

Entwicklung eines Testszenarios zur Analyse weiblicher Präferenzen bei Computerspielen

Diplomarbeit

zur Erlangung des Grades einer Diplom-Informatikerin
im Studiengang Computervisualistik

vorgelegt von
Britta Hennecken

Erstgutachter: Prof. Stefan Müller
AG Computergraphik, Universität Koblenz
Zweitgutachter: Dipl.-Inform.(FH) Stefan Zingel
LMR | Laboratory for Mixed Realities

Koblenz, im Juli 2007

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

.....
(Ort, Datum)

.....
(Unterschrift)

Danksagung

Im Folgenden möchte ich einigen Personen danken, die mir beim Erstellen dieser Arbeit behilflich waren.

Besonderer Dank gilt Stefan Zingel des LMR, er stand mir stets mit Rat und Tat zur Seite.

Herrn Prof. Dr. Stefan Müller möchte ich für zahlreiche Hilfestellungen und Ratschläge danken.

Darüber hinaus richtet sich mein Dank an alle Probanden, die sich für das Testszenario zur Verfügung gestellt haben.

Inhaltsverzeichnis

1	Einleitung	5
1.1	Motivation	5
1.2	Aufbau und Ziel der Arbeit	6
2	Theoretische Grundlagen	7
2.1	Allgemeine Bezeichnungen	7
2.2	Aktuelle Studien	8
2.3	Frauenbilder in Computerspielen	11
2.3.1	Die Entwicklung der weiblichen Spielfigur	11
2.3.2	Lara Croft	11
2.3.3	Kate Archer	13
2.3.4	April Ryan	14
2.4	Bisherige Erkenntnisse weiblicher Präferenzen bei Computerspielen	15
2.4.1	Erlernen der Spielregeln	16
2.4.2	Konflikt	17
2.4.3	Fehlertoleranz	19
2.4.4	Emotionale Bindung	19
2.4.5	Kognition	20
2.4.6	Avatar	21
2.4.7	Spielmotivation	22
2.4.8	Puzzle	23
2.4.9	Soziale Interaktion	24
3	Konzeption	25
3.1	Konzeption der Testapplikationen	25
3.1.1	Allgemeines	26
3.1.2	Erlernen der Spielregeln	28
3.1.3	Konflikt	29
3.1.4	Fehlertoleranz	30
3.1.5	Emotionale Bindung	31
3.1.6	Kognition	32
3.1.7	Avatar	33
3.1.8	Spielmotivation	34
3.1.9	Puzzle	35
4	Implementierung	38
4.1	Tools und Bibliotheken	38
4.1.1	Ogre 3D	38
4.1.2	CEGUI Framework	39
4.1.3	Der Game State Manager	40
4.2	Kongruente Implementierung beider Applikationen	42

4.2.1	Initialisierung der Anwendung	42
4.2.2	IntroState	44
4.2.3	PlayState	46
4.2.4	EndState	49
4.3	Spezielle Implementierung der <i>Feelou</i> -Anwendung	51
4.3.1	PlayState	51
4.3.2	EndState	63
4.4	Spezielle Implementierung der <i>Rellos</i> -Anwendung	65
4.4.1	PlayState	65
4.4.2	EndState	69
5	Ergebnisse	70
5.1	Ablauf	70
5.2	Konzeption des Fragebogens und Ergebnisse	70
5.3	Auswertung und Schlussfolgerungen	89
5.4	Abschließendes Fazit	94
A	Screenshots	95

Kapitel 1

1 Einleitung

Dieses Kapitel beinhaltet einführende Aspekte vorliegender Arbeit. Die konkrete Motivation zur Thematik sowie allgemeiner Aufbau und zentrale Ziele werden betrachtet.

1.1 Motivation

Die Computerspielbranche hat sich zu einem stetig wachsenden Markt mit steigenden Verkaufszahlen etabliert. Die Gewinnsummen haben die der Filmindustrie längst überschritten. Umso bemerkenswerter ist die Tatsache, dass die weibliche Spielerschaft bislang eine schlecht erschlossene Zielgruppe darstellt. Der Anteil weiblicher Spieler beträgt lediglich 37 Prozent. Es herrscht ein Mangel an Erkenntnissen bezüglich weiblicher Präferenzen bei Computerspielen. Eine Vielzahl aufwendiger Spiele, die auf den Markt gebracht werden, ist für eine junge, männliche Zielgruppe konzipiert. Es gibt zahlreiche Ansätze, die diesen Zustand zu ergründen versuchen. Ein schwindend geringer Frauenanteil in der Spieleentwicklung ist einer der zentralen Gründe, die zur Erklärung herangezogen werden. Lediglich zehn Prozent der Angestellten in der Spieleindustrie sind weiblich. Ein Großteil dieser zehn Prozent ist in Marketing und Werbung tätig. Es ist davon auszugehen, dass der Anteil weiblicher Entwickler, die maßgeblich an der Konzeption des Spielinhalts beteiligt sind, deutlich unter diesen zehn Prozent liegen.

Des Weiteren stellt es für Entwicklerfirmen ein erhöhtes Risiko dar, in eine unerschlossene Zielgruppe zu investieren. Das Kaufverhalten kann schlecht kalkuliert werden. Einige Arbeiten haben sich mit dieser Thematik befasst und erste Ansätze entwickelt, das weibliche Spielverhalten zu ergründen. Meist bieten Studien oder psychologische Theorien das Fundament. Aus diesen bisherigen Untersuchungen gehen zentrale Faktoren in der Konzeption eines Computerspiels hervor, die an weibliche Präferenzen angepasst werden sollten.

1.2 Aufbau und Ziel der Arbeit

Zu Beginn der Arbeit wird bezüglich weiblicher Präferenzen bei Computerspielen recherchiert. Es werden Bücher, Zeitschriften und Internetquellen herangezogen.

Im weiteren Verlauf werden zwei Testapplikationen entwickelt, die in den relevanten Punkten über Vorlieben und Abneigungen der Testpersonen Aufschluss geben sollen. Als Basis zur Konzeption der Testapplikationen werden die in der Recherche gewonnenen Erkenntnisse herangezogen.

Eine Applikation wird entsprechend den Thesen bzgl. weiblicher Spielpräferenzen umgesetzt. Die andere Anwendung orientiert sich nicht an diesen Erkenntnissen und beinhaltet Spielelemente, die für Frauen weniger ansprechend sein sollen. Die Applikationen sollen, mit Ausnahme der relevanten Faktoren, sehr ähnlich sein, um eine mögliche, nicht beabsichtigte Manipulation anderer Faktoren zu vermeiden.

Die Versuchspersonen spielen im Rahmen des abschließenden Testszenarios beide Applikationen. Durch einen geeigneten Fragebogen wird anschließend ermittelt, wie die verschiedenen Konzepte auf die Testpersonen wirken und ob die herangezogenen Thesen bestätigt werden können.

Das Ziel der Arbeit ist demnach, weibliche Vorlieben bei Computerspielen zu untersuchen. Im Einzelnen werden Thesen, die bereits in vorangegangenen Arbeiten aufgestellt wurden, durch ein geeignetes Testszenario in der Praxis untersucht.

2 Theoretische Grundlagen

2.1 Allgemeine Bezeichnungen

Im Bereich der Computerspiele gibt es einige grundlegende Begriffe, die im Folgenden kurz erläutert werden.

Genre Um das breite Angebot an Computerspielen gliedern zu können werden sie bezüglich ihrer Thematik verschiedenen Kategorien zugeordnet. Diese werden als Genre bezeichnet. Gängige Beispiele sind Rollenspiele, Sport- und Strategiespiele sowie Actionspiele. Da die Inhalte heutiger Computerspiele immer komplexer werden, ist es häufig nicht möglich, sie eindeutig einem Genre zuzuordnen.

PC Unter einem PC (Player Character) versteht man die Spielfigur, die den Spieler im Spielgeschehen repräsentiert. Der Nutzer bestreitet das Spielgeschehen in der Gestalt des PC.

NPC In Analogie zum PC wird eine Spielfigur, dessen Rolle der Spieler nicht einnehmen kann, NPC (Non-Player Character) genannt.

Item Unter einem Item versteht man ein Symbol, dem im Spiel eine definierte Bedeutung zukommt. Beispielsweise können Punkte gesichert, spezielle Fähigkeiten bereitgestellt sowie neue Spielwelten erschlossen werden.

2.2 Aktuelle Studien

Um einen Einstieg in die Thematik zu erhalten, werden aktuelle Studien betrachtet. Es werden Erkenntnisse des Spieleherstellers Electronic Arts zum weiblichen Computerspielverhalten herangezogen. Sie wurden 2006 in "Die Spielerin" [1] veröffentlicht.

Demnach betrage der allgemeine Anteil weiblicher Spieler 37 Prozent. Davon favorisieren 23 Prozent Rollenspiele oder Anwendungen, die den Aufbau einer Zivilisation verlangen. Sportspiele werden von 20 Prozent der Spielerinnen herangezogen. Action- und Abenteuerspiele finden ihre Begeisterten in 17 Prozent. Eine deutliche Mehrheit von 67 Prozent scheint kein bevorzugtes Genre zu haben.

Neben favorisierten Spielinhalten erfasst [1] das Alter weiblicher Spieler. Es ist hervorzuheben, dass die Gruppe der 31- bis 39-Jährigen mit 31 Prozent den deutlich größten Anteil einnimmt. Die 14- bis 19-Jährigen sind mit 21 Prozent vergleichsweise gering vertreten. 57 Prozent aller weiblicher Computer- und Videospiele sind über 30.

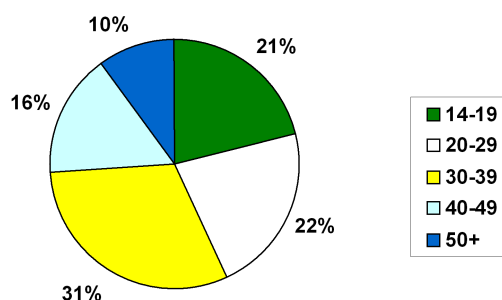


Abbildung 1: Verteilung der Altersgruppen

Im Rahmen der Untersuchung *Spielplatz Deutschland* [2] wird eine Typologie deutscher Computer- und Videospiele vorgenommen. Die Werbeagentur Jung v. Matt und das Spielmagazin GEE und Electronic Arts haben diese Studie in Zusammenarbeit mit Prof. Dr. Sabine Trepte von der Hamburg Media School erstellt. Merkmale des Spielverhaltens werden herausgearbeitet, in Gruppen unterteilt und prozentual erfasst. Demnach [2] lassen sich deutsche Computerspieler in fünf Gruppen klassifizieren: Freizeitspieler (54 Prozent), Gewohnheitsspieler (24 Prozent), Denkspieler (11 Prozent), Fantasiespieler (6 Prozent), Intensivspieler (5 Prozent).

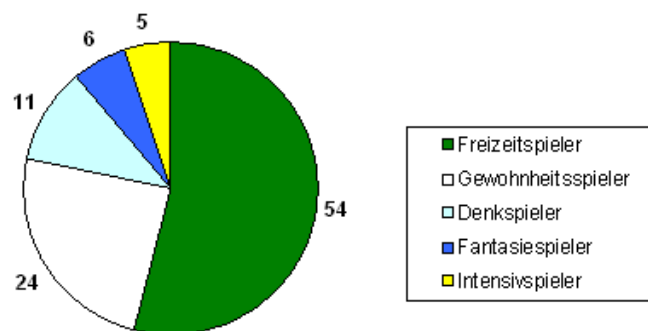


Abbildung 2: Kategorien, Spielplatz Deutschland

Der *Freizeitspieler* stellt mit 54 Prozent die größte Kategorie dar und widerspricht dem allgemeinen Klischee des Computerspielers. Es handelt sich um gelegentliche Spielsequenzen, die in der Freizeit ausgeübt werden. Das Computerspiel dient der abendlichen Entspannung oder gelegentlichen Abwechslung am Tag. Bevorzugte Genre sind Fun- und Sport- sowie Geschicklichkeitsspiele. Die zeitliche Investition ist die geringste aller Gruppen. Die Hälfte der Freizeitspieler ist weiblich. Computerspiele stellen im Alltag des *Gewohnheitsspielers* einen festen Bestandteil dar. Obwohl bei einem Durchschnittsalter von 30 Jahren neben dem Beruf und gegebenenfalls Familie nicht mehr so viel Zeit zur Verfügung steht, wird täglich gespielt. Der Frauenanteil beträgt 25 Prozent.

Für den *Denkspieler* ist das Computerspiel durch geforderte Taktik und Strategie eine geistige Herausforderung. Die Zeitinvestition und sozialen Umstände sind denen des Freizeitspielers sehr ähnlich. 40 Prozent der Denkspieler sind weiblich.

Die Motivation des *Fantasienspielers* ist das Eintauchen in virtuelle Welten. Dies ermöglicht das zeitweise Ausblenden der Realität. Der Frauenanteil beträgt hier 30 Prozent.

Das Bild des *Intensivspielers* entspricht dem allgemeinen Klischee des Computerspielers. Das Durchschnittsalter beträgt 20 Jahre, beliebte Genre sind Action- und Strategiespiele. Die im Volksmund verbreiteten *Ballerspiele* werden diesen Kategorien untergeordnet. Ein geringer Anteil von 20 Prozent wird von weiblichen Spielern eingenommen.

Abbildung 3 veranschaulicht die Verteilung weiblicher Spieler auf die genannten Gruppen.

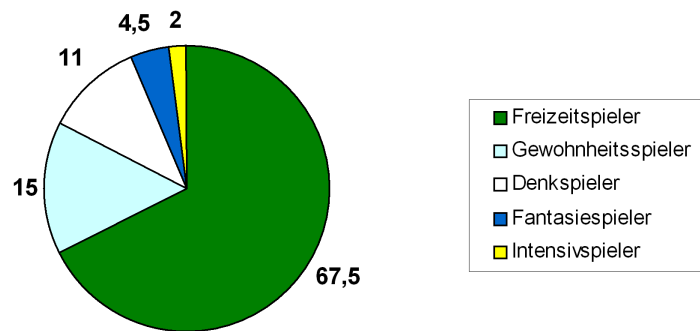


Abbildung 3: Verteilung weiblicher Spieler

Abschließend wird festgehalten, dass die weibliche Spielerschaft in Deutschland 37 Prozent aller Computer- und Videospiele einnimmt. Gut 67 Prozent davon werden der Gruppe der Freizeitspieler zugeordnet. Das Spielen stellt somit für die Mehrheit der Frauen Abwechslung oder Entspannung im Alltag dar. Soziale Kontakte und andere Freizeitaktivitäten werden auf Grund des Computerspielens nicht vernachlässigt oder beeinträchtigt.

Der ebenfalls verhältnismäßig hohe weibliche Anteil der Denkspieler von 40 Prozent lässt sich durch die strukturelle Ähnlichkeit zum Freizeitspieler erklären. Der Unterschied betrifft lediglich die konkreten Spielinhalte. Für eine deutliche Mehrheit von knapp vier Fünftel der Spielerinnen ist demnach das Spielen am Computer lediglich eines von vielen Hobbies, mit denen sie ihre Freizeit gestalten.

Das Extrem des Intensivspielers stellt in der weiblichen Spielerschaft einen sehr geringen Anteil von zwei Prozent dar.

Darüber hinaus ist das durchschnittliche Alter der Spielerinnen relativ hoch. Mehr als die Hälfte ist älter als 30.

Die betrachteten Aspekte geben einen ersten Einblick in das Spielverhalten weiblicher Computer- und Videospiele.

2.3 Frauenbilder in Computerspielen

2.3.1 Die Entwicklung der weiblichen Spielfigur

Das Buch *Sheroes* von Birgit Richard [9] aus dem Jahr 2004 betrachtet die Rolle weiblicher Charaktere in Computerspielen. Die folgenden Ausführungen sind dieser Quelle entnommen.

In älteren Computerspielen ist meist eine deutliche Reduzierung weiblicher Charaktere auf passive oder sexistische Darstellung festzustellen. Auf der einen Seite gibt es die hilflose Opferrolle, beispielsweise die der entführten Prinzessin, die auf die Rettung eines männlichen Helden wartet. Sie hat keinerlei Einfluss auf die Handlung des Spiels und stellt lediglich einen Aspekt der Hintergrundgeschichte dar. Eine solche Rolle ist in *Super Mario Land* vertreten. Der gesamte Spielverlauf dient der Rettung einer Prinzessin, die jedoch nur passiver Teil des Spielgeschehens ist.

Darüber hinaus lassen sich weibliche Figuren in sexistischen Darstellungen finden. Hier geht es darum, die überwiegend männliche Spielerschaft anzusprechen. Ein Beispiel für diese Darstellungsweise weiblicher Charaktere ist das Computerspiel *Biing*. Es handelt sich um eine Wirtschaftssimulation, in der Bewerber aufgrund weiblicher Rundungen eingestellt und Kunden durch strippte Mitarbeiterinnen zufrieden gestellt werden. Dass eine solche Repräsentation weiblicher Charaktere kaum Identifikationsmöglichkeiten für Computerspielerinnen bietet, liegt auf der Hand.

Des Weiteren gibt es die weibliche Heldenfigur, die meist mit vorwiegend männlichen Charakterzügen ausgestattet wird. Sie unterscheidet sich lediglich im Aussehen von ihren männlichen Kollegen.

2.3.2 Lara Croft

Die Entwicklerfirma Eidos bringt 1996 die erste Version des Spiels *Tomb Raider* auf den Markt. Besonders durch die weibliche Hauptfigur unterscheidet sich das Produkt von herkömmlichen Abenteuerspielen. Sie steht ihren männlichen Kollegen in nichts nach. Tollkühn und unbeirrt siegessicher erfüllt sie ihre Mission. Innerhalb kürzester Zeit wird Lara Croft zu einer Werbeikone.

Durch spezielle Sprünge überwindet sie gewaltige Hindernisse. Darüber

hinaus besiegt sie männliche Gegner mit Leichtigkeit. Die erste erfolgreiche weibliche Heldenfigur ist hervorgebracht. Der Entwickler Eidos erhofft sich, die Käufergruppe der weiblichen Spieler zu vergrößern. Frauen sollen sich mit Lara identifizieren können, sich durch die Stärke und List Laras für das Spiel begeistern. Die männliche Spielerschaft wird durch ihre enormen weiblichen Rundungen bei Laune gehalten, die neben Abenteuer und Spannung einen willkommenen Nebenaspekt darstellen. Mit Lara Croft wird eine neue Darstellungsform weiblicher Charaktere realisiert, die der unerschrockenen, tollkühnen Heldin.

Betrachtet man Lara Croft genauer, lässt sich feststellen, dass sie zahlreiche männliche Charakterzüge in sich vereint. Beispiele hierfür sind nach Birgit Richard Mut, Körperkraft und ein starker Willen. Im Umgang mit dem reichhaltigen Waffensortiment ist sie souverän und skrupellos. Bemerkenswert ist auch, dass sie ein überzeugtes Einzelgängertum lebt in den ersten sieben Teilen ist kein tiefgehender, sozialer Kontakt zu anderen Spielfiguren auszumachen. Sie hat keinerlei Bedürfnisse nach einem Ort, an dem sie sich geborgen fühlt. Ihre Kleidung entspricht einer Miniatur-Version männlicher Sportkleidung. Ihre weiblichen Kurven werden dadurch betont. Aufgrund dieses Mangels an weiblichen Attributen und sexistischen Tendenzen, sei es nach [9] schwer sich als weiblicher Spieler mit Lara identifizieren zu können.



Lara Croft (www.laraweb.de)

2.3.3 Kate Archer

Kate Archer ist eine der zahlreichen Nachfolgerinnen von Lara Croft. Im Gegensatz zu ihr verfügt sie nicht über gewöhnliche Waffen, die auch bei männlichen Heldenfiguren Anwendung finden. Vielmehr besitzt sie ein regelrechtes Sortiment an weiblichen Geheimwaffen. Eine Lippenstiftbombe, eine explosive Handtasche und eine Gürtelschnalle mit Seilwinde sind nur einige Beispiele. Ihrem Chef gegenüber gibt sie sich frech und aufmüpfig. Die Darstellung der Figur ist analog zu Lara Croft mit enormen weiblichen Kurven versehen. Ihre Kleidung ist an das weibliche Konzept angepasst. Sie trägt einen körperbetonenden, orange-weißen Lackanzug, der Blicke ins Dekolleté erlaubt. Insgesamt scheint sie sich nach [9] eher als Identifikationsfigur für weibliche Spieler zu eignen als Lara Croft.



Kate Archer (www.eigenheimstrasse.de)

2.3.4 April Ryan

April Ryans Mission in The Longest Journey ist die Wiederherstellung der Familienharmonie. Sie ist in ein festes Netz sozialer Beziehungen von Familie, Freundschaften und Kollegen eingebunden. Ihre Gegner schaltet sie nicht durch Waffengewalt aus, sondern durch List und durchdachte Taktik. Sie ist im Gegensatz zu anderen Protagonisten unsterblich. Um ihre Mission erfüllen zu können, muss sie durch Kommunikation mit anderen Charakteren Informationen sammeln. Diese Kommunikation erfolgt durch sogenannte Menü-Konversation (dem Spieler werden mögliche Äußerungen zu einem bestimmten Zeitpunkt eingeblendet, von denen er eine beliebige wählen kann). Durch ihr zeitweise ungeschicktes Verhalten gewinnt sie meist schnell die Sympathie des Spielers. Ihre Gestik und Körpersprache lässt oftmals auf ihr verhaltenes, ängstliches Gemüt schließen. Gelegentliches Kratzen oder das Richten der Frisur weisen des Weiteren auf zeitweise Verlegenheit hin. April ist auf die Hinweise und Hilfe anderer Charaktere angewiesen. Lara Croft hingegen bestreitet ihre Missionen völlig autonom.



April Ryan (www.kotaku.com)

2.4 Bisherige Erkenntnisse weiblicher Präferenzen bei Computerspielen

Obwohl sich einige Wissenschaftler und Computerspieleentwickler mit der Ergründung weiblicher Präferenzen bei Computerspielern beschäftigen, wird diesen Erkenntnissen bislang kaum Aufmerksamkeit zuteil. Bereits 1998 wird im Rahmen des Buches *From Barbie to Mortal Kombat*[4] die Thematik der weiblichen Vorlieben bei Computerspielen untersucht. Es werden einige Faktoren herausgearbeitet, die an die Bedürfnisse des weiblichen Publikums angepasst werden sollten, um ihr Interesse an Computerspielen zu steigern. Relevante Versuchsreihen und Interviews mit Computerspieleentwicklern bieten eine ansprechende Ergänzung der theoretischen Ausführungen.

In einem späteren Werk veröffentlicht Sheri Graner Ray 2003 mit *Gender Inclusive Game Design*[3] ihre Empfehlungen zur adäquaten Spielkonzeption für die weibliche Zielgruppe. Auch hier werden verschiedene Faktoren, wie beispielsweise Konfliktbewältigung, Spielmotivation und Fehlertoleranz aus geschlechtsspezifischer Sicht betrachtet. Wissenschaftliche Studien, Theorien aus der Psychologie und Fallbeispiele werden herangezogen, um weibliches Spielverhalten abzuleiten und zu ergründen. Nach Ray bringt das Anpassen von Spielkonzepten an die Ansprüche weiblicher Spieler eine allgemeine Verbesserung der Spielqualität mit sich, von der nicht nur das weibliche Publikum profitiert.

Im Folgenden werden die zentralen Aspekte betrachtet, die als Schlüssel zur Erschließung der weiblichen Spielerschaft angesehen werden. Hierzu zählen das Erlernen der Spielregeln, die Konfrontation mit Konflikten, die Konsequenzen von spielerischen Fehlern, die emotionale Bindung, die der Spieler zum Spielgeschehen aufbaut, geschlechterspezifische Kognition, die Beschaffenheit und Einbettung der Spielfigur ins Geschehen, die eigentliche Motivation, die zum Spielen animiert, die Rolle von Puzzles und die soziale Interaktion im Spiel.

2.4.1 Erlernen der Spielregeln

Im Folgenden wird das Erlernen von Spielregeln und möglichen geschlechtsspezifischen Unterschieden betrachtet.

Bevor ein Computerspiel genutzt werden kann, muss ein geeigneter Einstieg ins Spielgeschehen stattfinden. Der Nutzer erlernt das Spiel [3]. Neben den konkreten Spielregeln, müssen Interaktion, Oberfläche und konkretes Konzept klar sein.

Nach Vera Birkenbihl [17] existieren geschlechterspezifische Lernstile. Nach ihren Ausführungen tendieren Jungen zum experimentellen Lernen, das heißt, sie beginnen ohne Anweisungen erhalten zu haben mit ersten Schritten und Versuchen. Mädchen hingegen bevorzugen eine vorherige Einweisung in Thematik und Aufgabenstellung. Im Rahmen des Buches [17] wird ein Versuch nach Inge Schwank, Professorin für Mathematik-Didaktik, betrachtet. Schüler erhalten eine Programmieraufgabe am Computer. Bevor konkrete Aufgabenstellungen geklärt werden, beginnen die meisten Jungen mit ersten Versuchen und verfolgen spontane Ideen. Eine Vielzahl der Mädchen hingegen warte auf Anweisungen, an denen sie sich orientieren können. Die "weibliche Hirnstruktur" [17] sei der männlichen entgegengesetzt angelegt, es werden einweisende Schritte benötigt, um einen erfolgreichen Arbeitsfluss einzuleiten.

Im Rahmen dieser Arbeit ist von Interesse, welche Auswirkungen diese Aspekte auf den Umgang mit Computerspielen haben.

Hat sich ein Konsument für den Kauf eines Spiels entschieden, ist nach [3] der Einstieg ins Spiel ein entscheidende Schritt, der über Erfolg oder Misserfolg des Titels entscheiden könne. Stößt der Käufer auf scheinbar hohe Barrieren, kann das unter Umständen dazu führen, dass das Spiel erst gar nicht gestartet wird. Es wird zur Seite gelegt und nicht mehr aufgegriffen. Für den Entwickler ist dies auf den ersten Blick von sekundärer Bedeutung, da der Kauf schon getätigt wurde und somit das Geld der Firma zu Gute kommt. Beim Spieler jedoch bleibt ein negativer Eindruck, der in den meisten Fällen mit dem Image der Entwicklerfirma im Gedächtnis bleibt.

Für die Konzeption des Spiels bedeutet dies, dass das Erlernen der Spielregeln sollte keine intensive Einarbeitung in ein ausgedehntes

Regelwerk mit sich bringen sollte. Spielkonzepte sind im Idealfall schnell zu erlernen, aber niemals zu beherrschen ("A minute to learn, a lifetime to master."). Da das weibliche Lernverhalten meist das vorherige Studieren und Analysieren von Aufgaben und deren Lösungen vorsieht (*learning before doing*), ist die angemessene Gestaltung des Regelwerks besonders für die weibliche Spielerschaft von großer Bedeutung. Treten hier zu große Hürden beim Einstieg in den Spielverlauf auf, kann dies zur Kapitulation führen. Im Rahmen eines Experiments [12] wird festgehalten, dass Jungen der achten Klasse regelrecht stolz sind, das Regelwerk nicht gelesen zu haben. Mädchen hingegen suchen stets die leitende Hand und fordern genaue Anweisungen. Sie möchten das Risiko nicht eingehen, auf Grund mangelnder Informationen, das Spiel zu verlieren [12]. Wird das Spiel gestartet, stellt die Interaktion eine weitere potentielle Hürde dar. Tastenkombinationen und Navigationskonzepte sollen intuitiv gestaltet sein. Nach Ray [3] lassen sich besonders weibliche Spieler von komplizierten Interaktionsmöglichkeiten abschrecken. Das wahllose Testen beliebiger Tastenkombinationen entspricht nicht dem weiblichen Lernverhalten und mindert oftmals die Spielmotivation. Alle Aufgaben, die dem Spieler gestellt werden, sollen innerhalb der Anwendung, ohne Nachschlagen in externen Dokumenten, lösbar sein.

2.4.2 Konflikt

Zahlreiche Spiele basieren auf einem zentralen Konflikt, der während des Spielverlaufs zu lösen ist. Klassische Beispiele sind Territoriumskämpfe, die das Verteidigen des eigenen Landes oder das Erobern fremder Welten erfordert. Nach Sheri Ray [3] haben Frauen wenig Interesse an solchen direkten Konflikten. Meist verlangen diese Konzepte die Erfüllung einer skrupellosen Mission, die der Spielfigur einen entsprechenden Vorteil einbringt. Verhandlungsmöglichkeiten, oder andere alternative Konfliktlösungen werden nur selten bereitgestellt. Somit ist der Spieler mit einem Konflikt konfrontiert, den er meist durch kämpferischen Einsatz zu lösen hat. Solche direkten Auseinandersetzungen sollen Frauen abschrecken und das Spielvergnügen beeinträchtigen. Des Weiteren habe die weibliche Spielerschaft mehr Interesse daran, eine *beidseitig zufrieden stellende Lösung* ("mutually beneficially solution", [3]) herbeizuführen, das

heißt es soll nicht nur die eigene Spielfigur von dem erspielten Resultat profitieren, sondern auch die NPCs(Non Playing Characters).

Ray [3] erklärt diesen Umstand durch die historisch bedingte, geschlechterspezifische Rollenverteilung. Der Mann war schon in früherer Zeit Jäger. Er muss sich seiner potentiellen Beute im direkten Konflikt stellen. Eine kämpferische Auseinandersetzung ist nicht zu umgehen. Im Vergleich zur Frau muss der Mann ein höheres Risiko eingehen, um seiner gesellschaftlichen Rolle gerecht zu werden. Die Frau hat die Rolle der Sammlerin, direkten Konflikten ist sie in der Regel nicht ausgesetzt. Objektiv betrachtet hatte der Mann zur Erhaltung eines Volkes einen geringeren Wert und kann demnach in Konfliktsituationen ein höheres Risiko eingehen. Das Leben der heranwachsenden Kinder ist von dem der Frau abhängig. Diese Betrachtungsweise erkläre die geschlechterspezifische Konfliktbewältigung.

Um dem weiblichen Konfliktverhalten in einem Computerspiel gerecht zu werden, sollten alternative Konzepte zur Lösung eines Konflikts bereitgestellt werden. Direkte, kämpferische Auseinandersetzungen mit feindlichen Spielfiguren sind zu vermeiden.

Darüber hinaus bevorzugen weibliche Spieler nach [10] das Erschaffen und Aufbauen spielerischer Elemente, während Männer auch gern zerstören und vernichten.

Im Rahmen einer Versuchsreihe von Yasmin Kafai [6] entwickeln Mädchen- und Jungengruppen Konzepte für Computerspiele. Die Spiele der männlichen Testpersonen basieren zum Großteil auf dem Prinzip *Gut gegen Böse*, bei den von Mädchen entwickelten Konzepten war dies nicht der Fall. Abschließend kann festgehalten werden, dass nach aufgeführten Quellen die weibliche Spielerschaft Spielinhalte bevorzuge, die keine direkten Auseinandersetzungen und Zerstörung anderer spielerischer Elemente fordern.

Ergänzend wird in einer Arbeit von Jess Bates [14] festgehalten, dass sich das Beschießen feindlicher Spielfiguren gegen den emotionalen Instinkt weiblicher Spieler richte und somit Unbehagen hervorrufe.

Die Verknüpfung des Spielinhalts an gewalttätige Inhalte soll ebenfalls nach [5] vermieden werden. Das erfolgreiche Absolvieren des Spiels soll gewaltfrei möglich sein.

2.4.3 Fehlertoleranz

In vielen Spielen ist das *GameOver*-Prinzip ein festes Element des Konzepts. In Folge gravierender Spielfehler wird die gesamte Anwendung beendet und muss wieder von Beginn gestartet werden. Es gibt einen zentralen Kritikpunkt. Nicht nur die Aufgabe, an der der Spieler gescheitert ist, muss wiederholt werden, sondern auch alle vorangegangenen, die bereits erfolgreich absolviert wurden. Es handelt sich um ein radikales Konzept, das keinen direkten Bezug zum konkret begangenen Fehler aufweist. Während dieses Spielprinzip nach [3] für die männliche Spielerschaft als selbstverständlich betrachtet wird, ruft es bei vielen Spielerinnen Frust hervor. Führt ein Abbruch des Spiels durch *GameOver* zu Frust, besteht die Gefahr, dass der gesamte Spielspaß beeinträchtigt wird. In Bezug auf dieses Spielprinzip können 1993 im Rahmen einer Studie von Yasmin Kafi [6] interessante Erkenntnisse gewonnen werden. Jungen- und Mädchengruppen werden aufgefordert, ein Konzept für ein Computerspiel zu entwickeln. Als bemerkenswert geht hervor, dass keines der von Mädchen entwickelten Spiel das Bestrafen von Fehlern durch *GameOver* vorsieht. Stattdessen werden alternative Konsequenzen bereitgestellt. Beispiele hierfür sind das Wiederholen der konkreten Aufgabe, an der der Spieler gescheitert ist, das Hinwegnehmen erarbeiteter Werkzeuge oder das Erschweren anderer Faktoren. Grundsätzlich ist auch nach [19] festzuhalten, dass Spielerinnen Wert darauf legen, begangene Fehler durch zusätzlichen Einsatz im Nachhinein korrigieren zu können.

2.4.4 Emotionale Bindung

Um einen Spieler in ein Spielgeschehen einzubinden, werden meist eine ausführliche Geschichte, die Charaktereigenschaften der Spielfigur und soziale Interaktion (s. 2.4.9) herangezogen. Die emotionale Ebene spielt eine zentrale Rolle. Somit legen Frauen nach [3] gesteigerten Wert auf eine emotionale Bindung zum spielerischen Geschehen. Sie möchten sich in die Spielfigur hineinversetzen und mitfühlen. Die konkrete Umsetzung im Rahmen der Testumgebung ist problematisch, die Herstellung einer emotionalen Bindung benötigt Zeit, das Testszenario ist auf eine

kurzweilige Durchführung ausgelegt. Trotzdem wird der Aspekt in Ansätzen verfolgt (3.1.5).

2.4.5 Kognition

Neben inhaltlichen Aspekten spielt die Wahrnehmung des Spielgeschehens eine entscheidende Rolle. Doreen Kimura beschäftigt sich in ihrem Buch *Sex and Cognition* [18] mit geschlechterspezifischen Wahrnehmungsformen. Es werden die gesellschaftliche Entwicklung und Testszenarien herangezogen, um die Thematik zu beleuchten. Sie stellt zahlreiche Faktoren heraus, in denen sich die Wahrnehmung der Geschlechter unterscheidet. In dieser Arbeit werden nur einige betrachtet, die im Rahmen der Thematik von Interesse sind.

Männer haben deutliche Vorteile bei der Treffsicherheit eines bewegten Ziels. Kimura sieht dies in der früheren Jagdtätigkeit des Mannes begründet. Die männliche Wahrnehmung könne das bewegte Objekt schneller und exakter anvisieren, die Geschwindigkeit abschätzen und somit präzisere Treffer hervorbringen. Sie belegt dies mit einem Dart-Testszenario.

Es können keine geschlechtsspezifischen Unterschiede der Reaktionszeit infolge eines visuellen Reizes belegt werden.

Bei feinmotorischen Aufgaben haben die Frauen eindeutige Vorteile. Nach [18] sei dies in der früheren Tätigkeit des Kleider Nähens, Kochens und Kinder hütens begründet.

Des Weiteren stellt Sheri Graner Ray in [3] fest, dass schnelle, visuelle Effekte bei Männern den Jagd- und Spieltrieb auslösen. Die weibliche Spielerschaft hingegen sollte mehr Einfluss auf die zeitliche Abfolge der Geschehnisse haben, nicht nur auf einströmende Ereignisse reagieren müssen. Jedoch dürfe nach [5] die schnelllebige *Action* im Spiel nicht fehlen.

2.4.6 Avatar

Die Darstellung der Avatare scheint ebenfalls geschlechtsspezifische Wahrnehmungen und Empfindungen hervorzurufen. In Abschnitt 2.3 wurden bereits die Darstellung weiblicher Spielfiguren beispielhaft betrachtet. Im Folgenden werden die Auswirkungen der Darstellung auf männliche und weibliche Spieler näher beleuchtet.

Für Männer scheint nach [3] die Darstellung des Avatars, mit dem sie das Spiel bestreiten, eine untergeordnete Rolle spielen. Im Spielverlauf scheint es keinen Unterschied zu machen, ob sie sich durch männlichen oder weiblichen Spielfiguren repräsentieren. Befragte haben geäußert, sie schätzen starke, männliche Heldenfiguren genauso wie wohl proportionierte, weibliche Charaktere, sie seien schön anzuschauen und beeinträchtigen das Spielvergnügen nicht.

Bei der weiblichen Spielerschaft komme der Darstellung eine deutlich größere und kritischere Bedeutung zu. Zum einen lege sie Wert darauf, den Spielverlauf mit einer weiblichen Figur zu bestreiten. Sie möchte sich mit der Figur identifizieren, um einen stärkeren Bezug aufzubauen und sich intensiver in das Spiel einzufinden.

Im Rahmen des Versuchs von Yasmin Kafai [6] kann ebenfalls festgestellt werden, dass die Spiele, die von Mädchen konzipiert wurden, weibliche Spielfiguren bereitstellen. Die der Jungen berücksichtigen neben den meist männlichen Hauptfiguren auch die weiblichen.

Nach [12] wünschen sich Mädchen meist niedliche oder attraktive weibliche Spielfiguren. Hierunter fallen Tiere, niedliche Phantasiewesen oder realistisch gestaltete Mädchen oder Frauen. Jungen hingegen möchten mit einer *coolen* Spielfigur, beispielsweise einem Ninja-Kämpfer oder einem Dinosaurier die Aufgaben bestreiten.

Mädchen und Frauen möchten sich nach [5] mit der Spielfigur identifizieren. Dies ermöglicht eine bessere Bindung zum gesamten Spielgeschehen. Das Spielen mit einer männlichen Figur rufe nach Ray [3] bei der weiblichen Spielerschaft Unbehagen hervor.

Zur Erklärung dieses Umstands zieht Ray die *Pyramide Of Power* heran. Sie stellt die Machtverteilung einer gesellschaftlichen Gruppe dar. Der breite Fuß repräsentiert die größte Bevölkerungsschicht, die den niedrigsten Machtanteil besitzt. Zur Spitze hin steigt die Macht, der Anteil

in der Bevölkerung nimmt dementsprechend ab.

Soll ein Mensch die Aufgaben eines in der Pyramide Tiefergestellten übernehmen, sei dies kein Problem. Er verfüge über die nötigen Fähigkeiten, den Anforderungen gerecht zu werden. Wird er jedoch mit den Aufgaben eines Höhergestellten konfrontiert, sei er überfordert und scheitere an den Anforderungen.

Der Durchschnitt weiblicher Bevölkerung befinde sich in den meisten Gesellschaften in der Pyramide unter dem der männlichen. Wenn eine Spielerin nun die Rolle einer männlichen Spielfigur einnehmen soll, müsse sie einen in der Pyramide höheren Platz einnehmen. Durch diesen Rollentausch erfahre die weibliche Spielerschaft nach Ray eine unbewusste Überforderung, die das Unbehagen beim Spielen männlicher Charaktere erkläre.

2.4.7 Spielmotivation

Ein entscheidender Aspekt beim Computerspielen ist die Motivation, die den Spieler dazu treibt, ein Spiel zu spielen. Ray [3] stellt hier einen entscheidenden geschlechtsspezifischen Unterschied heraus. Für Männer sei das Spielergebnis der zentrale Aspekt. Der endgültige Punktestand entscheide über das allgemeine Spielerlebnis. Es sei wichtig, die eigene Spielleistung in einer Rangliste (*Highscore-Liste*) festzuhalten, um sich mit anderen Spielern messen zu können. Das Spiel soll eindeutige Verlierer und Gewinner hervorbringen.

Im Gegensatz dazu stehe bei der weiblichen Spielerschaft der Vorgang des Spielens an sich im Mittelpunkt. Nach einem Artikel von T. Hartmann und C. Klimmt [11], solle das Spiel an sich Freude bereiten, das endgültige Resultat sei bei der weiblichen Spielerschaft zwar nicht irrelevant, aber von sekundärer Bedeutung. Auf Grund dessen sollten alternative Konzepte zur traditionellen Rangliste entwickelt werden.

Der Sieg stelle für Frauen eher ein privates Erfolgserlebnis dar. Das Brüsten mit ihrer Spielleistung gegenüber anderen scheine nicht in ihren Verhaltensformen verankert [7].

In der Studie [11] wird die Wahrnehmung von Wettkampfsituationen in Computerspielen geschlechtsspezifisch untersucht. Es werden die Bedeutung von Wettstreitmotiv, dem Siegen an sich und der Effizienz des eigenen Spielverhaltens untersucht. Alle drei Faktoren haben für die

männliche Spielerschaft eine deutlich höhere Relevanz, als für die weibliche.

Darüber hinaus wird in [12] festgehalten, dass sich die weibliche Spielerschaft mehr Zeit nehme, in das Spielgeschehen einzutauchen. Hintergrundgeschichte und andere begleitende Elemente spielen eine große Rolle, um sie für das Spiel zu motivieren.

Nach [4] stehe für Frauen die Motivation im Spiel im direkten Zusammenhang zum Geschehen des Spiels. Das Ziel, das sie durch ihre Spielleistung erreichen können, sei von größerem Interesse als das Siegen an sich.

2.4.8 Puzzle

Der Begriff Puzzle beschreibt im Zusammenhang mit Computerspielen ein Genre, das nach Alexey Pajitov, Erfinder des Klassikers *Tetris*, kleine, abstrakte Spielkonstrukte bereitstellt, die den Geist fordern [15].

Der Puzzle-Designer Scott Kim fügt ergänzend hinzu, es handele sich um ein Problem, dessen Lösungsfindung Spaß bereite. Im Gegensatz zu konventionellen Computerspielen bestehe die Aufgabe nicht in der Bekämpfung anderer Spieler oder Spielfiguren [16]. Nach [3] ist das Puzzle kein direkter Wettbewerb, der das weibliche Publikum nach 2.4.2 eher abschrecke. Obwohl emotionale Aspekte in den Schatten der Logik rücken, habe die weibliche Spielerschaft große Freude am Lösen solcher Puzzle. Als Gründe könnte nach [3] die Vorliebe für kurzweilige Spielsequenzen herangezogen werden. Um die emotionale Ebene anzusprechen, müsse der Spieler mehr Zeit investieren.

Die weibliche Spielerschaft lege nach [?] großen Wert darauf, das Puzzle ins Spielgeschehen zu integrieren. Beispielsweise kann das erspielte Resultat in das eigentliche Spielgeschehen einfließen, oder das allgemeine Thema des Puzzles an das des gesamten Spiels angepasst werden.

Darüber hinaus sei nach [8] das Lösen von Rätseln für die weibliche Spielerschaft mit einem größeren Spaßfaktor verbunden, als das schnelle Reagieren auf visuelle Effekte, wie es in vielen konventionelle Computerspielen gefordert wird.

2.4.9 Soziale Interaktion

Unter sozialer Interaktion versteht man im Zusammenhang mit Computerspielen meist die Kommunikation zwischen einzelnen Spielfiguren oder Spielern.

Im Rahmen einer Studie [13] untersuchen Oliver Hartmann, Oliver Knörzer und Kevin O'Brien geschlechtsspezifische Vorlieben bei Computerspielen in Hinblick auf soziale Interaktion. Sie gehen von der These aus, Frauen legen mehr Wert auf soziale Interaktion im Computerspiel als Männer. Die Studie zieht 386 Testpersonen heran und basiert auf einem Fragebogen. Das Resultat widerlegt jedoch die anfängliche These, sowohl Frauen als auch Männer legen großen Wert auf Möglichkeiten zur sozialen Interaktion im Spiel. Tendenziell legen Männer sogar größeren Wert darauf als die weibliche Spielerschaft. Die Bereitstellung sozialer Interaktion zwischen Spielfiguren erfordert viel Zeit und Arbeitsaufwand. Die Entwicklung eines solchen Konzepts überschreitet den zeitlichen und inhaltlichen Rahmen dieser Arbeit, zumal es lediglich einen von vielen Aspekten repräsentiert. Aus diesen Gründen, wird die soziale Interaktion nicht ins Testszenario einfließen.

3 Konzeption

Bevor mit der konkreten Entwicklung geeigneter Testapplikationen begonnen werden kann, müssen grundlegende Elemente des Spiels festgelegt werden.

3.1 Konzeption der Testapplikationen

Die Applikationen sollen im späteren Testszenario dazu dienen, die relevanten Spielelemente in verschiedenen Varianten darzubieten. In Kapitel 2.4 sind die hierzu bedeutenden Elemente aufgeführt. In einer Version werden bisherige Erkenntnisse weiblicher Präferenzen bei Computerspielen berücksichtigt und entsprechend umgesetzt. In der anderen Version werden diese Präferenzen ignoriert oder sogar ins Gegenteil umgekehrt. Dies soll Aufschluss über eventuelle Abneigungen der weiblichen Spielerschaft geben.

Es wird darauf geachtet, die Anwendungen in möglichst vielen Faktoren identisch zu gestalten. Sie sollen sich lediglich in den relevanten Punkten aus Kapitel 3 unterscheiden. Weiterführende Unterschiede könnten die Testergebnisse unbeabsichtigt beeinflussen und verfälschen.

Im Folgenden werden die konkreten Spielelemente und deren Einbindung in die jeweiligen Testapplikationen im Detail betrachtet. Hierzu zählen beispielsweise das Erlernen der Spielregeln, die Darstellung und Art des Konflikts, der dem Spiel zu Grunde liegt. Die Art und Weise wie Fehler des Spielers behandelt werden. Kann eine emotionale Bindung der Spieler hervorgerufen werden, um die allgemeine Spielmotivation zu verstärken. Welche Darstellungsweisen des PC können herangezogen werden, um eventuelle geschlechterspezifische Vorlieben heraus zu filtern.

Die soziale Interaktion, die in Kapitel 2.4 aufgeführt ist, wird nicht in das Testszenario einfließen. Das Generieren sozialer Verhaltensweisen ist eine der größten Herausforderungen heutiger Forschung im Bereich der Künstlichen Intelligenz. Es könnte lediglich eine sehr rudimentäre Darstellung dieses Konzepts realisiert werden, die keine zufrieden stellende Grundlage für die Tests bietet.

3.1.1 Allgemeines

Hier werden allgemeine Aspekte der Konzeption der Applikationen betrachten.

Rellos In der ersten Testapplikation werden weibliche Präferenzen nicht berücksichtigt. Vielmehr werden einige Aspekte gegensätzlich realisiert, so dass Abneigungen herausgefiltert werden können. Das Spiel mit dem Titel *Rellos* handelt vom gleichnamigen Volk, das sich gegen feindliche Eindringlinge wehren muss.

Zu Beginn wird ein Hauptmenü geboten, das die Hintergrundgeschichte zum Spiel, die Anleitung, den Start des Spiels und das Beenden der Anwendung bereitstellt.

Der Spieler schlüpft in die Rolle des furchtlosen Jim, der möglichst viele Feinde vor den Mauern des Dorfes vernichten soll.

Zu Beginn des Spiels sind keine Eindringlinge zu verzeichnen und Jim wird mit drei virtuellen Leben ausgestattet.

Das Ziel besteht darin, möglichst viele Feinde am Eindringen zu hindern. Gelingt es mehr als 50 Feinden ins Dorf einzuwandern, kann Jim den Untergang der Rellos nicht verhindern. Gelingt es ihm jedoch mehr Feinde zu vernichten, geht er als Sieger des Kampfs hervor.

Es existieren drei verschiedene Arten von Feinden. Sie lassen sich durch die Farbgebung ihrer Helme unterscheiden; es gibt rote, grüne und graue. Feinde mit einem roten Helm können mit der rechten Maustaste beschossen werden, grüne mit der linken. Lediglich das exakte Treffen eines Feindes mit der richtigen Maustaste kann die feindliche Einwanderung verhindern. Große, graue Feinde müssen zehnmal beschossen werden, bevor sie besiegt sind. Außerdem kommt es bei einem Kontakt mit ihnen zum Verlust eines der drei virtuellen Leben.

Feelou In der zweiten Testapplikation werden die relevanten Aspekte nach weiblichen Präferenzen umgesetzt.

Das Ziel des Spiels ist die Aufmunterung der Feelou. Das gesamte Volk ist traurig und niedergeschlagen. Der Spieler hat die Möglichkeit, durch die Hilfe einer kleinen, tapferen Feelou namens Jiny, die Stimmung des Volkes zu verbessern.

Während des Spiels kommen Items auf den Spieler zugeflogen, die durch Berührung eingesammelt werden.

Durch einfache Mausbewegung kann der Nutzer *Jiny* beliebig horizontal und vertikal bewegen, um sie in die Position der entgegenkommenden Items zu bringen.

Die sogenannten Happy-Items helfen die Feelou aufzumuntern. Die Sad-Items haben keinen Einfluss auf das Spielgeschehen. Fängt der Spieler jedoch viele Sad-Items, bringt dies das zeitliche Versäumen von Happy-Items mit sich.

Happy Item Diese Items kommen der Aufmunterung der Feelou zu Gute. Sie sind durch ein lächelndes Gesicht gekennzeichnet. Für jeweils drei eingesammelte Happy-Items kann einer von insgesamt 80 Feelou aufgemuntert werden.

Sad Item Diese Items zeichnen sich durch ein trauriges Gesicht aus. Sie haben keinen direkten Einfluss auf das Spielgeschehen.

Sudoku Item Durch das Einfangen eines Sudoku-Items bekommt der Spieler die Aufgabe ein Sudoku-Rätsel zu lösen. Es können maximal drei Feelou aufgemuntert werden.

Memory Item Wurde ein Memory-Item gefangen, wird eine Memory-Anwendung gestartet. Es ist hier ebenfalls möglich durch gute Spielleistung drei Feelou aufzuheitern.

Als Spielresultat wird das Volk der Feelou eingeblendet. Es wird präsentiert, in welchem Maße es durch die vorangegangene Spielleistung aufgeheitert werden kann.

Im Folgenden werden die Konzepte des Spiels erläutert, die keinen direkten Bezug zu einem der relevanten Aspekte aufweisen, für das Entwickeln einer Spielapplikation jedoch essenziell sind. Hierzu zählt das Bereitstellen eines Hauptmenüs, das die Struktur der Applikation organisiert. Es werden Hintergrundgeschichte, Spielanleitung, der Wechsel in den Spielmodus und eine Option zum Beenden der

Applikation bereitgestellt. Der strukturelle Aufbau des Menüs ist in beiden Applikationen analog gestaltet.

Geschichte Die Hintergrundgeschichte gewährleistet den Einstieg ins Spielgeschehen.

Anleitung Die Aufgaben und Regeln des Spiels können über diesen Menüpunkt aufgerufen werden.

Zum Spiel Die Applikation wechselt vom Einstieg in den eigentlichen Spielmodus (zur detaillierten Beschreibung siehe 4.1.4.).

Beenden Die Anwendung wird beendet.

Gelangt der Anwender durch das Hauptmenü in das eigentliche Spielgeschehen, startet das eigentliche Spiel, das Aufschluss über das Spielverhalten der Testperson geben soll. In der Endphase des Spiels wird in beiden Applikationen erneut ein Menü dargeboten.

Zurück zum Spiel Ermöglicht das erneute Betreten des Spiels

Zum Menü Bringt den Anwender zurück zum Hauptmenü.

Beenden Beendet die gesamte Anwendung.

3.1.2 Erlernen der Spielregeln

Wie in Kapitel drei bereits erläutert, legt die weibliche Spielerschaft viel Wert darauf, bereits im Voraus zu wissen, worauf es während des Spielverlaufs ankommt, z.B. welche Ziele erreicht werden sollen, welche Hürden gemeistert werden und vor welchen Gefahren man sich in Acht nehmen sollte. Diese grundlegenden Aufgaben sollen bereits vor dem Spielantritt bekannt sein. Auf Grund dessen stelle die Spielanleitung für die weibliche Spielerschaft einen wichtigen Aspekt zu einem erfolgreichen Start ins Spielgeschehen dar.

Da es keine adäquaten Erkenntnisse zur frauengerechten Gestaltung von Spielanleitungen gibt, wird in diesem Zusammenhang lediglich geprüft, wie intensiv sich die Testpersonen mit der dargebotenen Spielanleitung befassen.

Nach den in Kapitel drei aufgeführten Thesen wird davon ausgegangen, dass sich die weiblichen Testpersonen intensiver mit der Anleitung auseinandersetzen. Die Anleitung beinhaltet Informationen über die Ausgangssituation des Spiels und die Aufgaben, die es zu bewältigen gilt. Im Unterschied zu den anderen Faktoren werden in diesem Fall keine zwei Varianten dargeboten. Das Ziel ist hier lediglich herauszufinden, wie viel Bedeutung eine Spielanleitung hat und ob Frauen in der Praxis mehr Wert darauf legen, im Voraus über die allgemeinen Spielbedingungen informiert zu werden?

3.1.3 Konflikt

Der Konflikt, der einem Spielgeschehen zu Grunde liegt, stellt einen zentralen Aspekt eines Computerspiels dar. Die Erkenntnisse, die in Abschnitt 2.4.2 betrachtet werden, legen folgende Konzeptionierung der Testapplikationen nahe.

Feelou In der Anwendung Feelou, in der weibliche Präferenzen berücksichtigt werden, wird kein konventioneller Konflikt dargeboten, den es in direkten Auseinandersetzungen mit feindlichen Spielfiguren zu lösen gilt.

Der Spieler erhält die Aufgabe, das traurige Volk der Feelou aufzuheitern. Im Rahmen der Hintergrundgeschichte wird berichtet, dass das Volk der Feelou traurig und niedergeschlagen sei. Der Spieler bekommt die Möglichkeit, durch das Einsammeln von Items und das erfolgreiche Lösen von Minispielen (2.4.8), den emotionalen Zustand des Volkes zu verbessern. Es werden keine kämpferischen oder zerstörerischen Elemente integriert. Darüber hinaus wird das Prinzip *Gut gegen Böse* nicht umgesetzt. Es existiert keine präsenste gegnerische Partei, die bekämpft werden muss. Die Gegenseite wird quasi durch die schlechte Stimmung im Dorf repräsentiert und somit nicht personifiziert.

Rellos Der zugrundeliegende Konflikt ist ein konventioneller Kampf um eigenes Territorium. Der Spieler schlüpft in die Rolle des furchtlosen Jim, der vor den Mauern seiner Stadt das Eindringen feindlicher Gegner verhindern muss. Durch gezielte Schüsse werden entgegenkommende Feinde eliminiert. Direkte Auseinandersetzungen stellen das zentrale

Spielprinzip dar. Der Konflikt *Gut gegen Böse* ist klar definiert und fordert die Vernichtung der gegnerischen Partei.

3.1.4 Fehlertoleranz

Die Fehlertoleranz in einem Spiel beschreibt, wie mit Fehlern oder Schwächen des Spielers umgegangen wird. Es kann eine sehr geringe Fehlertoleranz geboten werden, die schon bei kleinen Fehlern, beispielsweise das Verpassen eines Items, gravierende Folgen für den Spielverlauf mit sich bringt. Eine hohe Fehlertoleranz hingegen sieht weniger gravierende Konsequenzen bei Spielfehlern vor. Zum Beispiel können Zusatzaufgaben gestellt werden, die den vorangegangenen Fehler korrigieren können. Nach Kapitel 2.4 bevorzugen Frauen eine eher hohe Fehlertoleranz, da ihre allgemeine Frustrationsschwelle niedriger sei als die der Männer. Das *GameOver*-Prinzip ist eines der elementaren Konzepte einer niedrigen Fehlertoleranz in Computerspielen. Hierbei muss das Spiel, nach meist mehrmaligem Scheitern einer Aufgabe, von Beginn an neu gestartet werden.

Feelou Um eine hohe Fehlertoleranz zu schaffen, ist darauf zu achten, dass es möglichst viele positive Rückmeldungen durch Ereignisse im Spiel gibt. Die Spielleistung entscheidet, in welchem Maß der Spieler eine Verbesserung der allgemeinen Situation hervorbringen kann. Eine direkte Verschlechterung kann auch bei schwacher Spielleistung nicht eintreten. Sammelt die Testperson in einer Phase des Spiels relativ wenig Happy-Items, kann sie diese schlechte Leistung beheben, indem sie im weiteren Spielverlauf viele einfängt und die Minispiele konzentriert löst. Das *GameOver*-Prinzip findet keine Anwendung. Das Spiel stellt keine Elemente bereit, die das virtuelle Leben des PC Jiny negativ beeinflussen können. Das elementare Ziel besteht darin, möglichst viele positive Items (Happy-Items) zu sammeln.

Rellos Die Fehlertoleranz in der Applikation *Rellos* ist wesentlich geringer. Hier wird das *GameOver*-Prinzip realisiert. Kommt es zu dreimaliger Berührung mit einem großen, grauen Feind, verliert die Spielfigur Jim sein virtuelles Leben. Das Spiel ist verloren. Diese Berührung kann somit im Laufe des Spiels nicht korrigiert werden und

bringt die unwiderrufliche Konsequenz, das Ende des Spiels, mit sich. Das ungehinderte Passieren feindlicher Eindringlinge lässt sich ebenfalls nicht korrigieren. Wird eine fest definierte Anzahl von Eindringlingen überschritten, ist das Spiel verloren. In dieser Testapplikation werden Fehler registriert und können im weiteren Spielverlauf nicht korrigiert werden.

3.1.5 Emotionale Bindung

Nach bisherigen Erkenntnissen legt die weibliche Spielerschaft großen Wert auf eine emotionale Bindung zum Spielgeschehen. Frauen möchten sich nach den Ausführungen in 2.4.4 in die Spielfigur hineinversetzen und sich in die Hintergrundgeschichte des Spielgeschehens einfinden. Demnach spielt sowohl der PC als auch die Geschichte, in die das Spiel einbettet ist eine zentrale Rolle, um den Spieler emotional einzubinden. Der zugrundeliegende Konflikt ist ein weiterer Faktor, der unter diesem Aspekt relevant ist. Er entscheidet über die Motivation, mit der der Nutzer das Spiel bestreitet.

Es gibt ein zentrales Problem, das im Zusammenhang mit der Untersuchung emotionaler Bindung auftritt. Um Emotionen aufzubauen, ist in der Regel ein längerer Zeitraum nötig. Im Rahmen dieses Testszenarios werden lediglich zwei kurzweilige Testapplikationen bereitgestellt, die einen intensiven Einblick verwehren. Diese Einschränkung wird in der späteren Auswertung der Testergebnisse berücksichtigt. Um trotzdem aussagekräftige Tendenzen herausstellen zu können, wird im Test eine allgemeine Fragestellung zur emotionalen Bindung bei Computerspielen aufgeführt.

Feelou Im Rahmen der Testapplikation *Feelou* wird versucht, durch die Hintergrundgeschichte und einer kurzen Charakterisierung des Playing Character Jiny, die emotionale Ebene der Testperson anzusprechen. Durch die Traurigkeit und negative Stimmung der *Feelou* wird direkt zu Beginn ein emotionaler Bezug zum Geschehen aufgebaut. Die Testperson bekommt die Möglichkeit, im Laufe des Spielgeschehens, das Volk der *Feelou* aufzumuntern und somit Verantwortung für Ihre Entwicklung zu übernehmen. Das Spielresultat in *Feelou* wird nicht im konventionellen Sinn mit Hilfe einer konkret erreichten Punktzahl präsentiert. Der Spieler

bekommt das Volk der Feelou mit ihren jeweiligen Gemütszuständen dargeboten. Auch dieser Aspekt soll die Bindung zwischen Testperson und Spielgeschehen festigen.

Rellos In *Rellos* werden emotionale Aspekte außer Acht gelassen.

3.1.6 Kognition

Ausführungen in Abschnitt 2.4.5 betrachten allgemeine Unterschiede der Wahrnehmung. Daraus resultierende Thesen zu weiblichen Präferenzen bei Computerspielen sind schwer aufzustellen. Die Untersuchungen zeigen, dass es spezifische Vorteile gibt, die jeweilige Freude an diesen Tätigkeiten wird jedoch nicht untersucht. Demnach ist es möglich, dass Frauen beim Anvisieren bewegter Gegenstände Schwächen aufweisen, aber trotzdem große Freude daran haben. Folgende Konzeption der Testapplikationen sollen diese Fragen im späteren Testszenario klären.

Feelou In der Anwendung Feelou werden verschiedene Spielsequenzen bereitgestellt. Das Einsammeln der Items ist schnell und erfordert Reaktionsvermögen und Schnelligkeit. Die Minispiele Memory und Sudoku bieten eine ruhige Abwechslung. Das Lösen von Rätseln ist zeitunabhängig und stellt eine Art Ruhepause für den Spieler dar. Der Spieler kann das Spielgeschehen steuern, selbst entscheiden, wie und wann er interagiert. Es werden schnelle, aktionsreiche Spielelemente mit langsamen kombiniert.

Rellos Der Spieler wird mit einem schnellen Spielgeschehen konfrontiert, auf das er nur durch schnelle Interaktion Einfluss nehmen kann. Kann der Einstieg nicht unmittelbar erfolgen, ist das Spiel schnell verloren, da zu viele Eindringlinge ungehindert passieren konnten. Darüber hinaus kommt es durch das Abschießen bewegter Objekte der männlichen Kognition entgegen.

3.1.7 Avatar

Der Playing Character stellt in einem Computerspiel die direkte Schnittstelle zwischen Spiel und Spieler dar. In beiden Testapplikationen schlüpft der Spieler in die Rolle des Playing Charakter und löst in seinem Namen die vorgegebenen Aufgaben. Die jeweiligen Charaktereigenschaften spiegeln sich in der Art der gestellten Aufgaben wider. Der furchtlose *Jim* soll mutig sein Land verteidigen, die lebensbejahende *Jiny* ihrem Volk zu neuer Lebensfreude verhelfen. Beide besitzen keinen Körper, sie werden lediglich durch den Kopf repräsentiert.

Feelou Entsprechend weiblicher Präferenzen handelt es sich in *Feelou* um eine weibliche Hauptfigur. Der PC Jiny ist eine lebensfrohe, tapfere Feelou, die die Hoffnung auf eine bessere Stimmung im Volk nicht aufgeben möchte. Bisherige Erkenntnisse heben hervor, dass sowohl attraktive, realistische, als auch verniedlichte PCs beim weiblichen Publikum auf Zustimmung stoßen. Da die Entwicklung einer realistischen Spielfigur im Rahmen dieser Arbeit den zeitlichen Aufwand überschreitet, wird auf die niedliche Variante zurückgegriffen. Jiny strahlt eine positive, gut gelaunte Energie aus, die sich auf den Spieler übertragen soll. Ihre Aufgabe im Spielgeschehen beschränkt sich auf das Einsammeln der Items; Waffen oder andere Werkzeuge kommen nicht zum Einsatz. Sie verkörpert das *nette, junge Mädchen von nebenan*. Sie soll den Spieler durch ihre positive Energie in ihren Bann ziehen. Während des gesamten Spiels vollzieht sie keine Handlung, die anderen Spielfiguren Schaden zufügt.



Abbildung 4: Feelou Jiny

Rellos Der männliche PC names *Jim* verkörpert Tapferkeit, Siegeswillen und Furchtlosigkeit. Er hat keine Angst, sich dem feindlichen Volk entgegenzusetzen. Nach bisherigen Erkenntnissen (2.4.6) sollen diese eher männlichen Attribute eines PC, bei weiblichen Spielern Unbehagen hervorrufen. Die Möglichkeit der Identifikation mit dem PC wird gemindert und das allgemeine Spielvergnügen beeinträchtigt. Durch die Augenbinde, die Jim um den Kopf gebunden trägt, wird seine Kampfbereitschaft im Erscheinungsbild verankert.



Abbildung 5: Rello Jim

3.1.8 Spielmotivation

Im Folgenden soll die allgemeine Motivation betrachtet werden, die den Anwender zum Spielen animiert. Nach Kapitel drei sei dies geschlechterspezifisch.

Feelou Die im Vordergrund stehende Motivation weiblicher Spieler sei nach bisherigen Erkenntnissen der Spaß am Spiel allgemein. Das Spielen als Tätigkeit soll Freude bereiten und eine willkommene Abwechslung vom Alltag bieten. Der Konkurrenzgedanke stehe hierbei im Hintergrund. Um diese Art der Motivation in der Testapplikation für weibliche Präferenzen zu unterstützen, gibt es in *Feelou* kein Sieger-Verlierer-Prinzip. Der Spieler sammelt im Laufe der Anwendung Punkte, die ihm im endgültigen Resultat in Form von aufgemunterten Feelou präsentiert werden. Die Testperson kann direkt erkennen, in wie weit das Volk der Feelou zu besserer Stimmung verholfen wurde. Der Spieler soll nicht durch die Jagd auf einen hohen Punktestand motiviert werden. Vielmehr

wird er durch die Hintergrundgeschichte dazu animiert, sich für die Feelou einzusetzen und das Spiel nach bestem Vermögen zu bestreiten. Der Spieler kann durch seine Spielleistung den Zustand der Feelou lediglich verbessern, eine negative Manipulation ist nicht möglich. Der Konkurrenzgedanke steht außen vor.

Rellos Die Spielmotivation in *Rellos* entspricht dem konventionellen Muster. Der Spieler wird mit einem Territoriumskampf konfrontiert und schlüpft in die Rolle des tapferen Helden *Jim*. Das Spiel basiert auf einem klassischen Sieger-Verlierer-Prinzip. Wird das Volk gerettet, indem möglichst viele feindliche Eindringlinge vernichtet werden, kann sich der Spieler als Sieger feiern lassen. Gelingt es jedoch nicht, und es können zu viele Eindringlinge ins Dorf einwandern, ist das Volk der *Rellos* verloren. Das Spielresultat wird durch konkrete Werte präsentiert.

3.1.9 Puzzle

Puzzle bieten die Möglichkeit, einen kurzen Ausstieg des eigentlichen Spielgeschehens bereitzustellen. Im Folgenden wird die konkrete Umsetzung in den Testapplikationen erarbeitet.

Feelou In die Testapplikation *Feelou* werden zwei Minispiele eingebunden. Sie werden durch das Einsammeln des Sudoku-, beziehungsweise Memory-Items aktiviert. Es werden Sudoku und Memory Anwendungen bereitgestellt, die vom Spieler durch Mausinteraktion zu lösen sind. Durch gute Leistung können Feelou aufgemuntert werden. Die Anzahl der nötigen Versuche auf dem Weg zur richtigen Lösung entscheidet über das Maß der Aufmunterung. Es können maximal drei Feelou erheitert werden.

Sudoku Es handelt sich nicht um die übliche Zahlenvariante. Als Motive dienen vier verschiedene Gemütszustände eines Feelou: glücklich, wütend, erstaunt und traurig. In jedem Feld, als auch in jeder Zeile und Spalte müssen alle vier Motive vertreten sein. Aus dieser Bedingung ergibt sich, dass in keinem Feld, in keiner Spalte oder Zeile ein Motiv doppelt erscheint. Durch die geringe Größe von vier mal vier Feldern sind die Rätsel schnell lösbar und stellen in der Regel kein großes Hindernis

dar. Am rechten Bildschirmrand befindet sich eine Selektionsspalte, in der sich die einzelnen Motive auswählen lassen. Anschließend wird das selektierte Motiv durch Mausklick in einem entsprechenden Feld platziert. Falsch gesetzte Motive können im Nachhinein durch ein anderes Motiv ersetzt werden.

Wird das Rätsel fehlerfrei gelöst hat der Spieler drei Feelou zu guter Stimmung verholten. Ist dies nicht der Fall, bekommt er eine Fehlermeldung eingeblendet, die ihn darauf aufmerksam macht, dass mindestens ein Motiv fehlerhaft gesetzt wurde. Der Spieler hat nun die Möglichkeit, ohne zeitlichen Druck den Fehler zu korrigieren. Jedoch können auf Grund des begangenen Fehlers lediglich zwei Feelou aufgemuntert werden.

Zu Gunsten der Fehlertoleranz wird zusätzlich eine Option *Zurücksetzen* angeboten. Hat der Spieler beim Lösen der Anwendung einen falschen Ansatz verfolgt, wird durch Betätigung dieses Buttons das Sudoku auf den Anfang zurückgesetzt. Da das erste Rätsel nicht auf Anhieb gelöst werden konnte, kann lediglich ein Feelou glücklich werden.

Um zum eigentlichen Spielgeschehen zurückzukehren, hat der Spieler alternativ die Möglichkeit, durch den Button *Zurück zum Spiel* das Sudoku zu verlassen. Dann können keine Feelou aufgemuntert werden. Zur besseren Anschauung wird in Abbildung 28 ein Spiel dargestellt.

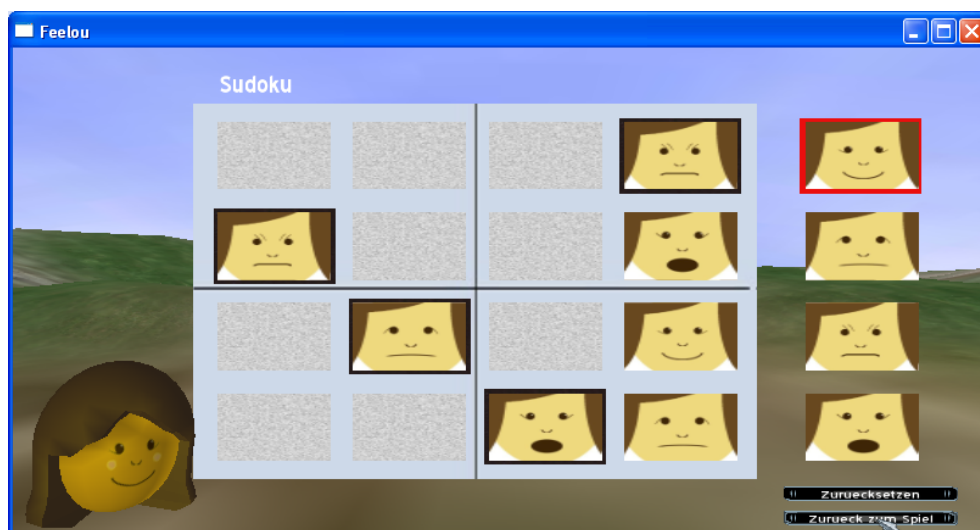


Abbildung 6: Sudoku

Memory Die zweite Variante des Minispiels ist ein konventionelles Memory. Hier werden 16 verdeckte Spielkarten in einem Raster dargestellt. Das Ziel besteht darin, alle Motivpaare ausfindig zu machen. Der Spieler deckt die einzelnen Karten durch Mausinteraktion auf. Sind zu einem Zeitpunkt bereits zwei Karten sichtbar, werden diese, im Fall von Ungleichheit, automatisch wieder verdeckt, sobald der Spieler eine weitere Karte aufdeckt. Handelt es sich jedoch bei den aufgedeckten Karten um ein Motivpaar, bleiben diese aufgedeckt. Das heißt es sind zu jeder Zeit maximal zwei aktive Karten aufgedeckt sowie alle bereits detektierten Paare. Auch hier werden verschiedene Gemütszustände der Feelou als Motive herangezogen. Möchte der Spieler das Memory verlassen, um zum eigentlichen Spielgeschehen zurückzukehren, kann der Button *Zurück zum Spiel* betätigt werden.

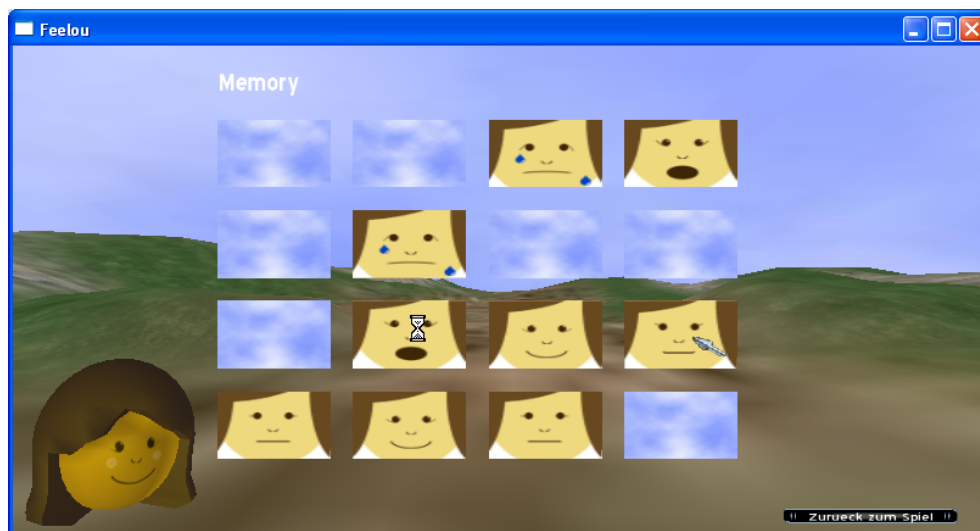


Abbildung 7: Memory

Rellos Im Spiel *Rellos* werden entgegen bisheriger Erkenntnisse weiblicher Präferenzen keine Minispiele integriert.

4 Implementierung

Das vierte Kapitel der Arbeit befasst sich mit programmiertechnischen Aspekten. Es werden zunächst verwendete Tools und Bibliotheken aufgeführt, die zur Realisierung herangezogen werden. Anschließend findet eine Beschreibung der konkreten Implementierung statt.

4.1 Tools und Bibliotheken

Zur Programmierung graphischer Anwendungen ist es sinnvoll, auf bestehende Tools und Bibliotheken zurückzugreifen. Sie stellen elementare Prozesse und Konzepte bereit und ermöglichen somit die Konzentration auf wesentliche Aspekte einer Anwendung. Im Folgenden werden diese Hilfsmittel, die zur Entwicklung der Testapplikationen herangezogen werden, betrachtet.

4.1.1 Ogre 3D

Zur Programmierung der Testapplikationen, die im Rahmen dieser Arbeit entwickelt werden, wird die freie, objekt-orientierte Graphics Engine *Ogre3D* [20] verwendet. Unter einer Graphics Engine versteht man eine spezielle API, die grundlegende Routinen zur graphischen Programmierung bereitstellt. Einige Beispiele des Funktionsumfangs sind das Laden von dreidimensionalen Modellen, das Generieren von Szenerien, die Beleuchtung virtueller Objekte oder das schnelle Transformieren in der Szene. *Ogre3D* ist C++-basiert und ermöglicht einen weitgehend intuitiven Einstieg in die Programmierung dreidimensionaler Welten. Das Framework arbeitet auf den gängigen Systemen OpenGL und Direct3D. Die Entwickler betonen, dass *Ogre3D* keine spezielle API zur Entwicklung von Spielen sei. Jedoch wurden bereits mehrere kommerzielle Computerspiele, beispielsweise *Ankh* [22], mit *Ogre3D* entwickelt.

Es handelt sich um eine freie Engine, die auch zu kommerziellen Zwecken beliebig erweitert und verändert werden kann. Die große Gemeinschaft, die sich um *Ogre3D* gebildet hat, stellt zahlreiche Features zur Verfügung, siehe beispielsweise den *GameStateManager* in 4.1.4. Durch diese individuelle Weiterentwicklung wächst der Funktionsumfang stetig und

bietet zahlreichen Nutzern eine stabile, effiziente Graphics Engine. Obwohl es sich nicht um eine Games Engine handelt, habe ich mich zur Entwicklung der Testapplikationen für Ogre3D entschieden. Ogre 3D deckt den nötigen Funktionsumfang, der zur Entwicklung des Testszenarios benötigt wird, ab. Bei Problemen, die während der Implementierungsphase auftreten, stehen andere Nutzer in zahlreichen Foren mit Rat und Tat zur Seite. Diese Aspekte haben mich dazu bewegt, Ogre3D zur Programmierung meiner Arbeit zu nutzen. Im Folgenden werden zum besseren Verständnis zentrale Objekte einer Ogre3D-Anwendung kurz erläutert [21].

Root-Objekt

Alle Objekte, die in eine Anwendung geladen werden, beispielsweise dreidimensionale Modelle oder die Kamera, werden dem Root-Objekt untergeordnet. Sie stellt demnach das Herzstück einer Applikation dar.

Scene Manager

Da es sich bei Ogre3D um ein szenegraphenbasiertes System handelt, wird eine Instanz benötigt, die das Konstrukt verwaltet und organisiert. Hierzu wird ein Objekt des Scene Managers herangezogen. Es ermöglicht unter anderem das Einfügen von beweglichen Objekten in den Szenegraphen, das Setzen von Weltgeometrien sowie die Bereitstellung von Beleuchtungsobjekten.

Framelistener

Mit Hilfe des Framelisteners hat der Entwickler die Möglichkeit, durch die Methoden `frameStarted()` und `frameEnded()` Anweisungen vor bzw. nach einem Frame auszuführen. Bevor das Programm erneut die Anweisungen innerhalb der Rendschleife berechnet, wird `frameStarted()` aufgerufen. Vor der Beendigung kommt `frameEnded()` zum Zuge.

Somit können stetige Anweisungen in die Ogre3D-Rendschleife integriert werden.

4.1.2 CEGUI Framework

Das CEGUI (Crazy Eddie's GUI) System [23] ist eine C++-basierte Bibliothek, die das effiziente Einbinden graphischer Nutzeroberflächen in Graphik APIs ermöglicht. Es existiert eine spezielle Schnittstelle, die die

Generierung graphischer Nutzeroberflächen in Ogre3D-Anwendungen bereitstellt. Das CEGUI Framework ist in der standardisierten Ogre3D-Version enthalten und wird bei der Kompellierung mit eingebunden. Zur Verwendung des Frameworks ist somit keine nachträgliche Installation nötig.

4.1.3 Der Game State Manager

Zu Beginn der Programmierung einer Spielapplikation muss eine grundlegende Struktur entstehen. Ein Computerspiel besteht aus mehreren Phasen, die den strukturellen Rahmen bilden. Zu Beginn wird meist durch ein Menü der Einstieg ins Spiel ermöglicht. Geschichte, Spielanleitung und andere Optionen werden bereitgestellt.

Die zweite Phase stellt den Spielverlauf an sich dar, meist wird sie durch einen Eintrag im Hauptmenü erreicht. Ist das Spiel beendet, wird die finale Phase, die Präsentation des Spielresultats, dargeboten. *Managing Game States in C++* [25] bietet einen Überblick über ein Framework, das diesen strukturellen Aufbau organisiert. In den einzelnen Phasen der Applikation wird jeweils die entsprechende Szene geladen und die Interaktion individuell bereitgestellt.

Beispielsweise ist die Interaktion in der Einstiegsphase anders ausgelegt als die in der eigentlichen Phase des Spiels. Im Einstieg soll die Interaktion an einige Menüpunkte gekoppelt sein (z.B. Geschichte, Anleitung), in der Spielphase steht die Navigation innerhalb der virtuellen Welt im Vordergrund. Somit unterscheiden sich die strukturellen Abschnitte grundlegend, was eine spezielle Struktur zur Organisation der einzelnen Phasen verlangt.

In Anlehnung an das Framework [25] wurde eine spezielle Struktur zur Implementierung mit Ogre3D entwickelt [24]. Diese stellt drei Game States bereit.

Im Folgenden werden die die Klassen des Frameworks betrachtet.

GameState Als `GameState` werden die einzelnen Phasen, die das Spiel strukturieren, bezeichnet. Die abstrakte Klasse bietet das Grundgerüst einer solchen Phase. Es werden Routinen zur Initialisierung und Interaktion bereitgestellt. Des Weiteren wird das

Wechseln in einen anderen `GameState` ermöglicht. Da zu jeder Zeit nur eine Instanz pro State benötigt wird, geschieht die Initialisierung nach dem Singleton-Prinzip. Beim Singleton-Prinzip wird der Konstruktor innerhalb einer Membermethode aufgerufen. Innerhalb dieser Methode wird geprüft, ob bereits eine Instanz des States existiert. Ist dies der Fall, wird diese zurückgegeben, anderenfalls eine neue erzeugt. Somit wird sichergestellt, dass jederzeit nur eine Instanz existiert. Die abgeleiteten `GameStates` können nach individuellen Ansprüchen innerhalb einer Applikationen definiert und erweitert werden.

GameManager Die Klasse `GameManager` organisiert die einzelnen Phasen. Es werden Methoden zum Umschalten zwischen den verschiedenen States und zur Weiterleitung von Events bereitgestellt. Während des Spielverlaufs werden Events, beispielsweise das Betätigen einer Tastatur- oder Maustaste, an den aktuellen State weitergeleitet und dort individuell behandelt.

Die Methode `changeState()`, die dem Wechsel zwischen zwei `GameStates` dient, sorgt für die Freigabe der Member des aktuellen States und leitet durch den Aufruf der `enter`-Methode des neuen States den Übergang ein. Beide Klassen beinhalten die Methode `frameStarted()`. Diese wird vor jeden Frame im Hintergrund aufgerufen. Demnach können hier Anweisungen integriert werden, die im folgenden Ablauf des Programms permanent in Abhängigkeit zur Framerate bearbeitet werden. In Analogie dazu existiert eine `frameEnded()`-Methode, die nach jedem Frame Anwendung findet.

4.2 Kongruente Implementierung beider Applikationen

In diesem Abschnitt werden die aus programmiertechnischer Sicht analogen Konzepte der Testapplikationen betrachtet. Alle aufgeführten Programmfragmente sind auf wesentliche Aspekte reduziert, es handelt sich nicht um lückenlose Auszüge der Applikationen.

Das in Abschnitt 4.1.3. beschriebene Prinzip des GameStateManagers wird in beiden Spielen identisch angelegt.

4.2.1 Initialisierung der Anwendung

Im Folgenden wird die konkrete Verwendung des GameStateManagers in den Testapplikationen betrachtet. Zu Beginn werden in der `main`-Methode, die als Einstiegspunkt einer Applikation dient, eine Instanz des GameManagerserzeugt und die `start`-Methode darauf angewandt.

```
//create GameManager object
GameManager game;
//call start-method to do necessary settings
game.start();
//enter IntroState to start application
game.changeState(IntroState::getInstance());
```

In der `start`-Methode werden initialisierende Schritte vorgenommen.

```
void GameManager::start(void)
{
    // initialize root-object
    mRoot = new Root();
    // initialize LogFile
    log = LogManager::getSingleton().createLog("Log");
    if (!configure()) return;
    // load given resource-files
    setupResources();
    //for event-handling
    mRoot->addFrameListener(this);
    //for input-handling
    mInputManager = new InputManager(mWindow);
    //combine input and event-handling
    mInputManager->getEventProcessor()->addKeyListener(this);
}
```

```

        mInputManager->getEventProcessor()->addMouseListener(this);
        mInputManager->getEventProcessor()->addMouseMotionListener(this);
    }

```

Das `Root`-Objekt fungiert als Vater aller Knoten, die im Laufe der Anwendung generiert werden. Alle grundlegenden Konstrukte werden ihm untergeordnet, beispielsweise das `Renderfenster`, der `SceneManager` und `FrameListener`. Das `LogFile` dient der Ausgabe beliebiger Werte, die während eines Programmablaufs von Interesse sind. Der Ausdruck `if (!configure())` stellt sicher, dass alle Anweisungen innerhalb der `configure()`-Methode korrekt durchlaufen wurden. Hier wird das `Renderfenster` initialisiert und an das `Rootobjekt` gebunden. Schlägt eine dieser Anweisungen fehl, kann die `Ogre3D`-Applikation nicht gestartet werden.

Der nächste essentielle Schritt in der `start()`-Methode des `GameManagers` ist das Laden der Ressourcen. Hierzu werden in einer Konfigurationsdatei, die im Verzeichnis der Anwendung liegt, verschiedene Pfade hinterlegt. Sie verweisen unter anderem auf Texturen, Skriptdateien zur Materialdefinition, Elemente zur Generierung graphischer Nutzeroberflächen und Dateien zur Beschreibung dreidimensionaler Modelle.

Darüber hinaus wird der `FrameListener`, ein Objekt zur Detektion von Events innerhalb eines aktiven `Renderfensters`, dem `Rootobjekt` übergeben. Der `InputManager`, der die Nutzereingabe, beispielsweise durch Maus oder Tastatur, behandelt, wird initialisiert und mit dem `KeyListener`, dem `MouseListener` und dem `MouseMotionListern` verknüpft. Das heißt, der `InputManager` reagiert im Rahmen der Applikation sowohl auf Tastatureingaben als auch auf das Klicken und Bewegen der Maus.

4.2.2 IntroState

Die konkrete Anwendung startet mit dem Einstieg in den `IntroState`. Die Anweisung

```
game.changeState(IntroState::getInstance());
```

sorgt für den Einstieg in die erste Phase der Applikation. Zur genauen Beschreibung der `changeState()`-Funktion siehe Abschnitt 4.1.3. Die `enter`-Funktion, die der `GameState`-Klasse angehört, initialisiert zahlreiche Instanzen und setzt Membervariablen auf ihren Anfangswert. Das Rootobjekt wird in einen Member geschrieben, der `SceneManager` initialisiert, die Kamera definiert und die graphische Nutzeroberfläche generiert. Der PC, das Hauptmenü und eine Ebene, auf die das Logo des Spiels gemappt wird, werden geladen.

Die Kamera verbleibt in den Standardeinstellungen und wird im Laufe des `IntroState` nicht verändert.

Das Hauptmenü zum Spiel wird mit Hilfe des CEGUI Systems (s. Abschnitt 4.1.2) entwickelt. Hierzu werden ein `GUIRenderer` und `GUISystem` initialisiert.

```
mGUIRenderer = new CEGUIRenderer(mWindow,
mSceneMgr);
mGUISystem = new CEGUISystem(mGUIRenderer);
```

Darauf folgend werden der Mauszeiger visualisiert und die zu verwendende Schrift definiert. Sind diese vorbereitenden Maßnahmen abgeschlossen, kann das konkrete Menü entwickelt werden. Hierzu werden `Button` als eine Art Fenster initialisiert und an der entsprechenden Position im Renderfenster platziert. Das Generieren eines Menübutton wird im Folgenden am Beispiel des *Geschichte*-Eintrags veranschaulicht.

```
//generate <Geschichte>-Button
CEGUI::PushButton* storyButton =
(CEGUI::PushButton*)CEGUI::WindowManager.createWindow "Story";
mEditorGuiSheet->addChildWindow(storyButton);
storyButton->setPosition(CEGUI::Point(0.35f, 0.8f));
storyButton->setSize(CEGUI::Size(0.3f, 0.03f));
```

```
storyButton->setText("Geschichte");
```

Das äußere Erscheinungsbild der Button ist definiert. Das Entscheidende, die Reaktion der Applikation beim Klicken auf einen der Menüeinträge, muss noch behandelt werden. Dies geschieht in der Methode `setupEventHandlers()`. Zum einfachen Zugriff auf die einzelnen Elemente der graphischen Oberfläche wird der `CEGUI::WindowManager` zwischengespeichert. Der Button *Geschichte* wird mit der Funktion `handleStory()` verknüpft, sodass in Folge eines Mausklicks auf den Menüeintrag `handleStory()` aufgerufen wird.

```
//save CEGUI::WindowManager
wmgr = CEGUI::WindowManager::getSingleton();
//bind Story-Button to function handleStory()
wmgr.getWindow((CEGUI::utf8*)"Story")
->subscribeEvent(CEGUI::PushButton::EventClicked,
::Event::Subscriber(&IntroState::handleStory, this));
```

Klickt der Nutzer auf den Menüeintrag *Geschichte*, wird ein Textfeld eingeblendet, das die entsprechende Hintergrundgeschichte des Spielgeschehens beinhaltet. In beiden Testapplikationen findet die Geschichte in einer einzelnen Einblendung Platz. Der Rahmen, der den Text beinhaltet, ist ein Objekt der Klasse `TextAreaOverlayElement`. Der Entwickler hat die Möglichkeit, einen beliebigen Bereich des Renderfenster mit diesem Element zu versehen, um Textbausteine zu visualisieren.

```
textArea->setPosition(mWindow->getWidth()* relation, 0);
textArea->show();
panel->show();
textArea->setCaption("beliebeigen Text eingegeben");
```

Betätigt der Nutzer den Eintrag *Anleitung*, wird der Text angepasst. Die Anleitung besteht aus drei Einblendungen, zwischen denen mit Hilfe spezieller *Weiter-*, *Zurück*-Buttons geblättert werden kann. Diese Schaltflächen sind nur innerhalb der Anleitung aktiviert und sichtbar. Befindet sich die Anwendung auf Seite eins der Anleitung, behandelt die Applikation den Klick auf den *Weiter*-Button durch die Einblendung der zweiten Seite. Befindet sich die Anwendung bereits auf der zweiten Seite,

wird die dritte eingeblendet. Realisiert wird dies durch die Statusvariable `instruction_state`, die die aktuell angeschlagene Seite erfasst. Ebenso ist darauf zu achten, dass sich die Button zum Durchblättern am aktuellen Status orientieren. Das heißt der *Weiter*-Button soll nur dann eingeblendet werden, wenn eine weitere Seite existiert. Andernfalls wird er ausgeblendet und deaktiviert.

Betätigt der Nutzer den Button *Zum Spiel* wird die Funktion `changeState()` aufgerufen und der `PlayState` eingeleitet.

```
game->changeState(PlayState::getInstance());
```

Weitere Anweisungen, die innerhalb des `PlayState` folgen, werden im nächsten Abschnitt näher betrachtet.

Der *Beenden*-Button führt den Abbruch der gesamten Anwendung herbei. Es wird die `exit()`-Funktion des `IntroState` aufgerufen, der alle initialisierten Objekte zerstört, um den Speicherplatz freizugeben.

4.2.3 PlayState

Wählt der Nutzer im `IntroState` den Menüeintrag *Zum Spiel*, wird der `PlayState` gestartet. In Analogie zu den anderen States wird auch hier zu Beginn die `enter()`-Methode aufgerufen. Es folgt eine Reihe an Funktionsaufrufen, die zur Initialisierung des `PlayState` von Nöten sind. Hier werden nur gemeinsame Methoden der beiden Applikationen aufgeführt. Alle anderen werden in den Abschnitten 4.3.1 und 4.4.1 betrachtet.

Das Rootobjekt, der `InputManager`, das `LogFile` und das `Renderfenster` werden jeweils in `Members` zwischengespeichert, um einen effizienten Zugriff innerhalb der Klasse zu gewährleisten. Im nächsten Schritt werden die übrigen `Member`variablen in `initializeMember()` mit dem gewünschten Wert initialisiert.

Kamera und `Viewport` werden definiert und mit entsprechenden Eigenschaften versehen. Die Kamera wird so konfiguriert, dass die Blickrichtung entlang der *x*-Achse der dreidimensionalen Szene verläuft. Somit blickt der Spieler entlang einer Schneise im Terrain, in der das Spielgeschehen stattfindet.

In `createScene()` werden der Playing Charakter und die Kamera in den Szenegraphen integriert. Die exakte Positionierung beider Objekte unterscheidet sich in den Anwendungen, daher wird dies in 4.3.1 und 4.4.1 näher betrachtet.

Die Einblendung des Spielstatus geschieht innerhalb der Funktion `createScoreStatus()`. Hierzu wird ein Textbereich definiert, der den aktuellen Spielstand gut sichtbar am unteren linken Bildschirmrand einblendet. Zunächst werden der `OverlayManager` der Applikation zwischengespeichert und ein Panel, ein Rahmenelement für den Textbereich, kreiert.

Im nächsten Schritt werden zwei Objekte der Klasse `TextAreaOverlayElement` generiert, die im weiteren Verlauf mit konkreten Textbausteinen versehen werden. Ein Textbereich stellt zahlreiche Eigenschaften bereit, die den Vorstellungen des Entwicklers entsprechend gesetzt werden können.

Die erste Instanz `textArea` beinhaltet den Bezeichner des Spielstatus *Happy Feelou* (in der Anwendung *Feelou*) bzw. *Eindringlinge* (in der Anwendung *Rellos*). Der zweite Textbereich, `status`, stellt den konkreten Wert des Spielstatus bereit. Er wird in der unteren, linken Ecke des Bildschirms dargestellt.

Das Terrain, auf dem das Spielgeschehen stattfindet, wird eingebunden.

```
mSceneMgr->setWorldGeometry( "terrain.cfg");
```

Die Anweisung integriert eine Ogre3D-Konfigurationsdatei mit Namen `terrain.cfg`, die Daten zur Darstellung des Terrains beinhaltet. Es wird eine Höhenkarte (`heightmap.raw`) eingelesen, um das Terrain nach individuellen Vorstellungen zu gestalten. Es handelt sich um ein Grauwertbild, das zur Definition von Höhenfeldern herangezogen wird. Der Wert null entspricht einer ebenen Koordinate, der Wert 255 einer maximal erhobenen. Darüber hinaus wird ein Material geladen, das das Einbinden von Alphakarten erlaubt. Somit kann die Terrainoberfläche aus mehreren, übereinander liegenden Texturen zusammengesetzt werden.

```
# heightmap-source  
Heightmap.image=heightmap.raw  
# heightmap size
```



```

Heightmap.raw.size=257
# set of tiles: must be (2^n)+1
PageSize=257
# size of each tile: must be (2^n)+1
TileSize=33
# size of a terrain page, in world units
PageWorldX=6425
PageWorldZ=3000
# maximum height of the terrain
MaxHeight=500
CustomMaterialName=Terrain

```

Die nächste Methode, die innerhalb von `enter()` aufgerufen wird, ist `createSky()`. Zur Simulation des Himmels, wird eine große Halbkugel um die gesamte Szenerie gespannt. Sie wird mit einer Wolkentextur versehen. Die Parameter beschreiben das Material, das auf die Halbkugel projiziert wird und entsprechende Skalierungsfaktoren der zugehörigen Textur.

```
mSceneMgr->setSkyDome( true, "Examples/CloudySky", 3, 2 );
```

Abschließend wird die Szenerie mit Pflanzen versehen. Es wurde darauf geachtet, dass die Modelle sehr klein sind, um die Performanz nicht zu beeinträchtigen.

Da das Terrain verschiedene Höhenabschnitte aufweist, müssen den Modellen individuelle y-Koordinaten zugewiesen werden. Dies wird mit Hilfe einer Schnittpunktberechnung erreicht. Für jeden Baum wird ein Strahl in die Szene verfolgt, er verläuft von oben nach unten durch das Terrain.

Im weiteren Verlauf wird der Schnittpunkt von Strahl und Terrain berechnet. Mit Hilfe dieses Werts kann nun der Baum an die entsprechende y-Position, den ermittelten Schnittpunkt, gesetzt werden.

```
new_tree->setPosition(x, intersection_y,z);
```

Abschließend wird die Funktion `startRendering()` aufgerufen, die die Ogre3D-RenderingLoop startet. In den vorangegangenen Beschreibungen wurden alle gemeinsamen initialisierenden Schritte des `PlayState` betrachtet.

Alle weiteren Ausführungen beziehen sich auf den direkten Spielverlauf, beispielsweise die Reaktion auf Nutzerinteraktion oder zeitliche Abläufe des Spielgeschehens. Diese unterscheiden sich in den Applikationen und werden demnach in den Abschnitten 4.3.1 und 4.4.1 detailliert aufgeführt.

4.2.4 EndState

Im Folgenden werden gemeinsame Elemente des `EndState` betrachtet. Auch hier wird zu Beginn die `enter()`-Methode aufgerufen, in deren Rahmen initialisierende Anweisungen vorgenommen werden.

Die Parameterliste beinhaltet das Spielresultat, das aus dem `PlayState` übergeben wird.

Kamera und Viewport werden initialisiert. Die standardisierte Position und Ausrichtung der Kamera bleiben im `EndState` unverändert, also befindet sie sich im Nullpunkt des dreidimensionalen Koordinatensystems und blickt entlang der negativen z-Achse. Innerhalb der Methode `createCEGUI()` wird ein Menü generiert, das eine erneute Spielsequenz, das Zurücksetzen ins Hauptmenü und das Beenden der Anwendung bereitstellt. Hierzu werden in Analogie zum Hauptmenü im `IntroState` zunächst die Maus aktiviert, Menüeinträge generiert und entsprechend konfiguriert.

Wählt der Nutzer im Menü den Eintrag *Zurück zum Spiel*, wird durch folgende Ereignisbehandlung

```
bool EndState::handlePlayAgain(const CEGUI::EventArgs& e)
{
    game->changeState(PlayState::getInstance());
    return true;
}
```

der `PlayState` aktiviert. Der Button *Zum Hauptmenü* veranlasst das Verlassen des `EndState` und das Eintreten in den `IntroState`.

```
bool EndState::handleMenu(const CEGUI::EventArgs& e)
{
    game->changeState(IntroState::getInstance());
    return true;
}
```

Wird die Option *Beenden* gewählt, folgt das Verlassen der Testapplikation. Alle weiteren Prozesse, beispielsweise die konkrete Darstellung des Spielresultats, werden in den Applikationen individuell vorgenommen. Genauere Beschreibungen sind den Abschnitten 4.3.2 und 4.4.2 zu entnehmen.

4.3 Spezielle Implementierung der *Feelou*-Anwendung

Im vorangegangenen Abschnitt wurden gemeinsame Aspekte der Programmierung beider Testapplikationen betrachtet. Es folgen Elemente, die lediglich in der Anwendung *Feelou* eingebracht werden. Der `IntroState` ist in beiden Anwendungen identisch strukturiert und wird hier nicht näher betrachtet.

4.3.1 PlayState

Initialisierende Schritte

Nachdem die Einstellungen für die Kamera vorgenommen wurden, folgt das Zusammensetzen der Szenerie. Die Methode `createScene()` behandelt das Einfügen des Playing Characters, das Generieren und Positionieren einer Lichtquelle und vorbereitende Schritte zur Einblendung von Zwischenergebnissen. Der Szenegraph wird so konfiguriert, dass die Kamera und der Playing Character *Jiny* einem gemeinsamen Vaterknoten zugeordnet werden. Dies bewirkt, dass zur Navigation im Spielgeschehen lediglich der gemeinsame Vaterknoten transformiert werden muss. Somit verändert sich der Blickwinkel des Betrachters im Einklang mit Jiniess Position.

Im Rahmen der Minispiele Memory und Sudoku werden verschiedene Menüeinträge benötigt, die bereits im Rahmen der `enter()`-Methode vorbereitet werden. Da das Initialisieren eines Menüs bereits in 4.2.2 näher betrachtet wurde, werden an dieser Stelle keine konkreten Codefragmente aufgeführt.

Der letzte initialisierende Schritt im Rahmen der `enter()`-Methoden ist das Generieren der Items. Dies wird innerhalb der Methode

```
void PlayState::createItems(int middle_y, int middle_z)
```

implementiert. Insgesamt werden im Szenegraphen 18 Items gesetzt: jeweils ein Sudoku- und Memory-Item, jeweils 8 Happy- und Sad-Items. Die Platzierung in der Szene geschieht nach einem Muster, das sich an den Parametern `middle_y` und `middle_z` orientiert. *//pattern of*

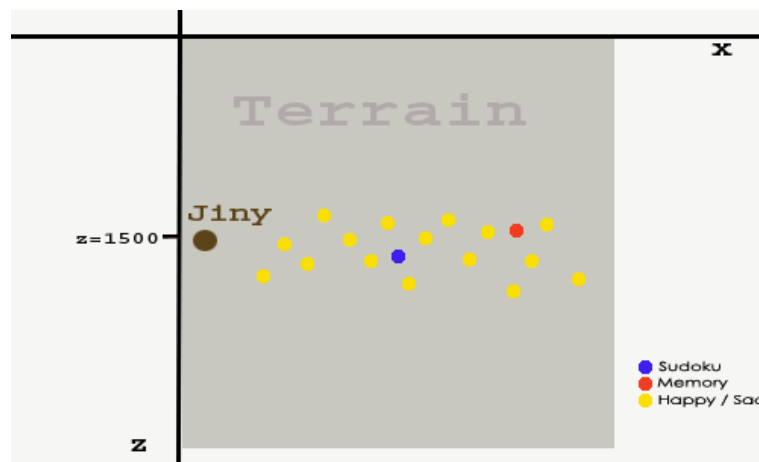
```
y-coordinates  
int y_pattern[]=middle_y+5, middle_y-10, middle_y,
```

```

middle_y+5, middle_y-5, middle_y, middle_y+10;
//pattern of z-coordinates
int z_pattern[]=middle_z+10, middle_z-20, middle_z-10,
middle_z+15, middle_z, middle_z-15, middle_z-10;

```

Die x-Werte der einzelnen Elemente werden in konstanten Abständen festgelegt. Abbildung 4.3.1 stellt die Anordnung in der Szene schemenhaft dar.



Alle Items werden dem gemeinsamen Vaterknoten `item_all` untergeordnet. Sowohl die Happy- und Sad-Items, als auch Sudoku- und Memory-Items werden wiederum separaten Vaterknoten angehängt. Der untenstehende Code veranschaulicht die Struktur im Szenegraphen.

```

item_all= mSceneMgr->getRoot()->createChild("ItemAll");
//special items (sudoku, memory)
item_special = item_all->createChild("ItemSpecial");
sudoku= item_special->createChild("Sudoku", Vector3(sx, sy, sz));
memory= item_special->createChild("Memory", Vector3(mx, my, mz));
//father of happy/sadItems
item_norm= item_all->createChild("NormItem");

```

Spielverlauf

Wie in Abschnitt 4.1.3 beschrieben, sind die Methoden `frameStarted()` und `frameEnded()` für den Spielverlauf der Testapplikationen von zentraler Bedeutung. Die Methoden werden zu Beginn bzw. im Anschluss an einen gerenderten Frame aufgerufen. Zeitabhängige Anweisung, wie das stetige Translieren von Objekten im dreidimensionalen Raum oder das

Prüfen von Schnittpunkten, werden in diesen Methoden vorgenommen. Das Einfangen der Items stellt in der Testapplikation *Feelou* das zentrale Spielelement dar. Um dies bereitzustellen, werden die Items pro Frame um einen fest definierten Vektor transliert.

```
item_all->translate(item_speed, 0, 0);
```

Die Testperson hat nach Abschnitt 3.1.1 die Aufgabe, die bewegten Items durch einfache Mausnavigation einzufangen. Hierzu wird in jedem Frame getestet, ob eine Kollision zwischen der Spielfigur Jiny und einem der Items vorliegt. Eine Kollision ist in diesem Zusammenhang mit dem erfolgreichen Einsammeln eines Items gleichzusetzen.

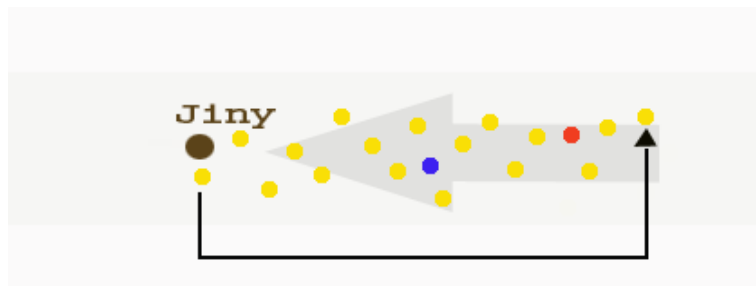
Das gefangene Item wird zwischengespeichert, um im darauf folgenden Frame den Zugriff zu sichern. Nach der erfolgreichen Überprüfung, ob es sich um ein Happy-Item handelt, wird der Spielstatus `happy_feelou` um eine Einheit inkrementiert. Des Weiteren werden dem Item und dem Skydome der Szenerie ein helles, leuchtendes Material zugewiesen, das im darauf folgenden Frame wieder zurückgesetzt wird. Somit entsteht beim Einsammeln ein kurzweiliges Aufleuchten der Szene. Abschließend wird der Spielstatus, der am linken, unteren Bildschirmrand dargestellt wird, mit dem neuen Wert aktualisiert.

```
//caughted item
    //register caught item for next frame
    caught_item= current_item;
    //is caught item an happy-item??
    if (StringUtil::match(current_item->getName(), "Happy*"))
    {
        //increment happy-feelou-counter for game-status
        happy_feelou++;
        //set a bright sky-material (flash)
        mSceneMgr->setSkyDome( true, "caughted_sky", 3, 2 );
        //set a bright item-material
        current_item->setMaterialName("caughted");
        //update status-display
        print_status();
    }
```

Aus Gründen der Effizienz werden, wie bereits beschrieben, lediglich 18 Items in die Szene eingebracht. Um unter diesen Umständen einen kontinuierlichen Fluss an entgegenkommenden Items zu gewährleisten,

wird eine Grenze definiert, an der die Items zurückgesetzt werden. Konkret bedeutet dies, innerhalb der `frameStated()`-Methode wird geprüft, ob sich ein Item auf gleicher Höhe oder bereits hinter der Spielfigur *Jiny* befindet. Ist dies der Fall, wird das entsprechende Item entlang der Schneise, relativ zur x-Achse, zurückgesetzt. Außerdem wird durch eine zufällig generierte z-Koordinate ein vorhersehbares Muster verhindert.

```
//check x-coordniate
if (item_pos.x < pc_pos.x )
{
    //translate, set back relative to x
    current_item->translate(set_back, 0, 0);
    //set random z-position
    current_item->setPosition
    (
        current_item->_getDerivedPosition().x,
        current_item->_getDerivedPosition().y,
        rand_position
    );
}
```



Minispiele

Sammelt der Spieler ein Sudoku- oder Memory-Item, wird ihm nach Abschnitt 3.1.9 ein Rätsel zur Lösung bereitgestellt und der entsprechende Modus gesetzt. `//caught Sudoku-Item`

```
sudoku_active=true;
//caught Memory-Item
memory_active=true;
```

Im darauffolgenden Frame wird auf Grund dieses Flags die entsprechende Funktion zur Erzeugung des Spiels aufgerufen.

```

if (memory_active)
    createMemory();
else if (sudoku_active)
    createSudoku();

```

Darüber hinaus wird der gewöhnliche Spielverlauf, das Translieren der Items oder das Navigieren der Spielfigur, unterbrochen. Der Mauszeiger wird eingeblendet. Die Funktion `mousePressed()` stellt für beide Spiele entsprechende Anweisungen bereit, die zur Interaktion benötigt werden. Da die Spielkarten beziehungsweise Spielfelder durch Ebenen im dreidimensionalen Raum repräsentiert werden, kann das Selektieren einzelner Elemente durch eine Strahlenverfolgung geschehen. Genauer heißt dies, dass infolge eines Mausklicks in einem der beiden Spiele ein Strahl von der Kamera durch die Spitze des Mauszeigers in die Szene verfolgt wird. Der Strahl wird in Bezug auf potentielle Schnittpunkte mit Objekten in der Szene überprüft.

```

Ray mouseRay = mCamera->getCameraToViewportRay(mx,my);
RaySceneQuery *memo_query= mSceneMgr->createRayQuery(Ray());
memo_query->setRay(mouseRay);

```

Die Anfrage liefert als Resultat eine Liste der geschnittenen Objekte. Es wird ermittelt, ob sich eine Spielkarte darunter befindet.

Memory

Dieser Abschnitt betrachtet die Funktion `createMemory()`, die zur Generierung einer Memory-Anwendung herangezogen wird. Zu Beginn werden die in Abschnitt 3.1.9 beschriebenen Button zur Rückkehr ins eigentliche Spielgeschehen aktiviert. Alle hier aufgeführten Aspekte der konkreten Programmierung wurden in 3.1.9 konzeptioniert. Das Spiel wird durch ein Raster der Größe 4*4 von einzelnen Ebenen im dreidimensionalen Raum repräsentiert. Den Ebenen wird ein einheitliches, neutrales Material zugewiesen, das die Rückseite der Spielkarten simuliert.

Es ist zu gewährleisten, dass das Muster der ausgelegten Karten nicht vorhersehbar ist. Es soll sich von Anwendung zu Anwendung unterscheiden. Hierzu werden in zwei Datenfeldern die Bezeichner der Materialien in verschiedener Reihenfolge aufgeführt. Diese Felder werden

bei der anschließenden Initialisierung des Spiels durchlaufen und im Hintergrund in einer speziellen Datenstruktur gespeichert. Da in beiden Feldern jedes Material genau einmal aufgeführt wird, ist gewährleistet, dass jedes Motiv zweimal im Spiel präsentiert wird.

Mit Hilfe einer Zufallszahl zwischen null und sieben wird sichergestellt, dass sich die jeweiligen Einstiegspunkte der Iteration unterscheiden. Die Struktur des Memory wird im Hintergrund in einer speziellen Datenstruktur hinterlegt. Für jede Spielkarte wird ein `memory_tile` (s. untenstehende Definition) angelegt, das die Entity der Ebene, das Motiv der Karte und den aktuellen Status beinhaltet. Die gesamte Struktur des Spiels wird wiederum im Vector `memory_structure` erfasst, der alle `memory_tiles` enthält.

```
struct memory_tile
{
    //movable object of tile
    Entity* tile_ent;
    //material_name
    String material;
    //covered/discovered
    bool active;
};

std::vector <memory_tile> memory_structure;
```

Sind diese initialisierenden Schritte des Memory-Spiels abgearbeitet, wird die Mausinteraktion bereitgestellt. Hierzu wird innerhalb der Methode `void PlayState::mousePressed (GameManager* game, MouseEvent *e)` ein Modus fürs Memory-Spiel implementiert.

Nach der Schnittpunktberechnung, die bereits im oberen Abschnitt betrachtet wurde, folgen die entsprechenden Anweisungen zur Interaktion. Generell folgt dem Anklicken einer Memory-Karte das Aufdecken des Motivs. Da zu jedem Zeitpunkt maximal zwei Karten aufgedeckt sein dürfen, wird der Zähler `discovered` eingeführt. Er registriert die aktuell aufgedeckten Elemente. Damit wird gewährleistet, dass während des Aufdeckens der zweiten Karte eine Prüfung auf Gleichheit mit der ersten erfolgt. Sind bereits zwei Karten aufgedeckt und der Spieler klickt eine weitere an, werden die zuvor aufgedeckten, im Fall der Ungleichheit, automatisch wieder verdeckt.

Im Folgenden wird ein beliebiger Spielzug betrachtet. Der Nutzer klickt auf eine der Spielkarten. Die Strahlenverfolgung detektiert einen Schnittpunkt mit einer Spielkarte.

Erste Karte aufdecken

Durch die im Hintergrund gespeicherte Struktur wird mit Hilfe des Bezeichners auf das entsprechende `memory_tile` zugegriffen. Aus der Struktur wird das Material ausgelesen und zugewiesen. Der Zeiger `discovered1` sichert den Zugriff im nächsten Spielzug. Der Status wird auf `true` gesetzt, um zu gewährleisten, dass bei einem erneuten Klick auf diese Spielkarte keine Reaktion hervorgerufen wird.

```
if(discovered==0)
{
    //set material -> discover card
    memo_itr->movable->setMaterial(memory_structure[i].material);
    discovered++;
    //set pointerto memory_structure
    discovered1= &memory_structure[i];
    (*discovered1).active=true;
}
```

Zweite Karte aufdecken

Deckt der Spieler eine zweite Karte auf, wird in Analogie zur vorherigen der Zeiger `discovered2` zwischengespeichert, der Status aktiviert und das entsprechende Material gesetzt. Darüber hinaus werden die zwei aufgedeckten Karten auf Gleichheit geprüft. Wird dies bestätigt, werden die Karten dauerhaft aktiviert und der Materialbezeichner durch *"pair"* definiert. Dies verhindert bei weiteren Spielzügen das Registrieren eines Klicks auf bereits aufgedeckte Paare. Ein letzter Test stellt fest, ob alle Motivpaare gefunden wurden und das Memory gelöst ist.

```
else if (discovered==1)
{
    //set material -> discover card
    static_cast <Entity*> (memo_itr->movable)
        ->setMaterialName(memory_structure[i].material);
    discovered++;
    discovered2= &memory_structure[i];
    (*discovered2).active=true;
}
```

```

//found pair??
if ((*discovered1).material==(*discovered2).material)
{
    just_found_pair=true;
    num_pair++;
    //mark the found pair
    (*discovered1).material="pair";
    (*discovered2).material="pair";
    //found all pair??
    if(num_pair>=8)
    {
        won_memory();
    }
}
}

```

Eine weitere Karte aufdecken

Wird bei zwei aufgedeckten Karten eine weitere angeklickt, werden wiederum die bereits betrachteten Einstellungen vorgenommen. Der Zähler, der die Anzahl aktuell aufgedeckter Spielkarten beinhaltet, wird auf den Wert eins gesetzt. Darüber hinaus werden die zuvor aufgedeckten Karten, falls es sich nicht um ein Pärchen handelt, mit Hilfe der Zeiger `discovered1` und `discovered2` zurückgesetzt. Der Status wird deaktiviert und das neutrale Material, das die Kartenrückseite simuliert, gesetzt. Darauf folgend wird ein Zeiger auf die im aktuellen Spielzug entdeckte Karte zwischengespeichert.

```

else if(discovered==2)
{
    //set material -> discover card
    static_cast <Entity*> (memo_itr->movable)
        ->setMaterialName(memory_structure[i].material);
    //update counter
    discovered=1;
    //cover discovered cards
    if(!just_found_pair)
    {
        (*discovered1).tile_ent
            ->setMaterialName("Examples/CloudySky");
        (*discovered2).tile_ent
            ->setMaterialName("Examples/CloudySky");
        (*discovered1).active=false;
    }
}

```

```

        (*discovered2).active=false;
    }
    //settings for just discovered card
    discovered1= &memory_structure[i];
    (*discovered1).active=true;
    just_found_pair=false;
}

```

Nach diesem Prinzip werden alle Spielzüge des Memory behandelt. Ist das Spiel beendet, das heißt es wurden alle Motivpaare vom Spieler detektiert, kommt die Methode `wonMemory()` zum Einsatz. Während des Spielverlaufs werden im Hintergrund mit Hilfe eines Zählers die Anzahl der Spielzüge registriert. Die Höhe entscheidet über das Spielresultat.

```

if (memory_clicks <= 30) happy_feelou+=3;
else if (memory_clicks <= 40) happy_feelou+=2;
else if (memory_clicks >= 41) happy_feelou++;

```

Hat der Spieler das Memory über den Button *Zurück zum Spiel* verlassen, werden keine Feelou aufgeheitert. Das Spielergebnis wird dem Nutzer eingeblendet. Darüber hinaus wird die Variable `memory_finished` gesetzt, die für die Beendigung des Memory-Spiels und Fortsetzung des eigentlichen Spielverlaufs, das Einsammeln der Items, sorgt.

Sudoku

Im Folgenden werden die Initialisierung und Interaktion der Sudokuanwendung im Detail betrachtet. Zur generellen Funktionsweise ist anzumerken, dass es sich nicht um eine dynamische Implementierung des Sudokuprinzips handelt. Dies ließ der zeitliche Rahmen der Arbeit nicht zu. Alternativ werden sechs verschiedene Sudokuspiele bereitgestellt. Eine Zufallszahl entscheidet, welches ausgeführt wird. Diese Vereinfachung genügt den Anforderungen der Testumgebung, da mehrfache Wiederholungen des Spielszenarios nicht vorgesehen sind. In der Methode `createSudoku()` wird zu Beginn, in Analogie zum Memory, ein Raster von Ebenen im dreidimensionalen Raum generiert. Die Sudokuspiele werden in Datenfeldern bereitgestellt. Jedes Spiel

besteht aus zwei verschiedenen Datenfeldern. Eins beinhaltet die Materialien der einzelnen Felder. Es dient im weiteren Spielverlauf zur Prüfung des vom Nutzer entwickelten Resultats. Das andere, `sudoku_start`, wird als initiales Muster des Sudoku herangezogen. Ihm kann entnommen werden, welche Felder bereits zu Beginn gefüllt sind. Undefinierte Felder, die vom Spieler zu füllen sind, werden durch *unknown* gekennzeichnet.

Darüber hinaus wird ein weiteres Datenfeld angelegt, das die Einträge des Nutzers registriert und dem späteren Vergleich mit der Lösung dient. Um das Sudoku kurzweilig zu gestalten, wird lediglich ein 4*4 Raster angelegt.

Die Generierung des Sudokurasters geschieht in Anlehnung an das gewählte Datenfeld. Es entscheidet darüber, welche Felder mit einem Motiv versehen werden und welche vom Nutzer zu setzen sind. Dem folgenden Codefragment kann entnommen werden, wie das Raster erzeugt wird. Es werden lediglich elementare Anweisungen aufgeführt, das heißt es handelt sich nicht um den lückenlosen Code der Anwendung.

```
for (int i=0; i<=3; i++)
{
    for(int j=0; j<=3; j++)
    {
        sudoku_struct current_struct;
        //generate sudoku_planes
        createPlane("Sudoku"+sudoku_counter);
        //set standard_material, sudoku-image unknown
        if(sudoku_start[sudoku_counter]== "unknown")
        {
            current_sudoku_ent->setMaterialName("verdeckt");
            //save just generated entity in structure_vector
            current_struct.fix=false;
        }
        else
        {
            //generate entity out of plane
            current_sudoku_ent
            ->setMaterialName(sudoku_start[sudoku_counter]);
            current_struct.fix=true;
            //create fix-sudoku-mark, black frame
        }
    }
}
```

```

        //increment position and counter
    }
    //increment position and counter
}

```

Am rechten Bildschirmrand wird eine Selektionsspalte bereitgestellt. Hier kann der Nutzer eines der vier Motive auswählen, um es in ein Sudokukästchen zu setzen. Das selektierte Motiv wird durch eine rote Umrandung gekennzeichnet. Die Implementierung wird hier nicht näher betrachtet. Die Variable `sudoku_selected` beinhaltet das aktuell selektierte Motiv. Sie wird zu Beginn des Spiels mit dem Wert "glücklich" definiert. Hiermit ist die Initialisierung des Sudoku abgeschlossen. Die folgenden Ausführungen beziehen sich auf den konkreten Spielverlauf. Die Interaktion erfolgt auch beim Sudoku durch eine Strahlenverfolgung. Zurückgelieferte Schnittobjekte werden durch diese Abfrage selektiert.

```

else if( sudoku_active && !sudoku_finished
        && memo_itr->movable
        && StringUtil::match(memo_itr
        ->movable->getName(), "VariableSudokuEnt*"))

```

Es werden folgende Bedingungen geprüft: befindet sich die Anwendung im Sudokumodus, ist das Rätsel noch nicht beendet, handelt sich um ein bewegliches Objekt im Raum und beginnt der Bezeichner der Entity mit dem String "VariableSudokuEnt". Liefert dieser Ausdruck den Wert `true`, wird das selektierte Motiv in das angeklickte Kästchen gesetzt. Dem Kästchen wird also das Material der Selektion zugeordnet und der Spielzug im Datenfeld `user_solution` gespeichert. Darüber hinaus wird nach jedem gültigen Spielzug die Methode `user_solution()` aufgerufen.

```

user_solution[index]= sudoku_selected;
memo_itr->movable->setMaterial(sudoku_selected);
check_user_solution();

```

In `check_user_solution` wird geprüft, ob das Raster vollständig gefüllt ist. Ist dies der Fall, wird die vom Spieler gelieferte Lösung geprüft. Handelt es sich um die richtige Lösung, wird das Spielresultat durch aufgeheiterte Feelou eingeblendet.

Treffen nicht alle der oben aufgeführten Bedingungen zu, folgt eine weitere Abfrage, die zur Änderung des selektierten Motivs dient. Im Unterschied zur obigen, beginnt der Bezeichner der Entity mit dem String "SelectionEnt".

```
else if( sudoku_active
        && memo_itr->movable
        && StringUtil::match(memo_itr->movable
                              ->getName(), "SelectionEnt*"))
```

Mit Hilfe des Bezeichners des selektierten Objekts, kann auf das entsprechende Motiv geschlossen werden, das im folgenden Spielverlauf als *selektiert* gilt. Die neu gesetzte Selektion wird durch die rote Markierung gekennzeichnet. Wird das Spiel erfolgreich gelöst, gelten folgende Bewertungskriterien.

```
//keinen Fehler gemacht
//nicht auf Anfang zurückgesetzt
if(!sudoku_mistake && !zurueck) happy_feelou+=3;
//mind. einen Fehler gemacht
//nicht auf Anfang zurückgesetzt
else if(sudoku_mistake && !zurueck) happ_feelou+=2;
//auf Anfang zurückgesetzt
else if (zurueck) happy_feelou++;
```

Die Implementierung des Sudokuspiels ist abgeschlossen. In den vorangegangenen Abschnitten wurden die zentralen Elemente der Programmierung betrachtet. Als allgemeines Abbruchkriterium des Spiels *Feelou* wird ein Zähler herangezogen, der das Zurücksetzen der Items registriert. Insgesamt passieren 100 Items, bevor das Spiel automatisch beendet und der *EndState* eingeleitet wird. Die Abfrage erfolgt jeweils zu Beginn der *frameStarted()*-Methode.

```
if(passed_items >= 100)
    game->changeState(EndState::getInstance(), happy_feelou);
```

4.3.2 EndState

Wie bereits in der Konzeption in Kapitel 3 beschrieben, wird im EndState der Anwendung *Feelou* der Gemütszustand des Volkes als Spielresultat eingeblendet. Zur besseren Übersicht sind die Bewohner in Kategorien, *traurig* und *glücklich*, eingeteilt. So kann auf einen Blick festgestellt werden, in wie weit das Volk durch die Spielleistung aufgemuntert wurde.

Insgesamt werden 80 Feelou-Modelle geladen, um das Resultat zu präsentieren. Entsprechend dem erspielten Ergebnis werden diese den Kategorien *traurig* oder *glücklich* zugeteilt.

```
String material1="happy_end1";
String material2="happy_end2";
String material3="happy_end3";
int x= start_row;
int y= y_start_happy;
int z= fix_value;
//display feelou
for(int j=0; j<row; j++)
{
    for(int i=0; i < column; i++)
    {
        if(made_happy == counter)
        {
            x = start_row;
            y = y_start_sad;
            material1="sad_end1";
            material2="sad_end2";
            material3="sad_end3";
        }

        current_feelou= feelou->createChild(name+j+ "_" + i);
        current_feelou->translate(Vector3(x,y,z));
        current_feelou_ent= mSceneMgr
            ->createEntity(ent + j+"_"+i, "feelou.mesh");
        //what hair color??          switch(counter%3)
        {
            case 0: current_feelou_ent->setMaterialName(material1);
            break;
            case 1: current_feelou_ent->setMaterialName(material2);
            break;
            case 2: current_feelou_ent->setMaterialName(material3);
```



```

        break;
    }
    current_feelou->attachObject(current_feelou_ent);
    //increment positions
}
//increment positions
}

```

Die *Feelou* werden aufgereiht in der Szene platziert, so dass ein einheitliches Bild entsteht. Zur Auflockerung werden die Modelle mit verschiedenen Materialien versehen. Es gibt *Feelou* mit schwarzen, blonden und roten Haaren, die in den Variablen `material1`, `material2` und `material3` bereitgestellt werden. Welches Material angewandt wird, wird in Abhängigkeit vom Zähler innerhalb der Schleife ermittelt.

In zwei verschachtelten `for`-Schleifen werden die *Feelou* in Reihen dargestellt. Eine Bedingung prüft bei jedem Schleifendurchlauf, ob noch weitere aufgemunterte *Feelou* geladen werden oder ob zur Darstellung der traurigen übergegangen wird. Sind alle Glücklichen generiert, werden die relevanten Variablen an die traurige Darstellung angepasst. Der Materialbezeichner wird ebenfalls neu definiert.

4.4 Spezielle Implementierung der *Rellos*-Anwendung

Folgende Ausführungen betrachten die aus programmiertechnischer Sicht speziellen Aspekte der *Rellos*-Anwendung. In Analogie zur Anwendung *Feelou* wird der `IntroState` nicht berücksichtigt. Relevante Elemente werden in 4.2.2 aufgeführt.

4.4.1 `PlayState`

Das Ziel des Spiels ist die Minimierung einwandernder Gegner, die eine feindliche Übernahme des Dorfs der *Rellos* beabsichtigen. Hierzu müssen die Gegner durch gezielte Schüsse, die durch Mausinteraktion ausgelöst werden, eliminiert werden. Im Folgenden werden die Konzepte der konkreten Implementierung näher betrachtet.

Initialisierende Schritte

In Analogie zum `PlayState` der Anwendung *Feelou* werden auch hier die initialisierenden Anweisungen in der `enter()`-Methode zusammengefasst. In diesem Zusammenhang werden lediglich Aspekte betrachtet, die in Abschnitt 4.2.3 noch nicht behandelt wurden. Zu Beginn wird die Kamera den Anforderungen entsprechend kreiert und konfiguriert. Der Betrachter nimmt das Spielgeschehen aus Sicht der Spielfigur *Jim* wahr, das heißt, das entsprechende Modell wird nicht als Objekt in die Szene geladen. Der Blick richtet sich parallel zur x-Achse. In der Methode `createEnemy()` werden die feindlichen Eindringlinge generiert. Auch hier sind nach Abschnitt 4.3.1 Parallelen zur Anwendung *Feelou* auszumachen. Es werden 16 feindliche Modelle kreiert, die an einer fest definierten Grenze wieder zurückgesetzt werden. Auch die Positionierung geschieht nach einem Muster, das sich an Richtwerten orientiert. Es gibt verschiedene Arten von Feinden. Eine Sorte trägt einen roten, die andere einen grünen und eine weitere einen grauen Helm. Dementsprechend werden ihnen verschieden Materialien zugeordnet. Ist die Initialisierung der feindlichen Objekte abgeschlossen, wird ein Strahl zur Visualisierung des Schießvorgangs bereitgestellt. Dies geschieht in der Methode `createShootRay()`. Hierzu wird eine lang gezogene Ebene generiert, die sich vom Betrachter, entlang der Blickrichtung, ausdehnt.

```

Plane shoot= Plane(Vector3::UNIT_Y, 10);
MeshManager::getSingleton().createPlane("shoot", shoot);
shoot_ray = mSceneMgr->createEntity( "ShootRay", "shoot" );
//add node to navigate -> view-direction
shootRay= navigate->createChild("Shoot", Vector3(0,-10,0));
shootRay->attachObject(shoot_ray);
//only visible when mousePressed (shoot)
shootRay->setVisible(false);

```

Durch Mausinteraktion kann im späteren Spielverlauf der Blickwinkel des Spielers manipuliert werden. Hierzu wird der Szeneknoten `navigate`, dem unter anderem die Kamera und der `shootray` untergeordnet sind, rotiert. Somit zeigt die Ebene, die den Schuss simuliert, immer entlang der Blickrichtung des Betrachters und muss keinen weiteren Transformationen unterzogen werden. Die finale Anweisung innerhalb der `enter()`-Methode startet die Render-Schleife des Ogre3D-Systems und leitet somit den eigentlichen Spielverlauf ein. Die initialisierenden Maßnahmen der Anwendung *Rellos* sind abgeschlossen.

Spielverlauf

In Analogie zur Testapplikation *Feelou*, werden auch hier die zentralen Aspekte des Spielverlaufs im Rahmen der `frameStarted()`-Methode realisiert. Hierzu zählen unter anderem das stetige Translieren der einwandernden Gegner und das Testen auf potentielle Schnittpunkte zwischen der Spielfigur *Jim* und feindlichen Objekten.

Zu Beginn der `frameStarted()`-Methode wird geprüft, ob eines der Abbruchkriterien zutrifft. Konnte der Spieler zu wenig Objekte durch gezielte Schüsse eliminieren, ist das Spiel verloren. Darüber hinaus werden die entgegenkommenden Feinde registriert und bei Überschreitung einer fest definiert Grenze wird das Spiel automatisch beendet. Konnten zu diesem Zeitpunkt weniger als 40 Feinde ungehindert passieren, gilt das Spiel als gewonnen und der `EndState` wird eingeleitet.

```

//170 enemies came along, less than 40 could entered
if(passed_enemy>= 170 && entered <= 40) wonGame();
//more that 40 could enter -> GameOver
else if(entered_enemies > 40) lostGame();

```

Um die Feinde kontinuierlich auf den Spieler zuzubewegen, werden sie in

jedem Frame entsprechend transliert.

```
//green and red enemies
enemy_father->translate(Vector3(speed, 0, 0));
//big, gray enemy
enemy_special->translate(Vector3(speed,0,0));
```

Des Weiteren wird auf einen möglichen Schnittpunkt mit dem *grauen Feind* getestet. Der Kontakt zwischen ihm und *Jim* führt zum Verlust eines virtuellen Lebens. Um dies bereitzustellen, werden in jedem Frame die Position beider Objekte miteinander verglichen. In Analogie zum Einsammeln der Items wird auch hier ein virtueller Radius um Jim gezogen, der zur Berechnung herangezogen wird. Wird eine Kollision detektiert, folgt der Verlust eines Lebens. Die Variable `lives` wird dekrementiert. Sind keine Leben mehr vorhanden, `lives` gleich null, wird der EndState nach einer kurzen Information eingeleitet. Das Spiel ist verloren.

```
// collision detected
    //decrement lives, display info for player ...
    if (lives ==0) gameOver();
```

Wie bereits in den *Initialisierenden Schritten* erläutert wurde, werden lediglich 16 Feinde in die Szene geladen. Um zu gewährleisten, dass ein stetiger Fluss an einwandernden Feinden auf den Spieler einströmt, werden sie an einer bestimmten Grenze zurückgesetzt und nähern sich erneut aus der Ferne. Darüber hinaus wird dem zurückgesetzten Objekt eine neu generierte z-Position zugeordnet, um ein vorhersehbares Positionsmuster zu vermeiden.

Zum Vernichten der Feinde wird zwischen zwei Modi unterschieden. Die grünen Feinde müssen mit der rechten, die roten mit der linken Maustaste beschossen werden. Nur so kann der Spieler sie am Eindringen hindern. Das Schießen geschieht durch Mausinteraktion und wird in der Methode `mousePressed()` behandelt. Diese Funktion wird innerhalb des GameManagers bei der Betätigung einer beliebigen Maustaste auf die Instanz des aktuellen States, hier der `PlayState`, angewandt. Wird eine der Maustasten gedrückt, wird der Schuss visualisiert. Hierzu wird der in der Methode `createShootRay()` generierte `shootray` sichtbar gemacht

und in Abhängigkeit zum Modus das entsprechende Material gesetzt. Sie unterscheiden sich in der Farbgebung.

```
mSceneMgr->getSceneNode("Shoot_Ray")->setVisible(true);
//right mouse-button
if(e->getButtonID()==16)
    mSceneMgr->getEntity("ShootRay")->setMaterial("shoot1");
//left mouse-button
else if(e->getButtonID()==32)
    mSceneMgr->getEntity("ShootRay")->setMaterial("shoot2");
```

Zunächst wird geprüft, ob ein feindliches Objekt getroffen wurde. Hierzu wird eine Strahlenverfolgung herangezogen, die eine Liste geschnittener Objekte liefert. Der Strahl hat seinen Ursprung beim Betrachter, dem PlayingCharacter *Jim*, und breitet sich entlang der Blickrichtung aus.

```
//generate RaySceneQuery
RaySceneQuery *shootQ= mSceneMgr->createRayQuery(Ray());
//generate ray from pc along view-direction
shoot_ray(pc_position, navigate->getOrientation()*Vector3(1,0,0));
shootQ->setRay(shoot_ray);
```

Ist der Strahl generiert und verfolgt, wird das Resultat nach Treffern feindlicher Objekte durchsucht. Es wird getestet, ob sich ein roter Feind bei linker Maustaste oder ein grüner Feind bei rechter Maustaste im Resultat der Strahlenverfolgung befindet. Um sicherzugehen, dass der Spieler nicht permanent beide Maustasten betätigt und sich somit das Spielprinzip erleichtert, darf die linke Maustaste beim Schießen auf einen grünen Feind nicht gedrückt sein. Sind beide Taste gedrückt, ist keine der Bedingungen erfüllt und der Treffer wird nicht registriert. Zum aufgeführten Programmcode ist anzumerken, dass die rechte Maustaste durch die ID 16 die linke durch 32 repräsentiert wird.

```
if ((match(shoot_itr->movable->getName(), "green_enemy*")
    && e->getButtonID()==16
    && e->getButtonID()!=32)
    ||
    (match(shoot_itr->movable->getName(), "red_enemy*")
    && e->getButtonID()==32
    && e->getButtonID()!=16))
```

Trifft eine der Bedingungen zu, wird der auslösende Spielzug als erfolgreicher Treffer registriert. Das entsprechende feindliche Objekt leuchtet durch Materialzuweisung hell auf und wird im darauf folgenden Frame unsichtbar gesetzt. Das Eliminieren der grünen und roten Feinde ist, wie geschildert, durch einen Treffer zu erzielen. Der graue Feind muss zehnmal erfolgreich mit der linken Maustaste getroffen werden. Der Zähler `big_enemy_shot` speichert die bereits erfolgreichen Treffer. Beim Zehnten kommt es zur Eliminierung des grauen Feindes. Er wird im nächsten Frame unsichtbar gesetzt und kann in diesem Durchlauf das Spielgeschehen nicht negativ beeinflussen.

```
else if( match(shoot_itr->movable->getName(), "Special_Enemy")
        && e->getButtonID()==32
        && e->getButtonID()!=16)
{
    if(big_enemy_shot>= 9) //eliminate big, gray enemy
    else big_enemy_shot++;
}
```

4.4.2 EndState

Im EndState der Anwendung Rellos wird lediglich ein Menü bereitgestellt und das Spielergebnis der Leistung entsprechend präsentiert. Es handelt sich um eine rein textuelle Darstellung, die hier nicht näher betrachtet wird.

5 Ergebnisse

Das Ziel der Arbeit ist die Analyse weiblicher Präferenzen bei Computerspielen. Das entsprechende Testszenario setzt sich aus zwei verschiedenen Spielsequenzen und einem abschließenden Fragebogen zusammen. Es werden die in Kapitel 2.4 herausgearbeiteten Aspekte aufgegriffen. Die in der Theorie herausgearbeiteten Faktoren werden somit in einem praktischen Test überprüft.

5.1 Ablauf

Die Testperson bekommt im Vorfeld keine Informationen darüber, welche Thematik dem Test zugrunde liegt. So wird eine unterbewusste Manipulation durch geschlechtsspezifische Stereotype vermieden. Das Szenario beginnt mit dem Spielen beider in Kapitel 4 entwickelten Applikationen. Die Reihenfolge spielt keine Rolle und liegt im Ermessen des Probanden.

Im Anschluss wird ein Fragebogen vorgelegt, der vollständig auszufüllen ist. Es werden die in Kapitel 2.4 aufgeführten, relevanten Aspekte aufgegriffen und hinterfragt.

5.2 Konzeption des Fragebogens und Ergebnisse

Die Konzeption des Fragebogens ist neben der der Testapplikationen für die Qualität der Testergebnisse von zentraler Bedeutung. Die relevanten Aspekte werden gezielt hinterfragt und die Auswahl möglicher Antworten wohl überlegt.

Zu jeder Frage sind die zugrunde liegenden Quellen noch einmal angegeben. Männliche Testpersonen werden als Vergleichswerte herangezogen, ihre Präferenzen werden nicht detailliert betrachtet. So kann herausgestellt werden, welche Tendenzen geschlechtsspezifisch sind. Insgesamt wurden siebzehn weibliche und elf männliche Testpersonen herangezogen. Dies entspricht keiner repräsentativen Menge, genügt jedoch zur Bestimmung von Tendenzen.

Allgemeines

Zu Beginn wird das Geschlecht, der zentrale Aspekt des Testszenarios, erfragt.

Darauf folgt die Bestimmung der allgemeinen Präferenz. Welches Spiel gefällt der Testperson im Gesamten besser. Bei diesem Urteil spielen alle Faktoren der Applikationen eine Rolle. Hierzu zählen auch zahlreiche, die im Rahmen dieser Arbeit nicht berücksichtigt werden.

Welches Spiel hat Ihnen insgesamt besser gefallen?

O Rellos O Feelou O keine Präferenz

Ergebnis

Abbildung 8 veranschaulicht die Ergebnisse der oben aufgeführten Frage. 71 Prozent der weiblichen Testpersonen hat das Spiel *Feelou*, in dem weibliche Präferenzen berücksichtigt werden, besser gefallen. Der Vergleichswert männlicher Probanden zeigt mit 55 Prozent eine Tendenz zum Spiel *Rellos*.

Lediglich 11 Prozent der Frauen und 18 Prozent der Männer favorisieren entgegengesetzt. 18 Prozent weiblicher und 27 Prozent männlicher Testpersonen können nach eigenen Angaben keine Präferenz ausmachen.

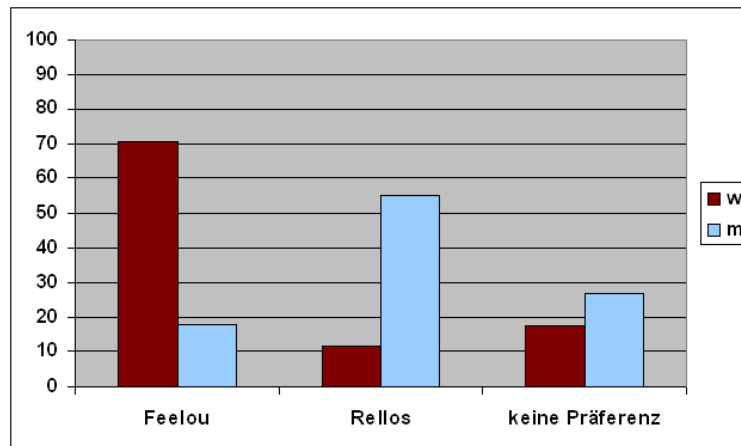


Abbildung 8: Allgemeine Präferenz

Erlernen der Spielregeln

Nach aufgeführten Erkenntnissen legt die weibliche Spielerschaft großen Wert auf die vorangehende Einarbeitung ins Spielgeschehen [3], [17].

Durch untenstehende Frage soll herausgefunden werden, ob dies in der Spielpraxis zutrifft. Welche der Testpersonen lesen die Anleitung, welche starten ohne Einstieg direkt ins Spielgeschehen?

Haben Sie die Anleitungen der Spiele gelesen?

ja, gründlich ja, flüchtig nein

Ergebnis

Abbildung 9 zeigt die Ergebnisse. 94 Prozent der weiblichen Probanden geben an, die Anleitung zum Spiel gelesen haben, davon 71 Prozent gründlich. Lediglich 6 Prozent äußern, die Anleitung nicht gelesen zu haben. Männliche Testpersonen haben zu 82 Prozent die Anleitung studiert, 41 Prozent flüchtig.

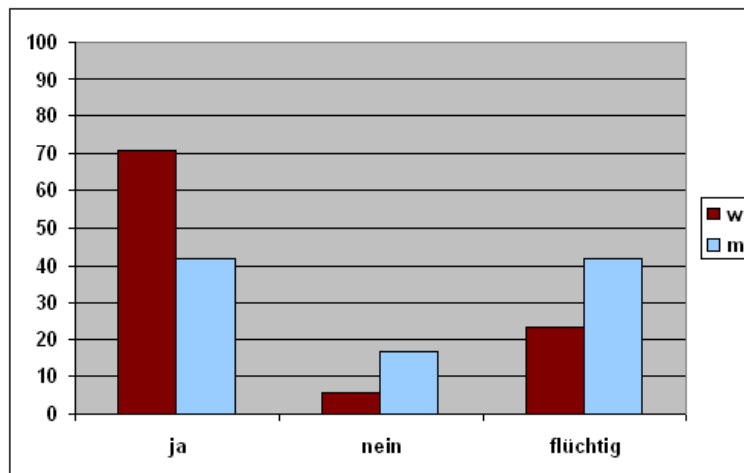


Abbildung 9: Anleitung

Konflikt

Der zugrundeliegende Konflikt eines Computerspiels begleitet das gesamte Spielgeschehen und hat demnach großen Einfluss auf das Spielerlebnis.

Hier wird auf die Art des Konflikts eingegangen. Bevorzugt die Testperson den direkten Konflikt (*Gut gegen Böse*) oder das Herbeiführen einer allseitig zufrieden stellenden Lösung [6].

In welchem Spiel hat Ihnen der zugrundeliegende Konflikt besser gefallen?

O Rellos (Rette Dein Volk vor feindlichen Eindringlingen)

O Feelou (Verhelfe Deinem Volk zu besserer Stimmung)

O keine Präferenz

Ergebnis

Im Folgenden werden die Ergebnisse oben aufgeführter Frage betrachtet. 58 Prozent der weiblichen und lediglich 18 Prozent der männlichen Testpersonen bevorzugen die Konfliktdarstellung im Spiel *Feelou*. Die entgegengesetzt konzeptionierte Testapplikation *Rellos* favorisieren 12 Prozent der Frauen und 55 der Männer. 30 Prozent der weiblichen, 27 Prozent der männlichen Testpersonen geben an, keine Präferenz zu haben.

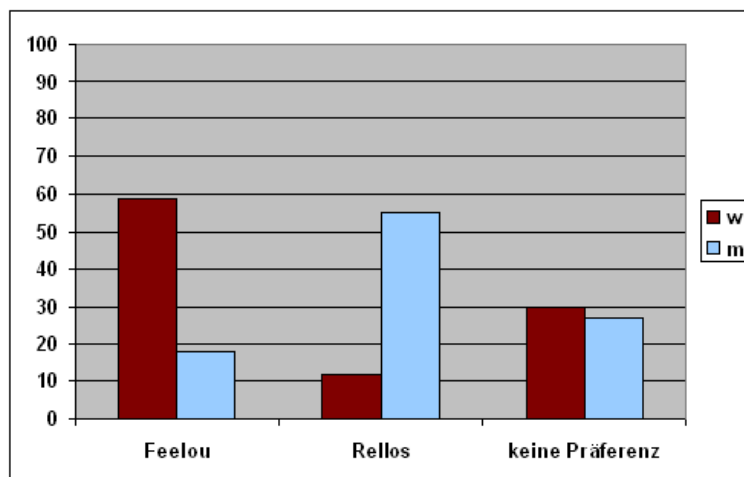


Abbildung 10: Konflikt

Der folgende Aspekt des Fragebogens geht gezielt auf die direkte Auseinandersetzung mit dem feindlichen Volk ein. Die Auseinandersetzung fordert die Zerstörung des feindlichen Volks, nach [10] widerspricht dieses Konzept weiblichen Präferenzen.

Hat Ihnen die direkte Auseinandersetzung mit dem feindlichen Volk gefallen?

ja nein keine Präferenz

Ergebnis

Im Zusammenhang mit dieser Frage, geben 59 Prozent der weiblichen Probanden und 27 Prozent der männlichen an, kein Gefallen an der direkten Auseinandersetzung zu finden. Ein geringer Anteil von 18 Prozent der Frauen bewertet die Konfrontation als positiv, bei den Männern ist es eine Mehrheit von 55 Prozent. Keine Präferenz äußern 23 Prozent der weiblichen und 18 Prozent der männlichen Probanden.

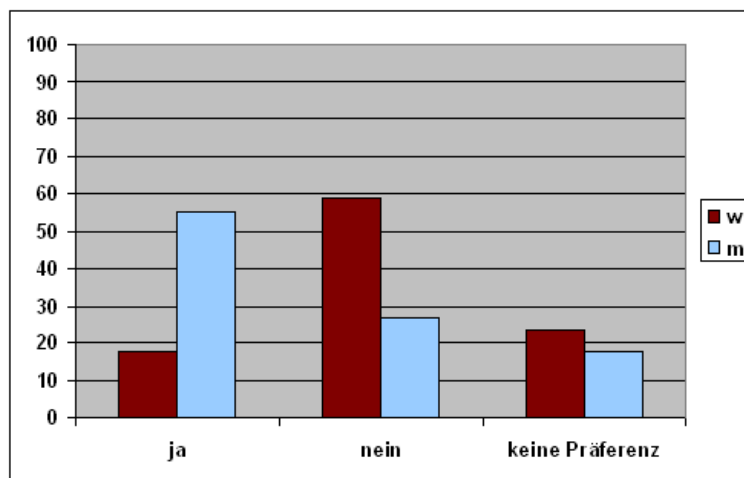


Abbildung 11: Direkte Auseinandersetzung

Fehlertoleranz

Ein zentraler geschlechtsspezifischer Aspekt der Fehlertoleranz sei das GameOver-Prinzip. Begangene Fehler im Spiel können demnach nicht korrigiert werden, was nach [19] das Spielvergnügen weiblicher Personen beeinträchtigt. Es wird erfragt, ob das konventionelle GameOver-Prinzip während der Spielsequenz Frust auslöse [6].

Kam in Rellos auf Grund des GameOver-Prinzips Frust auf?

ja nein ein wenig

Ergebnis

Das GameOver-Konzept rufe bei 52 Prozent der weiblichen und 45 Prozent der männlichen Probanden Frust hervor. 18 Prozent der Frauen sowie 27 Prozent der Männer geben an, das erneute Starten des Spiels auf Grund wiederholter Fehler als nicht frustrierend wahrzunehmen. 30 Prozent der weiblichen und 27 der männlichen Testpersonen verspüren durch das GameOver-Prinzip ein wenig Frust.

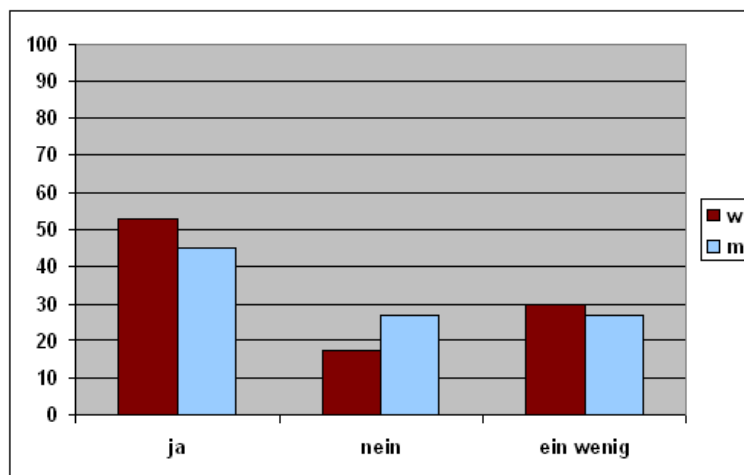


Abbildung 12: GameOver

In der Anwendung *Feelou* werden Fehler oder Schwächen im Spiel nicht bestraft [19]. Dies entspricht nach bisherigen Erkenntnissen weiblichen Präferenzen bei Computerspielen. Die Belohnung soll im Vordergrund stehen, nicht die Bestrafung.

Hat es Ihnen gefallen, dass in Feelou Spielfehler nicht bestraft wurden?

O ja O nein O keine Präferenz

Ergebnis

Im Rahmen dieser Frage geben 59 Prozent der weiblichen und 55 Prozent der männlichen Testpersonen an, das Konzept des Belohnens statt Bestrafens als positiv zu werten. Lediglich 12 Prozent der Frauen und 27 Prozent der Männer scheinen kein Gefallen daran zu finden. 29 Prozent der weiblichen, 18 der männlichen Probanden äußern keine Präferenz.

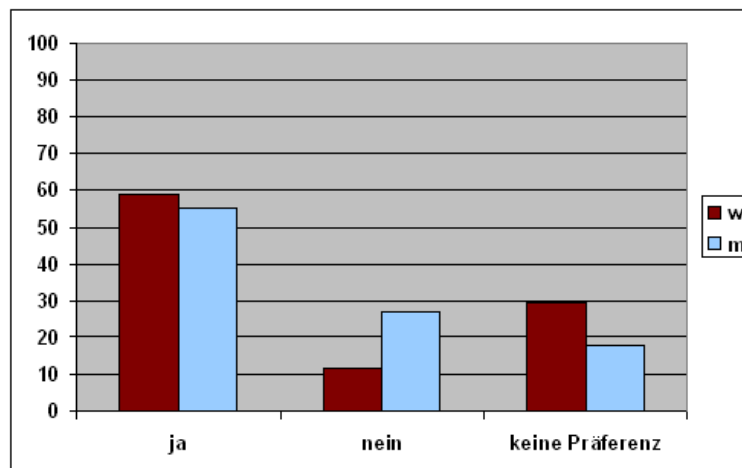


Abbildung 13: Bestrafung

Spielmotivation

Im Folgenden wird die Motivation, die eine Person zum Spielen animiert, hinterfragt. Nach bisherigen Erkenntnissen soll die weibliche Spielerschaft keinen großen Wert darauf legen, eigene Resultate mit denen anderer vergleichen zu können, um sich zu brüsten [3]. Es wird eine allgemein gültige Fragestellung formuliert.

Legen Sie Wert darauf, Ihre eigene Spielleistung mit der anderer Spieler zu messen (beispielsweise in einer Highscore-Liste)?

ja nein keine Präferenz

Ergebnis

Lediglich 12 Prozent aller befragten Frauen geben an, keinen Wert darauf zu legen, bei den Männern ist es eine Mehrheit von 54 Prozent. Dagegen stehen 59 der weiblichen Probanden und 19 Prozent der männlichen. Keine Relevanz habe dieser Aspekt für 29 Prozent der weiblichen und 27 der männlichen Testpersonen.

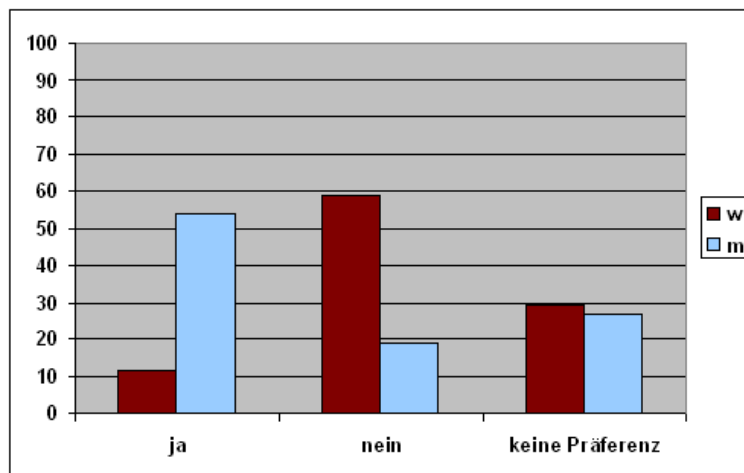


Abbildung 14: Highscore

Darüber hinaus geht folgender Punkt auf die Darstellung des Spielergebnisses ein. In *Rellos* wird die Anzahl der ungehindert ins Dorf einwandernden Feinde als Resultat herangezogen. Darüber hinaus entscheidet das Ergebnis über Sieg oder Niederlage. Nach [7] sei zu erwarten, dass diese Darstellung der weiblichen Spielerschaft nicht entgegenkomme. In der Anwendung *Feelou* wird der erspielte Zustand des Volks abgebildet. Das heißt, der Proband sieht konkret, was er durch seine Leistung im Spiel erreichen konnte [4]. Es wird nicht über Sieg oder Niederlage geurteilt.

Welches Spielresultat, unabhängig von Ihrer persönlichen Spielleistung, hat Ihnen besser gefallen?

- Rellos (Sieger/Verlierer, konkrete Werte)*
- Feelou (erspielter Zustand des Volks der Feelou)*
- keine Präferenz*

Ergebnis

71 Prozent der weiblichen Probanden favorisieren die Darstellung im Spiel *Feelou*, bei den Männern sind es lediglich 18. Keiner der weiblichen Befragten bevorzugte das Ergebnis in *Rellos*, jedoch 55 Prozent der männlichen. Keine Präferenz haben nach eigenen Angaben 29 Prozent der Frauen und 27 Prozent der Männer.

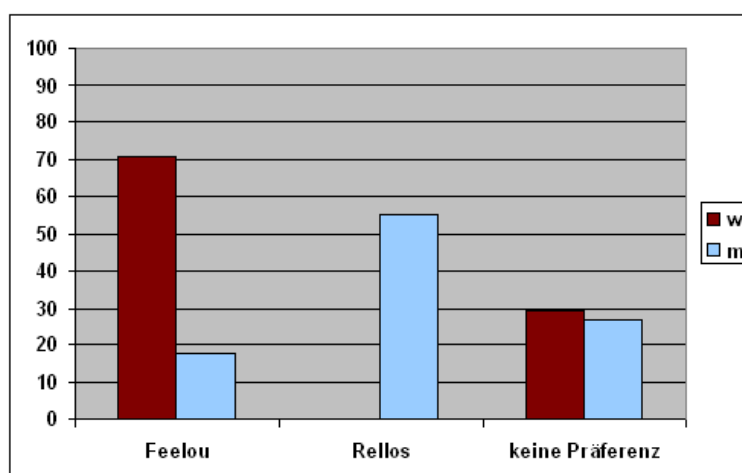


Abbildung 15: Resultat

Emotionale Bindung

Der Aufbau einer emotionalen Bindung im Rahmen dieses Testszenarios ist problematisch (s. 2.4.4). Es wird eine allgemein gültige Frage herangezogen, um die Thesen überprüfen zu können [3].

Legen Sie im allgemeinen Wert darauf, eine emotionale Bindung zum Spielgeschehen aufbauen zu können?

ja nein

Ergebnis

65 Prozent der weiblichen Testpersonen geben an, Wert darauf zu legen, eine emotionale Bindung zum Spielgeschehen aufbauen zu können. Männer vertreten diesen Standpunkt zu 55 Prozent. Demnach verspüren 35 Prozent der weiblichen und 45 Prozent der männlichen Probanden nicht den Wunsch, emotional eingebunden zu werden.

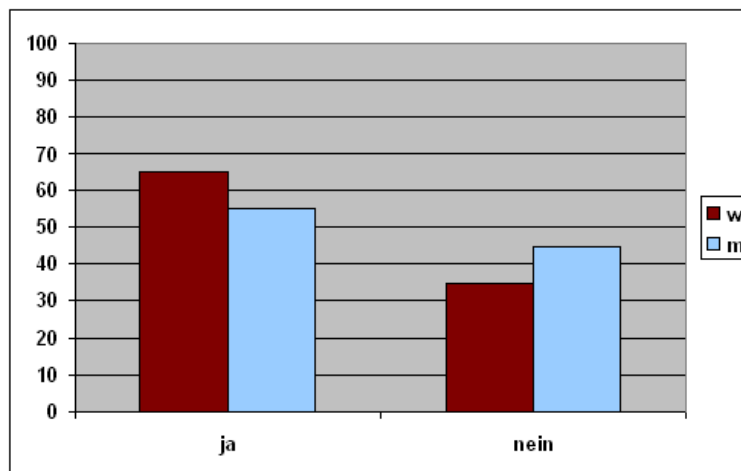


Abbildung 16: Emotionale Bindung (allgemein)

Ergänzend zur oben aufgeführten, allgemeinen Fragestellung wird festgehalten, ob die Testperson im Rahmen dieses kurzweiligen Szenarios eine emotionale Bindung aufbauen kann. Zur Problematik dieses Aspekts s. Abschnitt 2.4.4.

Konnten Sie in den Spielen eine emotionale Bindung aufbauen?

in keinem in beiden nur in Rellos nur in Feelou

Ergebnis

Eine deutliche Mehrheit von 83 Prozent der weiblichen und 100 Prozent der männlichen Probanden können nach eigenen Angaben zu keiner der Testapplikationen einen emotionalen Bezug aufbauen. 17 Prozent der Spielerinnen können sich emotional in die Anwendung *Feelou* einfinden. Keine Testperson gibt an, dies im Zusammenhang mit dem Spiel *Rellos* wahrzunehmen.

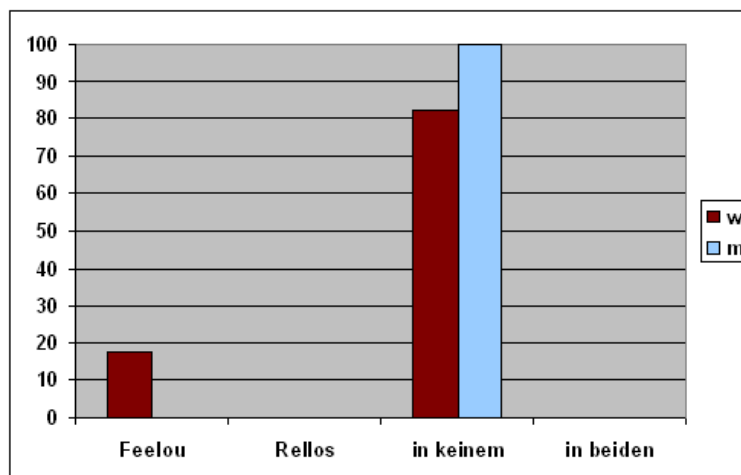


Abbildung 17: Emotionale Bindung zum Spiel

Ein weiterer Aspekt zum Aufbau einer emotionalen Bindung zum Spiel stellt die Hintergrundgeschichte dar. Es ist zu erwarten, dass der Großteil weiblicher Testpersonen die einleitende Geschichte liest.

Haben Sie die Geschichten der Spiele gelesen?

ja, gründlich ja, flüchtig nein

Ergebnis

Die Mehrheit von 71 Prozent der weiblichen und 50 Prozent der männlichen Probanden geben an, die Geschichte gründlich gelesen zu haben. 17 Prozent der Frauen, 25 Prozent der Männer haben sie überflogen. Keine Relevanz hat die Geschichte für 12 Prozent der weiblichen und 25 Prozent der männlichen Befragten.

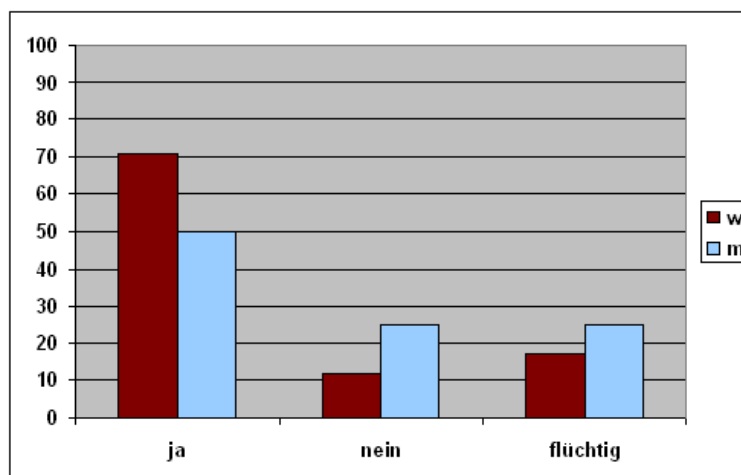


Abbildung 18: Geschichte

Kognition

Folgende Frage soll ermitteln, welches der grundlegenden Spielprinzipien die Testperson favorisiert. Das reaktionsschnelle Schießen, das schnelle, zielgerichtete Mausinteraktion verlangt oder das eher ruhige Einsammeln, das die Selektion der richtigen Items erfordert. Nach [18] und [3] ist zu erwarten, dass die weibliche Spielerschaft den PC Jiny bevorzugt. Die Frage lässt sich nicht eindeutig einer Kategorie zuordnen, die Präferenz bzgl. des Konflikts spielt ebenfalls eine Rolle.

Welche Aufgabe hat Ihnen besser gefallen?

Abschießen der Feinde Einsammeln der Items keine Präferenz

Ergebnis

47 Prozent der weiblichen, 18 Prozent der männlichen Befragten geben an, am Einsammeln der Items größeren Gefallen zu finden. Das Schießen in der Anwendung *Rellos* bevorzugen 55 der männlichen und 35 der weiblichen Versuchspersonen. Lediglich 18 Prozent der Frauen und 27 Prozent der Männer äußern in diesem Zusammenhang keine Präferenz.

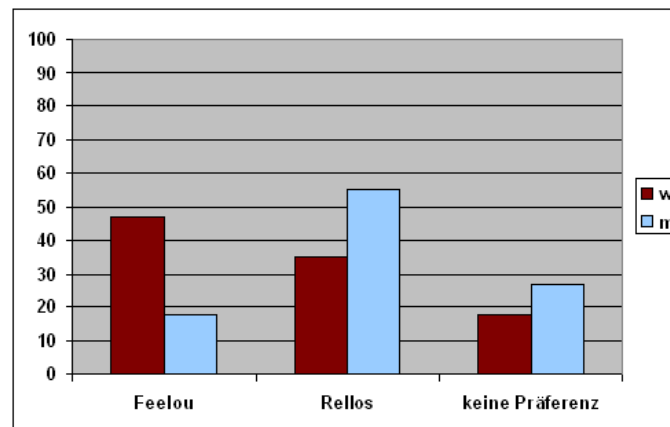


Abbildung 19: Aufgabe

Ein weiterer Aspekt geschlechtsspezifischer Kognition ist das Empfinden potentiell aufkommender Hektik im Spiel [3]. Zuerst wird die Anwendung *Rellos* betrachtet.

War Ihnen das Spielgeschehen in Rellos zu hektisch?

O ja O nein O ein wenig

Ergebnis

47 Prozent der weiblichen und 27 Prozent der männlichen Probanden geben an, das Spielgeschehen in *Rellos* als zu hektisch zu empfinden. Ebenfalls 27 der Männer und 12 Prozent der Frauen nehmen diesen Aspekt gegensätzlich wahr. Ein wenig zu schnell sei der Ablauf für 41 Prozent der weiblichen und 46 Prozent der männlichen Versuchspersonen.

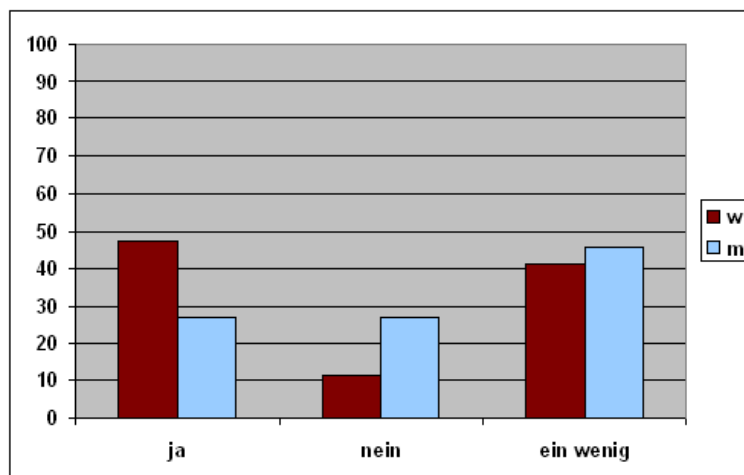


Abbildung 20: Hektik (*Rellos*)

In Analogie zur vorherigen Frage wird der Aspekt in Hinblick auf die langsamere Anwendung *Feelou* betrachtet.

War Ihnen das Spielgeschehen in Feelou zu hektisch?

ja nein ein wenig

Ergebnis

23 Prozent der Frauen, null Prozent der Männer geben an, das Spielgeschehen als zu hektisch zu empfinden. Ein wenig Hektik komme bei 18 Prozent der weiblichen und 19 Prozent der männlichen Testpersonen auf. Eine Mehrheit von 59 Prozent weiblicher und 81 Prozent männlicher Probanden geben an, keine Hektik zu empfinden.

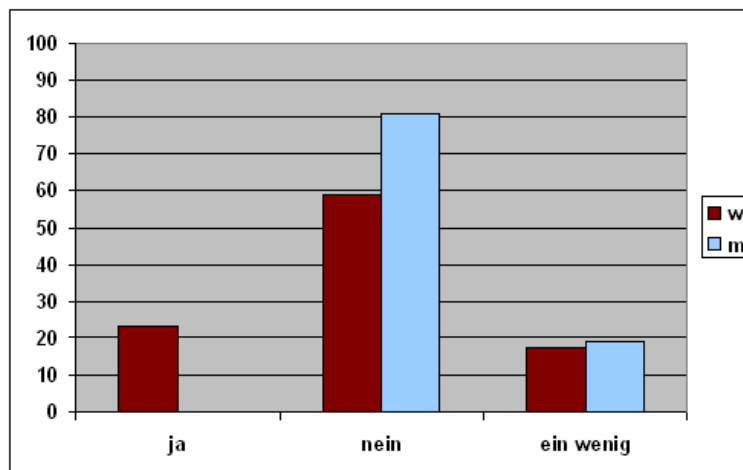


Abbildung 21: Hektik (*Feelou*)

Darüber hinaus wird gezielt auf die ruhigen Phasen im Spiel *Feelou* durch die Puzzle-Sequenzen eingegangen [18], [3].

Empfanden Sie die langsameren Phasen (Puzzle) in Feelou als angenehm?

ja nein keine Präferenz

Ergebnis

Eine deutliche Mehrheit der Befragten gibt an, die langsamen Sequenzen als angenehm zu empfinden. Es handelt sich um 88 Prozent der weiblichen und 64 Prozent der männlichen Testpersonen. Hingegen geben null Prozent der Frauen und 18 Prozent der Männer an, dieses Konzept nicht zu favorisieren. Keine Präferenz äußern 12 Prozent der weiblichen sowie 18 Prozent der männlichen Probanden.

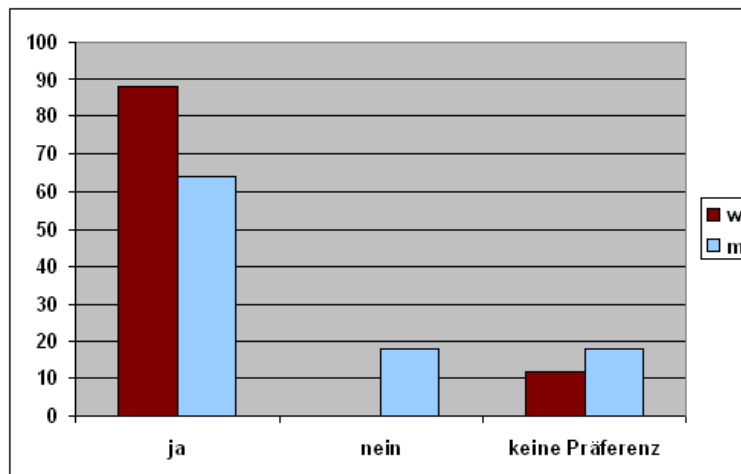


Abbildung 22: Langsame Phasen

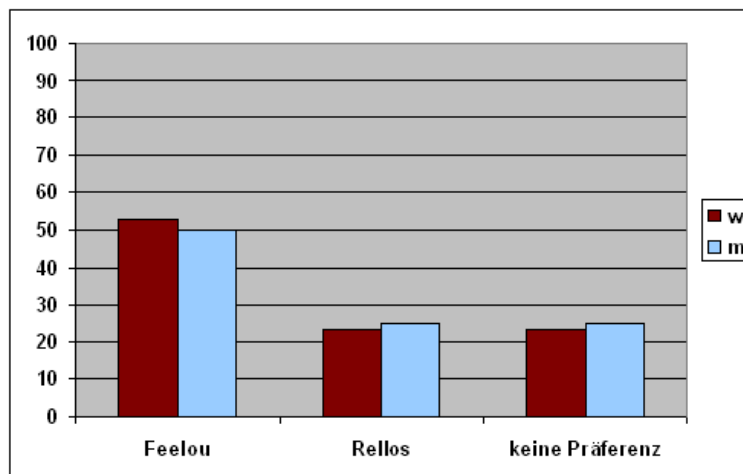
Hier wird die Art der Interaktion mit der Maus betrachtet. Im Spiel *Feelou* erfolgt diese intuitiv, Jiny wird lediglich durch einfache Mausbewegung navigiert. In *Rellos* wird die Blickrichtung, das exakte Anvisieren und die Koordination der zwei Maustasten gefordert [3]. Bei diesem Aspekt spielt auch das Erlernen der Spielregeln eine Rolle.

Favorisieren Sie die Art der Interaktion mit der Maus in einem der Spiele?

ja nein keine Präferenz

Ergebnis

Die Frage weist keine geschlechtsspezifischen Tendenzen auf. Gut die Hälfte beider Geschlechter - 54 Prozent der weiblichen sowie 50 Prozent der männlichen Befragten - bevorzugen die intuitive Interaktion im Spiel *Feelou*. Dagegen stimmen 23 Prozent der Frauen und 25 Prozent der Männer für die Steuerung im Spiel *Rellos*. Die gleiche Verteilung gilt für die Probanden, die keine Präferenz ausmachen können.



Avatar

Der Avatar repräsentiert den Spieler in der virtuellen Welt. Es wird erfragt, ob der Proband generell eine der beiden Spielfiguren Jiny oder Jim bevorzuge. Nach [5], [3] ist zu erwarten, dass die weibliche Spielerschaft das Spielen mit Jiny favorisiert.

Würden Sie das Spielen mit einer der beiden Spielfiguren bevorzugen?

Jiny Jim keine Präferenz

Ergebnis

Eine deutliche Mehrheit von 82 Prozent der weiblichen Testpersonen gibt an, Jiny als Spielfigur zu bevorzugen. Die Männlichen vertreten diesen Standpunkt zu 16 Prozent. Keine der Frauen favorisiert *Jim* als Vertreter im Spiel, bei den Männern sind es 42 Prozent. Ebenfalls 42 Prozent der männlichen Befragten geben an, bei dieser Auswahl keine Präferenz zu haben. Dagegen stehen lediglich 18 Prozent der Weiblichen.

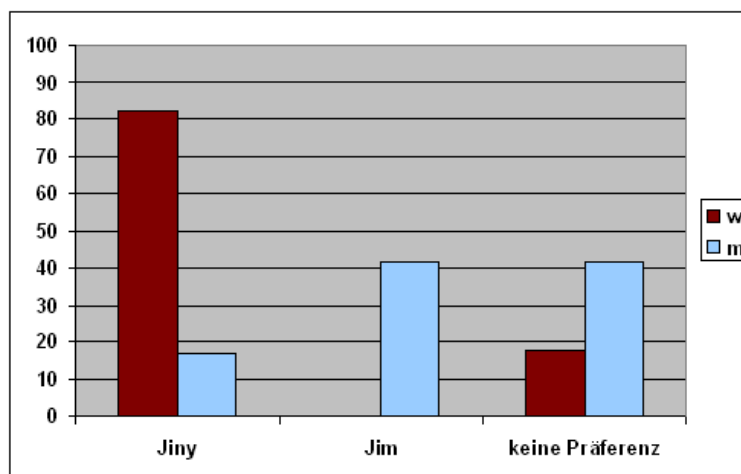


Abbildung 23: Spielfigur

Puzzle

Dieser Aspekt des Fragebogens widmet sich den Puzzle-Sequenzen der Testapplikation *Feelou*. Nach vorangehenden Erkenntnissen sollen diese Minispiele dem weiblichen Spielverhalten entgegenkommen [?].

Haben Ihnen die Puzzle als Zwischensequenzen in Feelou gefallen?

ja nein keine Präferenz

Ergebnis

88 Prozent der weiblichen und 81 Prozent der männlichen Probanden bewerten die Zwischensequenzen als positiven Aspekt des Spiels. Keiner der Befragten gibt an, die Puzzle als negativ zu empfinden. 12 Prozent der Frauen und 19 Prozent der Männer äußern keine Präferenz.

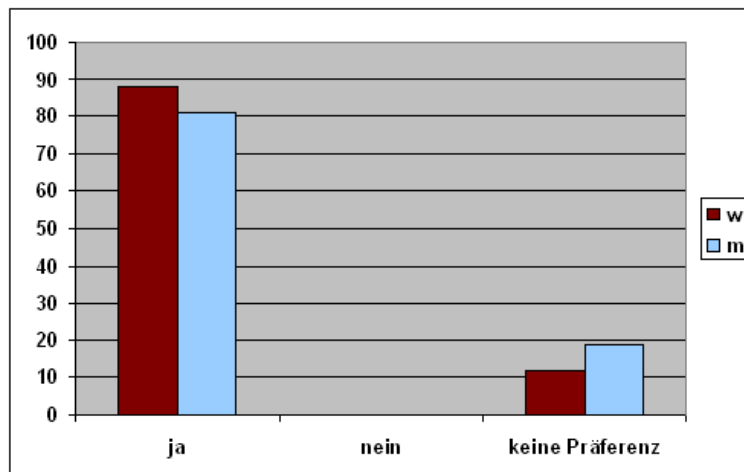


Abbildung 24: Puzzle

5.3 Auswertung und Schlussfolgerungen

Abschließend werden auf Basis der gewonnenen Ergebnisse Schlussfolgerungen gezogen. Es lassen sich lediglich Tendenzen ausmachen, allgemeine Gültigkeit kann aufgrund der Testbedingungen nicht gewährleistet werden. Der Kreis der Befragten stellt keine repräsentative Menge dar. Darüber hinaus entsprechen die Testapplikationen lediglich einer exemplarischen Umsetzung der relevanten Faktoren.

Allgemein

An Hand der Ergebnisse lässt sich eine deutliche Tendenz der weiblichen Befragten zur Testapplikation *Feelou* ausmachen. Da es sich um eine allumfassende Fragestellung handelt, die auch nicht untersuchte Aspekte einbindet, ist es nicht möglich eine konkrete Schlussfolgerung zu ziehen. Jedoch deutet es an, dass die zu untersuchenden Faktoren einen nach bisherigen Erkenntnissen erwartungskonformen Einfluss auf das gesamte Spiel haben.

Erlernen der Spielregeln

Die absolute Mehrheit weiblicher Versuchspersonen hat die Anleitung gelesen. Die These, dass die weibliche Spielerschaft großen Wert auf die vorangehende Einarbeitung ins Spielgeschehen legt, scheint bestätigt. Es ist anzumerken, dass die Testsituation ein erhöhtes Interesse an der Anleitung herbeigeführt haben könnte.

Bei der Entwicklung von Computerspielen sollte großer Wert auf die Gestaltung und Konzeptionierung des Lernprozesses gelegt werden. Ein intuitiver, kurzweiliger Einstieg ist nach dem Erwerb ein entscheidender Schritt zu einem erfolgreichen Titel.

Konflikt

Die Testergebnisse lassen darauf schließen, dass ein offener, direkter Konflikt auf kämpferischer Basis nicht den Präferenzen weiblicher Spieler entspricht. Direkte Auseinandersetzungen auf kriegerischer Ebene scheinen die Mehrheit weiblicher Computerspieler nicht anzusprechen. Es handelt sich um eine geschlechtsspezifische Präferenz, männliche Befragte finden tendenziell großes Gefallen an dieser Art des Konflikts.

Ein Großteil der Spielerinnen favorisiert eine Aufgabe, die das Verbessern einer gegebenen Situation verlangt, ohne einer anderen Partei Schaden zufügen zu müssen. Das konventionelle Schema *Gut gegen Böse* trete als grundlegendes Spielkonzept außer Kraft.

Fehlertoleranz

Beim untersuchten Aspekt der Fehlertoleranz ergeben sich keine geschlechtsspezifischen Präferenzen. Etwa die Hälfte beider Geschlechter geben an, auf Grund des GameOver-Prinzip Frust empfunden zu haben. Dies widerspricht tendenziell den Ausführungen aus Abschnitt 2.4.3. Es wurde davon ausgegangen, dass die weibliche Frustrationsgrenze deutlich niedriger liege als die der Männer.

Vielmehr stellt sich im Rahmen dieses Testszenarios heraus, dass ca. 80 Prozent beider Geschlechter durch das GameOver-Prinzip frustriert seien, jeweils 30 Prozent davon ein wenig.

Darüber hinaus hat es einer deutlichen Mehrheit beider Geschlechter gefallen, dass Spielfehler in der Anwendung *Feelou* nicht bestraft wurden. Demnach sollten alternative Bewertungskriterien für die Leistung im Spiel entwickelt werden. Das Belohnen soll im Vordergrund stehen, nicht die Bestrafung begangener Spielfehler.

Spielmotivation

Bezüglich der generellen Motivation, die eine Person zum Spielen bewegt, sind deutlich geschlechtsspezifische Präferenzen auszumachen. Ein schwindend geringer Anteil weiblicher Spieler legt nach eigenen Angaben Wert darauf, die eigene Leistung mit der anderer in Bezug zu setzen. Dem Konkurrenzgedanken bezüglich anderer Spieler komme demnach lediglich eine geringe Bedeutung zu. Bestenlisten zählen somit nicht zu den von Frauen favorisierten Aspekten eines Computerspiels.

Des Weiteren lassen sich bezüglich des Spielresultats Erkenntnisse festhalten. Die zahlenbasierte Darstellung im Spiel *Rellos*, die den Spieler als Sieger beziehungsweise Verlierer deklariert, wird von keiner der weiblichen Befragten präferiert. Tendenziell legen Frauen keinen Wert auf diese Kategorisierung. Die Bewertung der Spielleistung sollte nach weiblichen Präferenzen kein Urteil abgeben und sich am Geschehen des Spiels orientieren. Darüber hinaus soll im direkten Bezug zum

Spielgeschehen ersichtlich sein, was durch die eigene Leistung im Spiel erreicht wurde (Feelou).

Die Thesen aus Abschnitt 2.4.7, werden durch die Ergebnisse des Testszenarios gefestigt.

Emotionale Bindung

Aufgrund der Komplexität dieses Aspekts wurde unter anderem auf eine allgemeingültige Fragestellung zurückgegriffen. Es stellt sich keine deutlich geschlechtsspezifische Präferenz heraus. Sowohl Frauen, als auch Männer legen Wert darauf eine emotionale Bindung zum Spielgeschehen aufbauen zu können. Der Trend ist bei den weiblichen Testpersonen stärker ausgeprägt, die Differenz genügt jedoch keiner Schlussfolgerung. Darüber hinaus spricht die Testapplikation *Feelou* lediglich vereinzelt die emotionale Ebene der Spielerinnen an, die der Männer wird gar nicht erreicht. Eine deutliche Mehrheit beider Geschlechter kann keinen emotionalen Bezug zu den Testapplikationen aufbauen.

Die Problematik der zeitlichen Komponente (s. 2.4.4) dominiert diesen Aspekt. Obwohl 18 Prozent der Frauen angeben, einen Bezug zu *Feelou* entwickelt zu haben, genügt dies lediglich zur vagen Annahme, dass Spielerinnen weniger Zeit benötigen eine emotionale Bindung aufzubauen.

Die weibliche Spielerschaft zeigt die Tendenz, den Einstieg ins Spielgeschehen durch das Lesen der Geschichte zu festigen.

Für die Entwickler von Computerspielen stellt die emotionale Einbindung der Spielerschaft eine große Herausforderung dar. Dies resultiert aus der enormen Komplexität der Thematik. Hintergrundgeschichte, Verhaltensweisen und Charakterzüge der Spielfiguren sowie die Einbettung des Anwenders in das Geschehen spielen eine zentrale Rolle. Gelingt die emotionale Einbindung, kann das allgemeine Spielerlebnis beider Geschlechter optimiert werden.

Kognition

Die Fragestellungen zur Kognition im Spielgeschehen bringen nach 2.4.5 weitestgehend erwartungskonforme Ergebnisse hervor. Spielerinnen tendieren dazu, das Einsammeln der Items zu favorisieren, finden aber trotz des direkten Konflikts Gefallen am zielgerichteten Schießen. Wenn

man berücksichtigt, dass der direkte Konflikt die Aufgabe für die weibliche Spielerschaft eher negativ belastet, besteht Grund zur Annahme, dass der generelle Vorgang des Schießens weiblichen Präferenzen nicht widerspreche. Es könnten beispielsweise andere Zielobjekte - leblose Gegenstände - herangezogen werden, um das Spielerlebnis weiblichen Vorlieben anzupassen.

Darüber hinaus kann festgehalten werden, dass tendenziell bei weiblichen Spielern eher Hektik während des Spiels aufkommt. Der Erwartung entsprechend kommt in der Applikation *Feelou* weniger Hektik auf, da es generell langsamer konzipiert ist. Die entsprechende These [3], 2.4.5 kann gefestigt werden.

Die eindeutige Mehrheit beider Geschlechter bewertet die ruhigen Phasen des Spiels *Feelou* als angenehm. Es handelt sich nicht um eine geschlechtsspezifische Haltung.

Abschließend kann festgehalten werden, dass die zeitliche Kontrolle sowie das Vermeiden aufkommender Hektik in die Konzeption eines Spiels einfließen sollten. Der generelle Aspekt des schnellen, zielgerichteten Schießens richtet sich nicht gegen weibliche Präferenzen. Der negative Einfluss durch direkte, kämpferische Auseinandersetzungen mit anderen Spielfiguren sollte jedoch berücksichtigt werden.

Avatar

Die Ergebnisse zeigen eine eindeutige Präferenz der Frauen, das Spielgeschehen mit der Figur *Jiny* zu bestreiten. Neben den Charakterzügen wird das Geschlecht der Spielfiguren als zentrales Element betrachtet.

Tendenziell besteht bei weiblichen Probanden kein Interesse, eine männliche Figur heranzuziehen. Die in 2.4.6 aufgestellten geschlechtsspezifischen Erkenntnisse können durch dieses Ergebnis gefestigt werden.

Neben den weiblichen Präferenzen ist anzumerken, dass die männliche Spielerschaft dem Geschlecht der Spielfigur deutlich weniger Bedeutung zukommen lässt.

Demnach sollten Computerspiele weibliche Hauptfiguren bereitstellen, die eine möglichst gute Identifikation ermöglichen. Dies kann beispielsweise durch das nutzerdefinierte Gestalten der Figur erreicht

werden. Im Idealfall können beliebige Charakterzüge und das Erscheinungsbild vom Spieler entwickelt werden.

Puzzle

Die Ergebnisse belegen, dass sowohl Frauen, als auch Männer die Puzzlesequenzen mit einer deutlichen Mehrheit befürworten. Generell sollten solche Zwischensequenzen einen festen Platz in der Konzeption von Computerspielen einnehmen. Beide Geschlechter finden Gefallen an diesem Element. In reaktionsschnellen Spielen könnte somit die Gefahr der Hektik reduziert werden. Ein direkter Zusammenhang zwischen Puzzle und Spielgeschehen sollte hergestellt werden. Das Rätsel fungiert ergänzend, es soll keine separate Einheit bilden.

Puzzle ermöglichen eine Intensivierung des Spielerlebnisses beider Geschlechter.

5.4 Abschließendes Fazit

Zusammenfassend kann festgehalten werden, dass das im Rahmen dieser Arbeit entwickelte Testszenario zur Analyse weiblicher Präferenzen bei Computerspielen zahlreiche Thesen festigen konnte.

Zum allgemeinen weiblichen Computerspielverhalten ist anzumerken, dass es sich meist um kurzweilige Spielsequenzen im Alltag handelt. Das Spielen am Computer ist eines von vielen Hobbies, der soziale Kontakt und die Freizeitgestaltung werden nur sehr vereinzelt beeinträchtigt. Darüber hinaus hat die Mehrzahl weiblicher Spieler kein favorisiertes Genre.

Auf Grund der komplexen Inhalte heutiger Computerspiele wächst die Anzahl der Aspekte, die heranzuziehen sind, um Vorlieben herauszuarbeiten. Diese Arbeit beschränkt sich auf die nach bisherigen Erkenntnissen zentralen Faktoren der Thematik.

Mit Hilfe des entwickelten Testszenarios konnte eine Vielzahl betrachteter Thesen gefestigt werden. Deutlich geschlechtsspezifische Vorlieben stellten sich beispielsweise in Bezug auf den Konflikt und die Auswahl der Spielfigur heraus. Andere Aspekte wie die Puzzle-Sequenzen stießen bei beiden Geschlechtern auf große Zustimmung.

Die allgemeinen Erkenntnisse dieses Forschungssektors stellen für die Entwickler von Computerspielen ein großes Potenzial bereit. Werden relevante Spielelemente geschickt kombiniert, kann die Zielgruppe erweitert werden.

Die Idealvorstellung ist sicherlich nicht, dass es zukünftig sogenannte Frauen- und Männerspiele gibt. Vielmehr sollten die Entwickler den Bedürfnissen beider Geschlechter gerecht werden. Durch zahlreiche Optionen und Varianten während des Spielgeschehens kann dies realisiert werden. So sollte es beispielsweise möglich sein, einen Konflikt auf die persönlich bevorzugte Art und Weise zu lösen. Wenn der Spieler eine kämpferische Auseinandersetzung wünscht, sollte sie in gleichem Maße bereitgestellt werden, wie eine diplomatische.

A Screenshots



Abbildung 25: Feelou, Titel



Abbildung 26: Feelou, Spielverlauf



Abbildung 27: Feelou, Item gefangen

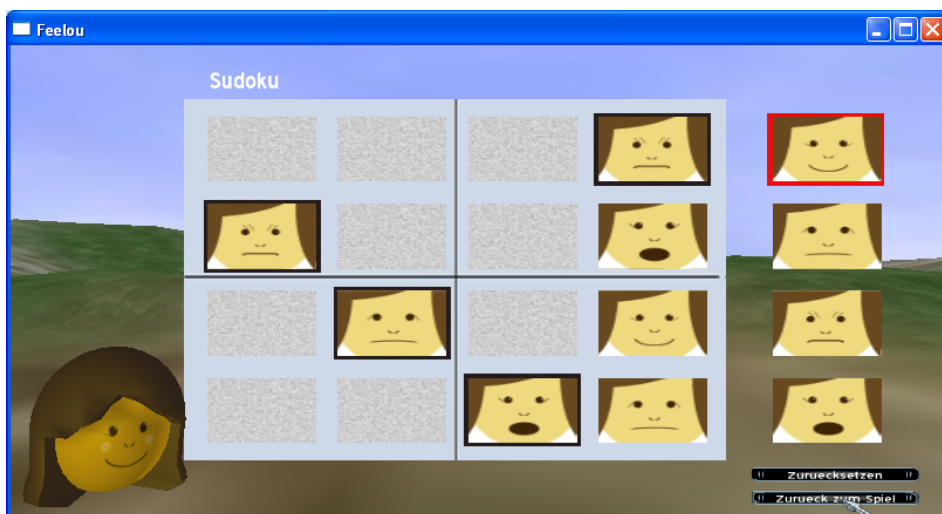


Abbildung 28: Feelou, Sudoku

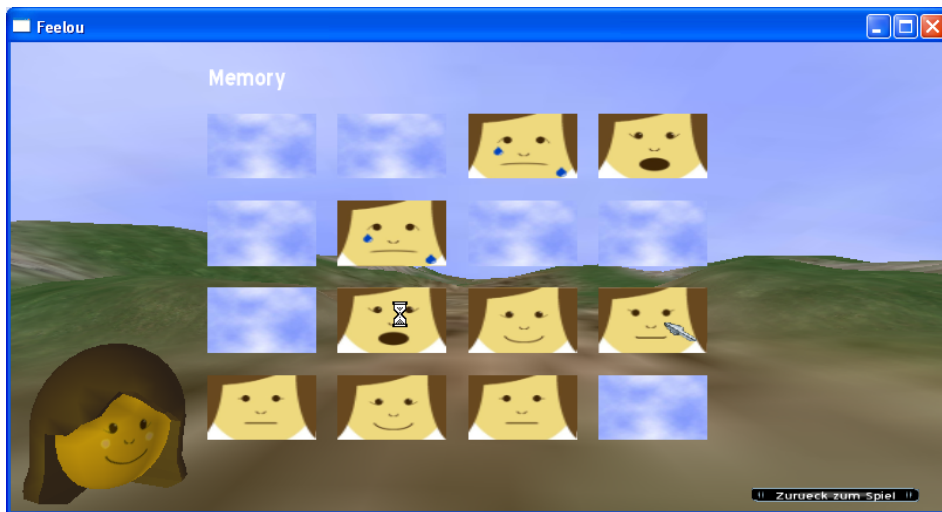


Abbildung 29: Feelou, Memory

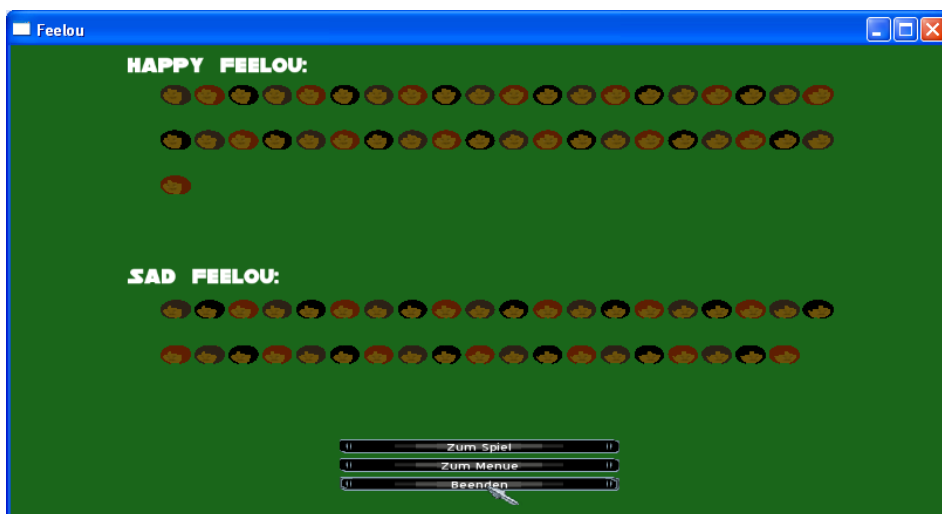


Abbildung 30: Feelou, Resultat



Abbildung 31: Rellos, Titel



Abbildung 32: Rellos, Spielverlauf

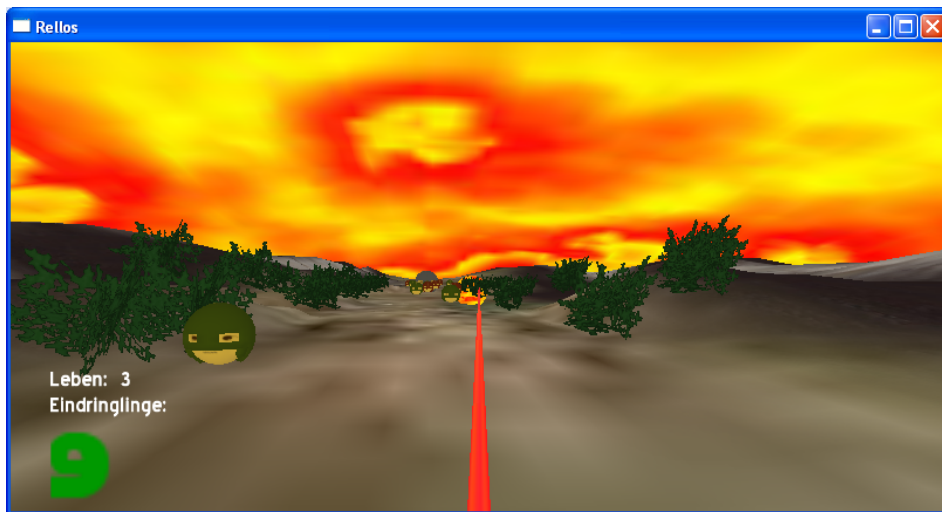


Abbildung 33: Rellos, Treffer



Abbildung 34: Rellos, Grauer Feind

Abbildungsverzeichnis

1	Verteilung der Altersgruppen	8
2	Kategorien, Spielplatz Deutschland	9
3	Verteilung weiblicher Spieler	10
4	Feelou Jiny	33
5	Rello Jim	34
6	Sudoku	36
7	Memory	37
8	Allgemeine Präferenz	71
9	Anleitung	72
10	Konflikt	73
11	Direkte Auseinandersetzung	74
12	GameOver	75
13	Bestrafung	76
14	Highscore	77
15	Resultat	78
16	Emotionale Bindung (allgemein)	79
17	Emotionale Bindung zum Spiel	80
18	Geschichte	81
19	Aufgabe	82
20	Hektik (<i>Rellos</i>)	83
21	Hektik (<i>Feelou</i>)	84
22	Langsame Phasen	85
23	Spielfigur	87
24	Puzzle	88
25	Feelou, Titel	95
26	Feelou, Spielverlauf	95
27	Feelou, Item gefangen	96
28	Feelou, Sudoku	96
29	Feelou, Memory	97
30	Feelou, Resultat	97
31	Rellos, Titel	98
32	Rellos, Spielverlauf	98
33	Rellos, Treffer	99
34	Rellos, Grauer Feind	99

Literatur

- [1] Die Spielerin, EA Das Magazin, 2006,
<http://www.electronic-arts.de>.
- [2] Spielplatz Deutschland, Typographie der Computer- und
Videospiele, Electronic Arts, Jung v. Matt, GEE, 2006,
<http://www.spielplatz-deutschland.de>.
- [3] Ray, S., R., Gender Inclusive Game Design, Charles River Media,
2003.
- [4] Cassell, J., Jenkins, H., From Barbie to Mortal Kombat, Gender and
Computer Games, The MIT-Press, 2000.
- [5] Subrahmanyam, K., Greenfield, P.M., Computer Games for Girls:
What makes them Play? , From Barbie to Mortal Kombat, 2000.
- [6] Kafai, Y.B., Video Games Designed by Girls and Boys: Variability
and Consistency of Gender Differences, 1993.
- [7] Yatim, M.H.M., Nacke L., Masuch, M., Improving Game Design by
Understanding the Gender Differences, AG Computerspiele,
Otto-von-Guericke Universität Magdeburg, 2006.
- [8] Gorriz, C. M., Medina C., Engaging girls with computers through
software games, Communications of the ACM, 2000.
- [9] Richard, B. Sheroes, Genderspiele im virtuellen Raum, Transcript,
Cultural Studies, 2004.
- [10] Chu, K., Egidio, R., Mishra, P., Graves-Wolf, L., Do Girls Prefer games
Designed by Girls? National Science Foundation New York, 2005.
- [11] Hartmann, T., and Klimmt, C. Gender and computer games:
Exploring females' dislikes. Journal of Computer-Mediated
Communication, 2006,
<http://jcmc.indiana.edu/vol11/issue4/hartmann.html>.
- [12] Chu, K., Gender Reactions to Games for Learning among fifth and
eighth grades, Department of Telecommunication, Information
Studies and Media

- [13] Hartmann, O., Knörzer, O., O'Brien, K., Geschlechterspezifische Vorlieben in Computer-und Videospiele im Hinblick auf soziale Interaktion und Alltagsbezug, Universität Ulm, 2005.
- [14] Bates, J., Girls and Video Games, Gamasutra, 2004.
- [15] Pajinov, A., Personal Interview with Sheri Graner Ray, 2002.
- [16] Kim, S., E-Mail-Interview with Sheri Graner Ray, 2002.
- [17] Brikenbiehl, V.F., Jungen und Mädchen: wie sie lernen, Knaur, 2005.
- [18] Kimura, D., Sex and Cognition, The MIT Press, 2000.
- [19] Miller, Girls' Preferences in Software Design: Insights from a Focus Group, Interpersonal Computing and Technology, an Electronic Journal for the 21th Century, April 1996.
- [20] Ogre3D, Graphics Engine. <http://www.ogre3d.org>
- [21] Junker, G., Pro Ogre3D Programming, Apress, 2006.
- [22] Ankh. <http://www.ankh-game.com/>
- [23] CEGUI System
http://www.cegui.org.uk/wiki/index.php/Main_Page
- [24] Game State Manager.
<http://gamedevgeek.com/tutorials/managing-game-states-in-c/>
- [25] Managing Game States in C++.
http://www.ogre3d.org/wiki/index.php/Managing_Game_States_with_OGRE