF02300323>.
- [17] George G. Robertson, Stuart K. Card und Jock D. Mackinlay. „The cognitive coprocessor architecture for interactive user interfaces“. In: *Proceedings of the 2nd annual ACM SIGGRAPH Symposium on User Interface Software and Technology*. (Williamsburg, Virginia, United States). New York: ACM, 1989, S. 10–18.
- [18] Jesse Schell. *The Art of Game Design — A Book of Lenses*. Burlington: Elsevier, 2008.
- [19] Brandon Sheffield. „First Person — Is it really more immersive?“ In: *Game Developer* 16.7 (August 2009), S. 2.
- [20] Tom Smith. „Intelligent Brawling — Analyzing the Genre to Build a Better Beat ’Em Up“. In: *Game Developer* 15.6 (June/July 2008), S. 15–24.
- [21] Sato Takayoshi. „The Dust of Everyday Life — The Art of Building Characters“. In: *Game Developer* 16.11 (December 2009), S. 16–22.
- [22] Steve Theodore. „Big Screen Blues“. In: *Game Developer* 17.8 (September 2010), S. 34–35.
- [23] Frank Thomas und Ollie Johnston. *The Illusion of Life — Disney Animation*. New York: Disney Editions, 1981.
- [24] Mick West. „Pushing Buttons“. In: *Game Developer* 12.5 (May 2005), S. 19–26. URL: <http://cowboyprogramming.com/2007/01/02/pushing-buttons>.

## Online-Quellen

- [25] URL: <http://www.britannica.com/EBchecked/topic/186011/empathy> (besucht am 22.09.2011).
- [26] URL: <http://www.playstationpro2.com/media/images3/The-Last-Guardian-3.jpg>.
- [27] URL: <http://udn.epicgames.com/Three/UsingSkeletalControllers.html>.
- [28] URL: [http://de.wikipedia.org/wiki/Uncanny\\_Valley](http://de.wikipedia.org/wiki/Uncanny_Valley).
- [29] URL: [http://www.valvesoftware.com/publications/2007/NPAR07\\_IllustrativeRenderingInTeamFortress2.pdf](http://www.valvesoftware.com/publications/2007/NPAR07_IllustrativeRenderingInTeamFortress2.pdf).
- [30] URL: <http://www.sfgalleries.net/art/sf4/sf4/series1/>.
- [31] URL: <http://pcmedia.ign.com/pc/image/article/693/693792/call-of-cthulhu-dark-corners-of-the-earth-20060306020328473.jpg>.
- [32] URL: [http://www.diablowiki.com/images/e/e2/Artisan\\_interaction.png](http://www.diablowiki.com/images/e/e2/Artisan_interaction.png).
- [33] URL: <http://worldofwarcraftscreenshots.com/showphoto.php/photo/687/size/big/cat/>.
- [34] URL: [http://en.wikipedia.org/wiki/File:360\\_controller.svg](http://en.wikipedia.org/wiki/File:360_controller.svg).
- [35] Evan Amos. URL: <http://commons.wikimedia.org/wiki/User:Evan-Amos/VideoGames>.
- [36] Marcus Andrews. *Game UI Discoveries: What Players Want*. Feb. 2010. URL: [http://www.gamasutra.com/view/feature/4286/game\\_ui\\_discoveries\\_what\\_players\\_.php](http://www.gamasutra.com/view/feature/4286/game_ui_discoveries_what_players_.php) (besucht am 04.08.2011).
- [37] Carlos Baena. *Status and Personal Space*. Feb. 2011. URL: [http://www.carlosbaena.com/resource/resource/\\_tips/\\_status.html](http://www.carlosbaena.com/resource/resource/_tips/_status.html) (besucht am 21.03.2011).
- [38] Steven Conway. *A Circular Wall? Reformulating the Fourth Wall for Video Games*. Juli 2009. URL: [http://www.gamasutra.com/view/feature/4086/a\\_circular\\_wall\\_reformulating\\_the\\_.php?print=1](http://www.gamasutra.com/view/feature/4086/a_circular_wall_reformulating_the_.php?print=1) (besucht am 19.04.2011).
- [39] Jonathan Cooper. *Basics: Animation Blending*. Juni 2005. URL: <http://www.gameanim.com/2005/06/19/blending-the-future-of-non-linear-animation/> (besucht am 22.05.2011).
- [40] Noah Falstein. *Interactive "Show, Don't Tell": Fundamental Principles of Interactive Entertainment*. 1996. URL: <http://www.theinspiracy.com/ArShowDT.htm> (besucht am 22.03.2011).
- [41] Steve Gargolinski, Phil Carlisle und Michael Mateas. *Suspending Disbelief — Bringing Your Characters to Life With Better AI*. 2010. URL: <http://www.gdcvault.com/free/gdc-10> (besucht am 11.05.2011).

- [42] Kevin Gifford. *Fumito Ueda Talks About Last Guardian's Impeccable Animation*. März 2011. URL: <http://www.1up.com/news/fumito-ueda-talks-guardian-impeccable> (besucht am 22.03.2011).
- [43] Carrie Gouskos. *Assassins Creed — Dealing with uncanniness*. Juli 2006. URL: [http://www.cnet.com.au/the-depths-of-the-uncanny-valley\\\_p3-240090842.htm](http://www.cnet.com.au/the-depths-of-the-uncanny-valley\_p3-240090842.htm) (besucht am 23.03.2011).
- [44] Benjamin Heckendorn. *Controller monitor built for Infinity Ward game testing*. Dez. 2008. URL: <http://benheck.com/12-05-2008/controller-monitor-built-for-infinity-ward-game-testing> (besucht am 15.03.2011).
- [45] Ivo Herzeg. *Crysis 2: Getting More Interactivity out of Animation Data*. 2011. URL: <http://www.gdcvault.com/free/gdc-11> (besucht am 10.05.2011).
- [46] Ed Hooks. *Ed Hooks' Monthly Newsletter*. Aug. 2002. URL: <http://www.edhooks.com/newslett/Aug02.htm> (besucht am 21.07.2011).
- [47] Mike Jungbluth. *Growing Game Animation — Transitions & Player Input*. Apr. 2011. URL: <http://altdevblogaday.org/2011/04/10/growing-game-animation-transitions-player-input/> (besucht am 14.04.2011).
- [48] Aleissia Laidacker und Nicolas Barbeau. *Living Crowds - AI & Animation in Assassin's Creed: Brotherhood*. 2011. URL: [www.gdcvault.com/free/gdc-11](http://www.gdcvault.com/free/gdc-11) (besucht am 10.05.2011).
- [49] Richard Leadbetter. *Console Gaming: The Lag Factor*. Sep. 2009. URL: <http://www.eurogamer.net/articles/digitalfoundry-lag-factor-article> (besucht am 15.03.2011).
- [50] McIntosh, Travis. *Animation and Player Control in Uncharted 1 & 2*. 2010. URL: <http://www.gdcvault.com/free/gdc-10> (besucht am 17.04.2011).
- [51] Jordan Mechner. *Karateka*. Jan. 2009. URL: <http://jordanmechner.com/karateka/> (besucht am 20.03.2011).
- [52] Jordan Mechner. *Prince of Persia*. Okt. 2008. URL: <http://jordanmechner.com/prince-of-persia/> (besucht am 20.03.2011).
- [53] Christiaan Moleman. *Animation as Gameplay #2*. Okt. 2010. URL: [http://www.gamasutra.com/blogs/ChristiaanMoleman/20101026/6290/Animation\\_as\\_Gameplay\\_2.php](http://www.gamasutra.com/blogs/ChristiaanMoleman/20101026/6290/Animation_as_Gameplay_2.php) (besucht am 21.03.2011).
- [54] Christiaan Moleman. *Anticipation in Games*. März 2009. URL: [http://www.gamasutra.com/blogs/ChristiaanMoleman/20090319/868/Anticipation\\_in\\_Games.php](http://www.gamasutra.com/blogs/ChristiaanMoleman/20090319/868/Anticipation_in_Games.php) (besucht am 21.03.2011).
- [55] Christiaan Moleman. *Good practice — Animation as Gameplay #1*. Apr. 2009. URL: [http://www.gamasutra.com/blogs/ChristiaanMoleman/20090405/1058/Good\\_practice\\_\\_Animation\\_As\\_Gameplay.php](http://www.gamasutra.com/blogs/ChristiaanMoleman/20090405/1058/Good_practice__Animation_As_Gameplay.php) (besucht am 21.03.2011).

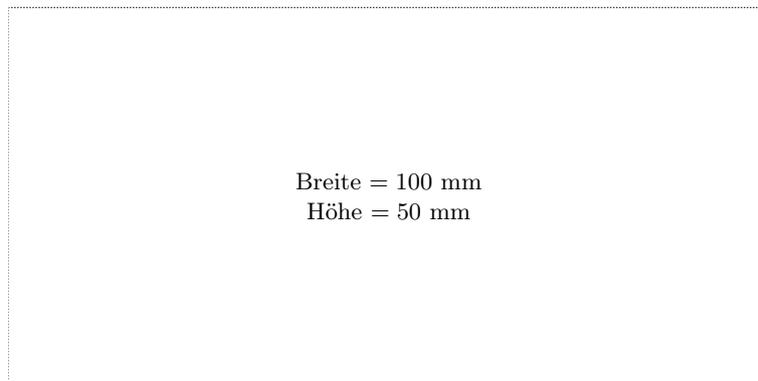
- [56] Christiaan Moleman. *IAnim – Team Ico*. Feb. 2009. URL: <http://www.ninjadodo.net/dodoblog/archives/2009/02/22/ianim-team-ico/> (besucht am 20.03.2011).
- [57] Christiaan Moleman. *IAnim – Valve*. März 2009. URL: <http://www.ninjadodo.net/dodoblog/archives/2009/03/08/ianim-valve/> (besucht am 17.03.2011).
- [58] Christiaan Moleman. *The Necessity Of Interactive Animation For Games*. Juni 2009. URL: [http://www.gamasutra.com/view/feature/4054/the\\\_necessity\\\_of\\\_interactive\\\_php?print=1](http://www.gamasutra.com/view/feature/4054/the\_necessity\_of\_interactive\_php?print=1) (besucht am 21.03.2011).
- [59] Kristan Reed. *ICO Review*. Feb. 2006. URL: [http://www.eurogamer.net/articles/r\\_icoreview\\_ps2](http://www.eurogamer.net/articles/r_icoreview_ps2) (besucht am 19.05.2011).
- [60] Barbara Robertson. *CGSociety – The Act*. Nov. 2006. URL: [http://features.cgsociety.org/story\\_custom.php?story\\_id=3826](http://features.cgsociety.org/story_custom.php?story_id=3826) (besucht am 17.03.2011).
- [61] Mark Serrels. *First Or Third Person – What’s Your Perspective?* Apr. 2011. URL: <http://www.kotaku.com.au/2011/04/first-or-third-person-w-hats-your-perspective/> (besucht am 05.05.2011).
- [62] David Sirlin. *Subtractive Design*. Juli 2009. URL: <http://www.sirlin.net/articles/subtractive-design.html> (besucht am 20.03.2011).
- [63] Joe Spataro und Tam Armstrong. *Bungie – The Animation of Halo: Reach*. 2011. URL: <http://www.gdcvault.com/free/gdc-11> (besucht am 10.05.2011).
- [64] Michael Sporn. *Walt-Graham communique*. Transcript here: <http://www.lettersofnote.com/2010/06/how-to-train-animator-by-walt-disney.html>. Juli 2009. URL: <http://www.michaelspornanimation.com/splog/?p=1917> (besucht am 06.05.2011).
- [65] Edge Staff. *Emotional Characters the Real Challenge*. Nov. 2006. URL: <http://www.next-gen.biz/features/emotional-characters-real-challenge> (besucht am 23.03.2011).
- [66] Edge Staff. *Half-Life 2*. Aug. 2005. URL: <http://www.next-gen.biz/features/half-life-2> (besucht am 23.03.2011).
- [67] Edge Staff. *The Making of... Prince of Persia*. Jan. 2009. URL: <http://www.next-gen.biz/features/making-of-prince-persia?page=4> (besucht am 23.03.2011).
- [68] IGN Staff. *Prince of Persia Interview – Alex Drouin*. Juli 2003. URL: <http://cube.ign.com/articles/428/428654p1.html> (besucht am 23.03.2011).

- [69] Hajime Sugiyama. *GDC06 Postmortem — The Emotional Character Control of Shadow of the Colossus*. Audio recording of presentation. März 2006. URL: <https://www.cmpevents.com/GD06/a.asp?option=C&V=11&SessID=2815> (besucht am 17.04.2011).
- [70] Steve Swink. *Deconstructing “Feel”*. Feb. 2007. URL: <http://www.steveswink.com/articles/deconstructing-feel-1-of-3/> (besucht am 15.03.2011).
- [71] Steve Swink. *Principles of Virtual Sensation*. Juli 2006. URL: <http://www.steveswink.com/principles-of-virtual-sensation/> (besucht am 15.03.2011).
- [72] Steve Swink. *Prototyping for Game Feel (v.2)*. Okt. 2007. URL: <http://www.steveswink.com/articles/prototyping-for-game-feel-version-2/> (besucht am 15.03.2011).
- [73] Max von Tettenborn. *Gothic – Spielen ohne Zahlen*. Okt. 2009. URL: [http://www.makinggames.de/index.php/magazin/120\\\_spielen\\\_ohne\\\_zahlen](http://www.makinggames.de/index.php/magazin/120\_spielen\_ohne\_zahlen) (besucht am 17.03.2011).
- [74] Jim Thacker. *Q&A: Autodesk’s Project Skyline team*. März 2011. URL: <http://www.cgchannel.com/2011/03/qa-autodesks-project-skyline-team/> (besucht am 19.05.2011).
- [75] Hiroki Ueno. *The Next Generation of Fighting Games — Physics & Animation in UFC 2009 Undisputed*. 2010. URL: <http://www.gdcvault.com/free/gdc-10> (besucht am 10.05.2011).
- [76] Jason Vandenberghe. *FMX 2011: Tales from Red Steel 2 — How Motion Control Will/Won’t Save Us All*. Video recording of presentation. Mai 2011. URL: <http://www.fmx.de/media/streaming-archive.html> (besucht am 10.05.2011).
- [77] Mick West. *Measuring Responsiveness in Video Games*. Mai 2008. URL: <http://cowboyprogramming.com/2008/05/30/measuring-responsiveness-in-video-games/> (besucht am 15.03.2011).
- [78] Mick West. *Programming Responsiveness*. Mai 2008. URL: <http://cowboyprogramming.com/2008/05/27/programming-responsiveness/> (besucht am 15.03.2011).
- [79] *Wikipedia: Character Animation*. URL: [http://de.wikipedia.org/wiki/Character\\_animation](http://de.wikipedia.org/wiki/Character_animation) (besucht am 21.07.2011).
- [80] *Wikipedia: Gestik*. URL: <http://de.wikipedia.org/wiki/Gestik> (besucht am 25.07.2011).
- [81] *Wikipedia: Half-Life*. URL: <http://de.wikipedia.org/wiki/Half-Life> (besucht am 21.07.2011).

- [82] Greg Wilson. *Off With Their HUDs!: Rethinking the Heads-Up Display in Console Game Design*. Feb. 2006. URL: [http://www.gamasutra.com/view/feature/2538/off\\_with\\_their\\_huds\\_rethinking\\_.php](http://www.gamasutra.com/view/feature/2538/off_with_their_huds_rethinking_.php) (besucht am 04.08.2011).
- [83] Xemu. *GDC 08: Crowds in Assassin's Creed*. Feb. 2008. URL: [http://xemu.blogharbor.com/blog/\\\_archives/2008/2/22/3540263.html](http://xemu.blogharbor.com/blog/\_archives/2008/2/22/3540263.html) (besucht am 16.03.2011).

# Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —