

Virtual Reality

Interaktion im Bereich der Innenarchitektur

Diplomarbeit

Ausgeführt zum Zweck der Erlangung des akademischen Grades
Dipl.-Ing. für technisch-wissenschaftliche Berufe

am Masterstudiengang Digitale Medientechnologien an der
Fachhochschule St. Pölten, **Masterklasse Postproduktion**

von:

Arian Jalaeefar, BSc

dm151513

Betreuer und Erstbegutachter: FH-Prof. Dipl.-Ing. (FH) Mario Zeller

Zweitbegutachter: FH-Prof. Dipl.-Ing. Dr. Peter Judmaier

Wien, 18.10.2018

Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

- ich dieses Thema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der vom Begutachter bzw. der Begutachterin beurteilten Arbeit überein.

.....

Ort, Datum

.....

Unterschrift

Danksagung

An dieser Stelle möchte ich mich bei meinem Betreuer FH-Prof. Dipl.-Ing. (FH) Mario Zeller für seine Geduld bedanken. Weiters möchte ich mich bei Milan E. Ammel bedanken, der mir während der Entstehung dieser Arbeit zur Seite gestanden und mir geholfen hat.

Ganz besonders möchte ich mich bei Simon Laburda bedanken. Simon hat mit seiner Unterstützung und Erfahrung einen erheblichen Beitrag zu dieser Arbeit geleistet.

Nicht zuletzt gilt mein Dank an meiner Familie (Julia und Kamran Jalaeefer). Ohne ihre Liebe und Vertrauen wäre das gesamte Studium nicht möglich gewesen.

Kurzfassung

Die vorliegende Arbeit kann als ein Nachschlagewerk gesehen werden, welches mit Hilfe von praxisnahen Beispielen versucht, Theorien vorzustellen, die als Basiswissen der Entwicklung interaktiver Konzepte im Bereich der Innenarchitektur notwendig sind.

Sie versucht diese Informationen in drei inhaltliche Bereiche zu unterteilen. Es werden zuerst Konzepte vorgestellt, welche entwickelt wurden, um Probleme der Interaktion in einer virtuellen Welt zu lösen. Dabei handelt es sich beispielsweise um Konzepte der Lokomotion, wie Redirected Walking oder die Erstellung relevanter Inhalte, welche eine Umgebung für den User glaubwürdig und immersiv gestalten lassen.

Dass diese Probleme existieren und dass es keine endgültige Lösung dafür geben kann, ist nicht notwendigerweise eine technische Herausforderung. An vielen Stellen liegen diese Probleme in der menschlichen Wahrnehmung, weshalb sich diese Arbeit ebenfalls intensiver mit diesem Thema auseinandersetzen wird. Es gilt zu verstehen, warum Motion Sickness entsteht oder wie bzw. aufgrund welcher Erfahrungen, ein User in einer virtuellen Umgebung ergründet, mit welchen Objekten und auf welche Art und Weise er in Kontakt treten kann.

Der dritte Teil schildert schließlich eine detaillierte Herangehensweise eines Usertests, welcher alle bis zu dem Zeitpunkt erlernten Inhalte und Konzepte zu integrieren versucht. Dieser Usertest erzielt einerseits einen möglichst genauen Beleg für die Hypothese dieser Arbeit, ob es tatsächlich am sinnvollsten wäre, sich bei dem Entwurf von Interaktionen in einer virtuellen Welt nach der Wirklichkeit zu richten und versucht andererseits die technischen Überlegungen bei der Umsetzung dieser Ideen darzulegen und anschließend zu eruieren.

Die Auswertung des Usertests liefert schließlich die Information, dass es sich bei den Methoden und Ideen um keine allgemeingültigen Konzepte handeln kann. Es wird herausgestellt, dass die Umsetzung des Wissens bei jedem Projekt, abhängig von Inhalt und anderen Komponenten, variieren kann.

Zum Schluss wird ein Projekt namens HOHO vorgestellt, eine Visualisierung eines Hotels. Dieses Projekt bietet eine gute Möglichkeit, die in der Arbeit vorgestellten Ansätze anhand einer professionellen Umsetzung zu analysieren.

Abstract

The present master's thesis can be considered as a reference work that uses practical examples to present the theoretical knowledge necessary for the development of interactive interior design concepts.

This thesis tries to divide the information into three areas. At first it introduces concepts that have been developed to solve problems of interaction in a virtual world. These are, for example, concepts of locomotion, such as redirected walking, as well as the creation of the relevant content that makes an environment feel realistic and immersive for the user.

That these problems exist and that there can be no final solution is not necessarily a technical challenge. In many places, problems occur due to human perception. This is why this thesis will deal more intensively with this topic. For example, it is all about understanding why or how motion sickness occurs and based on which past experiences a user figures out which virtual objects can be interacted with and in which ways.

The third part of this thesis describes a detailed approach of a user test, which attempts to integrate all the contents and concepts learned up to that point. On one hand, the user test provides the most accurate possible proof of the hypothesis and on the other hand tries to explain the technical considerations in the implementation of these ideas.

However, the evaluation of user tests showed that the methods and ideas can not be universally valid concepts. It is emphasized that the implementation of the gained knowledge in each project can differ depending on its content and other components.

And finally a project called HOHO is presented. HOHO is a visualization of a hotel. This project provides a good opportunity to analyze the approaches presented in this thesis on the basis of a professional implementation.

Inhaltsverzeichnis

Ehrenwörtliche Erklärung	II
Danksagung	III
Kurzfassung	IV
Abstract	V
Inhaltsverzeichnis	VI
1 Einleitung	1
1.2 Problemstellung und Thematik	3
1.3 Motivation	4
1.4 Ziele, Forschungsfrage und Hypothese	5
1.5 Methodik und Gliederung	7
2 Virtual Reality	9
2.1 Die Geschichte von Virtual Reality	9
2.2 Einsatzbereiche von Virtual Reality	12
2.3 Architekturvisualisierung	13
3 Lokomotion und Navigation	16
3.1 Motion Sickness	16
3.1.1 Vektion	17
3.2 Based on Physical Body Movement	19
3.2.1 Pressure-Based Board	20
3.2.2 Cycling Systems	20
3.2.3 Walk-in-Place	20
3.3 Redirected Walking	23
3.3.1 Gains	25
3.3.2 Impossible Space	30
3.4 Teleportation	33
Artificial Locomotion	36
3.5 Andere Methoden	36
3.6 Input Devices	38
3.6.1 HTC VIVE und die Funktionsweise	39
4 Interaktion mit der Umgebung	43
4.1 Human Centered Interaction	43
4.2 Norman's Prinzipien	45
4.2.1 Affordanz	45
4.2.2 Signifiers	45
4.2.3 Constraints	46

4.2.4	Feedback	46
4.2.5	Mapping	48
5	Die Virtuelle Welt / Content	50
5.1.1	Conceptual Integrity	50
5.1.2	Die Gestalttherorie	51
5.2	Das menschliche Denken	53
5.3	Die virtuelle Umgebung	54
5.3.1	DevTon Gefängnis	56
6	Umsetzung / Usertest	65
6.1	Vorbereitung des Tests	66
6.1.1	Software und Hardware	66
6.1.2	Contenterstellung	67
6.1.3	Zieldefinition	70
6.1.4	Eingabemöglichkeiten	73
6.1.5	Teleportation	75
6.1.6	Skalierung	78
6.1.7	Probanden	80
6.2	Contenterstellung	81
6.2.1	Küche	81
6.2.2	Schlafzimmer	84
6.2.3	Gefängnis	89
6.2.4	Garage	91
6.3	Umsetzung	94
6.3.1	Küche	94
6.3.2	Schlafzimmer	103
6.3.3	Gefängnis	108
6.3.4	Garage	113
6.4	Punktevergabe	117
6.5	IntroMap	119
6.6	Durchführung des Usertests	120
6.7	Tabellarische Auswertung	121
6.8	Fazit des Usertests	124
6.8.1	Allgemein	124
6.8.2	Küche	126
6.8.3	Schlafzimmer	127
6.8.4	Gefängnis	129
6.8.5	Garage	130
7	HOHO	132
8	Fazit	138
	Literaturverzeichnis	140

Abbildungsverzeichnis	152
Anhang	160
A. Interview mit Marcel Nürnberg SQUARE BYTES Interactive Media	160
B. Interview mit Axel Dietrich, Vrisch	168
C. Interview mit Christian Friesenegger, 3Motion Virtual GmbH	173

.....

1 Einleitung

Das Thema dieser Arbeit lässt sich als „Interaktionsmethoden in der virtuellen Realität“ zusammenfassen. Bei dem englischen Begriff *Virtual* (auf Deutsch virtuell) handelt es sich um ein Adjektiv um Objekte zu beschreiben, welche in der Wirklichkeit nicht vorhanden sind, jedoch echt zu sein scheinen. (Sherman & Craig, 2002) Unter *Reality* (auf Deutsch Realität), hier auch mit dem Begriff *Wirklichkeit* gleichgesetzt, versteht man wiederum Erscheinungen und Phänomene als wahr und existent in der Art wie sie erscheinen. Aus philosophischer Sicht kann diese Erklärung vage klingen. Leider erlaubt die Dimension dieser Arbeit keine tiefere Analyse über die Interpretation jener Begriffe, weshalb sie hier lediglich zum Zweck der Verständlichkeit beschrieben werden.

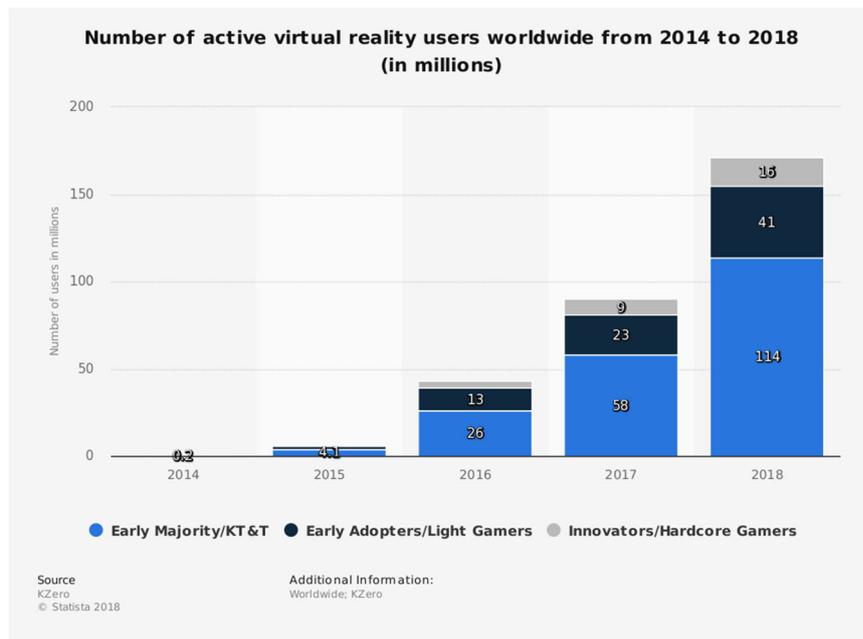


Abbildung 1: Anzahl der aktiven VR User, (Statista, 2014)

Wirft man einen kurzen Blick auf obiges Diagramm, offenbart sich die rasante Integration von Virtual Reality im Consumerbereich. Diese Intensität an Aufmerksamkeit, die einem Medium in so einer kurzen Zeit geschenkt wird, deckt

1 Einleitung

viele Fragen auf. Diese Fragen können einerseits, aus analytischer Perspektive, Vorteile und Nachteile des Mediums als auch den Einfluss auf die menschliche Wahrnehmung unter die Lupe nehmen. Andererseits können sie den Fokus auf dessen Fähigkeiten legen. Die vorliegende Masterarbeit wurde unter diesem Fokus verfasst. Denn unverzichtbar und unübersehbar ist in allen Fällen die Verbreitung von Virtual Reality in Bereichen der Medizin, Technologie, Industrie, Werbung, Film und interaktiven Medien.

VR-Healthcare-Simulationssysteme sind auf dem besten Weg Medizinstudenten eine praxisnahe Ausbildung zu ermöglichen ohne die Patientensicherheit zu gefährden. Dabei handelt es sich um eine neugewonnene Qualität im gesundheitlichen Bereich. (Pinker, 2018) Auch Fertigungsprozesse, Anlagen und industrielle Konstruktionen werden mittlerweile in einer virtuellen Welt zwecks Korrektur und Nachbearbeitung oder Präsentation (vor)visualisiert. Simulationen helfen bei den unterschiedlichsten Ausbildungen und eignen sich beispielsweise auch für handwerkliche Berufe. (Mohammady, 2018)

Die Liste von Einsatzbereichen der Virtual Reality ist offensichtlich lang. Aber egal in welchem Bereich und zu welchem Zweck, benötigt VR konzeptionellen und technischen Aufwand um den erwünschten Effekt zu erreichen. Diesen Konzepten und Ideen stehen Einschränkungen und Schwierigkeiten im Weg, die gelöst bzw. kompensiert werden müssen. Diese Probleme führen auf verschiedene Hintergründe zurück. Trotzdem befindet sich Virtual Reality auf dem besten Weg benutzerfreundlicher und günstiger zu werden. In wenigen Jahren verlieren die technischen Probleme, welche den Entwicklern und Nutzern heute im Weg stehen, an Relevanz, weshalb sich diese Arbeit gegen die Behandlung von solchen Herausforderungen entschieden hat.

Denn abgesehen von den technischen Herausforderungen, sind Entwickler und User von Virtual Reality-Anwendungen mit Problemen konfrontiert, die auf den psychologischen Wahrnehmungen der Rezipienten basieren. Diese benötigen andere Formen der Analyse und müssen somit anders behandelt werden. Daher befasst sich diese Arbeit auch mit Konzepten, die mit einem Fokus auf jene Form der Analyse erstellt sind.

1.2 Problemstellung und Thematik

Der Begriff *Interaktion* beschreibt in einer virtuellen Welt im Bereich der Innenarchitektur die Kommunikation mit interagierbaren Objekten, die sich in einer virtuellen Umgebung befinden. In der Praxis benötigt dieser Begriff jedoch genauere Erklärung. Was bedeutet Interaktion und wie findet man als User heraus, wie man mit den Objekten interagieren kann?

Spricht man von einer Visualisierung in VR, in der sich die Interaktion auf ein Beobachten der Umgebung beschränkt, müssen sich die Entwickler des Projekts mit keinen weiteren Aspekten der Interaktion auseinandersetzen. Die Ursache der Attraktivität dieses Mediums liegt jedoch nicht nur in der Visualisierung, sondern insbesondere in seinen Kommunikationsmöglichkeiten. Dabei versetzt man den User in eine Welt, in der versucht wird, Wissen, Gefühle, Absichten und Bewegungen eines Users zu empfangen und darauf reagieren zu können. All das, um ihm das Gefühl der Präsenz und Immersion in dieser neuen Welt zu geben.

Dieses Ziel wird von der Anwendung allerdings erst erreicht, wenn ein Zusammenspiel von allen beteiligten Elementen erfolgt. Idee und Ziel der Anwendung müssen daher mit einem Fokus auf die Möglichkeiten und Einschränkungen des Users gewählt werden. Alle Eingabemöglichkeiten müssen soweit wie möglich auf die Wahrnehmung des Users angepasst sein. Es ist nicht zielführend und wirkt nicht immersiv, einen User in einem virtuellen Wald herumlaufen zu lassen, ohne die physischen Gegebenheiten zu berücksichtigen. Denn man darf als Entwickler nicht vergessen, dass unerfahrene VR-User während Fahr- und Flugsimulationen oft gegen Simulation Sickness zu kämpfen haben. Diese und weitere Beispiele weisen auf konzeptionelle und technische Gegebenheiten hin, welche in dieser Arbeit erläutert werden.

Das größte Problem der Virtual Reality ist jedoch die **menschliche Wahrnehmung**. VR-Anwendungen müssen einerseits so konzipiert werden, dass sie von einer Form der Wahrnehmung akzeptiert werden, welche auf Erfahrungen der realen Welt abgestimmt ist. Andererseits müssen sie diese Wahrnehmung so weit manipulieren, dass die User in diese Welt eintauchen können und sie als immersiv empfinden.

Diese Arbeit wurde unter dem Fokus der Anwendungsentwicklungen verfasst, die für unerfahrene VR-User konzipiert sind. Somit stehen die Integration der Gewohnheiten und das Wissen dieser User in einer virtuellen Realität im Zentrum. Daher lässt sich die Problematik aus der Hypothese dieser Arbeit herleiten.

Hypothese: Es ist immer besser sich in der Interaktion mit der Umgebung und dem Verhalten der Objekte nach der Wirklichkeit zu richten. Diese Hypothese basiert auf folgender Aussage:

“If a user can interact using natural skills, then the application can take advantage of the fact that the user already has a great deal of knowledge about the world.“
(LaViola, Bowman, Kruijff, Poupyrev, & McMahan, 2017, S. 4)

„Es ist also für die Applikation in VR von Vorteil, wenn der Benutzer auf die gleiche Art und Weise mit der Umgebung interagieren kann, wie er mit der realen Welt interagiert. Es ist somit davon auszugehen, dass der Benutzer mit einem großen Vorwissen an Interaktionsmöglichkeiten in die virtuelle Welt einsteigt.“ (LaViola, Bowman, Kruijff, Poupyrev, & McMahan, 2017, S. 4)

Dies bedeutet, dass eine schnellerlernbare und leicht integrierbare Interaktion nur dann möglich ist, wenn diese sich so weit wie möglich nach der Realität richtet. Eine exakte Reproduktion der Realität ist jedoch in einer VR-Anwendung weder möglich und abhängig vom Kontext (nicht) erwünscht. Das daraus resultierende **Problem** lässt sich wiederum wie folgt ausdrücken: Die Interaktion kann nur dann schnell verstanden werden, wenn sie so realistisch wie möglich umgesetzt ist. Die virtuellen Umgebungen sind jedoch nicht immer realitätsnah gebildet.

Das Hauptthema ist somit die Untersuchung von Konzepten, welche diese Problematik zu lösen versuchen. Zum einen soll ein Überblick über diese Methoden geschaffen werden, zum anderen müssen sie in der Umsetzung genauer analysiert werden. Dies geschieht im praktischen Teil dieser Arbeit, welcher wiederum den Usertest beinhaltet.

1.3 Motivation

Ausgangspunkt dieser Arbeit war ein zweimonatiges Praktikum bei der Firma Vrisch, in der ein direkter Einstieg und eine Auseinandersetzung mit der Entwicklung von VR-Anwendungen möglich war. Entscheidend für das gewählte Thema war die starke Verbindung zwischen den psychologischen Aspekten der Interaktion und ihrer technischen Umsetzung.

Die vorliegende Masterarbeit dient als Nachweis eines abgeschlossenen Studiums der Digitalen Medientechnologie. Die Absolventen dieses Studiums besitzen Kenntnisse im Bereich der digitalen Technologien, welche für die heutige Welt eine

wichtige Rolle spielen. Technik soll dem Menschen und seiner Lebensqualität dienen. Technisches Wissen und Virtual Reality, als ein techniklastiges Medium, sollte soweit erweitert und perfektioniert werden, dass Forschung, Bildung und Medizin mit dessen Hilfe der Menschheit behilflich sein können. Aus persönlicher Sicht ist es die Interaktion und ihr direktes Feedback auf die Absichten des Users, welche VR so speziell und faszinierend machen. VR besitzt das Potenzial für viele Probleme in der nahen Zukunft aussichtsreiche Lösungsansätze zu bieten.

Wenn man von Interaktion spricht, greift man gleichzeitig zu mehreren relevanten Punkten, die beachtet werden müssen. In diesem Prozess ist Hardware integriert, mit der man sich als Entwickler vertraut machen muss. Es müssen die erwünschten Interaktionen konzipiert werden, wofür ein fundiertes psychologisches Wissen erwünscht ist, welches letzten Endes mit den Ansätzen des Designs als auch den mentalen Modellen des Users kompatibel sein sollten. Die Programmierung jener Konzepte stellt den nächsten Schritt in diesem Prozess dar. Dieser Teil folgt eigenen Denkmustern und Algorithmen. Zum Schluss erfolgt die Visualisierung des Ganzen, welche mit entsprechend weiteren Herausforderungen verbunden ist.

1.4 Ziele, Forschungsfrage und Hypothese

Ziel dieser Arbeit besteht darin, für die oben skizzierte Hypothese einen plausiblen Nachweis zu erhalten. Daher beginnt die Arbeit mit einem Überblick von verständlichen bzw. greifbaren Interaktionsmöglichkeiten in VR, insbesondere für unerfahrene User. Diese Thematik lässt sich in vier Kategorien unterteilen:

Konzeption: Es kann sich dabei beispielsweise um Einschulungen, Visualisierungen und Demonstrationen handeln. Hierbei spielt die Zielgruppe eine wichtige Rolle. Zusätzlich muss beachtet werden, mit welchen technischen Vorkenntnissen und mit welchen Fähigkeiten diese Gruppe auf das Medium reagiert.

Navigation: Die Möglichkeit(en), sich in einem Raum zu bewegen. (Carbotte, 2016)

Interaktion: Das Auswählen, Manipulieren und Ändern von Objekten, welche sich in einem Raum befinden. Hierbei geht es einerseits um die Art und Weise der Handlung und andererseits um jene Objekte, die man in einem Raum bewegen kann und darf. Beispielsweise ist das Schieben einer Wand ebenfalls auf die Thematik zurückzuführen, wie Objekte zu der Aktion des Users Feedback geben. (Jerald, 2015a, S. 328)

1 Einleitung

Visualisierung: Die Darstellung und Fertigstellung von der gesamten Anwendung. In einer virtuellen Welt spielen nicht nur die 3D-Visualisierungen und die Ästhetik eine Rolle. Damit sind Methoden gemeint, die dem User seine Möglichkeiten der Interaktion zeigen. Sie werden unter anderem mit Hilfe von Interfaces, Labels und Images visualisiert. Der User ist in einer virtuellen Welt auf sich gestellt. Somit müssen Informationen und Hilfsmittel, die ihm von der Anwendungsseite zur Verfügung gestellt werden, deutlich und verständlich sein.

Die zentralen **Forschungsfragen**, welche sich aus den beschriebenen Thematiken ergeben, sind:

Welche Methoden der Interaktion mit der Innenarchitektur in VR haben sich bis jetzt am besten bewährt und warum sind sie sinnvoller einzusetzen als andere Methoden? Wenn man beispielsweise das Bewegen von Objekten (Möbeln) in VR analysiert, bemerkt man, dass diese Objekte oft geschoben und nicht gehoben werden. Nachdem man sich in VR der Realität annähern möchte, stellt sich die Frage nach den Hintergründen solcher Entscheidungen.

Bei einer 2D-Interface Software weisen uns beispielsweise Click Buttons darauf hin, dass wir sie drücken können. **Wie weist man den Benutzer darauf hin, welche Objekte auf eine Interaktion reagieren und welche nicht?** Was sind die Ansätze zur Lösung von Problemen ähnlicher Art?

Ist es lohnenswert, sich über die physikalischen, psychologischen und ergonomischen Aspekte Gedanken zu machen oder sind die Entscheidungen in den meisten Fällen eine Auswahl von bestehenden Möglichkeiten, die sich bereits bewährt haben oder letztendlich eine „Geschmacksfrage“?

Aus dem gewonnenen Wissen als auch den Recherchen hat sich eine Hypothese ergeben, welche basierend auf obigen Aussagen formuliert wurde:

Hypothese: Es ist immer besser, sich bei der Interaktion mit der Umgebung und dem Verhalten der Objekte nach der Wirklichkeit zu richten.

1.5 Methodik und Gliederung

Diese Arbeit erzielt nicht nur eine Zusammenfassung von bereits existierendem Wissen. Ihr Ziel besteht schlicht und ergreifend darin, einen praxisnahen Einstieg in die Entwicklung der Interaktionsmethoden in VR-Applikationen zu ermöglichen.

Die ersten drei Kapitel beinhalten Darstellungen und Herleitungen der notwendigen Fachbegriffe und Definitionen. Nach einem kurzen Einstieg in die Geschichte der Virtual Reality werden unterschiedliche Einsatzbereiche vorgestellt. Infolgedessen werden die Methoden der Fortbewegung in einer virtuellen Umgebung, inklusive ihrer Vor- und Nachteile, erläutert.

Das fünfte Kapitel beschäftigt sich mit der Erstellung interaktiver Inhalte. Diese Begriffe mussten jedoch zuerst erklärt werden, was wiederum durch das vierte Kapitel erfolgt. Diese zwei Kapitel richten sich intensiver nach dem eigentlichen Thema, der Interaktion. Dort wird dargestellt was Interaktion im Allgemeinen und speziell in einer virtuellen Welt bedeutet. Zusätzlich erhält der Leser einen Überblick über die theoretischen Grundlagen, welche er benötigt um eine VR-Anwendung praktisch umsetzen zu können. Orientiert an diesem Ziel wurden die Informationen explizit partiell dargestellt, denn die Erläuterung aller wichtigen Aspekte und Theorien ist im Rahmen einer solchen Arbeit nicht möglich.

Das sechste Kapitel nimmt ein praktisches Projekt in die Hand, einen Usertest, welcher seinen Probanden in vier virtuellen Umgebungen Aufgaben stellt. Dieser Test wurde aus zwei Gründen in der vorliegenden Form durchgeführt: Einerseits sollte die Affordanz der Objekte in VR getestet werden. Dadurch kann auf die Hypothese genauer eingegangen und am Ende des Tests bestätigt bzw. abgelehnt werden. Andererseits sollte der Test als eine Umsetzung des gelernten Wissens gelten. Es wurde im Laufe der Erstellung versucht auf die vorher erwähnten Inhalte einzugehen und sie in die Entwicklung des Tests einzubauen. Das Ergebnis dieses Tests, ein persönliches Fazit und eine grafische Wiedergabe der Werte bilden die abschließenden Teile der Arbeit. Dies erfolgte in der Absicht, dass die dritte Forschungsfrage mittels der gewonnenen Erkenntnisse beantwortet werden kann.

Der Inhalt dieser Arbeit ist das Resultat aus Recherchen von Fachliteratur, dem Internet, des oben erwähnten Usertests sowie drei Interviews mit Experten. Die Interviews wurden so geführt, dass sie auf die Erfahrung der Experten eingehen und praxisnahe Lösungen sowie Ideen eruieren. Abgesehen davon wurden die Forschungsfragen so formuliert, dass die Erfahrung der Experten als eine gute Ergänzung zu den Recherchen dienen konnte. Die Experten wurden so ausgewählt, dass sie möglichst unterschiedliche Zugänge zu dem Thema der Interaktion aufweisen.

1 Einleitung

Für die Transkription wurde eine einfache Form gewählt, Versprecher, Wortdrehungen und Betonungen wurden berücksichtigt. Teile des Gesprächs, die keine Relevanz zu dem Thema beinhalteten, wurden nicht transkribiert.

Aus Gründen der leichteren Lesbarkeit wird in der vorliegenden Masterarbeit die gewohnte männliche Sprachform verwendet. Dies impliziert jedoch keine Benachteiligung des Weiblichen oder anderer Geschlechter, sondern soll im Sinne der sprachlichen Vereinfachung als geschlechtsneutral zu verstehen sein.

2 Virtual Reality

Webster's New Universal Unabridged Dictionary, 1989, definiert den Begriff *virtual* (auf Deutsch virtuell) als „*being in essence or effect, but not in fact.*“ und den Begriff *Reality* (auf Deutsch Realität) als „*the state or quality of being real*“. (Sherman & Craig, 2002, S. 6)

In dieser Arbeit ist mit Virtual Reality (VR) eine computergenerierte Umgebung gemeint, mit welcher der User in Echtzeit interagieren und kommunizieren kann. Kommunikation kann hier ebenfalls auf einer abstrakteren Ebene geschehen, d. h. sie beschreibt nicht nur eine zwischenmenschliche Kommunikation. In anderen Formulierungen sind die Aktionen und Reaktionen, die Ursachen und Wirkungen und das Geben und Nehmen, alles unterschiedliche Formen der Kommunikation.

In den Medien wird VR gerne als ein neues Medium beschrieben. Vielleicht ist die Verbreitung und Nutzung dieses Mediums im heutigen Consumerbereich etwas Neues, jedoch ist das Konzept der virtuellen Realität bis zu den Höhlenmalereien zurückzuverfolgen. Die Erwartungen als auch die Inhalte haben sich geändert, jedoch der Wunsch, Illusionen und Welten zu kreieren, ist immer noch erwünscht und interessant. (Jerald, 2015a, S. 51)

2.1 Die Geschichte von Virtual Reality

David Brewster entwickelte 1851 eine benutzerfreundliche und kleinere Version eines Stereoskops. Es ähnelte dem heutigen **Google Cardboard**.



Abbildung 2: Google Cardboard (links), Brewster's Stereoskop (rechts), (Harding, 2013)

2 Virtual Reality

1890 entwarf Amariah Lake den ersten **Haunted Swing**, welcher für einen Vergnügungspark entwickelt und umgesetzt wurde. Die Idee des Haunted Swing richtet sich nach einer Eigenschaft der menschlichen Wahrnehmung, der Self-Motion (Siehe Kapitel 3.1, *Motion Sickness*). Die Besucher saßen in einer Schaukel, die sich angeblich stark bewegte. In Wahrheit rotierte die Umgebung jedoch um die Besucher herum, während sich die Schaukel kaum von der Stelle bewegte. („The Haunted Swing illusion“, o. J.) Die menschliche Wahrnehmung neigt dazu, Bewegung von üblicherweise statischen Objekte (in diesem Fall der Raum) als eigene Bewegung zu interpretieren, weshalb die Besucher die Schaukel als das bewegende Objekt wahrgenommen haben.

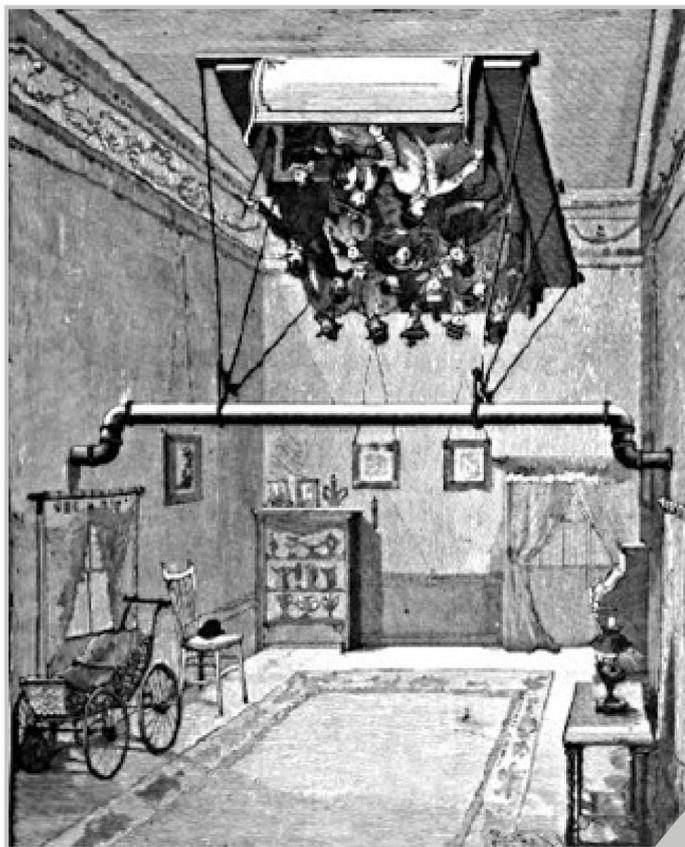


Abbildung 3: The Haunted Swing Illusion, (The Haunted Swing, o. J.)

1928 entwickelte Edwin A. Link den ersten **Flugsimulator**. Dieses Produkt simulierte die Bewegung und Steuerungen eines Flugzeuges. Trotz seiner Erwartung, erlebte Edwin Link mit dem Verkauf seines Produkts an das Militär keinen großen Erfolg, weshalb er seinen Simulator an einen Vergnügungspark verkaufte. Sieben Jahre später bestellte das US Army Air Corps sechs Exemplare dieses Systems und kurz nach Ende des zweiten Weltkriegs wurde der Verkauf von 10.000 Stück dieses Simulators registriert. Sein System fand in den späteren

2 Virtual Reality

Jahren in vielen weiteren Bereichen der Technik, Einschulung und Simulationen seinen Einsatz. (Jerald, 2015a, S. 55)

1960 meldete Morton Heilig ein Patent für den ersten **Head Mounted Display** (HMD) an. Anschließend baute er eine Apparatur die er **Sensorama** nannte. Sensorama wurde 1962 entwickelt, um dem Medium Film eine höhere Immersion zu verschaffen und ist somit das erste VR-Gerät der Welt. (DeMichele, 2016)

1961 entwickelte die Philco Corporation das erste Head Tracking System für HMD, das **Headsight**. Das System Headsight kann als erstes HMD mit Head Tracking System angesehen werden. Diese Brille beinhaltete zwei Screens, jeweils für jedes Auge und ein magnetisches Motion-Tracking System, welches mit einer externen Kamera verbunden war. Die Kopfbewegungen des Benutzers bewegten ebenfalls die externe Kamera und erlaubten dem User, auf natürliche Art und Weise, eine andere Umgebung zu besichtigen. (Virtual Reality Society, 2017)

Im Jahre 1965 wurde mit dem **Sword of Damocles** an der Harvard University of Utah durch Ivan Sutherland der erste Head Mounted Display mit computer-generierten Bildern entwickelt. (Jerald, 2015a, S. 57).

1985 wurde **VIVED** (Virtual Visual Environment Display) von der NASA als die erste kommerzielle Stereoscopic HMD vorgestellt.

Die Firmen Virtually, Division and Fakespace experimentierten in den 1990er Jahren mit verschiedenen Variationen von Virtual Reality-Projekten und Forschungen, hauptsächlich örtlich fixierten Installationen. Mittlerweile waren schon weitere bekannte Firmen wie Sega und Disney ebenfalls dabei neue Systeme auf den Markt zu bringen.

Bis Ende der 1990er ist eine rasante Weiterentwicklung von Virtual Reality-Systemen zu erkennen. Das Magazin *Wired* behauptete 1993, dass 1/10 der Bevölkerung in den nächsten fünf Jahren mit HMD in dem öffentlichen Verkehr zu sehen sein würden. (Jerald, 2015a, S. 62) Dieser Erfolg und Anstieg änderte seine Geschwindigkeit in der 2000er Jahren. Dieses Jahrzehnt wird als „VR-Winter“ bezeichnet. (Jerald, 2015a, S. 63)

2012 wurde auf der IEEE VR der **Low-cost Field of View To Go** (FOV2GO) in Kalifornien vorgestellt. Dieses Gerät hat den ‚Best Demo Award‘ gewonnen und wurde daraufhin ein Teil des MxR Lab’s Open Source Project, welches als Vorläufer der heutigen HMDs gesehen werden kann. Gleichzeitig nahm sich Palmer Luckey, ein Mitglied dieses Labors, vor, das Produkt nun öffentlich vorzustellen. Luckey begegnete kurz darauf John Carmack, dem CTO von Oculus

VR. Ein Kickstarter für die Entwicklung des **Oculus Rifts** startete kurz danach und eine neue Zeit für Virtual Reality begann. (Jerald, 2015a, S. 63)

Im April 2016 wurde **HTC VIVE**, ebenfalls ein Virtual Reality System von HTC, in Kooperation mit Valve vorgestellt. Valve stellte, abgesehen von ihrer Brille, eine neue Tracking-Technologie namens **Lighthouse** vor.

2.2 Einsatzbereiche von Virtual Reality

Wie bereits erwähnt, findet Virtual Reality in unterschiedlichen Bereichen ihren Einsatz. In diesem Kapitel werden einige Einsatzbereiche vorgestellt, die sehr vielfältig sind und daher nur kurz umschrieben werden können. Ziel ist, dass der vielfältige Nutzen dieses Mediums anhand dieser Beispiele verständlich wird.

In der Medizin bietet Virtual Reality Möglichkeiten, Phobien von Menschen zu behandeln. Dabei gibt es viele Arten der Phobien (Spinnen, Platzangst usw.), die erfolgreich überwunden oder stark reduziert werden können. Patienten werden in der virtuellen Welt Situationen ausgesetzt, in denen sie ohne Gefahr mit ihren Ängsten konfrontiert werden und somit versuchen können, diese unter Kontrolle zu bringen. (Dr. Andrew Rosen, 2016)

Da visuelle Reize in vielen Fällen reale Ängste auslösen, ist der Einsatz von Virtual Reality hilfreich um diese Auslöser zu simulieren. Somit findet VR auch in der klinischen Psychologie und Psychotherapie besondere Anwendung. (Eichenberg & Wolters, 2012) VR kann bei den Demenz- und Traumapatienten ebenfalls eine Anwendung finden. Laut Ärzten der Alzheimer Gesellschaft Deutschland kann mit Hilfe von Virtual Reality versucht werden, Lebensfreude und positive Erinnerungen zu wecken. Dies kann den Fortschritt der Krankheit verlangsamen. (Allgemeine, 2017)

Nach einem Schlaganfall ist die Lähmung einer Hand keine Seltenheit. Jährlich erleiden 20.000 Menschen in Österreich einem Schlaganfall. Oft erschweren die körperlichen Einschränkungen nach einem Schlaganfall den Patienten ihren Alltag. Psii.rehab nutzt eine spielerische Art, mit welcher Patienten selbstständig und mit Hilfe einer VR-Brille und einem EMG-Biofeedbacksystem motorische Fähigkeiten ihrer Hand wiedergewinnen. (Virtual Reality-App, 2017)

Im Militär kann Virtual Reality zu Trainingszwecken eingesetzt werden. Eine südkoreanische Firma hat eine Übung zum Fallschirmspringen virtuell simuliert. Dabei kann der Kandidat seinen eigenen Flug steuern und sich ohne Unfallrisiko an diese Situation gewöhnen. (Biocca, 2017) Abgesehen davon werden Soldaten

2 Virtual Reality

für viele Kriegssituationen vorbereitet: Gebiete, in denen sie später eingesetzt werden, werden mit Hilfe von VR nachgebildet, sodass sie sich einen Überblick verschaffen und ihre Effektivität steigern können. (Biocca, 2017)

In der Industrie als auch der Fertigung von vielen Produkten, bietet Virtual Reality eine gute Möglichkeit, Prozesse und Änderungen im Laufe der Herstellung effizienter zu simulieren. Damit kann man komplexe Vorgänge vereinfacht und verständlich verbildlichen, sodass unerfahrene Nutzer bessere Vorstellungen von dem Ergebnis bekommen.

Als ein Beispiel für den Einsatz von VR in der Berichterstattung bietet sich der 2016 stattgefundene Wahlkampf zwischen Hillary Clinton und Donald Trump an. Die New York Times veröffentlichte damals 360°-Aufnahmen der Kandidaten, in denen sie ihre Position erläutert haben. (Biocca, 2017) Zuschauer konnten dadurch eine so noch nie da gewesene Nähe zu den Kandidaten empfinden, welche andere Medien in den Schatten stellt.

Virtual Reality leidet immer noch unter dem Vorurteil, dass sie von einer Vielzahl an Menschen als eine Erweiterung von Spielekonsolen angesehen wird. Diese Beispiele und viele andere machen jedoch offensichtlich, dass die unterschiedlichen Einsatzbereiche, damit verbundene Vorteile und Forschungsgebiete, dieses Vorurteil überflüssig machen.

2.3 Architekturvisualisierung

Die Umgebung, verstanden als ein Bereich in der unmittelbaren Nähe, in dem sich der Mensch befindet, kann unterschiedlich definiert werden. Innenräume, Landschaften, MRT-Röhren und Ähnliches stellen Umgebungen dar, die man beschreiben und analysieren kann.

Computergenerierte Umgebungen ermöglichen das Gefühl der **Präsenz**, in welcher sich ein Mensch in der Wirklichkeit nicht befinden kann. Ein Beispiel dafür ist das Reproduzieren des Innenkörpers eines Menschen in VR als eine solche Umgebung, in der sich ein Benutzer praktisch seine eigene Arterie und den Blutfluss beobachten kann.

Wichtig ist, dass jede Umgebung eigene Objekte, Eigenschaften sowie Gegebenheiten beinhaltet und diese eine eigene Art der Interaktion beansprucht. Jene Interaktion mit dieser Umgebung kann abhängig sein von den Objekten, die sich in diesem Raum befinden. Wichtig ist ebenfalls die Frage, welche Form von **Interaction Fidelity** die Objekte aufweisen, d.h. die objektive Genauigkeit, mit der

2 Virtual Reality

ein System oder eine Anwendung versucht eine Realität zu reproduzieren. (Nabiyouni, 2015, S. 2)

Diese Arbeit konzentriert sich mit ihrer Thematik auf Interaktion im Bereich der Innenarchitektur. Klarerweise sind Räumlichkeiten im Bereich der Architektur nicht alle gleich. Es gibt sowohl in Entwurf, Konstruktion als auch der Konzipierung von den Interaktionsmethoden offensichtliche Unterschiede zwischen Umgebungen. Jedoch offenbaren sie auch gemeinsame Objekte mit gleicher Art der Interaktion und demzufolge bestimmte Verhaltensmuster der Menschen in der Kommunikation mit den virtuellen Umgebungen, welche man untersuchen und vergleichen kann.

Der Fokus dieser Arbeit liegt jedoch in einem anderen Grund: Wie erwähnt, steht die Abweichung der Interaktionsmethoden mit Objekten von der Wirklichkeit im Mittelpunkt. Somit ist es angebracht, Umgebungen in den Vordergrund zu stellen, die eine große Schnittstelle oder Ähnlichkeit zur Wirklichkeit bieten. Unrealistische Umgebungen erfüllen diese Bedingung nicht.

Dies soll keinesfalls ein Urteil darstellen, das Architekturvisualisierung in VR der wichtigste Bereich für die Zukunft dieses Mediums markiert oder dass andere Umgebungen weniger interessant sind. Viele von den hier untersuchten Methoden der Innenarchitektur mögen in allen Umgebungen gleich integriert sein, dies soll jedoch keine Voraussetzung für weitere Bereiche darstellen.

Die erste Interaktion mit einer Umgebung, welche alle VR-Anwendungen gemeinsam haben, ist das Herumschauen. Sobald der Nutzer eine VR-Brille aufsetzt, beginnt die Kommunikation, d.h. das Wahrnehmen der Umgebung. (Gillies & Xueni Pan, 2018, S. 1) Der Benutzer erhält ein Gefühl für die Größe des Raumes, seinem Abstand zu den Objekten als auch eine Empfindung der Lichtverhältnisse. Seine Umgebung wahrnehmen zu können, beeinflusst sein Gefühl der Präsenz zwar gewaltig, wird jedoch schnell eintönig und reicht nicht mehr aus. Die Umgebung sorgt vor der Interaktion für ein immersives Gefühl des Users.

„**Immersion** is the objective degree to which a VR system and application projects stimuli onto the sensory receptors of users in a way that is extensive, matching, surrounding, vivid, interactive, and plot informing. „(Jerald, 2015a, S. 79)

Um das Gefühl der Präsenz bei dem User zu stärken muss man an erster Stelle auf die Umgebung achten, welche die virtuelle Welt für ihn definiert. Im Bereich der Innenarchitektur wird dieses Gefühl mit der richtigen Auswahl von

2 Virtual Reality

Gegenständen und Farben unterstützt. Licht spielt ebenfalls eine wesentliche Rolle. Zusätzlich ist es wichtig, die Größenverhältnisse zu beachten, die ein User aufgrund seiner Erfahrung bei einem zu großem bzw. zu kleinem Raum sofort bemerken wird. Erfahrungsgemäß sind viele unerfahrene User nicht in der Lage das unangenehme Gefühl in der virtuellen Welt zu beschreiben. Sie bemerken jedoch, dass sich ein virtueller Raum mit unkorrekten Größenverhältnissen seltsam anfühlt. Dieses Gefühl kann die Immersion negativ beeinflussen und sollte deshalb vermieden werden. Beobachtet man das Verhalten der Menschen hinsichtlich ihrer Umgebung, ist Fortbewegung der nächste Schritt um mit ihrer Umgebung in Kontakt zu treten. Dieses Fortbewegen in einer virtuellen Umgebung findet auf unterschiedliche Arten statt: Im Allgemeinen animiert ein großer Raum den User, herumzugehen um seinen Raum zu erkunden. Somit kann man behaupten, dass Fortbewegung in einer Umgebung die zweite spezifische Interaktion mit ihr darstellt, nachdem das Herumschauen als erste Interaktion definiert wurde. Entsprechend relevant werden nun Muster der Fortbewegung und Navigation in VR.

3 Lokomotion und Navigation

Eine der fundamentalen Interaktionen in einer virtuellen Welt besteht darin, sich als User in dieser Umgebung bewegen zu können. Die Art und Weise der Fortbewegung hängt von ihrer Anwendung ab. Bei einer Anwendung, in der sich ein User beispielsweise auf einen Stuhl setzt und eine Inszenierung anschaut, stellt die Navigation eine andere Wichtigkeit dar, als bei einem Rundgang in einem vorvisualisierten Haus. (Gillies & Xueni Pan, 2018, S. 12) Als Entwickler möchte man Anwendungen kreieren, in denen man den User in einer größeren virtuellen Welt herumgehen lässt. Diese Räumlichkeiten sind meistens größer als der Bereich, der von den verwendeten Sensoren der physischen Welt abgetastet werden kann.

Zusätzlich stellt sich die menschliche Wahrnehmung als ein weiteres Problem in den Weg. Der Mensch nimmt die eigene Bewegung und die Bewegung der Umgebung mit unterschiedlichen Rezeptoren wahr. Eine Unstimmigkeit zwischen den Signalen, welche diese Rezeptoren ans Hirn weiterleiten, führt oft zu einer Warnung des Körpers. Diese Warnung zeigt sich meist in Form von Übelkeit bzw. Schwindel oder Signalen, die als **Motion Sickness** bezeichnet werden. Somit müssen sich die Entwickler notgedrungen eine Navigationsmethode überlegen, die einerseits zu der Anwendung passt und andererseits zu keinen unangenehmen Gefühlen wie Übelkeit oder Schwindel führt. Solche Navigationsmethoden in Virtual Reality können in drei Kategorien unterteilt werden: Lokomotionsmethoden, die auf natürlichen Körperbewegungen basieren, Teleportation und Artificial Lokomotion. Es gibt ebenfalls Kombinationen aus diesen Methoden bzw. die Möglichkeiten, mehrere dieser Methoden in einer Anwendung zu integrieren, um letztlich Symptome der Motion Sickness zu bekämpfen.

3.1 Motion Sickness

Motion Sickness (Bewegungskrankheit) ist eine Fehlinterpretation des Gehirns, welche Übelkeit, Schwindel, Kopfschmerzen und weitere unangenehmen Gefühle verursacht. Grund dafür liegt im Wahrnehmen der Fortbewegung des eigenen Körpers und den entsprechend erzeugten Signalen. Bei einer Bewegung des menschlichen Körpers wird diese Information mittels drei Systemen an das Hirn weitergeleitet: **Den Augen**, welche eine Bewegung sehen und erkennen, dem **Gleichgewichtsorgan** im Innenohr und dem **propriozeptiven System**, welches

3 Lokomotion und Navigation

wiederum der Wahrnehmung bestimmter Reize aus dem Innenkörper dient. (A. Springer, 2016) Ein Verständnis von Motion Sickness erfordert eine genauere Erklärung dieser und weiterer grundlegender Begriffe.

Das Gleichgewichtsorgan bzw. der Vestibularapparat befindet sich im Innenohr und besteht aus drei kreisrunden Kanälen, den Bogengängen sowie den Otolithenorganen. Die Bogengänge und die Otolithenorgane sind mit Flüssigkeit gefüllt. Wenn der Kopf bewegt wird, bewegt sich ebenfalls diese Flüssigkeit. Durch die Bewegung bemerken Sinneshärchen, welche von dieser Flüssigkeit umgeben sind, eine Veränderung, die sie als Signal an das Gehirn schicken. Jeder Bogengang ist für eine bestimmte Drehung zuständig. Sie erkennen, ob sich der Kopf nach oben, unten, rechts, links oder seitwärts gedreht hat. Diese Informationen werden durch das Gehirn an weitere Organe weitergeleitet, welche wiederum auf diese Informationen angewiesen sind. (Institut für Qualität und Wirtschaftlichkeit im Gesundheitswesen, 2017)

Ursachen von Problemen oder Fehler dieser Informationsübermittlung, welche zur Bewegungskrankheit führen können, sind visuell oder physisch induziert. Bei einer visuell induzierten Bewegungskrankheit hilft in den meisten Fällen das Schließen der Augen. Dies ist bei der physisch induzierten Bewegungskrankheit jedoch keine Hilfe. Reisekrankheit ist eine Form davon. Obwohl die Körperreaktion in beiden Fällen relativ gleich ist, können die Ursachen unterschiedlich sein. (Jerald, 2015a, S. 200)

Propriozeptive Wahrnehmung hilft den Menschen die Lage und die Position des eigenen Körpers, die Bewegung und die Stellung der Gelenke und Gliedmaßen bestimmen zu können, ohne hin zu sehen. („PropriozeptiveWahrnehmung“, o. J., S. 1)

3.1.1 Vektion

Vektion ist ein weiterer wichtiger Aspekt der Bewegungswahrnehmung. Dieser Effekt kann entstehen, wenn sich Objekte in dem eingeschränkten Sichtfeld des Betrachters so bewegen, dass sie ein Gefühl der Eigenbewegung hervorrufen. (Braining, 2018, S. 44) Vektion muss nicht unbedingt zu einer Bewegungskrankheit führen. Sie hängt vielmehr von mehreren Faktoren ab, die einzeln oder kombiniert die Bewegungskrankheit hervorrufen können.

Sensory Conflict Theory, welche am meisten verbreitet ist, besagt, dass sich bei einer passiven Bewegung, z.B. Fliegen, der menschliche Körper nicht bewegt, während die Landschaft als auch die Umgebung an dem Körper vorbeiziehen und das Auge diese Information an das Gehirn weiterleitet. Es wird angenommen, dass

3 Lokomotion und Navigation

Bewegungskrankheiten durch die Rezeption von widersprüchlichen Informationen hinsichtlich Bewegung und Körperlage entstehen. (A. Springer, 2016)

Bei einer VR Applikation entstehen die visuellen und auralen Reize durch eine Applikation, wobei vestibuläre und propriozeptive Reize auf die Bewegung des menschlichen Körpers in der realen Welt zurückzuführen sind. Der Körper spürt eine Unstimmigkeit, welche schließlich zu dem Gefühl der Übelkeit führt.

Evolutionary Theory greift auf die These zurück, dass der Körper bei einem Konflikt zwischen der Wahrnehmung und Motorik auf die gleiche Art und Weise reagiert, wie bei einer Vergiftung. Übelkeit und Schwindel helfen ihm, das Gift aus dem Magen zu schaffen.

Postural Instability Theory konzentriert sich auf die Haltung des Körpers. In Situationen, in denen er nicht in der Lage ist, das Gleichgewicht zu halten oder die eigene Haltung in eine passende Form zu bringen, erleidet der Mensch dem Gefühl der Übelkeit. (Jerald, 2015a, S. 203)

Die **Rest Frame Hypothese** begründet die Bewegungskrankheit aus einer anderen Perspektive. Laut dieser Hypothese entsteht das Gefühl bei Nichtübereinstimmung der menschlichen Wahrnehmung von stationären Objekten und ihrer realen Referenz. (Jerald, 2015a, S. 204)

Menschen entwickeln im Laufe ihres Lebens ein mentales Modell von den Objekten und Gegenständen, die sie in der Welt als statisch und stationär wahrnehmen. Ein **mentales Modell** bezeichnet die Definition eines Gegenstands oder eines Prozesses im Bewusstsein eines Menschen. Diese Objekte sind beispielsweise Wände, Bäume oder Säulen. Die Bewegung von anderen Objekten oder die eigene Körperbewegung wird in den meisten Fällen in Relation zu diesen stationären Objekten wahrgenommen. Die Bewegung oder die Geschwindigkeit eines Autos entsteht im Kopf des Betrachters somit im Vergleich zu den im Hintergrund statischen Häusern. Wenn sich diese Gegenstände in der Welt nicht wie das mentale Modell eines Menschen von diesen Objekten verhalten, entsteht ein Konflikt bei der Wahrnehmung der Rezipienten, was wiederum zu dem Gefühl der Übelkeit führt. Bezogen auf die Gegenstände einer virtuellen Welt ist es also zu empfehlen, dass manche Objekte im Hintergrund als statisch erkennbar sind. Sie helfen dem User, sich schneller zu orientieren und einen Überblick über die Lage des Raumes zu verschaffen.

Die **Eye Movement Theory** besagt, dass die Bewegungskrankheit entstehen kann, wenn das Auge durch eine unnatürliche Bewegung versucht das gesehene Bild zu stabilisieren. (Jerald, 2015a, S. 206) Wenn die Augen eine andere

Bewegung von den gerenderten Bildern in der virtuellen Welt erhalten, als sie es erwarten, müssen sie sich entsprechend anders bewegen, als sie es in der Wirklichkeit tun. Das ist ein weiterer Grund für die Bewegungskrankheit.

Es existieren somit mehrere Theorien und Ansätze, die Bewegungskrankheit im Allgemeinen und in Bezug auf Virtual Reality erklären. Grundsätzlich versucht der Körper Unstimmigkeiten, welche er wahrnimmt, mit eigenen Methoden auszugleichen. Er versucht seine Haltung zu ändern um die Balance wiederherzustellen, bewegt die Augen schneller bzw. langsamer, um mit der Geschwindigkeit mitzuhalten und in extremen Fällen mit einem Signal von Übelkeit oder Schwindel die unpassenden Umstände bemerkbar zu machen.

Abgesehen davon, dass man diese Wahrnehmungsstörungen beseitigen bzw. trainieren kann, gibt es die Möglichkeit, sie bewusst und kontrolliert zu manipulieren. Mit diesem Wissen versuchten Forscher unterschiedliche Methoden der Lokomotion in Virtual Reality zu integrieren.

3.2 Based on Physical Body Movement

Es ist naheliegend, dass der User eine virtuelle Welt am besten wahrnehmen kann, wenn er seinen Körper bei der Entdeckung dieser Welt auf die gleiche Art und Weise zum Einsatz bringt wie er es in der Realität tut. Der natürlichste Weg des Menschen, sich in einem Raum zu bewegen, ist das Gehen, sofern er es kann. (Langbehn, Lubos, & Steinicke, 2018, S. 2) Das Gehen unterstützt somit die Illusion der Präsenz in einem Raum und führt zu einem richtigen Empfinden über die Größe dieses Raumes und der Orientierung eines Users. (Gillies & Xueni Pan, 2018, S. 13) Basierend darauf, dass die Simulation der natürlichen Bewegungen in einer virtuellen Welt am besten funktionieren, versuchen Forscher Methoden zu entwickeln, die ihnen eine 1:1-Übertragung der Körperbewegungen in eine virtuelle Welt ermöglicht. Da die exakte Übertragung aus unterschiedlichen Gründen, wie zum Beispiel Platzgröße, nicht möglich ist, wurden Ideen entwickelt, wie diese Bewegungen auf eine virtuelle Welt gemappt werden können.

Im folgenden Abschnitt werden Methoden vorgestellt, welche auf der physischen Körperbewegung basieren. Dies bedeutet, dass diese Bewegungen des Users als seine Bewegungen in VR interpretiert werden. In den meisten Fällen benötigt man ein *Locomotion Interface*. Dabei handelt es sich um ein System, das dem User ein Gefühl der Bewegung vermittelt, welche in einer virtuellen Welt übertragen wird. (Nabiyouni, 2015, S. 13)

3.2.1 Pressure-Based Board

Bei dieser Methode muss der User auf einer interaktiven Fläche stehen, während er einen Joystick in der Hand hält. Gleichgewicht und Richtung werden kontrolliert indem er mit dem Fuß auf einen bestimmten Bereich des Interfaces drückt.

3.2.2 Cycling Systems

Bei dieser Methode bekommt der User meist die Möglichkeit, sich auf ein fix-installiertes Rad oder einen autosesselähnlichen Stuhl hinzusetzen. Die Möglichkeit, den Lenker des Rads bzw. das Lenkrad eines Autos in der Hand zu haben, zählt zu einem hilfreichen haptischen Feedback. Die Bremse, die simulierte Geschwindigkeit und die weiteren Informationen können mit Hilfe der Sensoren präzise an die Software weitergeleitet werden. Mit diesen Methoden ist eine immersive Fahrt in einer virtuellen Umgebung möglich. Es besteht ebenfalls die Möglichkeit, das Treten in Schritte umzuwandeln, sodass der User in der realen Welt auf einem fix-installierten Rad das Radfahren nachahmt, wobei er in der virtuellen Umgebung nach vorne schreitet. (Nabiyouni, 2015, S. 15)



Abbildung 4: Cycling System, (VR Research, o. J.)

3.2.3 Walk-in-Place

Diese Methode scheint das Problem der Lokomotion gelöst zu haben, da hierbei die Möglichkeit des unendlichen Gehens sehr gut simuliert werden kann. Diese Systeme schränken die Bewegungsmöglichkeiten des Users in der physischen Umgebung zwar ein, sind jedoch stark in der Simulation der natürlichen

3 Lokomotion und Navigation

Gehbewegung. Der User benötigt allerdings in den meisten Fällen weiterhin eine technische Hilfe um seine Balance zu halten. Mittlerweile experimentieren mehrere Hersteller an unterschiedlichen Ansätzen.



Abbildung 5: Kat Walk VR („Virtuo-VR | Kat Walk“, o. J.)

KAT Walk VR hat es geschafft, eine Möglichkeit zu implementieren, bei welcher der User auf einer konkaven Fläche gehen, laufen, drehen und sich in seinen Harness (Geschirr) setzen kann. Den Vorteil an *KAT Walk VR* stellt die Möglichkeit des Users dar, seine Hände frei zu bewegen. Es handelt sich hierbei um ein häufiges Problem, das bei vielen Walk-in-Place-Geräten zu finden ist. Die beiden folgenden Fotos zeigen zwei dieser Geräte als Beispiele.



Abbildung 6: Virtualizer (christoph Springer, 2016)

3 Lokomotion und Navigation



Abbildung 7: WizDish (Feltham, 2015)

Eine weitere Idee des Walk-in-Place wurde von der Firma *Infinadeck* vorgestellt. Infinadeck ist eine omnidirektionale Treadmill (Laufband), das ebenfalls seinem User eine große Handfreiheit zu lassen versucht.



Abbildung 8: Infinadeck Treadmill (C. Springer, 2016b)

Der Nachteil von diesen Geräten ist der von ihnen benötigte Platz und in den meisten Fällen der hohe Preis. Somit kommt ein neuer Ansatz ins Spiel, der zwar im Vergleich zu den Walk-in-Place-Geräten mehr Platz benötigt und dabei mehr Trackingbereich in Anspruch nimmt, jedoch im Vergleich zu den oben genannten Ideen das Problem an einer anderen Stelle zu lösen versucht. Redirected Walking versucht diese Herausforderung mittels Software zu lösen.

3.3 Redirected Walking

Redirected Walking versucht einen Weg zu finden, mit dem Menschen einen virtuellen Raum mit ihren eigenen Schritten begehen können, indem sie die Bewegung des Users in VR abbildet. Diese Methode kann enorme Vorteile bringen, ist aber mit einem großen Problem konfrontiert: die Eingeschränktheit durch die physische Welt. Grundsätzlich manipuliert Redirected Walking die Wahrnehmung des Users so, dass er einen geraden Weg zu folgen meint, obwohl er in der physischen Welt einen Bogen entlanggegangen ist.

Generell ist die Bewegung in einer virtuellen Welt mit einer Herausforderung konfrontiert: der menschlichen Wahrnehmung und Interpretation beweglicher Umgebung und eigener Bewegung. Der User ist in VR an Platzgröße und Bewegungsfreiheit, die ihm die reale Umgebung anbietet, gebunden und die Entwickler können diese Information für jeden einzelnen User nicht in die virtuelle Umgebung einbauen.

Natürlich muss man zu Beginn und bei der Konfiguration von VR-Devices den Bereich markieren und damit einen *Chaperone* erstellen, der den Benutzer auf die Distanz zu den realen Objekten hinweist. Der Chaperone visualisiert die Kanten des Spielbereiches, welcher von den Sensoren getrackt wird. Der Spieler wird von Sensoren wahrgenommen, solange er sich in diesem Bereich bewegt. (Brown, 2017)

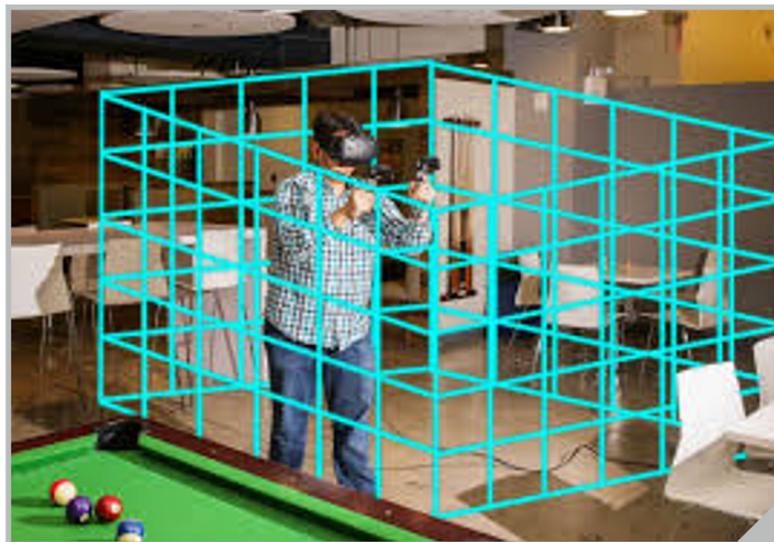


Abbildung 9: VR Chaperone (Hills-Duty, 2018)

3 Lokomotion und Navigation

Bei HTC VIVE befindet sich dieser Bereich zwischen den beiden Basisstationen des Lighthouses, die in einem Abstand von maximal 6 Metern voneinander stehen können.

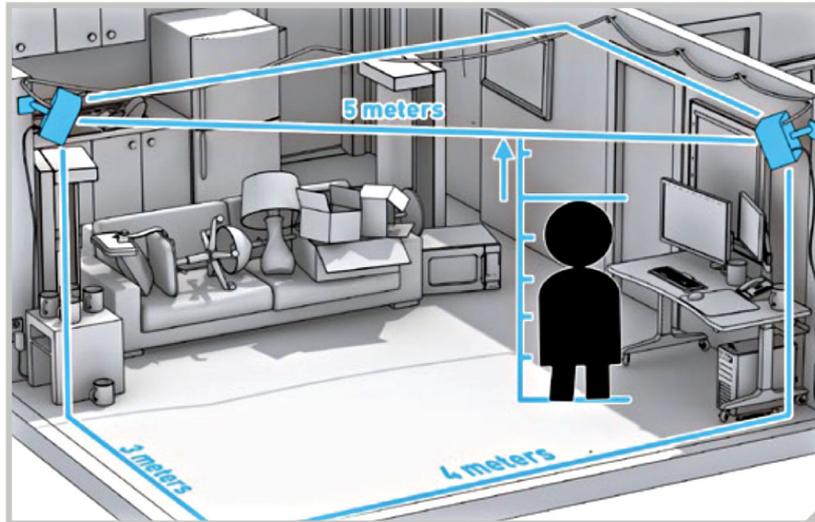


Abbildung 10: HTC VIVE Trackingbereich (C. Springer, 2016a)

Dieser Bereich bietet jedoch keine Freiheit. Es ist lediglich eine Einstellung um mit Hilfe der Sensoren einen Hinweis auf die Bewegungsmöglichkeiten zu erhalten. Manche Anwender schalten diese Anzeige nach einer gewissen Zeit aus. In VR werden jedoch Welten kreiert. Es ist gewollt und erwünscht, sich in einem kleinen Zimmer zu Hause zu befinden und in eine große, spannende Welt einzusteigen.

Die Idee des Redirected Walking basiert auf den Funktionen des menschlichen Orientierungsvermögens. Ihre Kernaussage ist, dass die Gehrichtung eines Anwenders in der virtuellen Welt manipuliert werden kann. Um zu wissen, wie diese Manipulation zu Stande kommt, muss man zuerst wissen, wie der Orientierungssinn eines Menschen im Allgemeinen funktioniert. Zu diesem Zweck kann man sich der **Sakkaden**, eines natürlichen Phänomens der visuellen Wahrnehmung, bedienen. („Neues Verfahren für Redirected Walking“, 2018) Mit ruckartigen und schnellen Augenbewegungen, die mehrmals pro Sekunde passieren, visieren die Menschen unterschiedliche Objekte in ihrem Sichtfeld an. Das Gehirn kaschiert diese Bewegungen mit einer Art Blindheit, die, genauso wie das Fokussieren, unbewusst und schnell passiert. Dieses Phänomen wird wiederum als sakkadische Suppression bezeichnet. („Neues Verfahren für Redirected Walking“, 2018)

Menschen, deren Augen verbunden sind, gehen höchstens 20 Meter geradeaus. Danach verlieren sie ihre Spur in einer zufälligen Richtung. Aus diesem Grund

3 Lokomotion und Navigation

verlaufen sich Wanderer in Wüsten und Wäldern ohne Wegbeschreibungen oder Hinweise. Irrtümlicherweise vermuteten Forscher jahrelang, dass dieses Phänomen auf die Dominanz eines Beins beim Menschen zurückführt. Die Wanderer irren sich jedoch zufällig und die kreisförmigen Abweichungen ergeben sich aus der Orientierungslosigkeit. (Jan, 2009)

Zudem stellten Forscher fest, dass der Mensch auf größerer Distanz die Wahrnehmung einer geraden Linie verliert. (Natalie, 2011) Auf die Sinneseindrücke und das Gleichgewichtsorgan im inneren Ohr ist in solchen Situationen kein Verlass mehr. Der Richtungssinn von Menschen und Tieren muss bei großen Distanzen immer wieder neu kalibriert werden. Dies geschieht auf einer auditiven und/oder visuellen Ebene über einen Abgleich mit der Umgebung. (Jan, 2009)

Redirected Walking nutzt diese Eigenschaft bzw. Schwäche aus, um die Neuorientierung eines Users in einer virtuellen Welt kontrolliert zu manipulieren. Das Ausgangsproblem besteht darin, dass der User meist an die Grenzen der realen, physischen Welt stößt, wenn er in der virtuellen Welt umhergehen möchte. Der physische Raum ist also kleiner als der virtuelle Raum und daher ist das Ziel von Redirected Walking, dass der User alle Bewegungen in der Welt als eigene Bewegung wahrnimmt. (Roberts, 2001, S. 2)

Mit anderen Worten: Die Idee von Redirected Walking ist eine Trennung der physischen und der virtuellen Bewegung, ohne, dass diese Trennung vom User gespürt wird. Die Bewegung in einer virtuellen und einer physischen Umgebung stehen in einem Verhältnis zueinander, welches bei Experimenten des Redirected Walking gezielt manipuliert wird, sodass die virtuelle Umgebung keine 1:1-Abbildung der physischen Umgebung darstellt.

3.3.1 Gains

Gain ist ein Multiplikationsfaktor hinsichtlich der Bewegung des Users in einer virtuellen Umgebung. Es gibt drei Arten von Gains:

Bei **Rotation Gain** dreht sich der User in der physischen Welt um 180° und nimmt diese Drehung aufgrund der visuellen Wahrnehmung der Umgebung in der virtuellen Welt als 90° wahr. (Abb. 11) Ähnlich legt er in der physischen Welt denselben Weg zurück, obwohl er in der virtuellen Welt nach rechts bzw. links abgebogen ist, wie aus den Abbildungen ersichtlich.

3 Lokomotion und Navigation

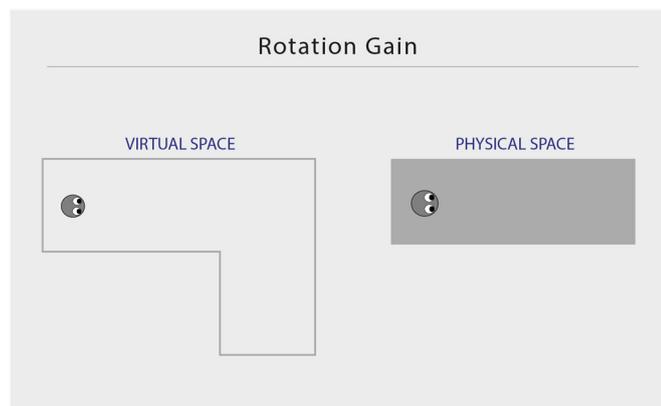


Abbildung 11: Rotation Gain-Startposition, (eigene Darstellung), (vgl. Suma, 2016)

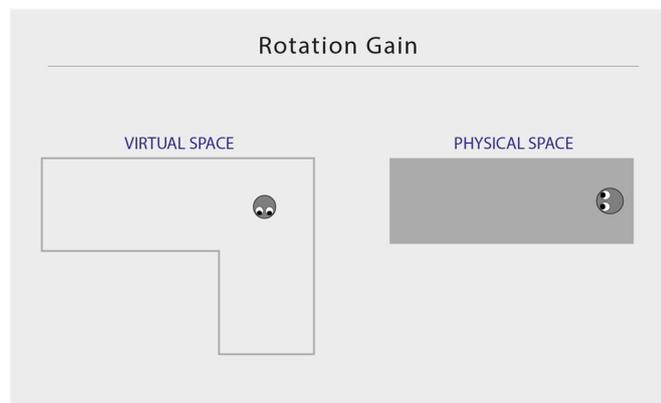


Abbildung 12: Rotation Gain-Drehung, (eigene Darstellung), (vgl. Suma, 2016)

Im Falle des **Curvature Gain** bewegt sich der User entlang einer Kurve, wobei er in der virtuellen Welt einen geraden Weg vor sich sieht. **Unlimited Corridor**, ein Experiment der Universität Tokyo, in Zusammenarbeit mit der Unity Game Engine Company, schildert ein Beispiel dieser Form des Redirected Walking. Hier bewegt sich der User entlang einer in der Realität existierenden, gebogenen Wand, wobei er in der virtuellen Welt einer Wand geradeaus folgt. (Robertson, 2016)

Das folgende Bild zeigt die Virtuelle Welt des Unlimited Corridor-Projekts. Die weiteren Bilder stellen eine Abbildung des Weges nach, den sie in der physischen Welt des Studios gehen.

3 Lokomotion und Navigation

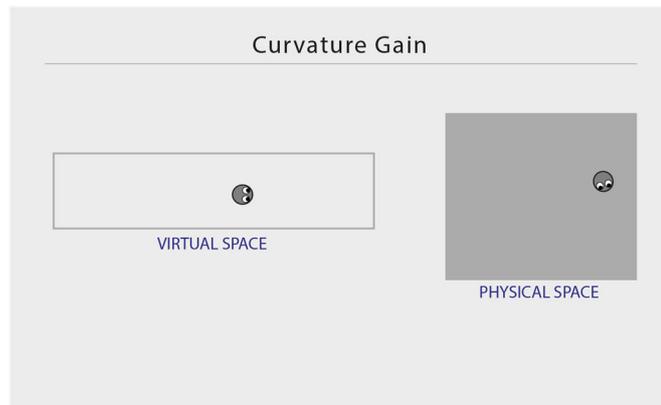


Abbildung 13: Simulation des Curvature Gain, (eigene Darstellung), (vgl. Suma, 2016)



Abbildung 14: Die virtuelle Umgebung des Unlimited Corridor (James, 2016)



Abbildung 15: Die physische Umgebung bei Unlimited Corridor (James, 2016)

3 Lokomotion und Navigation

Translation Gain ist ein Multiplikationsfaktor für Schrittgrößen. Wie auf der Abbildung ersichtlich, wird die Schrittgröße des Users in der virtuellen Umgebung beispielsweise verdoppelt.

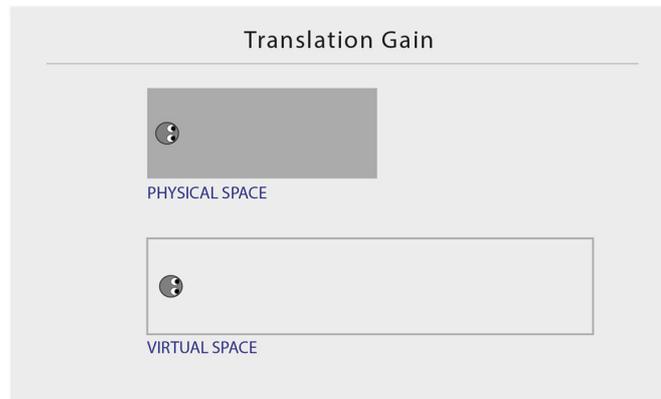


Abbildung 16: Translation Gain, (eigene Darstellung), (vgl. Suma, 2016)

1994 erforschten Michael Moshell und Dan Mapes an der University of Central Florida das Phänomen des Redirected Walking. Sie ließen ihre Probanden in einer manipulierten Umgebung in VR einen geraden Weg gehen, ohne dass sie bemerkten, dass sie in Wahrheit einen Bogen gemacht haben. Je enger dieser Bogen ist, desto größer kann der zu darstellende Bereich in der virtuellen Welt sein. Es ist aber zu beachten, dass die Verzerrung dem User vermehrt auffallen kann, je kleiner diese Bögen werden. (Roberts, 2001, S. 2) Ihr Experiment ist an der damals verwendeten Tracker-Technologie, die Simulationskrankheiten verursacht hat, gescheitert. (Roberts, 2001, S. 1) Shari Razzaque, Zachariah Kohn und Mary C. Whitton zeigten in einem Experiment, wie sie die Probanden in einer Anwendung entlang eines Wegs im Zick-Zack gehen ließen, wobei die Probanden in Wirklichkeit dieselbe Route hin und zurückgelaufen sind. (Abb. 17)

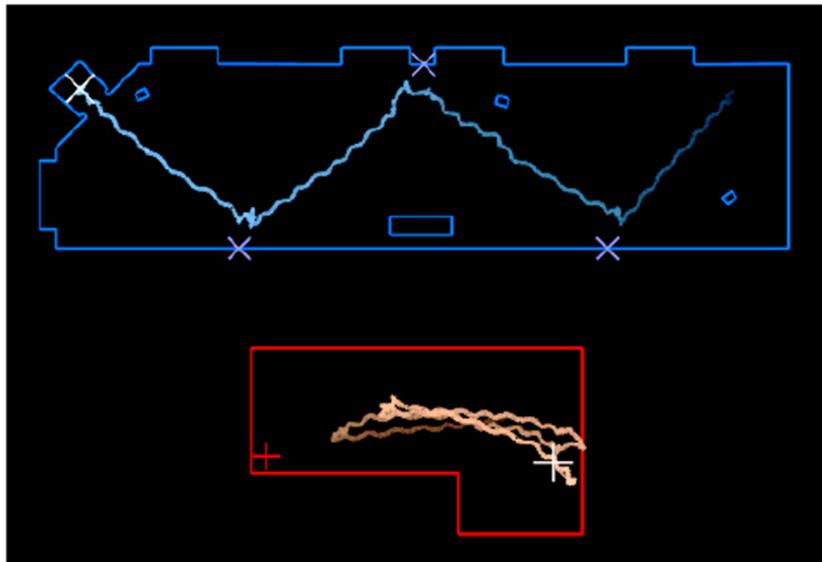


Abbildung 17: Vergleich zwischen dem virtuellen und physischen Weg bei einem redirected Walking versuch (Roberts, 2001)

Bei diesem Test wurden die Probanden beauftragt, dass sie mit ruhigen Schritten zu der vor ihnen stehenden Wand laufen sollten um einen grünen Knopf zu betätigen. Daraufhin mussten sie sich umdrehen, zu der Wand hinter ihnen laufen und die gleiche Aktion erneut durchführen. Indem sie diese Routine immer wiederholen mussten, wurden sie indirekt gezwungen und haben zugleich den Eindruck erhalten, eine bestimmte Route zu verfolgen, die nicht den physischen Gegebenheiten der Wirklichkeit entsprach. Es findet also eine erfolgreiche Manipulation der vestibulären Wahrnehmung statt. **Vestibuläre Wahrnehmung** ist für die Gleichgewichtsregulation zuständig. Sie hilft ebenfalls den Menschen sich in einem Raum orientieren zu können. (Saft, 2013)

Menschen reagieren auf visuelle und vestibuläre Reize zur Orientierung und Aufrechterhaltung des Gleichgewichts. Sie benutzen die gleichen Organe und Sinne um zu erkennen ob sie sich selbst bewegen oder ob Objekte um sie herum in Bewegung sind. Forschungen zeigen, dass Menschen Bewegungen als eigene Bewegung (Self-Motion) wahrnehmen, je mehr Faktoren in ihrer Umgebung als konstant wahrgenommen werden. Wenn visuelle, aurale und vestibuläre Signale in der Umgebung eines Users einheitlich bleiben, sollte dieser eine willkürliche Bewegung seiner Umgebung nicht spüren, da er sie entweder als eigene Bewegung wahrnimmt oder unbewusst mit der eigenen Bewegung ausgleicht. (Roberts, 2001, S. 2) Bei einer Unklarheit des empfangenen Signals scheint die visuelle Wahrnehmung über die Vestibuläre zu dominieren. (Jerald, 2015b, S. 109) Und durch diese Dominanz ist Redirected Walking schließlich so effektiv.

3 Lokomotion und Navigation

Menschliche Wahrnehmung benötigt Bewegungen, um Veränderungen zu erkennen. Wenn diese kontinuierlichen Wahrnehmungen unterbrochen werden, sodass der Mensch die Bewegungen aus den Augen verliert, fällt es ihm schwer, Veränderungen wahrzunehmen. (Suma, 2016) Dieses Phänomen ist in der Psychologie unter dem Begriff der *Veränderungsblindheit* bekannt. (Cherry, 2018) Erfahrungen zeigen, wenn Veränderungen in virtuellen Umgebungen eine Konsistenz aufweisen, sodass es den User nicht stört, werden diese als akzeptabel und angenehm empfunden. User können möglicherweise Verzerrungen erkennen, sind allerdings oft nicht in der Lage diese genau zu beschreiben. (Suma, 2016)

3.3.2 Impossible Space

Um die **Distanzabschätzung** von Usern in VR zu testen, wurden sie bei einem Versuch von Evan Suma Rosenberg in einen von ihm **sogenannten Impossible Space** geschickt. Dessen Tests zeigen in zwei unterschiedlichen Umgebungen, inwiefern die räumliche Wahrnehmung von Probanden in einer virtuellen Umgebung manipulierbar ist. Im ersten Test veränderte er die Position des Eingangs kurz nachdem der User den Raum betrat und auf einen Schreibtisch in der Ecke des Raumes zuing. (Abb. 18)

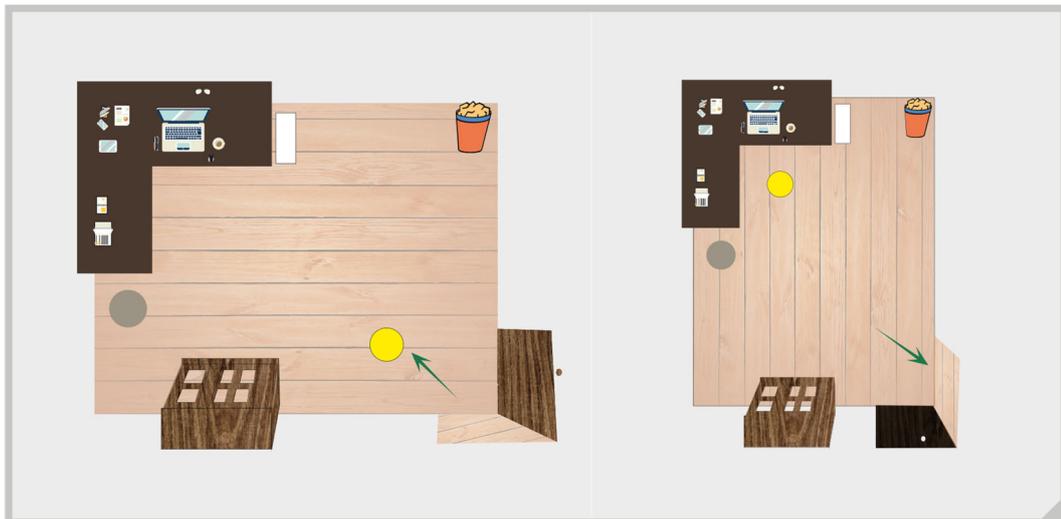


Abbildung 18: Impossible Spaces, Änderung der Ausgangsposition, (eigene Darstellung), (vgl. Suma, 2016)

Damit konnte er die virtuellen Räume so gestalten, dass der User in einem unendlichen Kreis umhergehen konnte. (Suma, 2016) Nach dem Test befragte er seine Probanden, ob ihnen Veränderungen oder Merkwürdigkeiten aufgefallen sind, was alle verneinten.

3 Lokomotion und Navigation



Abbildung 19: Darstellung des unendlichen Kreises, (eigene Darstellung), (vgl. Suma, 2016)

Im zweiten Test wurden die Probanden gebeten, sich auf einen bestimmten Punkt zu stellen und von dort aus zu einer anderen Stelle zu laufen. Daraufhin sollten sie sich umdrehen und zu der Ausgangsposition schauen. In diesem Moment wurde das Bild im Head Mounted Display ausgeblendet und die Probanden sollten blind und eigenständig zu diesem Punkt gelangen. Diese Methode wird als *Direct Blind Walking* bezeichnet und stellt eine gängige Methode dar, um Distanzabschätzungen zu überprüfen. (Siegel, 2015, S. 10)

Die folgenden Abbildungen zeigen die Bewegung der virtuellen Wand, sobald der User den Raum verlässt. (Abb. 20) Während sich der User in dem Flur in Richtung des anderen Raums befindet, wird die innere Wand (außerhalb seines Sichtfeldes) verstellt. Diese Positionsänderung weist eine Überlappung der beiden Räume von 0 bis 75% auf.

Die Probanden wurden nach Betreten des neuen Raums gefragt, ob sie die vorherige Stelle zeigen könnten. Daraufhin sollten sie sich auf diesen Ausgangspunkt stellen. In einem Fall von Überlappungen war diese Position sehr nah an der Wand bzw. ein oder zwei Schritte davon entfernt. 90% der Probanden in diesem Test sind sogar bei 75% Überlappung jenen Weg zurückgegangen, welcher in ihrem Kopf als zurückgelegter Weg registriert war und nicht den physisch korrekten Weg. (Suma, 2016) *Direct Blind Walking* offenbart also, dass Menschen nicht in der Lage sind, Distanzen unter diesen Umständen realistisch einschätzen zu können.

3 Lokomotion und Navigation

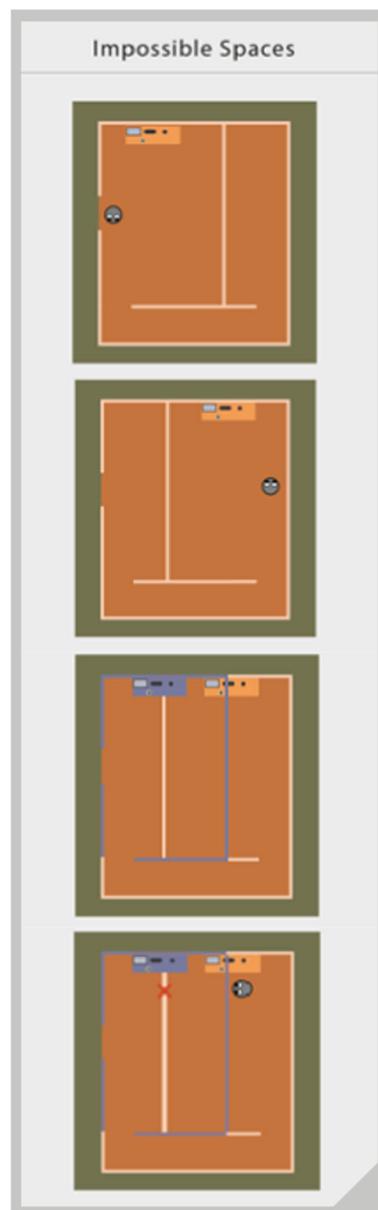


Abbildung 20: Impossible Spaces, (eigene Darstellung), (vgl. Suma, 2016)

Redirected Walking ist somit eine Möglichkeit der Lokomotion in einer virtuellen Welt, welche dem User eine hohe Immersion ermöglicht. Das Wahrnehmen eines Raumes mit Hilfe der eigenen räumlichen Bewegung hilft der Orientierung. Redirected Walking benötigt im Vergleich zu anderen Lokomotionsmethoden einen größeren Trackingbereich, da es auf die physische Distanz, welche zurückgelegt wird, angewiesen ist. Das kann, im Gegensatz zu anderen Methoden, wie *Walk-in-Place*, *Fliegen* oder *Treadmills* als ein Nachteil von Redirected Walking gesehen werden. (Roberts, 2001, S. 5) Da es aber eine in Software implementierte Methode darstellt, ist es in vielen Fällen ökonomischer.

3 Lokomotion und Navigation

Mehrere Spiele und Anwendungen implementieren diese Methode. **Unseen Diplomacy** ist ein auf Steam verfügbares Spiel, welches Redirected Walking als die Lokomotionsmethode ausgewählt hat. Dieses Spiel nutzt Redirected Walking um den Spieler in einem größeren Bereich herumlaufen zu lassen. („Unseen Diplomacy bei Steam“, 2016) Es ist bemerkenswert, wie die Bewegung so gesteuert wird, dass die physischen Grenzen kaum beachtet werden. Es benötigt jedoch zusätzlichen Aufwand, Umgebungen passend zu dieser Methode, zu designen.

3.4 Teleportation

Die zweite Methode ist Teleportation. Sie ist die beliebteste Lokomotionsmethode in VR. (Carbotte, 2018) Es erlaubt dem Nutzer problemlos Bereiche zu besuchen, welche größer sind als der eigentliche physische Ort. Es gibt zwar mehrere Variationen von Teleportation in VR, im Allgemeinen aktiviert der User die Teleportation jedoch mit Hilfe eines vorgesehenen Buttons, einer Markierung bzw. eines Interface, welches auf ein Teleportationsziel in einem Navigationsbereich hinweist. Nach Durchführung der Aktion befindet sich der Spieler auf der ausgesuchten Stelle. Diese Form der Teleportation kann in unterschiedlichen Anwendungen etwas variieren.

Tunneling ist beispielsweise eine solche Variation. Die Tunneling-Technik reduziert den animierten, beweglichen Teil auf einem kleineren Fenster in der Sicht des Users. Man kann in diesem kleinen Fenster zu dem neuen Ort navigieren, während die Umgebung um dieses Fenster stationär bleibt. Sobald in dem Inneren ebenfalls keine Bewegungen mehr zu erkennen sind, wandelt sich die Welt um den User in den neuen Ort um. (Carbotte, 2018)



Abbildung 21: Tunneling, Screen Shot aus dem Spiel „Informer podcast“

3 Lokomotion und Navigation

In 2016 stellten Forscher der University of Columbia fest, dass die dynamische Reduktion des Sichtfeldes während der Teleportation in Virtual Reality ebenfalls zur Reduktion der Motion Sickness führt. Dies geschieht indem der sichtbare Bereich des Users während der Bewegung kleiner wird, was wiederum dazu führt, dass die Menge an Details, welche einer Person in der virtuellen Welt auffallen, reduziert wird. Diese Details stellen die Ursachen für vestibuläre Unstimmigkeit dar, die letztendlich zu Übelkeit führen. (Carbotte, 2018)

The Gallery, ein weiteres Spiel und verfügbar auf Steam, verwendet eine Art der Teleportation, bei welcher der User die Umgebung „reorientieren“ kann. Damit ist eine Entscheidungsfreiheit während der Rotation zu einem Teleportationsziel gemeint. Der User wählt nicht nur das Ziel (Target) aus, sondern auch seine anschließende Blickrichtung (Target Rotation). („The Gallery - Episode 1“, 2016)

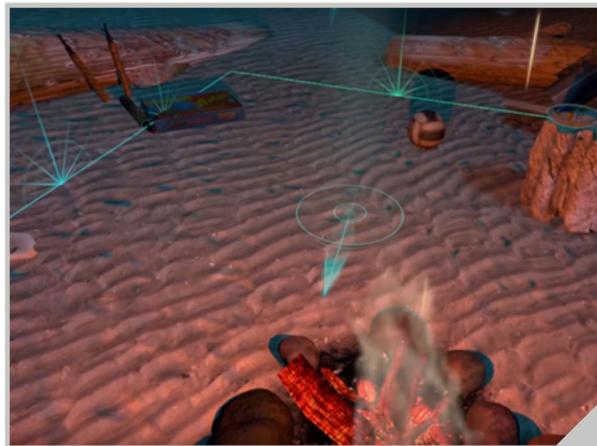


Abbildung 22: Teleportation in „The Gallery“ (Carbotte, 2018)

Spells ‘n’ Stuff, lässt seinen Spieler das Teleportationsziel mit Hilfe eines Würfels aussuchen, welchen er zu einer bestimmten Stelle wirft. Zwischen der Spielerposition und der Landeposition des Würfels wird eine Parabel gezogen, die auch zur besseren Orientierung dienen soll. („Spells ‘n’ Stuff bei Steam“, 2016)



Abbildung 23: Teleportation, Screen Shots aus dem Spiel „Spells ‘n’ Stuff“

3 Lokomotion und Navigation

Budget Cuts verwendet eine intuitive Methode. Hier geht es darum, dass der Spieler vom Teleportationsziel aus in der Lage ist, die Umgebung beobachten zu können. Er wählt ein Ziel und kann eine Spionagekamera an genau dieser Stelle aktivieren. Die Kamera kann nun von dem Teleportationsziel aus bewegt werden. Dies bedeutet, dass der Spieler die Umgebung um diesen Punkt untersuchen und sich danach entscheiden kann, ob er sich dorthin teleportieren möchte. („Budget Cuts Demo bei Steam“, 2016)

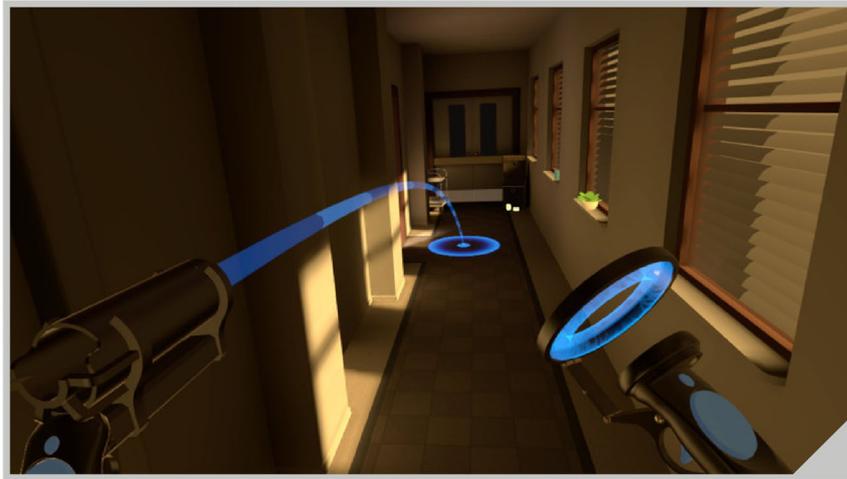


Abbildung 24: Auswahl der Teleportationsziel bei „Budget Cuts“ (Lang, 2016)



Abbildung 25: Visualisierung der Umgebung an Teleportationsziel bei „Budget Cuts“ (Lang, 2016)

Artificial Locomotion

Als letzte Variante findet Artificial Locomotion Erwähnung. Diese Methode ist unter 3D-Spielern als klassische Fortbewegungsmethode bekannt. Hier bewegt sich der Spieler in die gewollte Richtung, indem er den *Thumbstick* seines Gamepads dorthin richtet, betätigt und anschließend die neue Position annimmt, nachdem er losgelassen hat. Die Aktion muss nicht unbedingt mit einem Gamepad durchgeführt werden. Der Thumbstick des Oculus Touch oder das Touchpad von HTC VIVE erfüllen ebenso diese Aufgabe.

VR zGame ist eines der Spiele, die sich für eine Kombination zwischen Teleportation und Artificial Locomotion entschieden haben. Dabei wählt der User ebenfalls als Erstes ein Teleportationsziel aus, die Bewegung findet dann jedoch mit einem sehr langsamen *Schieben* zu dieser Stelle statt. Wie bereits erwähnt, spielt die Geschwindigkeit bei dem Gefühl der Motion Sickness oft eine wichtige Rolle (Game Cooker US Rocks VR, 2016). Diese Methode verursacht bei vielen Menschen eine Kinetose, weshalb sie selten als einzige Lokomotionsmethode in den VR-Anwendungen implementiert wird. Beispielsweise kann bei **Spellfighter** zu Beginn zwischen den Lokomotionsmethoden Teleportation bzw. Artificial Locomotion ausgewählt werden. („Spell Fighter VR bei Steam“, 2016)

3.5 Andere Methoden

Leave The Nest von Kaio Interactive bietet eine neue Fortbewegungsmethode: das Fliegen. Der User ist in diesem Spiel ein Vogel, der sich mit Hilfe seiner Flügel nach vorne bewegen soll. Er muss dafür die Controller in den Händen halten und mit seinen Armen das Fliegen in der realen Welt nachahmen. Damit kann er sich durch die Welt bewegen und erlebt sie aus der Vogelperspektive. („Leave The Nest bei Steam“, 2016) Es kann für den Spieler eine körperlich anspruchsvolle Übung darstellen, da er seine Arme eine verhältnismäßig lange Zeit bewegen muss. Der Spieler fliegt gegen Objekte und verliert, sobald seine Bewegungen nachlassen.

Paintey, entwickelt von Brian Lindenhof, ist ein weiteres Spiel, welches ebenfalls von Steam für HTC VIVE zur Verfügung steht. Das Spiel experimentiert mit einer weiteren Art der Fortbewegung. Hier muss der User in einer Stadt und zur Erfüllung der angegebenen Aufgaben mit seinen Händen nach vorne klettern. Es ist zu Beginn etwas ungewohnt, wenn man durch die Straßen klettern muss und das Klettern nach vorne ist an und für sich eine Umstellung, da hier die Perspektive eine andere ist. Es ist jedoch eine kreative, gelungene Methode, um das Problem der Platzgröße zu lösen. („Paintey bei Steam“, 2016)

3 Lokomotion und Navigation

Windlands von Psytec Games Ltd bietet eine Fortbewegung bestehend aus einer Kombination aus Artificial Locomotion und Schwingen. Hier kann sich der Spieler in den Ruinen einer Welt fortbewegen und diese entdecken. Dabei kann er sich mit einem Seil an Bäume hängen und nach vorne schwingen. Die Möglichkeit der Fortbewegung in der *Artificial-Form* ist möglich, sobald sich der Spieler von den Bäumen löst. („Windlands bei Steam“, 2016)

David Dewhirst entwickelte eine Lokomotionsmethode, welche die Armbewegung im normalen Gang als eine Simulation des Fortschreitens in VR überträgt. Der Spieler hält die Controller in den Händen, bewegt seine Arme als ob er Spazieren gehen würde und bewegt sich somit vorwärts. („daviddewhirst | VR Movement System“, o. J.)

RIPMotion wendet die Walk-in-Place Methode an, in welcher der Spieler auf einer Stelle gehen muss während er seine Controller in die gewünschte Richtung hält. Die eigene Körperbewegung macht diese Lokomotionsmethode etwas angenehmer für ein VR-Spiel. („RIPMotion“, 2016).

The Walkabout Locomotion System konzentriert sich darauf, dass der User Distanzen mit eigenen Schritten gehen kann, um ein besseres Gefühl für die Umgebung zu bekommen. Der User muss die Wege mit physischen Schritten bis zum Ende des Trackingbereichs gehen. An diesen Stellen hält er die Teleportationstasten beider Controller gedrückt und dreht sich einmal in der physischen Welt um. Die Engine erkennt diese Aktion als eine Erweiterung des geraden Wegs. Hier friert die virtuelle Umgebung ein und wird leicht verschwommen. Dies geschieht um durch die Diskrepanz zwischen den Gegebenheiten der zwei Welten (physische und virtuelle) keine Übelkeit und Schwindel beim Spieler zu verursachen. Nach Loslassen der Tasten stellt sich die Umgebung erneut ein. Somit kann sich der User nach dem Umdrehen weiter nach vorne bewegen. Walkabout ist eine Art Redirected Walking, wobei der User diese Reorientierung bewusst wahrnimmt und sogar selbst kontrolliert. Die Methode wurde von Tekton Games entwickelt und findet unter anderem im Spiel **The Dark Within** ihren Einsatz. („r/Vive - WalkAbout - a new locomotion system for VR games. Presented in a teaser for future horror game called The Dark Within.“, o. J.)

All diese Variationen der Fortbewegung in VR, andere Interaktionen des Users mit seiner Umgebung als auch seine Absichten müssen in Echtzeit an die Engine weitergegeben werden. Hier kommen die Input Devices ins Spiel. Sie sind jene Geräte, die eine Verbindung zwischen dem User und dem Rechner (Engine) ermöglichen.

3.6 Input Devices

Input Devices, auf Deutsch Eingabegeräte, sind physikalische Werkzeuge, die Informationen und die Absichten des Users an die Anwendung übertragen. (Jerald, 2015b, S. 346) Bei ihrer Auswahl müssen, abhängig von der Anwendung, unterschiedliche Parameter beachtet werden. Spricht man von Controllern der verschiedenen Plattformen, spielt die Größe dieser Geräte eine wichtige Rolle, denn sie hat großen Einfluss auf das Handling und die Haptik. Abgesehen davon beanspruchen diese Geräte unterschiedliche Muskeln der Hand bzw. des Arms. Größere Eingabegeräte werden von größeren Muskeln kontrolliert und Kleinere von kleineren Muskeln.

Eingabegeräte werden generell anhand des DOFs kategorisiert. **Degrees of Freedom** bezeichnet die Anzahl der Dimensionen, die durch dieses Gerät manipuliert werden können. (Snyder, 2016) Eine klassische Maus wird auf zwei DOFs eingeschränkt. Die Motion Controller von HTC VIVE haben 6 DOFs. Viele VR-Applikationen, welche Whole-Body Tracking Systeme implementieren, besitzen mehr als 6 DOFs. Eingabegeräte können relativ bzw. absolut sein. Relative Eingabegeräte berechnen und geben den Unterschied zwischen der jetzigen und der vorherigen Messung. Eine Maus ist ein Beispiel für ein relatives Eingabegerät. Ein absolutes Eingabegerät misst die Position bzw. den Unterschied des jetzigen Werts mit einem konstanten vordefinierten Referenzpunkt. **Head Mounted Displays**, wie die Brille von HTC, sind ein Beispiel für ein absolutes Eingabegerät. Buttons gehören zu den Eingabemöglichkeiten die meistens für das Aufrufen einer bestimmten Aktion verwendet werden. Aufgrund der eingeschränkten Anzahl an Zuständen (gedrückt oder nicht gedrückt) sind sie optimal für die Angabe von binären Informationen. Die Verwendung von Buttons ist in einer VR-Applikation mit Vorsicht zu genießen, denn viele Buttons können den User schnell verwirren. (Jerald, 2015b, S. 347) Die Verwendung von Eingabegeräten und ihre Anzahl ist immer abhängig von der Anwendung und der Komplexität der Informationen, welche vom User empfangen werden und letztlich auch von der Zielgruppe.

Es gibt eine Vielzahl unterschiedlicher Eingabegeräte. Der Usertest der vorliegenden Arbeit und detailliertere Recherchen wurden auf eine HTV VIVE System aufgebaut. Als Input Devices wurde auf die Motion Controller von HTV-VIVE zurückgegriffen. Zur Option stand ebenfalls ein Oculus Rift mit entsprechenden Touch Controllern.

Valve, die Entwickler der Lighthouse-Technologie, sind an einem offenen Umgang mit anderen Hard- und Software-Entwicklern interessiert und stellen Details als

auch ihr technisches Know-How für eine Weiterentwicklung zur Verfügung. Dieser selten anzutreffende, offene Umgang führte zu der Entscheidung mit HTC-VIVE zu arbeiten.

3.6.1 HTC VIVE und die Funktionsweise

Das HTC VIVE-System beinhaltet zwei Basisstationen, eine VR-Brille und zwei Motion Controller. Die Technologie, welche Valve in Zusammenarbeit mit HTC entwickelt hat, wird als Lighthouse bezeichnet.

Beide Basisstationen werden auf jeweils zwei Seiten eines Raumes, am besten unter der Decke und mit einem Abstand von höchstens 6 Metern zueinander, installiert. Sie scannen jeweils einen Bereich von 120°. Diese Basisstationen sind nicht miteinander verbunden und müssen nicht gleichzeitig während einer Applikation aktiv sein. Das Tracking mit nur einer Basisstation kann an manchen Stellen ungenau werden, da Brille und Controller durch die Rotationen und Bewegungen des Users leicht in einem Winkel zu der Station stehen können, welche von ihr nicht erkannt bzw. gescannt werden.

Lighthouse ist ein optical navigation system. Diese Systeme nutzen optische Physik um den Grad einer Bewegung zu messen, die sich zwischen einer Quelle und einer Fläche befindet. Wenn ein Lichtstrahl eine Fläche trifft, werden Teile davon absorbiert, weitere Teile werden verstreut und der Rest wird reflektiert. Das Verhältnis dieser Aufteilung ist von der Wellenlänge als auch dem Material der verwendeten Fläche abhängig. Aus den reflektierten bzw. verstreuten Teilen können hilfreiche Informationen gewonnen werden. (Garg, 2013) Grundsätzlich werden dabei nur Winkel und keine Distanz gemessen. Anschließend wird die Zeit gemessen und diese Informationen werden weiterverarbeitet. (Yates, 2016)

In einer Basisstation (Lighthouse 1.0) befinden sich zwei Motoren und Infrarot-LEDs. Infrarotlicht wird von den IR-LEDs erzeugt. Mit dem Aufleuchten signalisieren sie einen Startpunkt. In den Basisstationen befinden sich ebenfalls zwei weitere Motoren, welche einen Laseremitter mit einer Geschwindigkeit von 60 Hz, also 16.6 ms, drehen. Ein Motor deckt somit die vertikale und der andere die horizontale Drehung ab. Kurzum leuchten die LEDs auf und ein Laser scannt die horizontalen 120° ab, LEDs leuchten erneut. In der Zeit, in welcher der Laseremitter eines Motors die Drehung auf 360° bringt, ist der Laser des zweiten Motors an der Reihe und scannt seinen vertikalen Bereich ab. Somit haben die Lighthouses ihre Aufgabe erledigt. Sie sind nicht mit dem Rechner verbunden und senden keine Daten weiter. Der Empfänger jedoch, sei es das HMD (Head Mounted Display) oder die Motion Controller, sind mit mehreren Photosensoren

3 Lokomotion und Navigation

auf der Oberfläche versehen. Diese Sensoren reagieren auf das Aufleuchten und die Infrarotstrahlen.



Abbildung 26: HTC VIVE Head Mounted Display (Stone, 2015)

Nach dem Aufleuchten der LEDs beginnen der HMD und der Controller die Zeit zu messen, bis ein Photosensor einen Strahl erkennt. Das Verhältnis zwischen jener Stelle, wo ein Sensor getroffen wurde und der Zeit, welche bis zu dem Moment vergangen ist, ermöglicht eine mathematische Berechnung um die exakte Position in Relation zu der Position der Basisstation zu ermitteln.

Diese Informationen müssen anschließend der Engine vorgestellt bzw. in diese eingegeben werden. Unreal Engine übernimmt diese Einstellungen so, dass man die Eingaben in zwei Bereichen unterteilen kann. 1: Axis Mapping 2: Action Mapping.

Axis Mapping ist für die Eingabe der Richtungen zuständig.

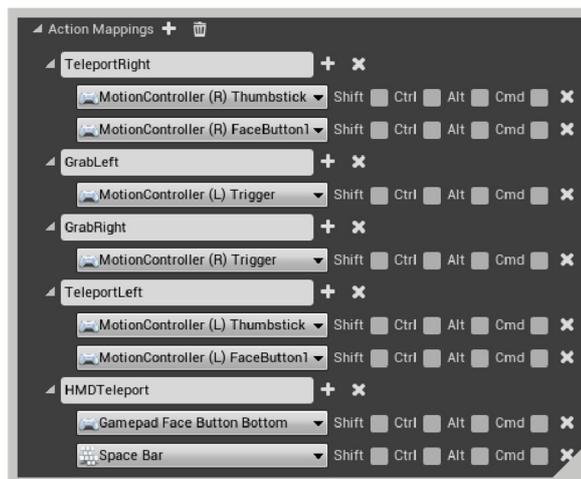


Abbildung 27: Einstellung der Befehle, (Screen Shot aus Unreal Engine 4)

3 Lokomotion und Navigation

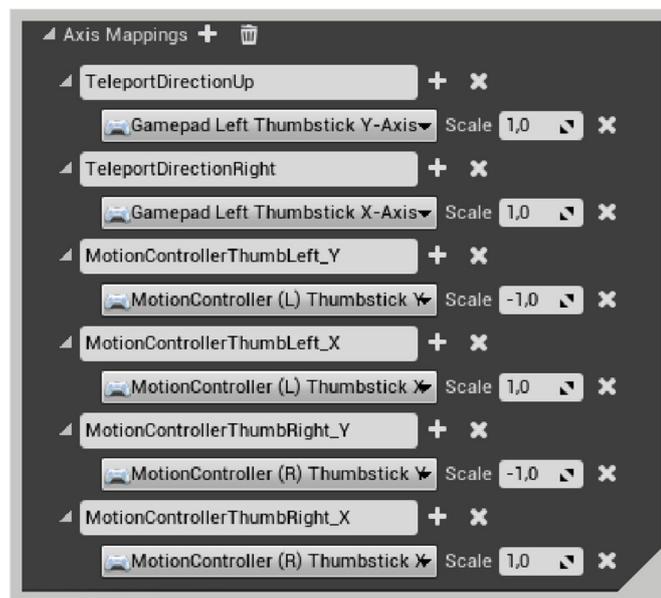


Abbildung 28: Einstellung der Bewegungsachsen, (Screen Shot aus Unreal Engine 4)

Action Mapping ist für Aktionen (bspw. Greifen) verantwortlich.

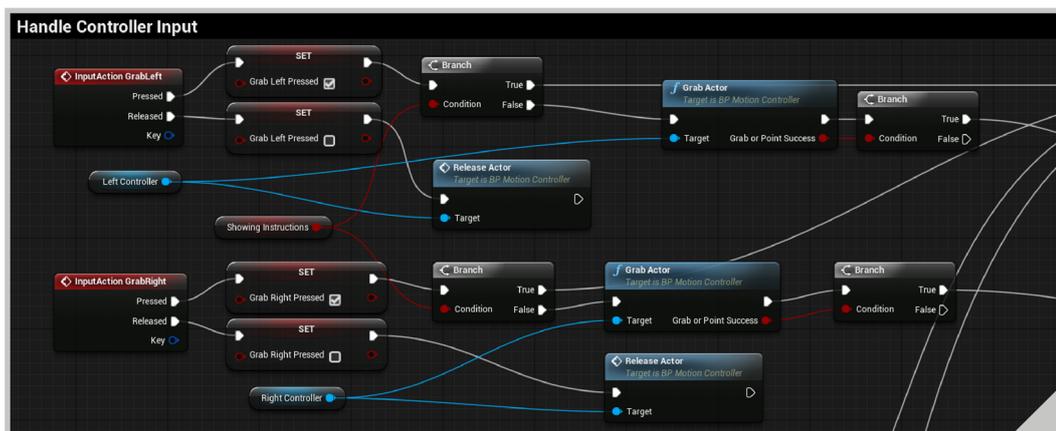


Abbildung 29: Einstellung der Befehleingaben, (Screen Shot aus Unreal Engine 4)

Diese Abbildung ist ein Beispiel für die Implementierung der *InputActions* in einem Projekt. Hier ist zu sehen, dass die Funktion *Grab Actor* aufgerufen wird, sobald die *InputAction GrabLeft* des linken Controllers aktiviert wurde. Dies bedeutet, dass der User, welcher eine darauf gemappte Taste betätigt hatte, das Aufrufen dieses *Event* verursacht. Gleiches gilt für den Controller der rechten Hand sowie anderen Tasten und ihren Belegungen.

3 Lokomotion und Navigation

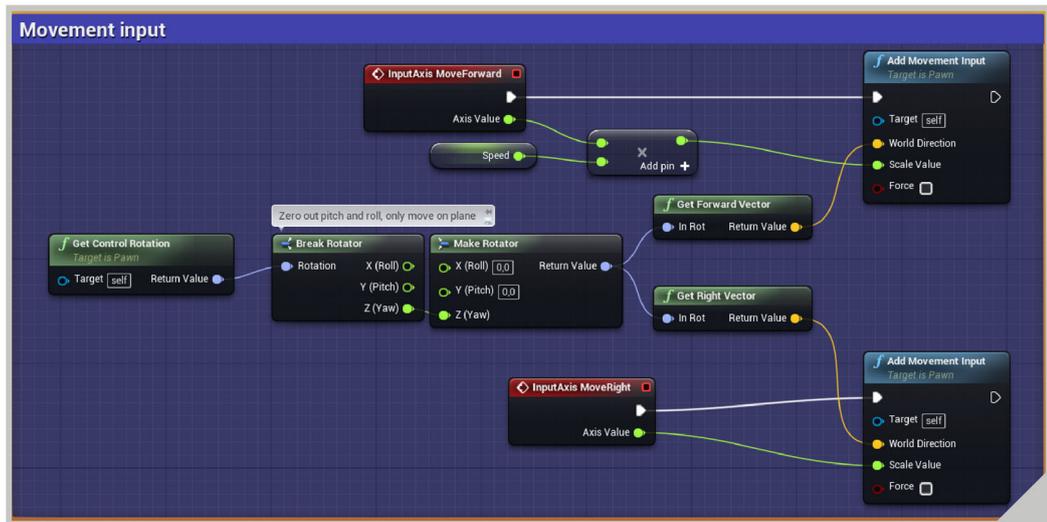


Abbildung 30: Einstellung der Fortbewegung, (Screen Shot aus Unreal Engine 4)

Obere Abbildung ist ein Beispiel für die Steuerung der Bewegung des Characters. Dabei handelt es sich um einen Character, der mit WASD-Tasten gesteuert werden kann. *InputAxis Move Forward* führt zu einer Fortbewegung, die aus der Geschwindigkeit (Speed) und der ausgerechneten Richtung (World Direction) resultiert.

4 Interaktion mit der Umgebung

Warum benötigt man Interaktionen in einer virtuellen Welt? Die Immersion und das Gefühl der Präsenz könnten in VR ebenfalls ohne Interaktion mit der Umgebung funktionieren, sie werden jedoch schnell eintönig. Vergleicht man eine virtuelle Welt in Bezug auf das Verhalten von einem Menschen mit seiner realen Umgebung, realisiert man, dass ein User in einer virtuellen Welt, genauso wie ein neugeborenes Kind, von Anfang an mit ihr kommuniziert. Diese Kommunikation basiert auf Geräuschen, Aktionen und Haptik. (Weitzman, 2016) Die Interaktion mit der Umgebung beginnt somit vor der Sprache. Interaktion als solche zeugt von einer wechselseitigen Beeinflussung des Verhaltens von Individuen oder Gruppen. Im sozialen Bereich wird sie mit Hilfe von Kommunikationsmethoden wie Sprache, Gesten und Symbolen verdeutlicht. („Interaktion“, o. J.) Sie ist also eine Notwendigkeit, um eine Beziehung zu jener Welt zu erzeugen, in welcher man sich befindet.

In diesem Kontext offenbart sich für Interaktion in VR folgende Fragestellung: Wie ist das gesamte Projekt, mit Hilfe eines Zusammenspiels von Hardware und Software, in der Lage, obige Kommunikationsmethoden des Benutzers in zugängliche Arten umzuwandeln? Wie ist es möglich, dem User den Eindruck zu vermitteln, dass er mit dieser Welt eine Beziehung eingehen kann? Dies würde gleichzeitig bedeuten, dass die Interaktion aus Sicht des Benutzers einen einfachen, sofort verständlichen und umgänglichen Entwurf aufweisen sollte. Damit öffnet sich das Tor zu einer der wichtigsten Designansätze bei Interaktionen und speziell bei Interaktionen in VR: *Human Centered Interaction*, der Nutzerorientierten Gestaltung.

4.1 Human Centered Interaction

Bei einer Interaktion zwischen Menschen und den Objekten in VR konzentriert sich die Nutzerorientierte Gestaltung auf die menschliche Seite der Interaktion. Diese Gestaltung gilt als erfolgreich, wenn der Benutzer nicht nur das gegebene Ziel versteht, die Methoden begreift und seine Aufgaben erledigt, sondern ebenfalls, wenn er die Anwendung mit Spaß und Begeisterung, quasi intuitiv, verlässt. Interaktion in VR muss intuitiv sein. „Unter Intuitiv versteht man in der Regel Gedanken oder Eingaben, welche auf unserem Unterbewusstsein beruhen und ohne Reflexion [Nachdenken] zustande kommen.“ („Was ist Intuition oder

4 Interaktion mit der Umgebung

intuitiv?“, o. J.) Es besteht also keine Notwendigkeit, dass dem Benutzer die komplexen Vorgänge und Berechnungen im Hintergrund bewusst sein müssen. Diese komplexen Algorithmen sollten letztendlich der Einfachheit und Verständlichkeit dienen. (Jerald, 2015b, S. 315)

Intuitiv ist ein Objekt oder eine Anwendung, wenn der Nutzer während des Gebrauchs unbewusst auf sein Vorwissen zurückgreift. Dabei kann er mit geringem Aufwand eine effektive und zufriedenstellende Interaktion starten. (Burmester, 2017) Man spricht hier oftmals von der **Interaktionsmetapher**, einem Konzept das auf vorhandenem Wissen des Users aufbaut und den Vorteil offenbart, dass dem User dadurch ein schneller Einstieg ermöglicht wird. Interaktionsmetaphern helfen umgehend, mentale Modelle zu entwickeln, um die Interaktion zu verstehen. **Mentale Modelle** sind hingegen analoge Wiedergaben von Wissen. Sie repräsentieren komplexe Aspekte und Problembereiche in Form von modellhaften Vorstellungen, womit sich die Sachverhalte und Abläufe veranschaulichen lassen. (Spektrum Akademischer Verlag, 2000b)

Donald Norman, Professor für Kognitivismen und Informatik an der University of California und mehreren weiteren Universitäten, beschreibt in seinem Buch „**The Design of Everyday Things**“ Prinzipien, die als *Norman's Principles of Interaction Design* bekannt sind. Sie erklären die wichtigsten Prinzipien des Designs aus Sicht des Benutzers. (Rekhi, 2017) Folgt man dem Ziel, dass der User in einer virtuellen Welt eine hohe Immersion erleben soll, sollte man davon ausgehen, dass er während dieser Zeit von der physischen Welt abgetrennt ist. Informationen, Hilfestellungen oder andere Formen der Unterstützung aus der Außenwelt sind entweder nicht vorhanden oder gelten als Störfaktor. Der Benutzer taucht in eine virtuelle Welt ein und muss sie selbst entdecken. Diese Entdeckungsreise ist entscheidend für sein immersives Erlebnis.

Ein Fundament dieser Interaktion ist der Mensch, sein Äquivalent bilden die Objekte, mit denen er in VR interagiert. Was müssen diese Objekte aufweisen, damit die Interaktion möglich wird? Antworten auf diese Frage finden sich mit den erwähnten *Norman's Principles of Interaction Design*.

4.2 Norman's Prinzipien

4.2.1 Affordanz

Der Begriff Affordanz (Angebotscharacter) wurde vom amerikanischen Psychologen James J. Gibson geprägt, welcher ihn vom englischen *afford* (anbieten) abgeleitet hat. Laut Gibson handelt es sich dabei um Angebote oder Handlungsanregungen der Umgebung, die auf Informationen und funktionsrelevanten Eigenschaften basieren. Sie ermöglichen ein bestimmtes Verhältnis zwischen dem Benutzer und sich selbst, sie signalisieren, was Benutzer mit ihnen *tun* können. (Spektrum Akademischer Verlag, 2000a)

Eine Türklinke weist den User auf eine bestimmte Art der Kommunikation mit der Tür hin. Ein Kippschalter zeigt dem User anhand seiner Form, auf welche Art er mit ihm kommunizieren kann. Um Missverständnisse vorzubeugen sei erwähnt, dass Affordanz keine Eigenschaft eines Objekts ist. Sie beschreibt vielmehr das Verhältnis zwischen dem User und den Eigenschaften eines interagierbaren Objekts. Die Affordanz von einem Objekt in VR wird, im Vergleich zu einer Webapplikation, *realitätsnäher* gestaltet. (Gillies & Xueni Pan, 2018, S. 5)

4.2.2 Signifiers

Zwecks Effektivität muss eine Affordanz wahrnehmbar sein. (Jerald, 2015b, S. 317). Ein *Signifier* ist ein wahrnehmbarer Indikator, der besagte Signale für eine bestimmte Art der Interaktion sendet. Dabei sind Funktionen, Strukturen und Verhalten gemeint, die in Bezug auf dieses Objekt durchgeführt werden sollten. Ein gut ausgewählter Signifier informiert den User, was mit der Affordanz eines bestimmten Objekts möglich ist, bevor er die Interaktion startet. (Jerald, 2015b, S. 317) Signifiers sind in Form von Schildern, Bildern, Labels und 3D-Interfaces in der virtuellen Welt bekannt. Sie können einen User allerdings auch schnell frustrieren, insbesondere wenn sie falsch ausgewählt worden sind. Ein falscher Signifier weist auf eine Affordanz hin, die letztendlich falsch ist. Marcel Nürnberg, Gründer von Firma **SQUAREBYTES Interactive Media**, ergänzte diese Aussage mit dem Beispiel eines Grills, den sie in den Garten eines virtuellen Hauses gestellt haben:

„Wir haben zum Beispiel immer unseren Grillen draußen stehen [...] und er war immer geschlossen [...] und 80% der Leute wollten diesen Grillen aufmachen. Das war für uns ein Zeichen: wir müssen diesen Grillen interaktiv machen. [oder] Jeder Mensch will einen Klodeckel aufmachen. Das

4 Interaktion mit der Umgebung

sind Experiences, die wir durch Probieren mit verschiedenen Gegenständen sammeln konnten, dass Menschen sich so verhalten, wenn sie es sehen“.
(Nürnberg, 2018)

Solche Signale, die von dem User anders als beabsichtigt verstanden werden, können nach einigen Versuchen zu Frustration führen. Wobei man nicht vergessen darf, dass viele Signifiers mit Absicht so gewählt werden, um eine Steigerung der Interaktion zu bewirken. Damit ist beispielsweise eine Tür gemeint, die nur durch bestimmte User geöffnet werden kann. Diese Intentionen sind jedoch Teil der Contenterstellung und müssen dementsprechend entworfen werden, sodass der User diese Absichten begreift.

„Signifiers sind wichtig, um zu verstehen, ob und wie man mit einem Objekt interagieren kann. Egal was sie kommunizieren, sie müssen verständlich sein.“ (Jerald, 2015b, S. 318)

4.2.3 Constraints

Constraints sind Limitierungen der Möglichkeiten. Sie sorgen für Einfachheit und Überschaubarkeit und helfen dem Spieler die Logik des Spiels schneller nachzuvollziehen. Ein bekanntes Beispiel für Constraints in VR sind die Limitierungen bei DOF, den **Degrees of Freedom**. Dabei handelt es sich um Freiheitsgrade, d.h. einer Anzahl von Möglichkeiten eines Users, um sich in einem virtuellen Raum zu bewegen. Ein dreidimensionaler Raum weist sechs Degrees of Freedom auf: Rotationen (yaw, pitch, roll) und die Bewegung entlang der drei Achsen (x, y, z). (Snyder, 2016)

Constraints können sowohl software- als auch hardwarebedingt sein. Manche Eingabegeräte bieten weniger DOF, wie zum Beispiel ein Joystick, der nur Bewegungen entlang zweier Achsen ermöglicht. Einschränkungen wie diese sollten nicht unbedingt als schlecht angesehen werden. Viele dieser Limitierungen sind nützlich und unterstützen realistische Eigenschaften. Beispielsweise kann eine virtuelle Hand gestoppt werden, sobald sie in eine Wand hineinbewegt wird. Bei einer solchen Einschränkung handelt es sich um eine softwarebedingte Limitation, welche der Anwendung eine realitätsnahe Eigenschaft gibt.

4.2.4 Feedback

Unter **Feedback** versteht man die Reaktion oder das Signal einer Anwendung nach einer Aktion des Users, welches ihn auf ein klares Ergebnis hinweist. (Jerald,

4 Interaktion mit der Umgebung

2015b, S. 319) **Force Feedback** stellt wiederum eine physikalische Kraft dar, die auf Steuerelemente ausgeübt wird. („Force Feedback Definition & Erklärung | Informatik Lexikon“, o. J.)

Haptik stammt aus dem griechischen Verb *háptein* und bedeutet *Angreifen* oder *Berühren*. („Duden | Hap-tik | Bedeutung, Definition, Herkunft“, o. J.) Haptik ist also ein Adjektiv, welches die Eigenschaften eines Objekts beschreibt, die den menschlichen Tastsinn ansprechen. **Haptisches Feedback** wird in zwei Bereiche unterteilt: **Kinästhetisches Feedback** und **Taktils Feedback**.

Kinästhetik beschäftigt sich mit Erfahrungen und Wahrnehmungen des Menschen, die durch Körperbewegungen gewonnen werden. Sie ist die Lehre der Bewegungsempfindung. (Stelzl, 2004, S. 3) **Taktil** bezieht sich auf den Tastsinn, d.h. Berührungen. Von Muskeln aufgenommene Reize fallen in den Bereich des kinästhetischen Feedbacks. Hält man beispielsweise eine Tasse in der Hand, übermittelt das kinästhetische Feedback Informationen über Gewicht, Größe und Form dieser Tasse an das Gehirn. Im Gegenzug ist das taktile Feedback für Sinneserfahrung zuständig, in diesem Fall über die Haut weitergeleitete Informationen. So wie hier mit der Tasse sind dies Eindrücke über Material, Textur und Temperatur.

Bei einer VR-Applikation können Feedbacks visuell, auditiv oder haptisch eingebaut sein. Unabhängig davon, welche Art von Feedback in einer Anwendung zum Einsatz kommt, ist es wichtig, dass sie auf eine verständliche Art mit dem User kommunizieren müssen. Sie sollten so entworfen und eingesetzt sein, dass sie den User nicht erraten lassen, ob er eine Aktion erfolgreich abgeschlossen hat oder das Gegenteil der Fall ist. (Rekhi, 2017)

Feedbacks werden zwecks Verständlichkeit eingebaut. Als Entwickler läuft man Gefahr, zu viele Feedbacks in seiner Anwendung zu integrieren. Dies kann den User irritieren, sodass sie nach einer gewissen Zeit ignoriert werden. Die Feedbacks müssen priorisiert werden. Zusätzlich sollte man darauf achten, dass sie nicht aufdringlich sind. In einer visuellen Form können sie beispielsweise rechtsbündig oder linksbündig auftauchen. Auditive Feedbacks mit Überlappungen erscheinen dem User wiederum stressig und störend oder sie werden nach einer gewissen Zeit überhört. (Jerald, 2015c, S. 319) Nimmt man die Vibration von einem Controller als Beispiel für haptisches Feedback, ist es wichtig, dass sie bei einer klaren, verständlichen Aktion vibrieren.

4.2.5 Mapping

Mapping bezeichnet die Beziehung zwischen Aktion und Reaktion. Zwischen einer Aktion, dem Ergebnis und der Intention muss ein verständliches Mapping stattfinden. (Jerald, 2015c, S. 320) Ein gutes Beispiel dafür ist das Verhältnis zwischen den Herdplatten und ihren Drehreglern (Abb. 31).

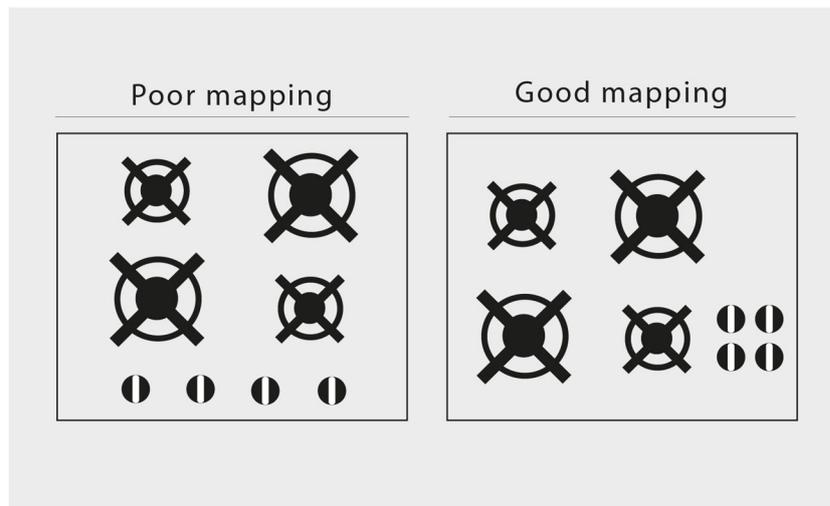


Abbildung 31: Vergleich zwischen gutem und schlechtem Mapping (Rekhi, 2017)

Betrachtet man die rechte Abbildung, begreift man sofort, welche Regler für die oberen und welche für die unteren Platten zuständig sind. Eine solche Aussage ist bei der linken Abbildung nicht möglich.

Interaktion kann direkt bzw. indirekt geschehen. Eine direkte Interaktion bezieht sich auf eine direkte Reaktion von einem Gegenstand nach einer Aktion des Users. Virtual Reality bietet mehr Möglichkeiten direkter Interaktion im Vergleich zu anderen digitalen Technologien. Virtuelle Objekte können direkt auf die Bewegung und Haltung einer Hand gemappt werden. Indirekte Interaktionen sind hingegen auf mehr Kommunikation zwischen dem User (input) und dem Ergebnis (output) angewiesen. Der User müsste ein Interface verstehen, eine Aktion durchführen, auf das Ergebnis warten, dieses Ergebnis analysieren und die Aktion wiederholen. Dieser Zyklus wird als **Cycle of Interaction** bezeichnet. Zwischen direkter und indirekter Interaktion befindet sich die Ebene der **semi-direct interaction**. Das beste Beispiel für eine semi-direct interaction ist die Verwendung eines Sliders für die Steuerung des Lichts. Greift der User den Slider an und realisiert die Lichtintensität, bemerkt er die Beziehung zwischen dem Slider und dem Licht. Je mehr und öfter er diese Aktion durchführt, desto familiärer wird er mit dem Interface und mit der Zeit entwickelt er ein mentales Modell für diese Interaktion. (Jerald, 2015c, S. 322)

4 Interaktion mit der Umgebung

Das Konzept der Interaktion in dem Kopf eines Users kann in zwei Bereiche unterteilt werden: In der **Ausführung** wird er ihm die Funktionalität des Objektes bzw. des Interfaces bewusst. In dieser Phase möchte der User wissen, wie er mit einem Objekt umgehen kann.

Als zweite Phase erfolgt die **Evaluierung**, in welcher der User Sinn, Zweck und Nutzen der Ausführung fokussiert: Er möchte wissen, was er damit verändern und beeinflussen kann. In diesem Kontext ist es die Rolle des Designers, diese zwei Phasen miteinander zu verbinden. (Norman, 2013, S. 38) Bewusst oder unbewusst sind die Gedanken des Users in VR mit diesen Zusammenhängen beschäftigt. Die Menge an Wissen und Erfahrung, die ein User durch Interaktion mit realen Objekten in VR, also virtuellen Objekten, anwenden kann, sagt etwas über die **Interaction Fidelity** aus: Der Ähnlichkeit zwischen der Aktion in der virtuellen Welt mit der entsprechenden Aktion aus der physischen Welt. (Jerald, 2015c, S. 328) Das Spektrum der Interaction Fidelity ist groß: Auf der einen Seite finden sich realistische Formen, auf der anderen entsprechend Unrealistische. - In der Mitte dieses Spektrums finden sich die meisten gängigen Konzepte der Interaktion, **the magical interactions** (Jerald, 2015c, S. 329)

5 Die Virtuelle Welt / Content

Erzählen ist ein Grundpfeiler menschlicher Kommunikation. Menschen kreieren Geschichten und beschreiben Inhalte verschiedener Dinge. Das Interesse der Menschen an Inhalt und Botschaft von Geschichten, d.h. der Narration, ist eine Eigenschaft, die nie außer Acht gelassen werden darf. Ihre Stärke lässt sich in VR erneut beobachten. Botschaften in Form von Geschichten zu vermitteln, ermöglicht es, mehrere Teile des Hirns zu aktivieren. Reduziert man den Inhalt einer Geschichte auf das Wesentliche, finden sich eine Reihe von Ursachen und Wirkungen. Kausalität ist das, was die Menschen immer suchen und zu verstehen versuchen. (Widrich, 2012) Zusätzlich neigt der Mensch dazu, in jeder Geschichte Parallelen, Analogien oder Ähnlichkeiten zu finden und sich mit ihnen, mehr oder weniger, zu identifizieren und diese Informationen miteinander zu verbinden. (Sapolsky, 2010)

5.1.1 Conceptual Integrity

Conceptual Integrity ist einer der wichtigsten Komponenten beim Design eines Systems. Conceptual Integrity beschreibt ein Designsystem, bei dem alle Konzepte und deren Beziehungen ein einheitliches System ergeben. Dafür sollte das Konzept eine durchgehend stringente Idee beinhalten, anstatt mehrere Zusammenhanglose zu präsentieren. („The importance of Conceptual Integrity“, 2011) Fred Brooks fasste 1975 in seinem Buch **The Mythical Man-Month** seine Definition von Conceptual Integrity in Bezug auf Systementwicklung zusammen:

“Conceptual integrity is the most important consideration in system design. It is better to have a system omit certain anomalous features and improvements, but to reflect one set of design ideas, than to have one that contains many good but independent and uncoordinated ideas...Conceptual integrity in turn dictates that the design must proceed from one mind, or from a very small number of agreeing resonant minds.“ („Conceptual Integrity“, o. J., S. 99)

Wenn mehrere zusätzliche Ideen notwendig sind, um das Gesamtkonzept zu verdeutlichen, ist es besser das Grundkonzept zu überarbeiten, statt die Zusatzideen einzubauen. Denn in einer virtuellen Umgebung müssen Strukturen, Zusammenhänge und zusammengehörigen Elemente klar und deutlich sein.

Kompliziertere Strukturen und Ideen können erst eingebaut werden, wenn der User über ausreichend Erfahrung verfügt.

5.1.2 Die Gestalttherorie

Laut der Gestalttheorie werden psychische Phänomene nur dann verstanden, wenn sie als ein organisiertes Ganzes aufgefasst werden können und nicht in einfache perzeptuelle Elemente zerlegt sind. (Bragull, o. J., S. 1) Diese Theorie wurde zum ersten Mal durch den Psychologen, Physiker und Philosophen Ernst Mach im Jahr 1886 definiert. Kernaussage ist der ständige Versuch des Menschen, seine Umwelt in einer für ihn zugänglichen Anordnung wahrzunehmen. Diese Tendenz zum Gruppieren von Objekten anhand ihrer Form, Funktionalität und anderen Gemeinsamkeiten folgt bestimmten Prinzipien, welche in seiner Gestalttheorie in sechs Kategorien gegliedert wurden. (Jerald, 2015b, S. 273)

1. **Gesetz der Nähe:** Objekte, die näher zusammengestellt werden, können aus der Sicht des Users auch als zusammengehörig interpretiert werden.
2. **Gesetz der Ähnlichkeit:** Dieses Prinzip beschreibt die Tendenz dazu, Objekte, welche eine Ähnlichkeit zueinander aufweisen, als zusammengehörig zu erkennen. Diese Ähnlichkeit kann von Farbe, Form, Größe, Helligkeit und weiteren Eigenschaften abhängen.
3. **Gesetz der guten Gestalt (oder Einfachheit bzw. Prägnanz):** Besagt, dass die menschliche Wahrnehmung versucht, unbekannte Formen in bekannte vertraute Formen umzuwandeln. Dadurch benötigt das Gehirn weniger Energie um neue Formen in sein Gedächtnis aufzunehmen. („Gesetz der guten Gestalt und Webdesign“, o. J.)
4. **Gesetz der guten Fortsetzung (oder der durchgehenden Linie):** Das menschliche Gehirn neigt dazu, einen erlangten Richtungsimpuls weiterzuverfolgen. Grund dafür liegt im Interesse der Wahrnehmung an Kontinuität. Dadurch werden Elemente, die sich entlang einer Linie befinden, als einheitlich wahrgenommen. (Abb. 32) (Tanja, 2013)

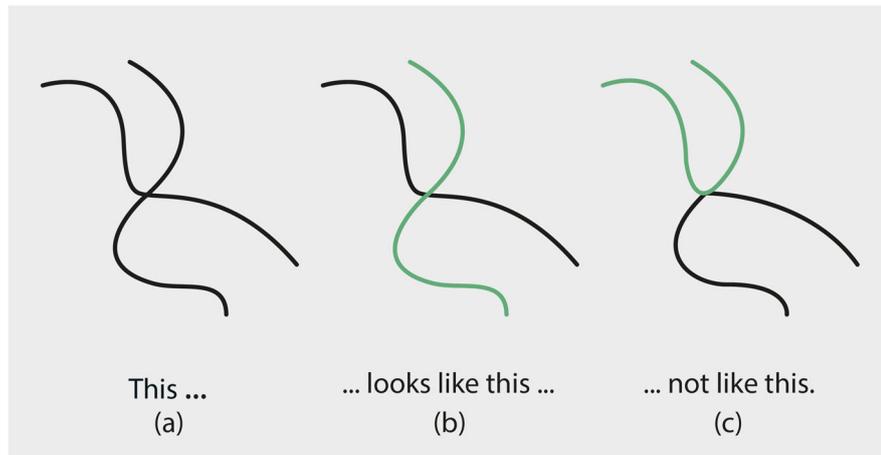


Abbildung 32: Darstellung des Fortsetzungsprinzips (Jerald, 2015a, S. 272)

5. Gesetz der Geschlossenheit:

Strukturen werden in einer geschlossenen Form präferiert. Objekte, die einen Umriss besitzen oder von anderen Elementen umrandet sind, werden als einheitlich und zusammengehörig wahrgenommen. Dieses Phänomen dient der Gruppierung und Zuordnung von Elementen. (Klickkomplizen, 2010)

6. Gesetz des gemeinsamen Schicksals:

Bei Bewegungen eines Elements fällt deren Geschwindigkeit und die Richtung sofort auf. Elemente, die sich in eine ähnliche Richtung und mit einer ähnlichen Geschwindigkeit bewegen, werden von der menschlichen Wahrnehmung als eine Gruppe angesehen.

7. Gesetz der gemeinsamen Region:

Wenn eine Anzahl von Elementen in einem gemeinsamen Bereich von anderen Elementen abgetrennt werden, werden sie als zusammengehörig wahrgenommen.

8. Gesetz der Gleichzeitigkeit:

Wenn bei Elementen eine gleichzeitige Veränderung feststellbar ist, neigt die Wahrnehmung dazu, diese Gruppe von Elementen als eine Einheit zu interpretieren.

9. Gesetz der verbundenen Elemente:

Verbundene Elemente werden ebenfalls durch die Wahrnehmung des Betrachters miteinander in Verbindung gebracht. („Methode_Gestaltgesetze“, o. J., S. 2)

Alle Gesetze dieser Gestalttheorie weisen die Gemeinsamkeit auf, dass sie die Wahrnehmung des Menschen beeinflussen. Menschen neigen dazu, Strukturen, Ordnungen und Zusammengehörigkeiten in ihrer Umgebung mit diesen Gesetzen zu definieren. Ein User versucht anhand der ähnlichen Formen eines Schlosses zu erkennen, welche Tür er öffnen kann. Nun stellt sich die Frage, wie er herausfindet, vor welcher Aufgabe er steht. An dieser Stelle bemerkt man die Wichtigkeit eines Ziels in der Contenterstellung.

Disney unternahm Mitte der 1990er mit ca. 45.000 Usern einen Versuch mit einem Aladdin-Spiel in Form einer Experience. Dabei versuchten sie sich auf die Geschichte und Attraktion zu konzentrieren, obwohl bei dieser Experience ein fliegender Teppich eingebaut war. Die Entwickler sind dabei zu folgenden Einsichten gekommen:

- Für unerfahrene VR-Spieler spielt das Erlebnis eine wichtigere Rolle als die Technologie.
- Eine Hintergrundgeschichte hilft der Immersion.
- Die Geschichte soll klar und simpel aufgebaut sein, ein klares Ziel muss definiert sein.
- Die Glaubwürdigkeit ist wichtiger als eine photorealistische Umgebung.
- Auf Objekte und Charaktere, die nichts zu der Geschichte beitragen, sollte verzichtet werden.

(Pausch, Snoddy, Taylor, Watson, & Haseltine, 1996, S. 201)

Berechtigt ist jedoch die Frage, warum die User ein vordefiniertes klares Ziel brauchen.

5.2 Das menschliche Denken

Finalität besagt, dass das menschliche Handeln immer zielgerichtet ist. (Berner, 2015) Bevor man über jegliche Art des menschlichen Verhaltens spricht, sollte die Frage beantwortet werden, aus welchem Grund die Menschen ein Bedürfnis für diese Form der Kommunikation haben. Der Mensch benötigt immer ein **verfolgbares Ziel**, das direkt oder indirekt seine Handlungen bestimmt. Unter einem direkten Ziel versteht man beispielsweise das Greifen zu einer Kaffeetasche,

als ein indirektes Ziel bietet sich das Bestellen eines Taxis an, um zum Flughafen zu fahren. Das indirekte Ziel ist eine Zusammenstellung aus mehreren kleinen Zielen, die zu einem großen Ziel führen.

Das Verhalten von Menschen kann grundsätzlich aus zwei unterschiedlichen Perspektiven analysiert werden: Einerseits aus der Ursache heraus, d. h. man fragt nach den Gründen einer Person und deren Handlungen. Andererseits hinsichtlich der Ziele, welche das Individuum zu bestimmten Handlungen bewegen.

Die erste Herangehensweise wird als kausale (Lat. *causa*: Grund, Ursache) und die Zweite als finale (Lat. *finis*: Ende, Ziel, Zweck) Erklärung bezeichnet. Beide schließen sich gegenseitig nicht aus. In den meisten Fällen liefert der finale Erklärungsansatz bessere Gründe und stellt das Verhalten des Menschen verständlicher dar. Es muss erwähnt werden, dass dieses zielgerichtete Verhalten nicht immer bewusst stattfindet. (Berner, 2015)

Dieses Wissen unterstützt Entwickler, die Bedeutsamkeit eines Ziels in der Anwendung zu erkennen. Denn es ist für einen User notwendig, den Grund für eine Aktion innerhalb einer Anwendung zu erkennen bzw. in diesem Fall innerhalb einer virtuellen Welt. Ursache und Wirkung können hier ebenfalls mit dem im vierten Kapitel erwähnten Thema *Ausführung und Evaluierung* in Verbindung gebracht werden, denn hier findet der *Cycle of Interaction* seine Definition wieder:

Der User formuliert bzw. setzt sich als erstes ein Ziel oder es wird ihm vorgegeben. Anschließend führt er Aktionen durch, um dieses Ziel zu erreichen. Die nächste Stufe beinhaltet das Evaluieren, d. h. er analysiert das Ergebnis seiner Aktion in Bezug auf das Ziel. (Jerald, 2015b, S. 323)

5.3 Die virtuelle Umgebung

Eine virtuelle Szene kann als die gesamte Umgebung verstanden werden, in der sich ein User befindet und mit den Objekten interagiert. Objekte, die in einer Szene gesehen werden, können grundsätzlich in vier Kategorien unterteilt werden:

Der Hintergrund besteht aus Objekten, welche meist sehr weit entfernt und statisch sind. Statisch weist in diesem Zusammenhang darauf hin, dass man mit diesen Gegenständen nicht interagieren kann. Diese sind beispielsweise der Himmel, die Berge oder die Sonne.

Contextual Geomtry beinhalten Objekte, die zur Definition einer Umgebung führen. Interaktionsmöglichkeit mit diesen Objekten ist ebenfalls eine Seltenheit. Sie helfen dem User bei der Orientierung, verschaffen ihm einen Überblick und

erklären ihm die Struktur dieser Umgebung. Bäume, Straßen, Häuser, Informationsschilder sind Beispiele von Contextual Geometry.

Fundamental Geometry hat, im Gegenteil zu den anderen, eine Affordanz, sie stehen dem User näher. Diese Objekte definieren meist den Interaktionsbereich. Ein Türrahmen, ein Tisch oder die Innenwände eines Hauses können als fundamentale Objekte gesehen werden.

Die letzte Kategorie beschreibt die **Interactive Objects**, sie sind interagierbar, sind meist kleiner und signalisieren ihre Affordanz anhand ihres Aussehens, Farbe und Form. Eine Tasse, Ein Schlüssel und ein Buch können in einer Szene als interagierbare Objekte bezeichnet werden. (Jerald, 2015b, S. 277)

Es ist sowohl eine konzeptionelle als auch eine dramaturgische Frage, wie und an welcher Stelle Objekte gestellt werden. **Segregation** bezeichnet die Trennung eines Objekts von seinem Hintergrund.(Jerald, 2015b, S. 276) Jene Trennung muss dem User so klar wie möglich aufgezeigt werden. Dieses Ziel wird erreicht, indem das Signal dieser Objekte deutlich und aussagekräftig ausgewählt wird. Farbe, Position, Sound und Licht bzw. Schatten können dem Entwickler helfen dieses Ziel zu erreichen.

Die Fertigstellung und Vorbereitung einer Szene entspricht einem gesamten Workflow. Nach der Ideenentwicklung folgt in den meisten Fällen ein Prototyp. Daraufhin geschieht das Modellieren der Objekte. Abhängig von dem Projekt werden die 3D-Assets modelliert bzw. von unterschiedlichen Datenbanken heruntergeladen. An nächster Stelle müssen die Objekte texturiert und ihre Materials bearbeitet werden. All diese Schritte nehmen viel Zeit in Anspruch. Während der Vorbereitung von Assets und 3D-Objekten, die letztendlich in einer VR-Anwendung verwendet werden, müssen sich die 3D-Artists an verschiedenen Richtlinien orientieren.

In der Phase der Fertigstellung jedes Projekts erreicht jede Entwicklung die Phase des Debuggings. Hier müssen Probleme gelöst werden, bei denen oft längere Zeit darin investiert wird, Ursachen zu finden. VR-Anwendungen sind hier keine Ausnahme. Es ist ratsam, dass der Entwickler kontinuierlich regelmäßige Tests durchführt, um gröbere Probleme zu vermeiden. Es ist jedoch verständlich, dass diese Tests in der Endphase des Projekts öfter durchgeführt werden. Die Game Engines bieten gute Möglichkeiten und Tools um Ergebnisse dieses Tests untersuchen und analysieren zu können.

Erstellung von hochqualitativen 3D-Assets für VR-Anwendungen kann an manchen Stellen kritisch werden. Wenn es sich dabei um die Game Logic handelt,

müssen die Probleme an einer anderen Stelle gelöst werden. Hier spricht man von Problemen, an welchen letzten Endes die Performance leiden wird. Dies kann dazu führen, dass Spieler die erwünschte Immersion in einer Szene nicht empfinden bzw. mit einem Gefühl von Schwindel und Übelkeit die virtuelle Umgebung verlassen.

Der folgende Abschnitt erklärt die skizzierte Herangehensweise anhand eines konkreten Beispiels, einer Szene, die für den Usertest vorbereitet werden musste. Die Ansätze, die in diesem Abschnitt angesprochen werden, beziehen sich auf die möglichen Probleme, die bei einer VR-Anwendung behoben werden sollten. Diese können bei anderen Projekten irrelevant sein.

5.3.1 DevTon Gefängnis

Die Umgebung wurde durch den Unreal Engine Marketplace bezogen. DevTon Abandoned Prison ist eine Umgebung, die für eine VR-Anwendung optimiert werden musste. Stellt man das Projekt in einem VR-Template so ein, dass sich ein VR-Player in diese Umgebung teleportieren kann, führt eine nichtflüssige Wiedergabe der Szene schnell zu einem unangenehmen Gefühl. Da dieses Gefängnis sehr realistisch aussieht, kann man von hochauflösenden Texturen ausgehen. Somit ist es naheliegend, dass man an dieser Stelle mit der Analyse des Problems beginnt.

Abbildung 33 ist eine Shader Complexity Darstellung des unteren Bildes, Abbildung 34. Die Informationsleiste unter dem ersten Bild zeigt, dass diese Szene viel zu komplexe Shader beinhaltet. Beobachtet man die Szene in dem Game View, erkennt man Decals, welche in den meisten Bereichen für die rote Anzeige sorgen (Abbildung 33). Die roten Bereiche zeigen die zu hochauflösenden Bereiche, welche vielleicht zu einer ruckeligen Wiedergabe führen.

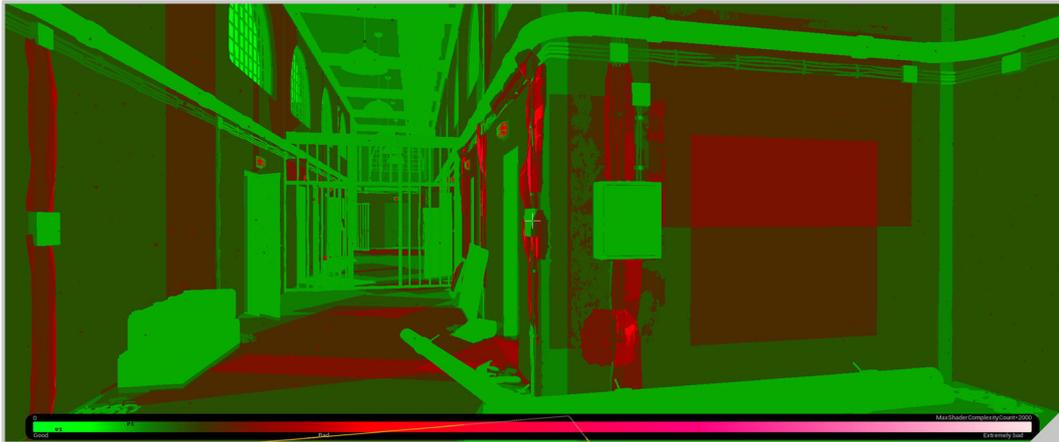


Abbildung 33: Shader Complexity, (Screen Shot aus Unreal Engine 4), (DevTon Studio, 2017)



Abbildung 34: DevTon Prison, (Screen Shot aus Unreal Engine 4),(DevTon Studio, 2017)



Abbildung 35: Darstellung der Decals, (Screen Shot aus Unreal Engine 4),(DevTon Studio, 2017)

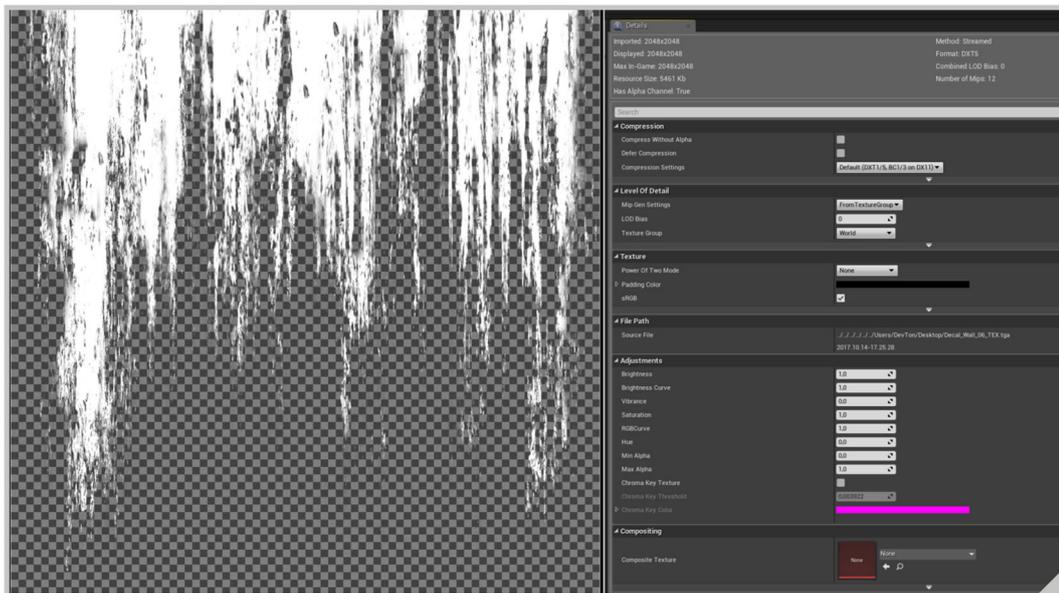


Abbildung 36: Decals und ihre Einstellungen, (Screen Shot aus Unreal Engine 4), (DevTon Studio, 2017)

Hier ist ein Beispiel von einem Decal aus der erwähnten Szene. (Abbildung 36) Dieser Schmutz wurde in Form von abgetrockneten Tropfen, in einer Auflösung von 2048x2048, auf mehreren Wänden des Gefängnisses platziert. Werden zwei dieser Decals ähnlicher Auflösung an einer Wand aufeinandergelegt, wird der Shader von der Engine an dieser Stelle als komplexer interpretiert.

Hochkomplexe Shader in einer VR-Szene müssen nicht unbedingt ein Problem darstellen. Sie können jedoch in Kombination mit mehreren anderen Faktoren, die ebenfalls anspruchsvolle Darstellungen benötigen, das System schnell überfordern, worunter letztlich die Performance leiden wird. Nach dem Entfernen der Decals zeigte ein weiterer Test, dass dies nicht die Ursache des Problems darstellte. Die Wiedergabe war immer noch nicht flüssig.

5 Die Virtuelle Welt / Content

Das GPU-Profilung zeigte bei DevTon Prison keine anspruchsvolle Post Processing Volume. (Abbildung 39) zeigt die Informationen des GPU-Visualizer des dahinterliegenden Frames.

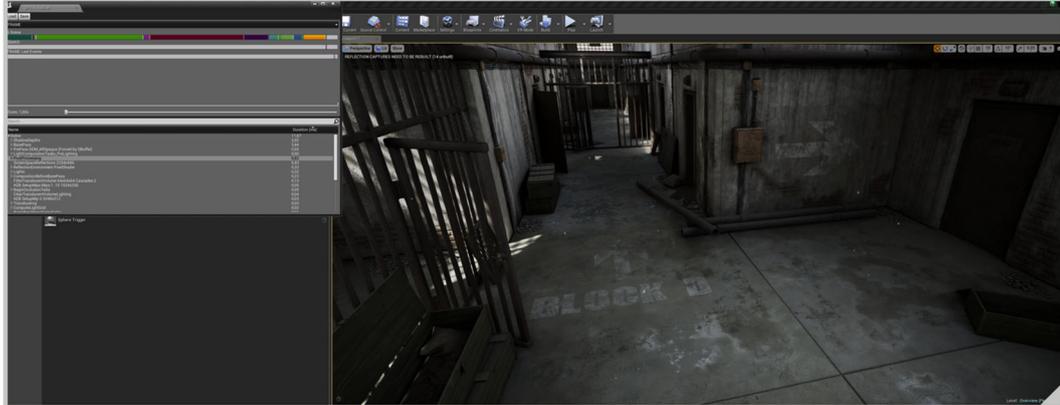


Abbildung 39: GPU-Visualizer für einen Frame, (Screen Shot aus Unreal Engine 4), DevTon Prison (DevTon Studio, 2017)

Wie aus den Abbildungen ersichtlich, wurde klar, dass die Information ebenfalls zu keiner Lösung geführt hat. Nach einigen Recherchen und Versuchen war Trial-and-Error die letzte praktikable Methode. Die ersten Vermutungen gründeten auf den Lichtverhältnissen. Das *Licht* wurde nach Entfernung mancher Lichter testweise erneut berechnet (Baked) Es konnte festgestellt werden, dass die Lichter ebenfalls keine Ursache für das Problem darstellten.

Der erste Versuch der Deaktivierung des Post processing Volume zeigte ein erfolgreiches Ergebnis. Dadurch wurde offensichtlich, dass eine Einstellung an dieser Stelle für aufwendige Rechnungen sorgte. Es hat einige Versuche gekostet um herauszufinden, dass der Screen Percentage auf 200% gestellt war. Die Auflösung wird mit diesem Prozentsatz gerendert. Daraufhin wird sie auf die Bildschirmauflösung herunterskaliert. D. h, dass statt 1080x1200 pro Auge eine Auflösung von 2160x2400, also die vierfache Menge an Pixeln, gerendert wurde, was letztendlich die Wiedergabe verlangsamt hatte.

Während der Versuche gab es, wie oben erwähnt, die Vermutung, dass das Licht in dieser Umgebung ebenfalls zu anspruchsvoll sein kann. Denn nachdem die Lichter testweise entfernt und erneut gebildet wurden, ist das Problem mit den dynamischen Objekten des Gefängnisses auffällig geworden. Dynamische Objekte empfangen das Licht anders als statische Objekte. Sie empfangen das indirekte Licht der Samples und berechnen keine Information in den Lightmaps. Sie berechnen ebenfalls die Schatten anders als die statischen und stationären

Objekte. Dies führt letztlich zu einem anderen Verhalten gegenüber ihrem Umgebungslicht.

DevTon-Projekt beinhaltete statische Objekte und erst nachdem manche Wände und Türen zu den dynamischen Objekten umgewandelt wurden, zeigte sich das Problem in Form von dunkleren bzw. helleren Bereichen. Das unten abgebildete linke Bild zeigt eine Tür in statischer Form. Das rechte Bild zeigt wiederum eine Tür in derselben Umgebung, jedoch umgewandelt zu einem dynamischen Objekt, damit der User mit ihr interagieren kann.



Abbildung 40: Reaktion der dynamischen Objekte (Screen Shot aus Unreal Engine 4), DevTon Prison (DevTon Studio, 2017)

Recherchen zeigten, dass viele Entwickler dieses Problem mit Hilfe der Texturen lösen. Dies bedeutet, dass sie die Farbunterschiede in der Texturfarbe kompensieren. Die bisherigen Recherchen im Laufe der Fertigstellung dieses Projekts ergaben ebenfalls keine bessere Lösung.

Fokussiert man sich auf die Architekturvisualisierung, kann man verallgemeinert sagen, dass das Echte Licht als Richtlinie genommen wird. Dies bedeutet, dass Stimmung und Lichtstärke, Richtung des Lichts und alle weiteren Komponenten so gewählt und miteinander kombiniert werden, dass die Umgebung realistisch wie möglich aussieht. Eine Möglichkeit besteht darin, sich nach der D65 Norm Light zu richten, um eine nach Norm angepasste Richtlinie als Referenz aufweisen zu können.

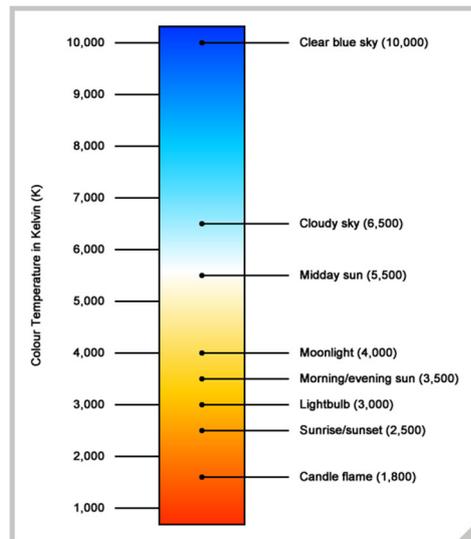


Abbildung 41: Farbtemperaturen in Kelvin, Referenz für Helligkeitseinstellung (Owen, o. J.)

In diesem Kontext wird grundsätzlich von zwei Herangehensweisen gesprochen. Entweder erhalten die Entwickler zu Beginn des Projekts **Referenzbilder**, nach welchen sie sich richten und dementsprechend das Licht in der Szene aufbauen können. Oder sie entscheiden sich für ein **Ziel** und unternehmen eine entsprechende Entwicklung. Sie verwenden beispielsweise natürliche Lichtverhältnisse und versuchen dieses Licht entsprechend zu optimieren. Es sei zu erwähnen, dass das Licht in einer Umgebung eine sehr wichtige Rolle spielt. In VR-Anwendungen kann man davon ausgehen, dass der User in den meisten Fällen zum Licht tendiert. Er schaut in die Richtung des Lichts und leuchtende Objekte wecken seine Aufmerksamkeit. (Kaulich, 2015, S. 42) Schlechte und unpassende Lichtstimmungen können das Gefühl der Immersion zerstören. (Milde, 2017) und gut gesetztes Licht sorgt wiederum für ein starkes Gefühl der Präsenz. An dieser Stelle muss auf den Ausdruck „echtes Licht“ etwas genauer eingegangen werden. Echtes Licht bedeutet nicht, dass die Lichtsetzung realistisch sein muss. Denn dies ist in einer computergenerierten Anwendung unmöglich. In einer virtuellen Umgebung werden Lichter an Stellen gesetzt, die in der Realität nicht möglich wären. Man könnte sagen, dass der Entwickler mit Hilfe seiner Tools das Gefühl des echten Lichtes bewusst *Faken* muss.



Abbildung 42: Screen Shot aus Lighting Academy – Session 3.1 (Milde, 2017)

Das obere Bild, die Abbildung 39, zeigt ein Beispiel der Lichtsetzung in einer Galerie, welche für eine VR-Anwendung erstellt wurde. Es gibt zwei Quellen für das gesamte Licht in dieser Umgebung: Als erstes das Licht, welches von unten auf die Wand und entlang des Bodens nach oben scheint. Zweitens das Tageslicht, welches aus dem Fenster hineinscheint.

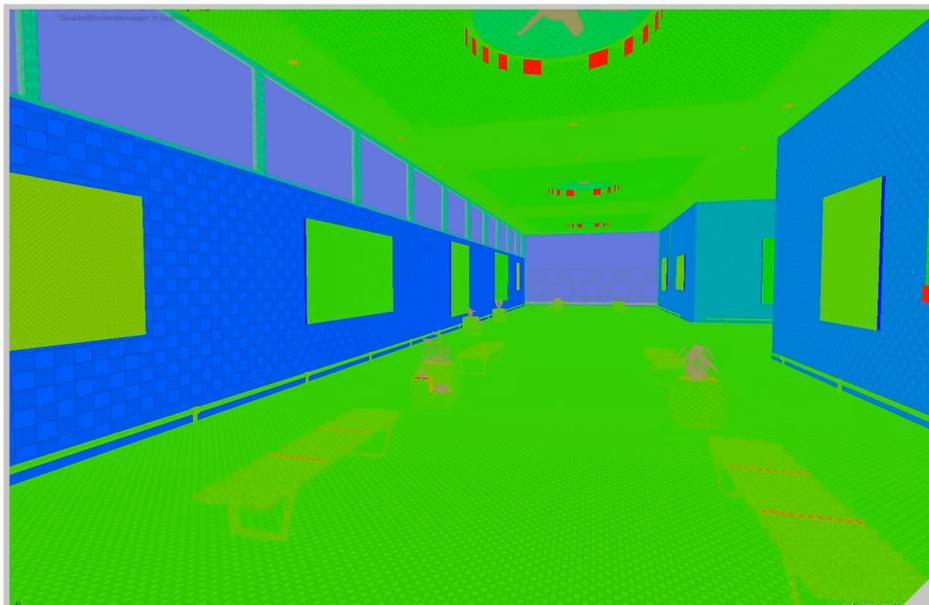


Abbildung 43: Darstellung des Lightmaps, Screen Shot aus Lighting Academy – Session 3.1 (Milde, 2017)

5 Die Virtuelle Welt / Content

Die Lightmap Resolution dieser Galerie weist auf einen Unterschied zwischen den Auflösungen des Lichtes an den rechten bzw. linken Wänden sowie der Decke hin. Dies liegt daran, dass das hineinscheinende Licht durch das Fenster ebenfalls die Schatten der Fensterrahmen auf die Wände wirft, sodass die Schatten der „Metalleisten“ wenige Zentimeter auf die untere Wand treffen und letztlich die gesamte Ästhetik etwas beeinträchtigen. Die Lightmap Resolution dieser Wand ist somit niedriger als jene der gegenüberstehenden Wand. Sie wurde runtergesetzt, sodass der Empfang des Schattens immer noch fein und realistisch aussieht (niedrige Auflösungen können verpixelt erscheinen). In der finalen Version ist zu erkennen, dass die rechte Wand etwas dunkler zu sein scheint, was in den Augen des Betrachters als natürlich und „richtig“ erkannt wird. (Milde, 2017)

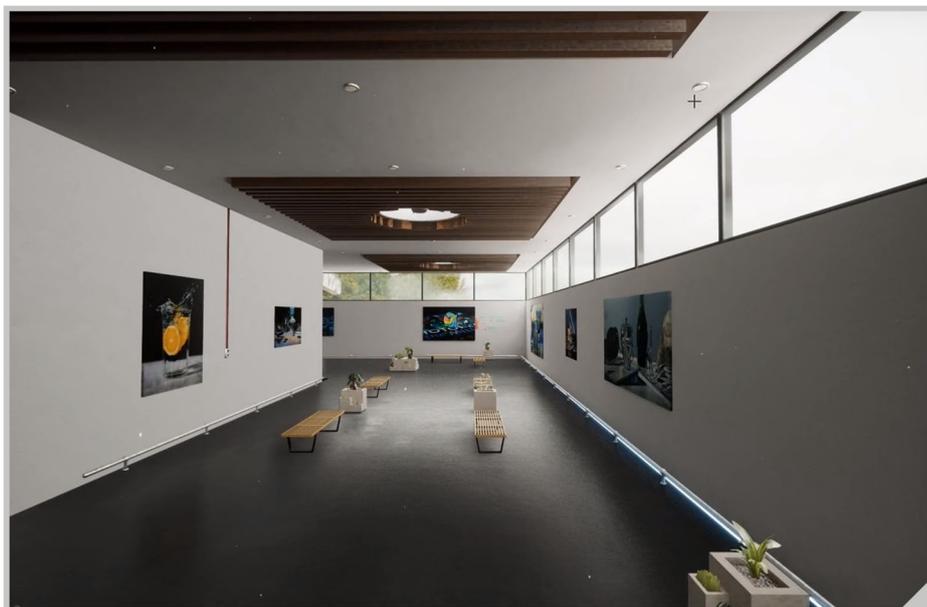


Abbildung 44: Finale Lichtberechnung (*baking*), Screen Shot aus Lighting Academy – Session 3.1, (Milde, 2017)

6 Umsetzung / Usertest

Dieses Kapitel beinhaltet den Usertest anhand dessen das gewonnene Wissen umgesetzt und geprüft wird. Laut der Hypothese dieser Arbeit ist es immer besser, sich bei der Interaktion mit der Umgebung und dem Verhalten der Objekte nach der Wirklichkeit zu richten. Um diese Aussage bestätigen bzw. ablehnen zu können, wurde bei diesem Test die Affordanz der Objekte in einer virtuellen Welt unter die Lupe genommen. Im Kapitel 6.2 (Contenterstellung) wird der Hintergrund für die Auswahl dieses Tests näher erklärt.

Im ersten Abschnitt werden Ideen und Gedanken über die Erstellung inhaltlicher Themen beschrieben. Nach einer kurzen Erklärung über die Gemeinsamkeiten und Bedingungen, welche der Test in allen vier Umgebungen einheitlich liefern soll, werden die Ideen jedes Levels einzeln vorgestellt. Es wird dargestellt aus welchem Grund und aufgrund welcher Informationen und Hypothesen die Inhalte konzipiert sind, wer bzw. was getestet wird und letztlich warum es in dieser Art passiert. Der zweite Abschnitt beschäftigt sich mit dem technischen Teil dieser Aufgabe. Es handelt sich dabei um die Umsetzung jener Ideen, welche im ersten Abschnitt erklärt wurden.

Eine Auswertung des Tests findet im Anschluss statt. Er besteht vor allem aus tabellarischen Wiedergaben jener Informationen, die aus den durchgeführten Tests mit 25 Probanden entstanden sind. Es ist einerseits aus zeitlichen Gründen und andererseits aufgrund der begrenzten Dimension dieser Arbeit nicht möglich, Feedback von Experten, die ebenfalls bei diesem Test partizipiert haben, einzubauen und den Test somit zu perfektionieren. Deren Meinungen und Vorschläge werden im Anschluss erläutert. Der Test wird für alle Interessenten für eine Weiterentwicklung online gestellt.

Probanden mit wenig Erfahrung in VR benötigen stets die Hilfe einer weiteren Person um ihren Start problemlos zu meistern. Ihre Anwesenheit ist notwendig um sich sicher zu fühlen und aus den gleichen Ausgangspositionen zu starten.

Somit war die Autorin dieser Arbeit selbst bei allen Tests anwesend, beobachtete das Verhalten ihrer Probanden und deren Reaktion auf verlangte Aufgaben. Aus diesem Grund bildet ein persönliches Fazit den informativen Abschluss dieses Kapitels und dient ebenfalls dazu, die Hypothese dieser Arbeit anhand von Beispielen zu analysieren.

An dieser Stelle sei erwähnt, dass Entscheidungen bei diesem Test auf eigenen Erfahrungen und Gedankengänge ruhen. Es werden nicht nur die Probanden getestet, sondern das eigene Wissen als auch die Richtigkeit der Gedankengänge der Entwicklerin. Der von ihr vorgeschlagene Weg ist weder alternativlos noch der Beste. Er stellt nur einen Möglichen dar, mit denen Projekte ähnlicher Art umgesetzt und ausgewertet werden können.

6.1 Vorbereitung des Tests

6.1.1 Software und Hardware

Der Test wurde im Rahmen dieser Masterarbeit im Juni 2018 mit Epic Games Unreal Engine, Version 4.19.2 („Unreal Engine | Blog“, o. J.) von Simon Laburda und Arian Jalaeefar, die Autorin dieser Arbeit, erstellt.

Die Umgebungen in diesem Usertest basieren auf Assets und Produkten, die über den Marketplace von Epic Games erworben wurden.

Das erste Level ist eine Küche, die von EVERMOTION modelliert und über den Marketplace bezogen wurde. (Evermotion Kitchen, o. J.)

Das zweite Level ist eine vereinfachte Version von einem ebenfalls heruntergeladenen Schlafzimmer von Evermotion. („Archinteriors for UE vol. 3 - Evermotion“, o. J.)

Bei dem dritten Level handelt es sich um ein modulares Modell eines Gefängnisses, das von DevTon Studio entwickelt wurde. (DevTon Studio, 2017)

Gabro Media sind die Entwickler der Assets des vierten Levels dieses Tests, einer Werkstatt. („Garage Workshop by Gabro Media in Environments - UE4 Marketplace“, o. J.)

Wie zu erwarten, müssen die Umgebungen für eine VR-Anwendung optimiert werden. Dies erfolgt in zwei Schritten: Als erstes werden die Assets in einem Virtual Reality Template von Unreal Engine importiert und die nötigen Anpassungen durchgeführt, damit sich ein VR-Player in diesen Räumlichkeiten bewegen kann. Dieser Schritt wird im zweiten Abschnitt näher erläutert. Anschließend sind die Assets in der Reihe. Da diese Assets für Visualisierungen entworfen sind, müssen sie meistens für VR-Anwendungen optimiert werden, d. h. Skalierungen, Kollisionen und Texturgrößen müssen angepasst werden. Die notwendigen Änderungen an diesen Assets wurden für diesen Test mit Autodesk Maya 2018 durchgeführt.

Der User kann sich während des Tests in einem physischen Bereich von 4.1 Metern bewegen. Dieser Bereich entspricht dem Trackingbereich zwischen den Basisstationen des HTC VIVEs. Der Test wurde mit einer HTC VIVE HMD entwickelt und ist mit Oculus Rift nicht kompatibel. Für die Schnittstelle zwischen HTV VIVE und dem Computer sorgt SteamVR.

Ein Betreuer, in diesem Fall die Entwicklerin, ist während des Tests anwesend, sorgt für die Sicherheit des Users und für einen gleichbleibenden Start des Tests bei allen Probanden.

Die physischen und technischen Bedingungen wurden so ausgewählt, dass sie im Laufe des Tests gleichbleiben können. Denn jede Veränderung kann die Plausibilität der Ergebnisse beeinträchtigen.

6.1.2 Contenterstellung

Die bisherigen Recherchen haben gezeigt, dass sich die Möglichkeiten von Visualisierung und Interaktion in einem Spektrum bewegen, in welcher die eine Seite der Realitätstreue sowie dem Fotorealismus und die andere Seite der Abstraktion entspricht.

Der User wird sich im Laufe dieses Spiels in vier unterschiedlichen Räumlichkeiten bewegen. Seine Interaktion mit den darin befindlichen Gegenständen soll sich in einer Spanne von *realistisch* bis zu *unrealistisch* bewegen. Damit ist nicht nur die Erscheinung von Objekten in einer Form gemeint, welche der Realität entspricht, sondern auch die Affordanz der Objekte.

Wie bereits in Kapitel 4.2.1 erwähnt, handelt es sich bei der Affordanz nicht um die Charaktereigenschaft eines Objekts, sondern um die Beziehung zwischen den Möglichkeiten bzw. den Umgangsfähigkeiten des Benutzers und den Eigenschaften eines Objekts. (Jerald, 2015b, S. 317) Auf einen gebräuchlichen Gegenstand bezogen kann man behaupten, dass der Henkel einer Tasse keine Affordanz dessen ist. Denn Affordanz kommuniziert die Art und Weise, wie man die Tasse als Benutzer anheben kann, d. h. diese Beziehung zwischen der Tasse und ihrem Benutzer stellt die Affordanz der Tasse dar.

Der Grund für die vier Umgebungen bzw. Level liegt in dem Verlangen begründet, den Schwierigkeitsgrad der Aufgaben zu steigern. Natürlich kann diese Steigerung als eine subjektive Wahrnehmung angesehen werden, was sich am Ende des Tests auch feststellen lässt. Man könnte hinterfragen, wonach sich dieser Schwierigkeitsgrad richtet. Die Entwicklung orientiert sich an dem Drang eines jeden Menschen, solche Aktionen durchzuführen, zu denen er in der Realität nicht

in der Lage ist bzw. der Schwierigkeit, diesen Wunsch in der echten Welt in die Tat umzusetzen. Aktionen dieser Art beziehen sich bei dem vorliegenden Test ausschließlich auf Objekte, welche in Verbindung zur Innenarchitektur stehen. Daher stellt der Test seine Spieler vor **vier Aufgaben**, welche diese, in einem entsprechenden Level und jeweils unabhängig und eigenständig, lösen soll:

Das Experiment möchte feststellen, ob die Probanden ihr Wissen über die reale Welt in VR zu Rate ziehen können.

“If a user can interact using natural skills, then the application can take advantage of the fact that the user already has a great deal of knowledge about the world.“
(LaViola u. a., 2017, S. 4)

Der Test soll daher ebenfalls eine Überprüfung dieser These darstellen, denn basierend auf dieser These wurde die Hypothese formuliert.

Bei einem Experiment müssen alle Elemente, außer dem zu testenden Faktor, gleichbleiben, damit das Ergebnis plausibel ist. (Rossmann, o. J., S. 6) Es ist demgemäß wichtig, dass die Art und Weise der Zielsetzung, die Eingabemöglichkeiten, die physische Welt und alle Faktoren, welche einen Einfluss auf die Auswertung haben können, durchgehend identisch bleiben. Die Steigerung bezieht sich auf die Affordanz der Gegenstände in einer Art, die man von der realen Welt nicht erwartet.

Zum Beispiel: Eine Wand soll realistisch texturiert werden und anderen Wänden in dieser Umgebung ähneln. Diese Wand zeigt aber dem Benutzer anhand ihrer Position, dass man sie verschieben kann. Das Verschieben als eine Interaktion mit der Umgebung als Ziel, die Wand und ihre Form, ihre Funktionalität und Affordanz, sind Elemente dieses Experiments. Außer der Affordanz bleiben die restlichen Faktoren gleich und realistisch. Und dieser Faktor wird bei jeder Stufe des Tests realitätsfremder. Damit sollte der Test überprüfen können, ob das Wissen über die Realität in Bezug auf die Interaktionsmöglichkeiten mit diesen Objekten in einer virtuellen Welt hilfreich ist. Es wird somit vorausgesetzt, dass die Affordanz der Objekte in diesen Umgebungen keinen großen Unterschied zu dem darstellt, was das gleiche Objekt in der Wirklichkeit andeutet. Die Eigenschaften der Objekte müssen, soweit es durch die technischen und ergonomischen Einschränkungen möglich ist, eingehalten werden.

Die Affordanz der Objekte zu testen weist ebenfalls darauf hin, dass das Spiel auf jegliche Hinweise für den Benutzer verzichten muss. Mit Hinweisen sind visuelle bzw. auditive Signale gemeint, die auf eine Interaktion hindeuten. Beispiele dafür

sind blinkende Lichter, bunte oder leuchtende Einrahmungen von interagierbaren Objekten oder sogar Informationen in Form von Texten oder Menüs. Solche Hinweise sind stark und präsent, sodass sie die Affordanz der Objekte in den Hintergrund schieben würden und die Aufgabe der Informationsvermittlung übernehmen.

Als Entwickler eines interaktiven VR-Spiels darf man jedoch den Umgebungsentwurf aus Benutzersicht nicht außer Acht lassen. Es ist nicht nur von Bedeutung, was die Objekte signalisieren, sondern auch wie die menschliche Wahrnehmung diese Signale empfängt. Letztendlich weisen diese Umstände auf die Fragestellung hin, wie der Mensch sich in einer VR-Umgebung fühlt und wie er, auf sich allein gestellt, die ihm verfügbaren Möglichkeiten erkennt.

Wie denkt und handelt der Mensch? Teleologisch gedacht benötigt jeder Spieler ein klar definiertes Ziel um sich im Klaren darüber zu sein, in welche Richtung und mit welchen Hintergedanken er sich fortbewegen soll. Ein Ziel führt also dazu, dass eine Richtung und die dafür notwendigen Handlungen dem Spieler bewusst werden. Dieses Ziel soll den Spieler in VR zum Handeln animieren, denn ohne dieses Ziel würde sich der Spieler in einer virtuellen unbekanntem Welt schnell verloren fühlen.

Gleichzeitig bedeutet dieser Ansatz jedoch auch, dass dem User ein Ziel gesetzt wird um sein Verhalten an einer anderen Stelle und beim Erreichen eines anderen Ziels zu analysieren. Somit ist es von Bedeutung, dass die Ziele so definiert werden, dass seine Bewegungen und sein Verhalten in Bezug auf Objekte, die gezielt für den Test vorbereitet wurden, analysiert werden können. Zum Beispiel wird der User vor die Aufgabe gestellt einen Raum zu betreten. Sein Ziel ist in diesem Fall, dass er den Raum betritt. Der Test überprüft jedoch die Affordanz einer Türklinke, d. h. ob diese dem User das Betreten in den anderen Raum verständlich übermittelt oder nicht. Diese Frage könnte zu Beginn einer Aufgabe etwas übertrieben scheinen, denn das Öffnen einer Tür besitzt eine relativ große Realitätsnähe. Je unrealistischer die Aufgaben allerdings werden, desto wichtiger wird es, die Diskrepanz zwischen dem zu erreichendem Ziel und der zu überprüfenden Affordanz bewusster wahrzunehmen.

Wenn das Öffnen einer Tür mit Hilfe einer Türklinke in VR das Ziel für den Benutzer wäre, müsste der Test im Hintergrund die Art und Weise jener Bewegungen analysieren, mit denen er dieses Ziel erreicht. Wie lang braucht er um die Tür zu öffnen? Entwickelt er umgehend ein Verständnis wodurch er die Tür öffnen kann? Woher weiß der User welche Türe er öffnen und welche er nicht öffnen kann? Welche Tür signalisiert ihm diese Funktionalität?

Je unrealistischer und abstrakter Ziele in VR werden, desto schwieriger wird es, eine direkte Kommunikation verständlich darzustellen. Wenn die Interaktion mit der Zeit in VR für den Benutzer das Ziel wäre, musste der Test im Hintergrund sein Verhalten unter die Lupe nehmen, wie er auf die Funktionalität von den Gegenständen kommt, die ihm bei diesem Ziel behilflich sein können. Hier wird diese Diskrepanz deutlich. Oft kommen in solchen Fällen Hilfsmittel wie Slider, 3D Interfaces, Menüs, Informationstexte usw. ins Spiel. An dieser Stelle sind die User meist mehr damit beschäftigt, mit diesen Hilfsmitteln zu hantieren, als die Affordanz der Objekte, welche eigentlich im Mittelpunkt stehen soll, zu erkennen. Dabei geht es nicht mehr darum, eine unrealistische Interaktion durch zu führen, sondern die Funktionalität dieses Interfaces zu begreifen. In solchen Fällen ist es empfehlenswert, dass die Interfaces leicht, deutlich und so minimalistisch wie möglich entworfen werden. Ein Beispiel für eine realistische Interaktion ist das Anheben von leichten greifbaren Objekten, wie eine Tasse, eine Gabel oder ein Teller.

Wie zeigt aber eine Wand -ohne direkte Hinweise darauf-, dass man sie ziehen oder schieben kann?

6.1.3 Zieldefinition

Hier wurde auf das Wissen und die Herangehensweise hinsichtlich der Zieldefinition bei der Filmdramaturgie zurückgegriffen. Während der Contenterstellung wurde darauf geachtet, den Probanden die Motivation zu geben, sich in dem Raum bewegen zu wollen. Dafür muss sich der Spieler zuerst in einer Umgebung sicher fühlen. Dadurch kann man davon ausgehen, dass er sich zutrauen wird, sich auf ein vordefiniertes Ziel zuzubewegen.

An dieser Stelle muss jedoch kurz erwogen werden, was die wichtigsten Faktoren sind, damit sich ein Mensch in einer Umgebung sicher fühlt. Das Gefühl der Sicherheit in einer Umgebung steht in Verbindung zu der Kontrolle über diesen Raum und die Geschehnisse. Das bedeutet für einen Spieler, dass er sich in einem Raum befinden soll, in welchem er schnell und leicht dieses Gefühl gewinnen kann.

Ein Spieler ist ein Charakter in einer Welt, der das Geschehen vorantreibt, der die Handlung in der Hand hat und Entscheidungen trifft, die zielgerichtet und verständlich sind. Das sind alles Eigenschaften eines Protagonisten in einem Film. (Hopp, o. J.) Ein Zuschauer kann gegenüber dem Protagonisten eines Films Empathie empfinden, wenn er in der Lage ist, sein Verhalten und seine

Überlegungen hinsichtlich des vordefinierten Ziels nachzuvollziehen und dadurch in die Geschichte eintaucht. (Hopp, o. J.)

In VR ist der Zuschauer selbst der Protagonist: Er handelt, kontrolliert und trifft Entscheidungen. Wenn ihm in einer scheinbar kontrollierbaren Welt eine Aufgabe gestellt wird, könnte man behaupten, dass das Gefühl der Präsenz in dieser Umgebung vorhanden sein kann.

Man muss jedoch auch wieder in Erinnerung rufen, dass die Art der Aufgabenstellung in allen Leveln gleichbleiben muss, da die Aufgabe als Ziel des Spiels ein entscheidender Faktor dieses Experiment ist.

Think Negative Technik

Dieser Test läuft, wie bei vielen Simulationen dieser Art, Gefahr von den Probanden abgebrochen zu werden, wenn ein Gefühl der Motion Sickness, Angst oder Unsicherheit bei unerfahrenen Spielern auftritt. Aus diesem Grund müssen die Ziele so schnell wie möglich verstanden werden.

Think Negative besagt, dass viele Emotionen und Kommunikationen in einer Szene nur mit Hilfe ihrer gegenteiligen Inszenierungen möglich sind. Ein bekanntes Beispiel ist in dem Fall das Zeigen von Hunger. An und für sich ist dieses Gefühl mit keinem Bild darzustellen. Eine Möglichkeit, um Hunger zu inszenieren, ist das Zeigen einer Mahlzeit. (Wishin, o. J., S. 2)

In anderen Wörtern beschäftigt sich das Medium Film oft mit dem Thema, das Unsichtbare sichtbar zu machen. Emotionen und Gefühle sind unsichtbare, jedoch bekannte Begriffe für alle Menschen. Untersucht man diese Aussage etwas genauer, kann man das Gefühl von Hunger beispielsweise in „Der Wunsch nach Essen“ umformulieren. Bei der Think Negative Methode geht es darum, die unsichtbaren Emotionen und Situationen in einer allen zugänglichen Form umzuwandeln um sie für den Zuschauer und im Fall von VR für den User greiflich und verständlich zu machen.



Abbildung 45: Charlie Chaplin in *The Gold Rush* (Charlie, 1925)

In diesem Beispiel (Abbildung 45) bereitet Charlie Chaplin in einer bekannten Szene des Films *The Gold Rush*, 1925, ein Festessen aus seinem Schuh, welcher am Ende aufgegessen wird. Er zeigt quasi das Gegenteil von dem, was er aussagen möchte, um die Aufmerksamkeit auf das Ziel zu lenken. Bezeichnet man das was er sagen möchte als das „Unsichtbare“ Element in der Szene, kann man behaupten, dass er mit dieser Methode den großen Wunsch nach Essen in Erfüllung zu bringen versucht. Je größer und stärker der Drang nach Erfüllung eines Wunsches inszeniert wird, desto stärker und intensiver fällt das Fehlen dieses Aspekts auf.

Überträgt man die Idee dieser Szene für die Zielsetzung des Usertests, könnte man versuchen den Spieler in einen kleinen Raum zu stellen, um ein Gefühl der Einengung verstärkt zu stimulieren. Mit diesem Gefühl sollte er das Bedürfnis erhalten, sich mit Hilfe seiner „magischen“ Kräfte aus diesem Raum zu befreien. Damit wird die Freiheit als das „unsichtbare“ Element in die Hand genommen, um das Bedürfnis nach Erreichen des Ziels in einem Gefängnis, welches das Gegenteil von Freiheit präsentiert, spürbar zu machen.

Basierend auf der Think negative Technik sind Ziele bzw. Aufgaben in den vier Umgebungen/Levels wie folgt definiert:

6 Umsetzung / Usertest

Level 01: Kannst du bitte die Küche aufräumen? Ich hatte leider keine Zeit. Danke!

Level 02: Bitte richte das Zimmer ein!

Level 03: Befreie Dich!

Level 04: Auf dem Tisch neben der Wanduhr liegt ein Zettel, worauf deine Aufgabe für dieses Level steht.

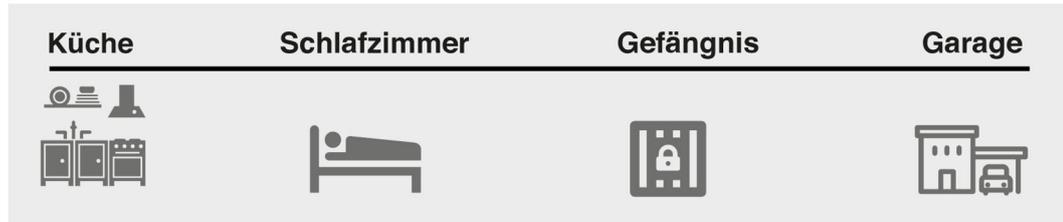


Abbildung 46: Reihenfolge der Umgebungen des Usertests, (eigene Darstellung)

Es ist offensichtlich, dass nicht nur die Art der Interaktion anspruchsvoller wird, je weiter man sich von der Realität entfernt, sondern auch die Möglichkeiten der Entwicklung. Je abstrakter die Interaktionen werden, desto schwieriger wird es auch die Aufgaben zu formulieren. Dieses Problem wurde vor allem während des Entwerfens des Ziels von Level 04 offensichtlich.

6.1.4 Eingabemöglichkeiten

Aus inhaltlicher Sicht betrachtet, sind Eingabemöglichkeiten die Schnittstellen zwischen dem Benutzer und der Engine, welche als erstes erlernt werden müssen. Sie sind *DIE* Möglichkeit, die Intention des Users an die Engine/Anwendung weiterzugeben. Die Zielgruppe dieses Tests setzt leichte, schnell erlernbare Interaktionen voraus und diese Eingabemöglichkeiten müssen so angelegt sein, dass sie im Laufe des Spiels gleichbleiben können.

In jedem Level des Tests ist der Spieler mit einer Art der Affordanz konfrontiert, die ihm aus seiner Realität und aus anderen, bis jetzt gespielten, Umgebungen nicht bekannt sind. Also müssen die Interaktionen in einem Rahmen definiert werden, welcher mit den Eingabemöglichkeiten für den Spieler eine leicht zugängliche Art darstellen und vor allem in allen Leveln gleichbleiben.

Zu Beginn war erwünscht, dass alle Bewegungen mit nur einer Eingabetaste der Motion Controller stattfinden. Die erste Interaktion mit einer Umgebung ist eine passive Interaktion: Die Übertragung der Kopfbewegung in die virtuelle Welt. Der Benutzer setzt die VR-Brille auf, beobachtet und scannt dadurch die virtuelle Welt um sich herum. Diese Kommunikation mit seiner Umwelt ist mittlerweile eine

6 Umsetzung / Usertest

Selbstverständlichkeit, sodass sie kaum noch als Kommunikationsform angesehen wird. (Gillies & Xueni Pan, 2018, S. 4)

Ziel war an dieser Stelle die Möglichkeit des Teleportierens, sobald der Controller auf den Boden gerichtet wurde. Zusätzliches Ziel stellte die Möglichkeit des Greifens dar, wenn die Hand per Knopfdruck zu einer Faust geballt wird und ein interagierbares Objekt näherkommt. Das Touchpad wurde ebenfalls durch diesen Trigger ersetzt, damit der Benutzer mit nur einem Knopf alle Aktionen durchführen kann. Schnell wurde klar, dass das Teleportieren oft aus Versehen stattfand, während die Benutzer den Trigger für das Greifen betätigt haben. Plötzlich sprang der User an eine andere Stelle, obwohl er eigentlich die Hand zum Greifen benutzen wollte.

Einige wenige Versuche waren ausreichend, um festzustellen, dass das Betätigen von nur einer Taste eine für den User leicht zu merkende Aufgabe ist. Die unterschiedlichen Reaktionen einer Taste führen jedoch zu Verwirrung. Abgesehen davon wurde das Spiel schnell frustrierend, weil die erwünschte Aktion von der Engine falsch interpretiert werden kann und dies wird einen unerfahrenen Spieler schnell überfordern.

Die Änderungen wurden daraufhin auf die von UE4 vorgeschlagenen Einstellungen zurückgesetzt. Die Einstellungen definieren zwei Eingabetasten, eine für das Teleportieren und die andere für das Ballen der Faust, d. h. um zu Greifen und mit interagierbaren Objekten zu interagieren. Diese zwei Tasten entsprechen auf den HTC VIVE Motion Controllern dem Trigger und dem Touchpad.

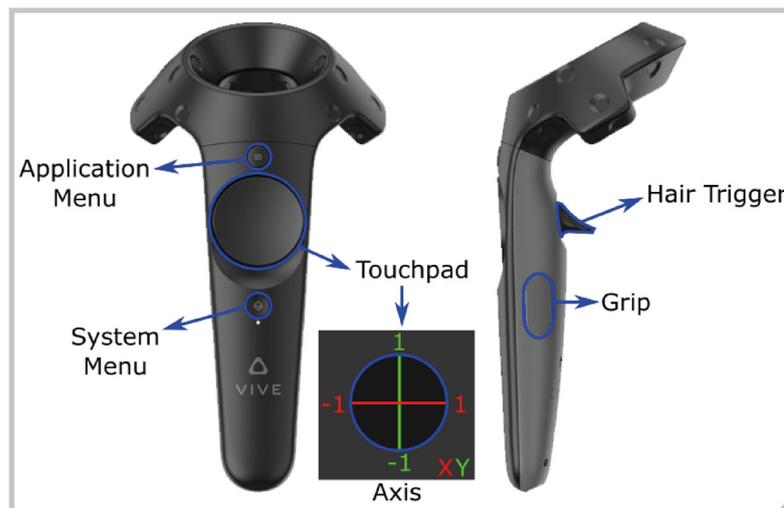


Abbildung 47: Tastenbelegung der Motion Controller (Van de Kerckhove, 2016)

Als nächste Stelle muss die Frage beantwortet werden, was ein Benutzer tun muss, um schließlich mit einem interagierbaren Objekt zu interagieren. Reicht es aus, wenn man die Bewegungen ansatzweise durchführt und den Rest der Engine überlässt? Dieser Gedanke basiert auf der Programmierung von vielen Spielen und Interaktionen. In diesen Fällen greift der Benutzer beispielsweise zu einer Türklinke und aktiviert damit dessen Triggerbox. Dadurch wird eine Animation gestartet, welche die Tür auf den festgesetzten Rotationsrange öffnet. Sobald der Benutzer einen vordefinierten Bereich verlässt oder die Türschwelle passiert, schließt sich diese automatisch. Das ist zwar leicht verständlich und sogar von allen Computerspielen akzeptiert und integriert. Ist es aber realistisch? Denn in einer realistischen Umgebung sollte man schließlich in der Lage sein eine Tür kontrollierter öffnen und schließen zu können.

Letztendlich hat sich die Entwicklerin des Tests für eine Art der Interaktion entschieden, die so gering wie möglich von der Engine selbst übernommen wird. Im Verlauf des Tests wurde darauf geachtet, ob dies von den Benutzern als mühsam empfunden werden sollte oder den gegenteiligen Eindruck, von mehr Kontrolle, erweckt.

Darauffolgend müssen die Einstellungen für die andere Taste angepasst werden: Das Touchpad ist für die Teleportation zuständig.

6.1.5 Teleportation



Abbildung 48: Visualisierung der Teleportation, Screen Shot aus Unreal Engine (Evermotion Kitchen, o. J.)

Die Teleportation stellt eine Möglichkeit dar, sich in den vier angefertigten Umgebungen fortzubewegen. Als zweite Möglichkeit bietet sich Gehen an. Der Spieler schreitet in der Wirklichkeit vorwärts, gleichzeitig wird seine Position von den Sensoren (Lighthouses) erfasst und aktualisiert. Diese Methode bietet zwar

eine hohe Immersion, ist jedoch mit Einschränkungen verbunden. Denn der Bereich befindet sich zwischen zwei Sensoren und hat eine bestimmte vordefinierte Größe. Abgesehen von dieser Begrenzung hängt die VR-Brille mit einem Kabel an einem Rechner, der in diesem Fall bis zu ca. 7 Meter von der Brille entfernt sein kann.

Zu Beginn seiner Entwicklung wurde der Test mit drei Probanden durchgeführt. Auffällig an ihrem Verhalten war das Bedürfnis, Wege so weit wie möglich *zu Fuß* zu gehen, soweit es Kabellänge und Trackingbereich erlaubten. Teleportation war offensichtlich ihre zweite Wahl. Das brachte die Entwickler auf die Idee auf Konzepte des Redirected Walking zurückzugreifen.

Entsprechend lautete nun der Plan, die virtuellen Schritte so zu skalieren, dass sie 1,5 Mal größer als die Schritte innerhalb der physischen Welt werden. Diese Entscheidung wurde getroffen, damit der Bereich entlang der Küche des ersten Levels größer wurde, sodass die Gefahr der Überschreitung der Kabellänge oder des Verlassens des Trackingbereichs verringert wird. Allerdings wurden diese Überlegungen in der späteren Entwicklungsphase wieder verworfen. Die Methode des Redirected Walking wäre nur in dem ersten Level nützlich gewesen und hätte für jedes weitere Level individuell optimiert werden müssen. Zusätzlich hätten die Größen der weiteren drei Umgebungen diese Option nur bedingt erlaubt.

Navigieren

Das Teleportationsziel befindet sich in einem vordefinierten Bereich, welcher in der genutzten Unreal Engine als *NavMeshVolume* gekennzeichnet ist. Innerhalb dieses Bereichs kann sich der Spieler mit Hilfe der Teleportationstaste (Touchpad) bewegen. Erwünscht war zuerst ein feiner und genau definierter Bereich. Im Laufe der Entwicklung wurde der NavMesh so fein kalibriert, dass die Spieler während des Teleportierens zu nah an Objekte wie Wände, Fenster oder Säulen herangekommen sind. Dieser Umstand erschreckt unerfahrene Nutzer, da sie sich plötzlich in die Wände gesteckt bzw. eingengt fühlen.



Abbildung 49: Einstellung der Eingabetasten, (Screen Shot aus Unreal Engine 4)

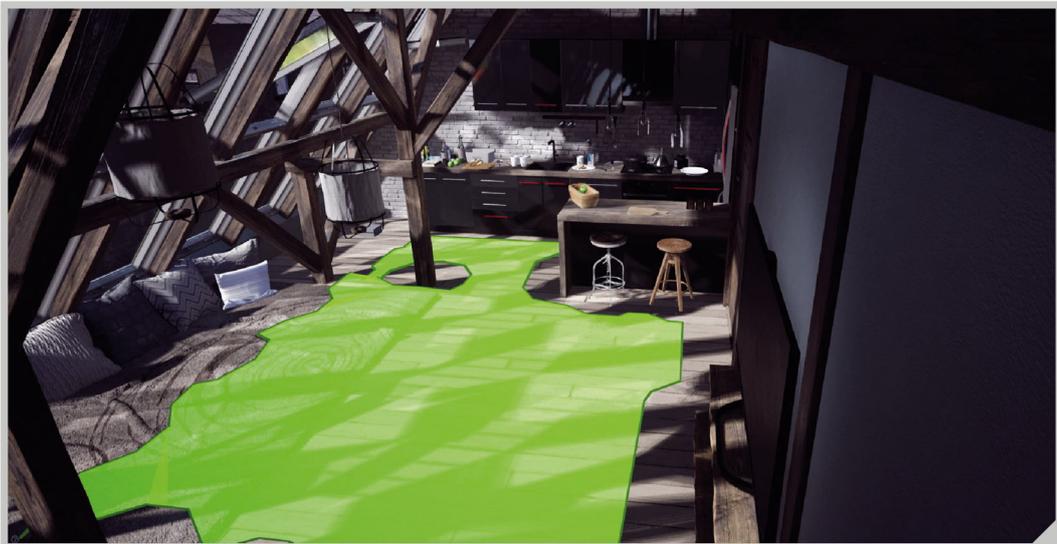


Abbildung 50: Darstellung des NavMeshVolume, (Screen Shot aus Unreal Engine 4), (Evermotion Kitchen, o. J.)

Die grüne Fläche bei Abbildung 47 illustriert den Bereich, in welchem sich der Spieler durch Teleportation bewegen kann. Dabei ist es wichtig, dass die Kanten des NavMeshVolume in diesem Raum eine relativ größere Distanz zu den Fenstern haben sollten.

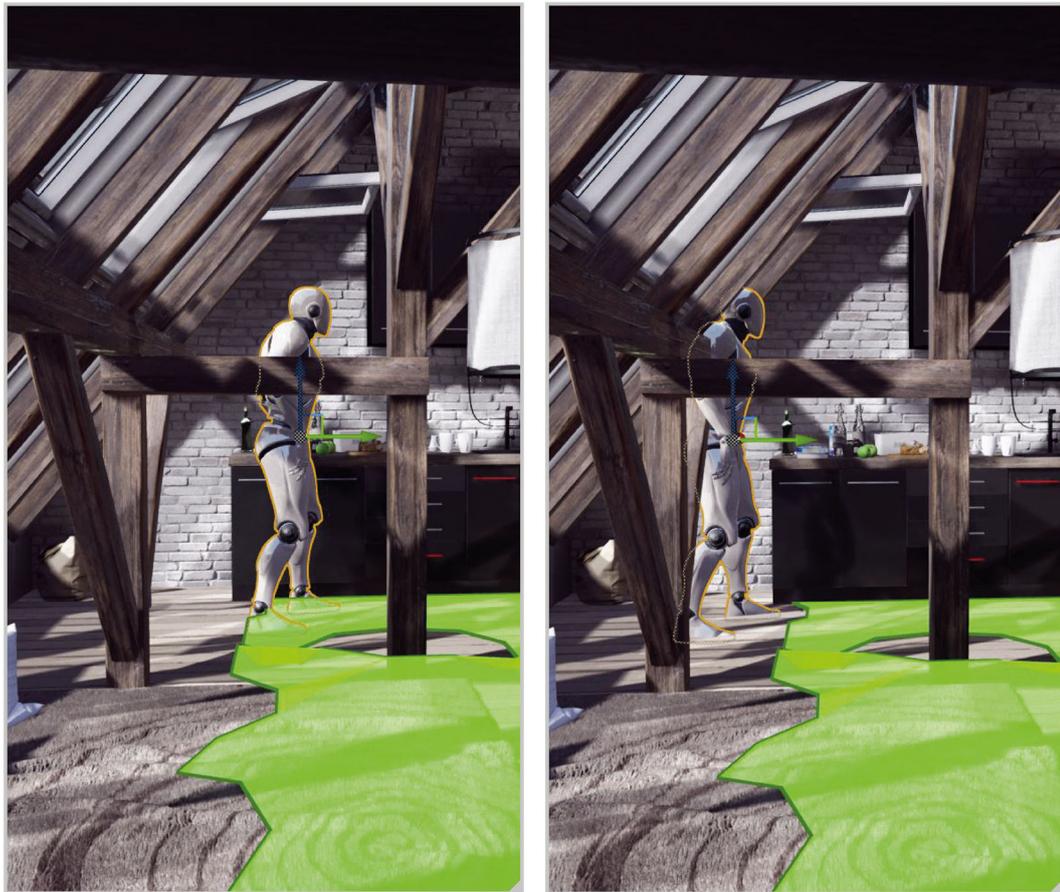


Abbildung 51: Anpassung der Raumgröße, (Screen Shot aus Unreal Engine 4), (Evermotion Kitchen, o. J.)

Erkennbar ist, dass ein Spieler mit einer Körpergröße von ca. 170 cm mit seinem Kopf im Fenster stecken wird, falls er sich zur Kante des Navigationsbereiches teleportiert. Nachdem dieser Bereich richtig eingestellt ist, sollte der VR-Player in der Lage sein, sich in dem dafür vorgesehenen Bereich zu bewegen.

6.1.6 Skalierung

Da eine virtuelle Welt aus der Perspektive eines Menschen gesehen wird, ist es wichtig, dass VR-Umgebungen sich nach dessen Körpergröße richten.

Leider muss man bei den Assets und Umgebungen, die man erwerben und herunterladen kann, mit dem Problem rechnen, dass sie nicht unbedingt für ein Virtual Reality-Projekt geeignet sind. Denn die mitgelieferten Texturen können zum Teil sehr groß sein und die Meshes sind meistens detailgetreu, sodass manche von ihnen in einer anderen 3D-Applikation, etwa Autodesk Maya oder 3ds Max, zu einer etwas reduzierteren Variante umgewandelt werden müssen.

6 Umsetzung / Usertest

Eine mögliche Herangehensweise, um dieses Ziel in der Unreal Engine zu erreichen, bietet die Anpassung der Größen an den Third Person Character, welcher mit einer Körpergröße von 180 cm von der Engine selbst angeboten wird.

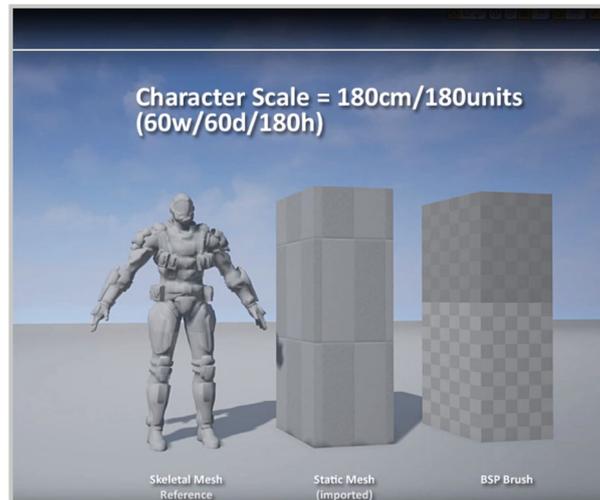


Abbildung 52: Größenvergleich für realistische Skalierung (World of Level Design, 2015)

An die Größe dieses Charakters können die Assets angepasst werden. Abbildung 50 zeigt die Anpassung der Schrankgrößen im ersten Level des Tests:

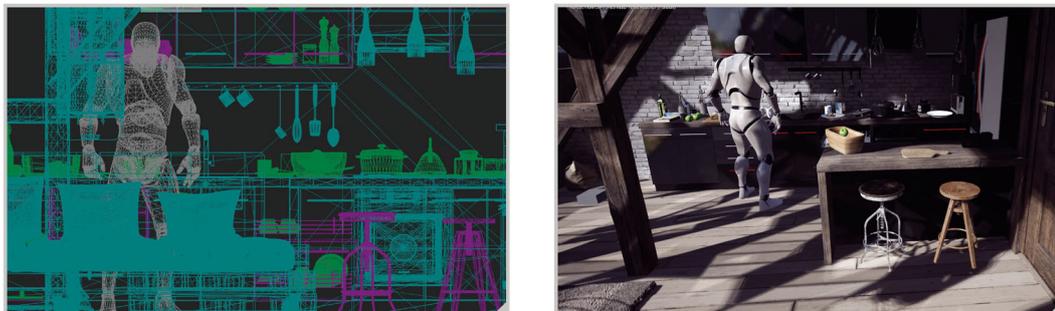


Abbildung 53: Anpassung der Schrankgrößen an Character (Evermotion Kitchen, o. J.)

Für eine VR-Applikation sind jedoch weitere Schritte nötig. Das VR-Template von Unreal Engine liefert ebenfalls eine voreingestellte VR-Kamera als auch die nötigen Interaktionen mit ihrer Umgebung, welche man als Basis verwenden kann.



Abbildung 54: Einstellung der Player-position in der Umgebung, (Screen Shot aus Unreal Engine 4), (Evermotion Kitchen, o. J.)

Importiert man die VR-Kamera in die erstellte Welt, rastet sie auf den Boden ein. Es ist wichtig auf die Höhe dieser Kamera zu achten, da sie sich beim Starten des Spiels an die Höhe der Brille anpasst und während der Teleportation auf derselben Höhe bleibt.

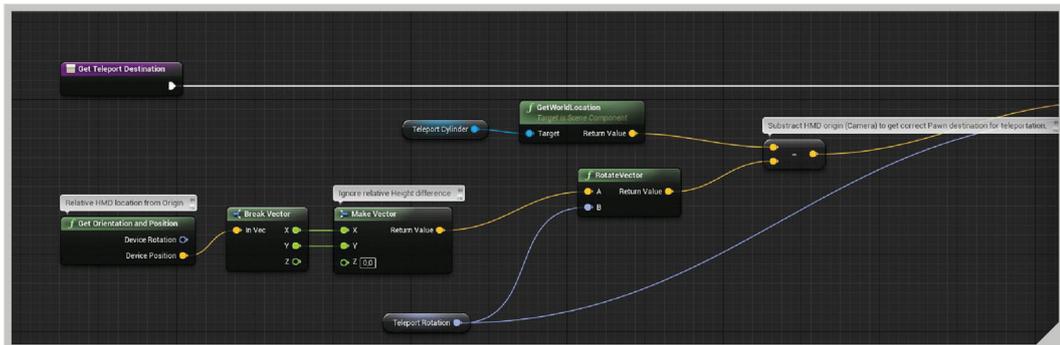


Abbildung 55: Fixierung der Playerhöhe in VR, (Screen Shot aus Unreal Engine 4)

Diese Abbildung zeigt, dass die Z-Achse dieser Position bei der Berechnung jeder neuen Position nicht mitberechnet wird.

6.1.7 Probanden

Damit bei allen Probanden ein Start unter gleichen Bedingungen erfolgen kann, werden sie vor dem Beginn auf einige wichtige Punkte hingewiesen:

Ihnen wird mitgeteilt, dass sie in diesem Test in vier unterschiedlichen Umgebungen jeweils eine größere Aufgabe zu lösen haben. Es ist wichtig, dass sie das Spiel unterbrechen, sobald sie sich nicht gut fühlen. Zudem wird ihnen der

Bereich gezeigt, in welchem sie sich in der physischen Welt bewegen können. Sie halten die beiden Controller in ihrer Hand, bevor sie die Brille aufsetzen. Außerdem erhalten sie die Information, dass sie von der Aufsichtsperson auf den Trackingbereich hingewiesen werden, falls sie sich zu weit in eine Richtung bewegen.

Es wird darauf Wert gelegt, dass den Probanden bewusst ist, was genau bei dem Test geprüft wird. Sie erfahren, dass die virtuelle Umgebung aus ihrer Sicht ebenfalls auf einem externen Bildschirm zu sehen ist. Zu Beginn wird jedem Probanden erklärt, was Affordanz bedeutet und warum dieser Test die Affordanz von Objekten überprüft. Sie sollen sich darauf einstellen, dass die Aufgaben in jedem Level eine Steigerung in ihrem Schwierigkeitsgrad aufweisen. Die Tasten der beiden Controller sind mit gleichen Funktionen belegt. Sie sind in der Lage sich in jedem Level mit dem Touchpad zu teleportieren und mit dem Trigger auf die interagierbaren Objekte zu reagieren.

6.2 Contenterstellung

6.2.1 Küche



Abbildung 56: Aufgabentafel in der ersten Umgebung des Usertests, (eigene Darstellung)

Der erste Raum ist eine unaufgeräumte Küche, Ziel ist das Aufräumen dieser Küche. Damit wird der Spieler indirekt dazu aufgefordert, sich in diesem Raum zu bewegen und die Objekte anzufassen, da er sie verschieben, verstellen, stapeln und entsorgen muss.

Hier wird auf Gesetze der Gestalttheorie zurückgegriffen. Ihre Prinzipien weisen auf das menschliche Verständnis von Ordnung, Zusammengehörigkeit und Gemeinsamkeit hin. Eigenschaften, die Menschen in der Natur und ihrer

6 Umsetzung / Usertest

Umgebung erkennen und erzielen. Dies bietet eine gute Grundlage um davon auszugehen, dass ein Wiedererkennen dieser Prinzipien in einer virtuellen Welt auf dem Wissen basiert, welches aus der physischen Welt mitgebracht wird.

Laut dem Gleichheitsgesetz der Gestaltungstheorie erkennen Menschen eine Gemeinsamkeit zwischen gleichaussehenden Objekten. („Grundlagen_der_Gestaltpsychologie“, o. J., S. 3) Um dies zu überprüfen, wurden manche Schränke in dieser Küche mit einem roten Griff versehen. Diese markierten Schränke können nicht geöffnet werden, wobei alle Schränke mit einem metallfarbenen Griff, wie in einer echten Küche, geöffnet werden können. Zusätzlich wurde darauf geachtet, dass die Tür der Spülmaschine auf der *Nick-Achse* rotiert, wie es die Probanden von der eigenen Küche kennen.

Interessant war an dieser Stelle die Aussage von Dr. Marco Gillies zu überprüfen: Laut Gillies ist die Affordanz eines Griffs stärker im Kopf des Benutzers verankert als seine Farbe. (Gillies & Xueni Pan, 2018, S. 6) Ausgehend von dieser Aussage sollte der Benutzer in dieser Küche die Schränke mit den roten Griffen trotzdem zu öffnen versuchen. Entweder würde dem User diese Erkenntnis einige Versuche kosten oder er wird frustriert sein.



Abbildung 57: Der Zustand der ersten Umgebung, wenn der Spieler in das Level einsteigt. (Screen Shot aus Unreal Engine 4), (Evermotion Kitchen, o. J.)

Zum Aufräumen sollte der Benutzer unbewusst auf sein Wissen zugreifen. Er sollte verteilte Tassen neben andere Tassen stellen, das Besteck in die Schublade zu dem restlichen Besteck legen. Währenddessen wird darauf geachtet, ob er die Messer zu den Messern, Gabeln und Löffel zu Gabeln und Löffel legen wird. Dabei testet man den erwähnten Punkt der Gestalttheorie, das Gesetz der Ähnlichkeit. („Grundlagen_der_Gestaltpsychologie“, o. J., S. 3)

Die Milchpackungen kommen in den Kühlschrank. Der User sollte nach wenigen Versuchen bewusst werden, dass die roten Schränke nicht interagierbar sind und er nur die rechte Tür des Kühlschranks öffnen kann.

Gemessen wird die Zeit zwischen dem Beginn des Spiels bis zur ersten Bewegung bzw. ersten Interaktion mit einem Objekt aus einer bestimmte Kategorie.

Es gibt drei Kategorien in diesem Level:

01: *The PickupItems*: Objekte, welche man in die Hand nehmen kann (beispielsweise Tasse, Besteck und Geschirr).

02: *Drawers*: Dabei handelt es sich um Schubladen und alle Objekte, die man, ähnlich einer Schublade, ziehen oder schieben muss.

03: *HingedDoor*: Objekte, die auf einer Achse (Pitch, Yaw oder Roll) rotieren. Beispiele dafür sind die Schranktüren und die Spülmaschine.

Diese Küche wurde aus mehreren Gründen als Einstiegslevel ausgesucht:

- Mehrere Gesetze der Gestaltungstheorie wurden hier getestet:
 - Gesetz der Gleichmäßigkeit
 - Gesetz der Nähe
 - Gesetz der Ähnlichkeit
 - Gesetz der Geschlossenheit
 - Gesetz der gemeinsamen Region

Wenn diese Prinzipien vom Spieler erkannt und eingesetzt werden, kann man behaupten, dass die Hypothese bestätigt ist. (Es ist besser, sich bei der Interaktion nach der Wirklichkeit zu richten)

- Eine unaufgeräumte Küche bietet viele kleine direkte Interaktionen, welche eine gute Übung als Einstieg bieten.

Diese vielen möglichen Interaktionen vermitteln indirekt eine starke Kontrolle über den Raum, in welchem sich der Benutzer befindet und diese Kontrolle sollte das Gefühl der Präsenz stärken. Am Ende dieses Tests sollte diese These analysiert werden können.

- Das Aufräumen in dieser Küche wurde so geplant, dass es in keiner notwendigen Reihenfolge erfolgen kann. Damit wurde die Fehlerwahrscheinlichkeit verringert. Der Spieler kann an irgendeiner Stelle beginnen und von dort aus weitergehen. Das eigentliche Aufräumen wurde ebenfalls an die Vorstellungen des Users angepasst. Ihm wird zu Beginn subtil mitgeteilt, dass er die Küche auf seine eigene Art aufräumen kann.

6 Umsetzung / Usertest

- Die Art und Weise wie die Objekte angefasst werden, als auch ihre Affordanz, sind, soweit es die Möglichkeiten der Engine erlaubte, realistisch gebaut. Der Wunsch, sich zu Beginn dieses Tests nach einer realitätsnahen Umgebung zu richten, ist somit erfüllt.

Erwartungen

Es wird erwartet, dass der Benutzer dieses Level leicht und ohne große Schwierigkeiten meistern kann. Die Interaktionen basieren alle auf seinem Wissen und die Objekte haben eine verständliche, klare Affordanz. Die Küche gilt nach dem individuellen Empfinden des Spielers als aufgeräumt, sodass Länge und Endzeit in diesem Fall keine Rolle spielen. Getrackt und Gemessen wird die Zeit zwischen dem Beginn des Spiels und der ersten Interaktion.

Es wird ebenfalls erwartet, dass der User zu Beginn langsam und vorsichtig mit den Gegenständen in Kontakt tritt und mit der Zeit sicherer und schneller wird. Es ist interessant zu erfahren, wie viele unerfahrene User in dieser Küche auf spielerische Ideen kommen statt ihre Aufgabe zu erfüllen.

6.2.2 Schlafzimmer



Abbildung 58: Aufgabentafel in der zweiten Umgebung des Usertests, (eigene Darstellung)

Der zweite Raum ist ein noch nicht eingerichtetes Schlafzimmer. Genauso wie im ersten Level wurde ein Ziel definiert, was schnell verständlich ist und das Gegenteil von jenem Zustand verlangt, welcher anfangs gesehen wird, d. h., wenn der Spieler in das Level einsteigt.

Dieses Schlafzimmer richtet sich, im Gegensatz zum ersten Level, mehr nach dem Wunsch des Users, magische Kräfte in einer virtuellen Welt zu besitzen und in der Lage zu sein, Aktionen durchzuführen, zu denen er in Wirklichkeit nicht fähig ist.

Wie geschieht eigentlich das Einrichten eines Zimmers in der Wirklichkeit?

Man entscheidet sich für bestimmte Wandfarben und Böden. Die Wände müssen gestrichen werden. Man entscheidet sich möglicherweise für ein bestimmtes Muster oder eine bestimmte Farbe und hofft, dass sie zu dem Raum, den Lichtverhältnissen und dem Boden passt. Daraufhin konzentriert man sich auf die Möbel. Diese werden im Zimmer aufgestellt sowie hin- und hergeschoben.

Oft entwirft man einen groben Plan im Kopf, fertigt kleine Zeichnungen auf einem Zettel an und richtet sich nach einem Bild, das man sich als Referenz ausgesucht hat. Anschließend verschiebt man die Möbel in eine Ecke und stellt sich an eine Position, von der aus man das Zimmer aus der größtmöglichen Entfernung beobachten zu können. Diesen Ablauf wiederholt man einige Male bis alles optimal platziert ist.

Die Möglichkeit, den Wänden eine gewünschte Farbe zu geben und die Möbel zu verschieben, ist natürlich auch in der realen Welt vorhanden, doch in VR kann dieser Wunsch praktisch sofort und lediglich mit einem Klick erfüllt werden.

Wie bereits erwähnt, muss der Benutzer ein virtuelles Schlafzimmer einrichten, die Farben der Wände sowie den Boden verändern. Es ist davon auszugehen, dass er nach wenigen Versuchen herausfinden wird, was er hier verschieben und bewegen kann. Aber wie zeigt man ihm nun, ohne Hinweise und Menüs, dass er die Farbe der Wände und den Boden ebenfalls verändern kann?

„Ein Tablet verstehe ich von der realen Welt und da weiß ich was ich tun soll. Das entspricht mittlerweile dem Verhalten der realen Welt. Und ich glaube, dass gerade bei Interaktionen solche Überlegungen das Kernthema sein werden. Wir überlegen auch, dass wir an der Wand einen Monitor aufhängen, wo man etwas auswählen kann. Weil... Touch Screen Monitore kennt man..“ (Friesenegger, 2018)

Der Spieler steht zu Beginn dieses Levels vor einer Wand, an der ein großes Bild hängt. Dieses Bild zeigt dasselbe Zimmer in einem Zustand wie es sich der Entwickler wünscht. Schaut sich der Spieler um, sollte ihm anhand der Form des Zimmers und der Möbel bewusst werden, dass er sich in demselben Raum wie auf dem Bild befindet. Die Idee der Zieldefinition richtet sich nach der Aussage von Christian Friesenegger.



Abbildung 59: Darstellung des erwünschten Zustands des Zimmers, (Screen Shot aus Unreal Engine 4)

Abbildung 56 beschreibt die Farbe und die Einrichtung dieses Zimmer in der fertigen Form.

Nun musste der User erkennen wo die Möbel hingeschoben werden müssen und er sollte ebenfalls erkennen, dass die Wände, Türrahmen und Fußleiste im Vergleich zu dem Bild eine andere Farbe besitzen.

Für den Spieler hat dieses Level, im Gegensatz zum Ersten, eine Ähnlichkeit mit einem Puzzle. Bei einem Puzzle sind Platz, Rotation und Farben der Puzzlestücke die wichtigsten Merkmale, auf welche der Spieler achten muss. Und wie bei jedem Puzzle richtet man sich nach einem Plan, Bild oder Ähnlichem.

Analog zum Verhalten in der Realität müsste der Spieler jedes einzelne Möbelstück in eine Ecke schieben, einige Schritte zurücktreten, nochmals hingehen und die Möbel etwas zurechtrücken. Und er müsste dies einige Male wiederholen. Der Vorgang selbst klingt aufwendig und mühsam, kann aber in VR ganz anders sein, denn man kann sich etwas entfernter stellen, einen *Zeigestab* in der Hand halten und damit die Objekte an ihre richtige Position schieben.

Daher bot sich in der Entwicklung auch genannter Zeigestab an. Dieser geht von der Handposition des Spielers aus, wird auf die gewünschten Objekte gerichtet und ermöglicht dem Spieler, sie an ihre Position zu verschieben.

Hier wird eine Eigenschaft getestet, die laut Christian Friesenegger, keine Einheit in VR darstellt: Gewicht.

Arian Jalaefar: Aber wie simuliert man in VR eigentlich Gewicht?

Christian Friesenegger: Gar nicht momentan. In dem Anwendungsbereich, wo wir tätig sind, ist Gewicht keine relevante Größe. Da geht es darum, ob ich es verschieben kann... Ob es sich schwer anfühlt oder nicht, das ist weniger wichtig. Was für uns viel wichtiger ist, ist zu überlegen, wie kann ich die realen Bedingungen, und das reale Verhalten virtuell abbilden. (Friesenegger, 2018)

Diese Aussage gab die Idee für ein Experiment. Könnte man das Gewicht simulieren, wenn man die Objekte nur auf ihren zwei Achsen bewegen lässt, sodass der Spieler die Objekte zwar in allen Richtungen bewegen aber nicht anheben kann?

Diese Frage wurde zu Beginn dieser Arbeit ebenfalls als eine der Forschungsfragen an den Experten gestellt und wurde von Marcel Nürnberg, Gründer von SQUAREBYTES Interactive Media, so beantwortet:

„Also man muss immer denken, was das Ziel ist, was will ich damit machen? Ich will eine realistische Möblierung hier darstellen. Und da ist es nicht wirklich realistisch, wenn das Sofa dann auf dem Tisch ist. Natürlich kann ich dem Benutzer die Freiheit geben, aber dann ist es halt ein lustiges Spiel. Bauen Sie sich einen Sesselturm oder so aber das verfehlt komplett das Ziel, dass ich diesen Raum komplett einrichten kann.“ (Nürnberg, 2018)

Es ist dem Entwickler also bewusst, dass er dem Spieler eine Möglichkeit wegnimmt, welche das Spiel realitätsfremder machen würde. Diese Entscheidung wird aber zielgerichtet getroffen und führt am Ende dazu, dass der Spieler zum Erreichen seines Ziels auf gewisse Art unterstützt wird.

Die Frage nach jener Freiheit, die man dem Spieler geben bzw. nehmen möchte, wird offensichtlich in vielen Phasen der Entwicklung, sowohl technisch als auch inhaltlich gestellt. In Bezug auf das Schlafzimmer hat man sich am Ende für den Zeigestab in der Hand entschieden. Er ist auch ein Sinnbild für mehrere Aspekte, auf die in diesem Level Wert gelegt wurde:

- Er erzeugt die Kommunikation mit den Möbeln trotz eines relativ großen Abstands.
- Der Spieler kann schnell agieren und somit wird die Möglichkeit der Frustration verringert.

- Er hilft dem Entwickler Gewicht zu simulieren indem der User die Objekte auf dem Boden *sliden* lässt.
- Je schneller das Level vorbei ist, desto größer ist die Wahrscheinlichkeit, dass einem unerfahreneren VR-Spieler nicht schlecht wird.
- Jedes Level versucht letztlich seinen User zu begeistern und somit zum kompletten Durchlauf zu motivieren. In so einem Fall gibt die mitgestoppte Zeit der Interaktion nicht zwingend brauchbare Ergebnisse, denn viele User können aus Langeweile langsamer oder generell träge agieren.

Erwartungen

Es wird erwartet, dass der Spieler in diesem Level so schnell wie im ersten Level seine erste Interaktion mit den Objekten erkennen kann. Der unaufgeräumte Raum und die Möbel, welche nebeneinander den Ausgang versperren, sollten einen klaren Hinweis bieten, dass sie verstellt werden müssen. Durch die offensichtliche Ähnlichkeit, die sie mit den Objekten im Bild haben, soll der User erkennen, dass er sich zum Sortieren nach diesem Bild zu richten hat.

Aufgrund der Anzahl von Objekten, die bewegt und verändert werden müssen, wird, im Vergleich zum ersten Level, von einer kürzeren Gesamtspielzeit ausgegangen.

Es wird vermutet, dass der User den Raum einrichtet, ohne zuerst die Möglichkeit zu beachten, dass er die Farben der Wände und den Boden ändern kann. Da diese Aktion mit einem *Klick* passiert, kann man vermuten, dass er zufällig und während einer anderen Aktion darauf kommen könnte.

6.2.3 Gefängnis

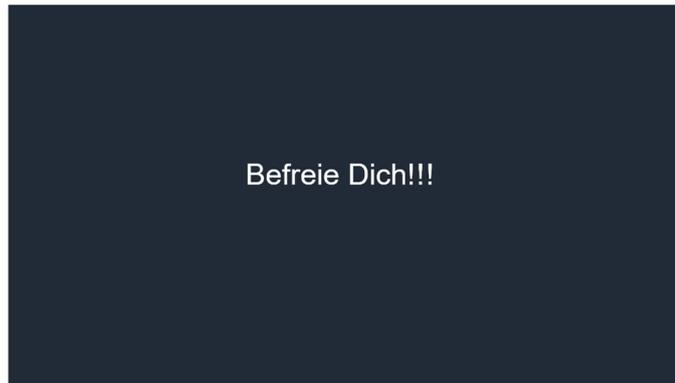


Abbildung 60: Aufgabentafel in der dritten Umgebung des Usertests, (eigene Darstellung)

Das Ziel in diesem Level liegt darin, die Affordanz von Gegenständen zu testen, welche, bezogen auf ihrer Interagierbarkeit, noch eine Stufe entfernter von der Realität stehen als Möbel und Gegenstände in der Küche von Aufgabe 1. Es handelt sich um Objekte, die in der Wirklichkeit als statisch wahrgenommen werden, d. h. Wände, Zimmerdecken, Fenster, Bäume usw.

Wie könnte man einen Benutzer ohne Hinweise auf den Gedanken bringen, dass er die Wände bewegen kann? Wie könnte er auf die Idee kommen, dass er beispielsweise die Decke hochschieben kann? Laut der *Think Negative Technik* könnte man davon ausgehen, dass ein Mensch auf die Idee kommen könnte, um sich herum mehr Platz zu schaffen, weil er sich eingeengt fühlt. (Wishin, o. J., S. 2) Wenn sich also der Spieler zu Beginn in einem kleinen, engen Raum befindet, würde er versuchen, sich aus dem Raum zu bewegen.

Reicht diese Intention als Ziel aus? Kann man das Bedürfnis von Raumgröße, Platz und Freiheit hervorrufen, wenn man das Gegenteil darstellt? Basierend auf dieser Frage und dem Interesse an ihrer Antwort fiel die Entscheidung für die dritte Aufgabe auf ein Gefängnis. Die Aufgabe lautete: *Befreie Dich!!!*

Es ist auffällig, dass es Level für Level schwieriger wird eine Aufgabe zu stellen, welche keinen Hinweis auf die Affordanz gibt und zu den Umgebungen passt, die für die Level ausgesucht wurden. Es wird nicht nur eine Herausforderung für den Benutzer darstellen, sich in der Umgebung zurechtzufinden. Es ist ebenfalls eine Herausforderung, sich während der Entwicklung streng an alle Angaben und Richtlinien zu halten. Es ist beispielsweise schwer die Aufgabe ohne Hinweise verständlich zu kommunizieren.

In diesem Level muss der Spieler eine Wand zur Seite schieben, um den Raum seiner Ausgangsposition zu verlassen. Daraufhin betritt er einen weiteren,

verschlossenen Raum, der ihn zu Beginn mit einer zu tiefen Decke einengt. Wenn er die Decke nach oben verschiebt, wird er eine weitere Tür öffnen können. Als Letztes muss er ein Fenster in einer Wand nach unten schieben, durch welches er das Gefängnis verlassen kann. Vermutlich wird ein sehr kleines Gefängnis zu offensichtliche Hinweise auf die Möglichkeiten geben, wodurch das Ergebnis beeinflusst werden kann. Daher wird es in diesem Level ebenfalls andere Räume geben, welche der User betreten kann. Zusätzlich existieren weitere Türen und Wände, die jedoch nicht verschoben oder geöffnet werden können. Abgesehen davon ist es wichtig, dass das Empfinden eines verlassenen Gefängnisses vermittelt wird. Somit kann man davon ausgehen, dass die Immersion und das Gefühl der Gefangenschaft das Bedürfnis nach Freiheit stärken.

Teleportation und die Kommunikation mit allen interagierbaren Objekten finden in diesem Level auf dieselbe Art und Weise statt, wie sie der Spieler von den zwei vorherigen Umgebungen kennt.

Erwartungen

Es wird erwartet, dass der User in diesem Level zum Erreichen seines Ziels größere Schwierigkeiten haben wird. Es fällt einem Entwickler schwer, das Verhalten des Spielers im Vorhinein einzuschätzen. Es wird davon ausgegangen, dass die Benutzer entweder nach einer gewissen Zeit nach Hilfe fragen oder das Spiel unterbrechen werden.

Andererseits wird er in diesem Level zum ersten Mal in eine Umgebung geschickt, die ihm gänzlich unbekannt und neu erscheint, da er sie aus der Wirklichkeit nicht kennt. Das spricht für Neugier und Interesse und somit für eine längere Gesamtspielzeit. Es kann allerdings ebenso eine Verweigerung des Users stattfinden, falls er diese Umgebung so schnell wie möglich verlassen möchte oder von diversen Versuchen frustriert ist.

6.2.4 Garage

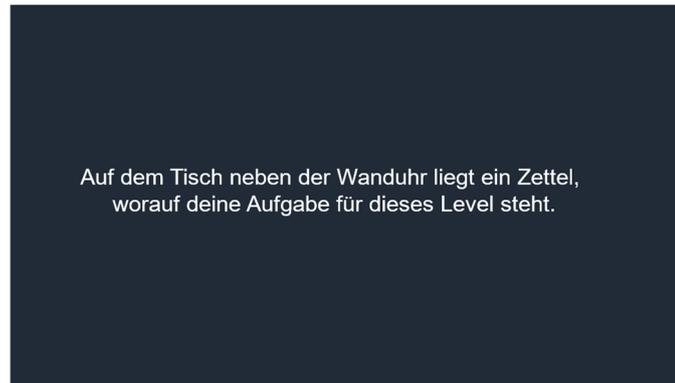


Abbildung 61: Aufgabentafel in der vierten Umgebung des Usertests, (eigene Darstellung)

In diesem Level ist nun das Ziel von dem gesamten Test erreicht. In jedem Level trainiert man den Spieler, die Affordanz von Objekten zu begreifen, welche sich so stark von dem Verhalten der Wirklichkeit unterscheiden, dass sie den Bezug zur Realität verlieren. Daher ist es das Ziel dieses Levels, die Zeit zu manipulieren. Aufgrund dieser Steigerung ist es offensichtlich, dass dieses Level aus inhaltlicher Sicht am schwierigsten war und der User sich ziemlich schnell mit *Trial-and-Error* anfreunden musste.

Wie kann man den Benutzer auf die Zeit aufmerksam machen ohne direkte Hinweise zu geben? Schon die Art der Aufgabenstellung unterscheidet sich von den Aufgaben der anderen drei: Hier befindet sich der Spieler in einer dunklen, mit zwei Fenstern ausgestatteten, Garage. Zusätzlich wurde ein kleiner Schlitz über ihrem Tor eingebaut. Diese Öffnungen lassen Strahlen einer, sich außerhalb und sehr schnell bewegendes, Sonne durchscheinen. Die Aufgabe wurde in einer Art formuliert, welche den Spieler subtil dazu zwingt, zuerst einen Blick auf die Wanduhr zu werfen, um anschließend den erwähnten Tisch zu finden. Allerdings ist erwähnter Zettel und dessen Aufgabe für das Ziel dieses Levels irrelevant. Daher ist auf ihm auch lediglich die Tastenkombination abgebildet, welche das Spiel beenden wird. Es ist viel wichtiger, dass der User auf die Idee bzw. den Trick kommt, wie bzw. womit er das Licht bewegen kann, um den Zettel überhaupt lesen zu können. Er ist nämlich in Dunkelheit gehüllt.

Die Uhrzeiger bewegen sich genauso schnell wie besagte Sonne. Abgesehen davon bewegen sich die Sonnenstrahlen kontinuierlich sobald man das Level betritt. Dadurch erschweren sie sogar das Lesen der Aufgabenstellung.

Der Spieler macht sich auf dem Weg zu dieser Uhr bzw. zu dem danebenstehenden Tisch. Auf diesem Tisch wird er den Zettel, neben einem Foto

von den Entwicklern dieses Tests, finden. Er wird sich umschauen, um den Tisch zu finden und in dieser Zeit wird ihm die schnelle Veränderung des hineinstrahlenden Lichts auffallen.

Wie kann man davon ausgehen, dass der Spieler versteht, was mit diesem Licht suggeriert werden soll? Wenn man aus der eigenen Erfahrung an die Aufgabe herangeht, wäre es Naheliegend, dass er versucht den Zettel zu finden und sich damit an einer Stelle mit ausreichender Helligkeit begibt. Diese Reaktion ist in diesem Level jedoch nicht zielführend, da man das Licht zu den Gegenständen bewegen sollte und nicht die Gegenstände zum Licht. Damit dieser Gedanke bei dem User so schnell wie möglich ausgeschlossen wird, wurde die Möglichkeit der Interaktion mit den Gegenständen in diesem Level nicht integriert. Damit kann man hoffen, dass der User keine lange Zeit darin investiert, eine Taschenlampe oder eine andere Lichtquelle zu suchen.

Zusätzlich muss man daran denken, dass Menschen nie mit Licht interagieren sondern mit der Lichtquelle. Soll der Spieler also die Sonne anfassen?

Ein schneller Test in der Entwicklungsphase hat gezeigt, dass es für den User aus zwei unterschiedlichen Gründen eine falsch wäre die Sonne anzugreifen:

1: Der Benutzer befindet sich in einer geschlossenen, abgedunkelten Garage. Wenn er mit Hilfe des *Zeigestabs* aus dem Fenster hinaus auf die Sonne zielen würde, könnte er die Sonne bis zu einer Position bewegen, in der sie von der Decke der Garage maskiert wird. Spätestens ab diesem Zeitpunkt wäre es für ihn nicht mehr möglich zu erkennen worauf er hinzeigt.

2: Es stellt eine benutzerunfreundliche Methode der Interaktion dar, wenn der Entwickler vom User erwartet, dass er direkt in die strahlende Sonne schaut um sie zu bewegen.

Die Sonne zu greifen kann also nicht richtig sein. Versucht man den Begriff und das Verständnis der meisten Menschen von Zeit zu analysieren, bemerkt man schnell, dass Menschen kaum die Zeit definieren. Es ist nicht die Zeit, die beschrieben wird, sondern meistens das Vergehen der Zeit. Sie wird mit Uhren, Kalendern, Jahreszeiten, Altwerden, Wartezeiten und Ähnlichem assoziiert. Man könnte behaupten, dass der User die Zeit anhalten würde, wenn er ihr Fortschreiten begreift.

Das hilft zwar um zu wissen, was man genau zu gestalten versucht, jedoch hilft es dem Spieler nicht herauszufinden, wie er es schafft.

Bezüge aus der Realität können kaum als Hilfsmittel oder Referenzen gelten, da die Zeit nie eine Affordanz hat. Mit der Zeit interagiert niemand. Wenn man dieses Level etwas genauer unter die Lupe nimmt, kann man behaupten, dass auf einen Hinweis bzw. auf ein Interface zurückgegriffen wurde: die Wanduhr.

Der User *muss* mit einem Gegenstand interagieren. In diesem Fall ist dies die Wanduhr. Sie weist nicht nur auf das Vergehen der Zeit hin, sondern ermöglicht auch eine Interaktion. Der Spieler kann ihren Zeiger anhalten und dadurch ebenfalls die Zeit sowie die Bewegung des Lichts manipulieren. Denn mit dem Drehen der Zeiger verändert sich die Position der Lichtquelle und somit der Lichtstrahlen. Er sollte die Zeiger so lange bewegen, bis die Lichtstrahlen direkt auf den Zettel fallen, damit er den Text lesen kann.

Erwartungen

Es wird erwartet, dass der Spieler in diesem Level mehr Zeit benötigt, da er nach einer Lösung sucht und viel Zeit in die Verbindung zwischen Uhr und Sonne investieren muss. Es ist schwer, das Verhalten des Users in diesem Fall vorherzusagen, jedoch wird vermutet, dass eine Uhr ein eindeutiges Interface ist, um die Reaktion des Lichts zu mappen.

6.3 Umsetzung

6.3.1 Küche

Nun muss der Spieler mit den Gegenständen in der Küche interagieren können. Was bedeutet Interaktion in diesem Fall? Inhaltliche Punkte in Bezug auf die Interaktion wurden bereits erklärt, jetzt müssen sie umgesetzt werden.

Um das Gesetz der Ähnlichkeit zu testen, wurden die Farben der Schränke zur Hälfte geändert. Schränke, mit welchen man nicht interagieren kann, besitzen einen roten Griff. Objekten, mit denen ein Spieler wiederum interagieren kann, muss eben diese Eigenschaft gegeben werden. Nachfolgend wird eine Möglichkeit in UE4 vorgestellt.

Es wurden zwei Funktionen, *Pick Up* und *Drop*, in einem Interface erstellt. Dieses Interface ist eine Programmierschnittstelle, die ein Objekt bieten muss, mit dem man interagieren möchte. Auf diese zwei Funktionen kann jedes Objekt dieser virtuellen Welt zugreifen, welches als ein *Blueprint Actor* definiert und in dieses Interface implementiert.

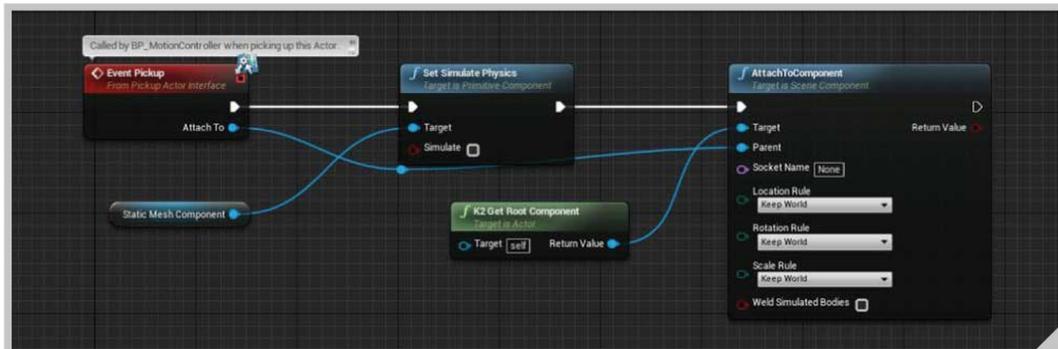


Abbildung 62: Blueprint zum Deaktivieren der Physiksimulation eines Objekts beim Aufheben, (Screen Shot aus Unreal Engine 4)

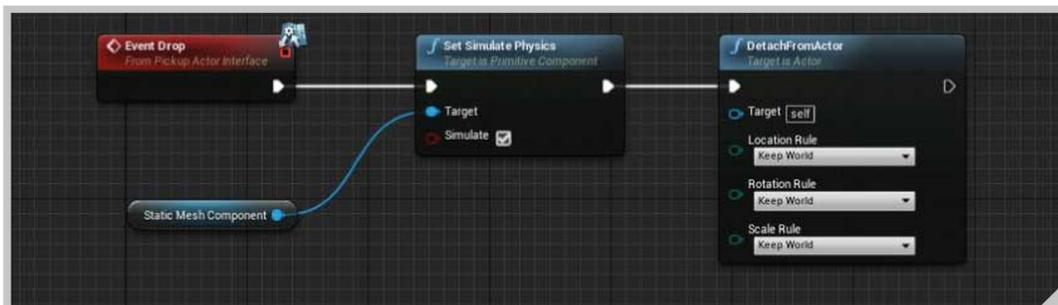


Abbildung 63: Blueprint zum Reaktivieren der Physiksimulation eines Objekts beim Ablegen, (Screen Shot aus Unreal Engine 4)

6 Umsetzung / Usertest

Beide Funktionen dienen dazu, dass die *Physics* eines Objekts (static mesh) deaktiviert werden, sobald die Hand diesen Gegenstand greift und reaktiviert wird, sobald die Hand dieses Objekt loslässt. Diese Aktion ist nötig, damit die Gegenstände an der Hand angehängt und mit ihr bewegt werden können.

Nun taucht ein anders Problem auf. Wie man es aus einer echten Küche kennt, werden Tassen in dem Schrank aufeinandergestapelt. Macht man das Gleiche in der virtuellen Küche, müsste man sich für bzw. gegen die *Physic simulation* dieser Gegenstände entscheiden. Sobald man die Physics der Tasse aktiviert, wird die obere Tasse in die untere hineinfallen und es folgen Kollisionsprobleme. Deaktiviert man die Physics, ist es möglich, die untere Tasse zu entnehmen, ohne dass die Obere hinunterfällt. Dieser Test verwendet die zweite Variante.

Der Spieler kann durch diese Einstellung viele Gegenstände wie Gläser, Tassen, Teller oder Besteck in der Küche bewegen und sie an die richtige Stelle bringen. Es ist jedoch nicht erwünscht, dass die Schubladen, Kühlschrankschranktür und Spülmaschine ebenfalls diese Eigenschaften besitzen. Denn bei der Tür einer Spülmaschine ist es wünschenswert, dass sie sich um eine Achse rotiert und eine Schublade sollte nur entlang einer Achse gezogen bzw. geschoben werden. Für die Schublade und alle weiteren Objekte, die sich wie eine Schublade verhalten, wurde ein Blueprint erstellt. Der Konstruktor dieser Klasse stellt die Position dieser Schublade und die maximale Extension auf der Achse (in dem Fall die x-Achse) ein, damit die Schublade nur bis zu einer bestimmten Stelle gezogen werden kann. (Abbildung 64)

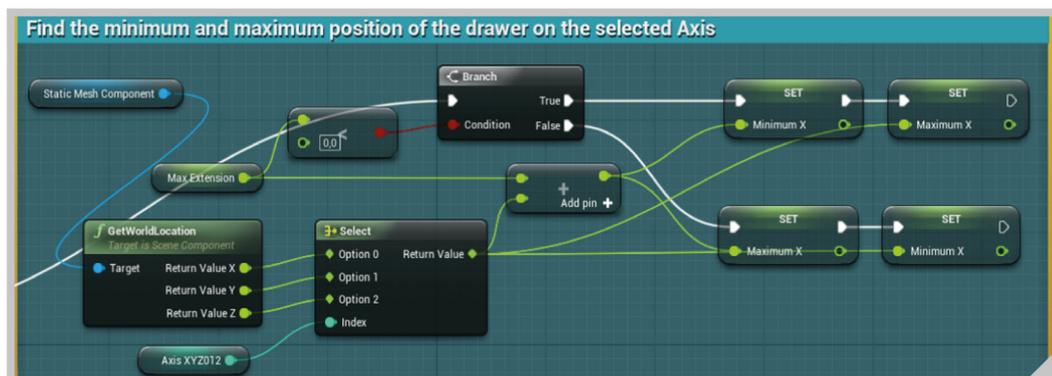


Abbildung 64: Blueprint zur Ermittlung des Bewegungsbereichs entlang der gewählten Achse (Screen Shot aus Unreal Engine 4)

6 Umsetzung / Usertest

Davor musste jedoch überprüft werden, entlang welcher Achse die Bewegung stattfinden soll.

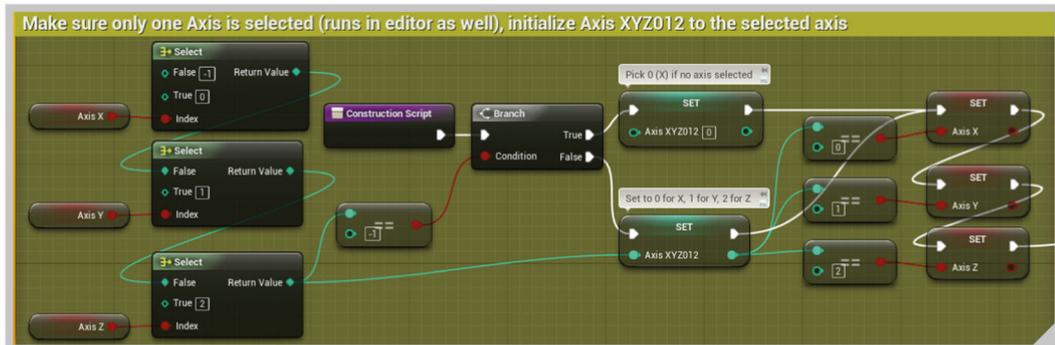


Abbildung 65: Blueprint zur Ermittlung der ausgewählten Achse (Screen Shot aus Unreal Engine 4)

Da die Schublade ebenfalls mit dem *Pickup Actor Interface* erweitert wurde, greift sie auf die Funktionen dieses Interfaces zurück. Zusätzlich muss die Position der Schublade beim Ziehen und während jedes Frames aktualisiert werden. Dafür ist dieser Teil des Codes zuständig:

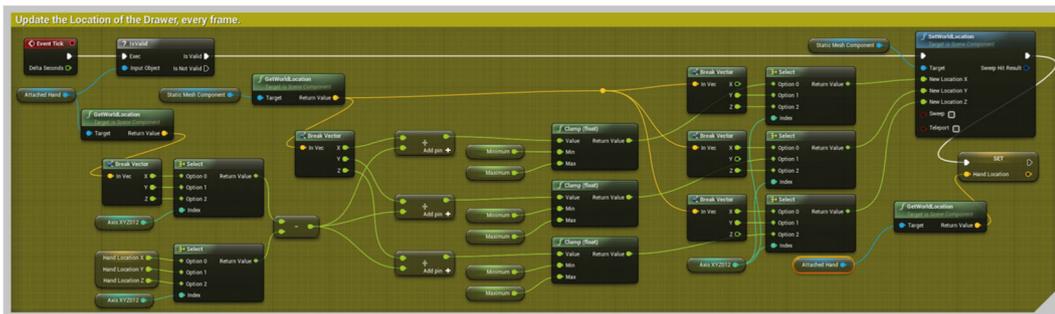


Abbildung 66: Blueprint zur laufenden Aktualisierung der Position der Lade, (Screen Shot aus Unreal Engine 4)

Danach sind die Schränke in der Reihe. Sie müssen von ihrem *Pivot Point* aus rotiert werden. Sobald sie an der Hand angehängt werden, wird ihre Position bei jedem Frame aktualisiert. Die *get Angle-function* berechnet mit Hilfe der Arkustangens-Funktion den Pitch- oder Yaw-Winkel, je nach Auswahl.

6 Umsetzung / Usertest

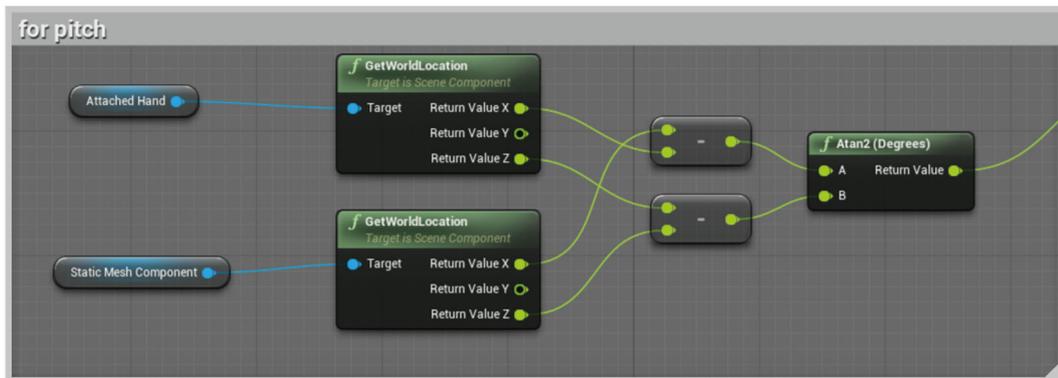


Abbildung 67: Blueprint zur Berechnung des Nick-Winkels zwischen Hand und Objekt (Screen Shot aus Unreal Engine 4)

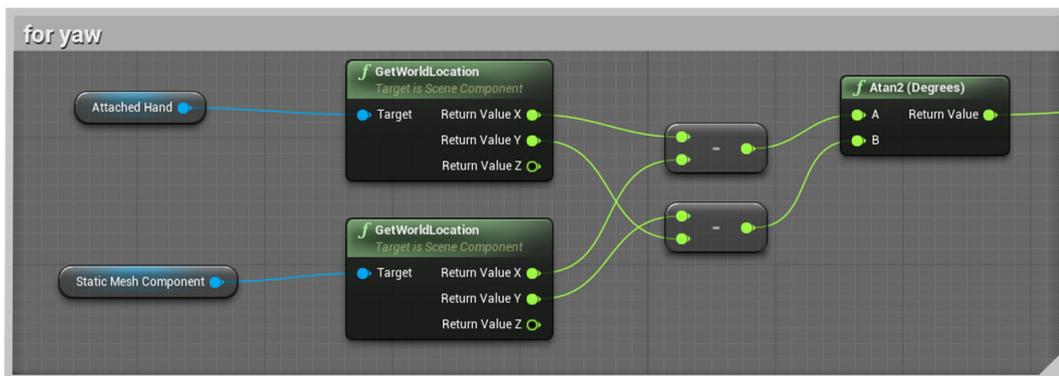


Abbildung 68: Blueprint zur Berechnung des Gier-Winkels zwischen Hand und Objekt (Screen Shot aus Unreal Engine 4)

Die Länge der Seiten wird durch den Abstand zwischen der *Grabsphere* der Hand, *Attached Hand* und dem Pivot des *Static Mesh Components* (in dem Fall die Schranktür) berechnet.

So wie sich eine Tür in der Wirklichkeit verhält, muss sich die Tür der virtuellen Küche ebenfalls in einem Range bewegen. Dieser Bereich entspricht beispielsweise Werten zwischen 0 bis 110° für eine Schranktür. Das *Get Hand Angle* Macro berechnet bereits den Winkel, in welchem die Hand steht. Damit die Bewegung funktioniert, muss die Variable *Angle* bei *Grabben* initialisiert werden.

6 Umsetzung / Usertest

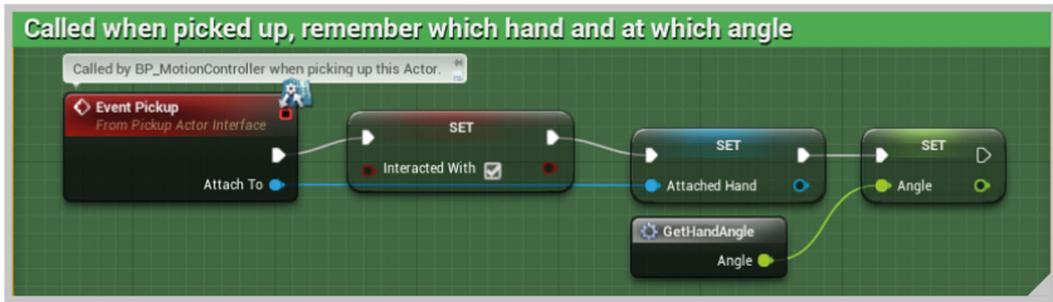


Abbildung 69: Blueprint um den Initialwinkel zu speichern (Screen Shot aus Unreal Engine 4)

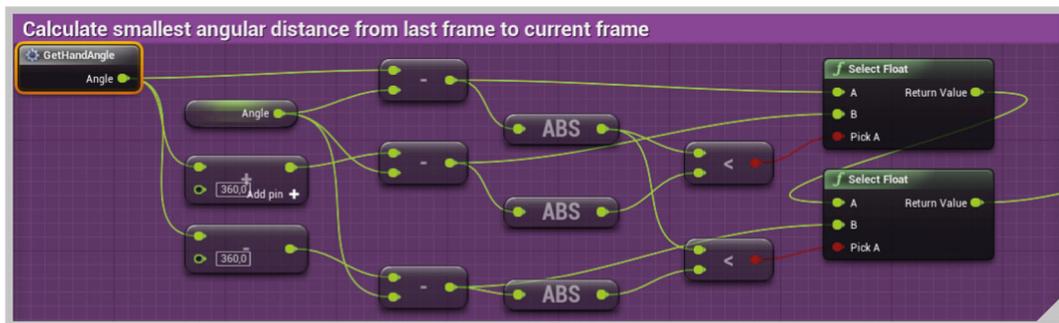


Abbildung 70: Blueprint zur Berechnung der kürzesten Distanz zwischen zwei Winkeln (Screen Shot aus Unreal Engine 4)

Diese Funktion nimmt die aktuelle Position der Schranktür (den momentanen Winkel) und zieht die Position des vorherigen Frames davon ab. Das Ergebnis wird mit (Ergebnis + 360) und (Ergebnis - 360) verglichen. Die Zahl, welche den kleinsten Absolutwert ergibt, wird im nächsten Schritt weiterverarbeitet.

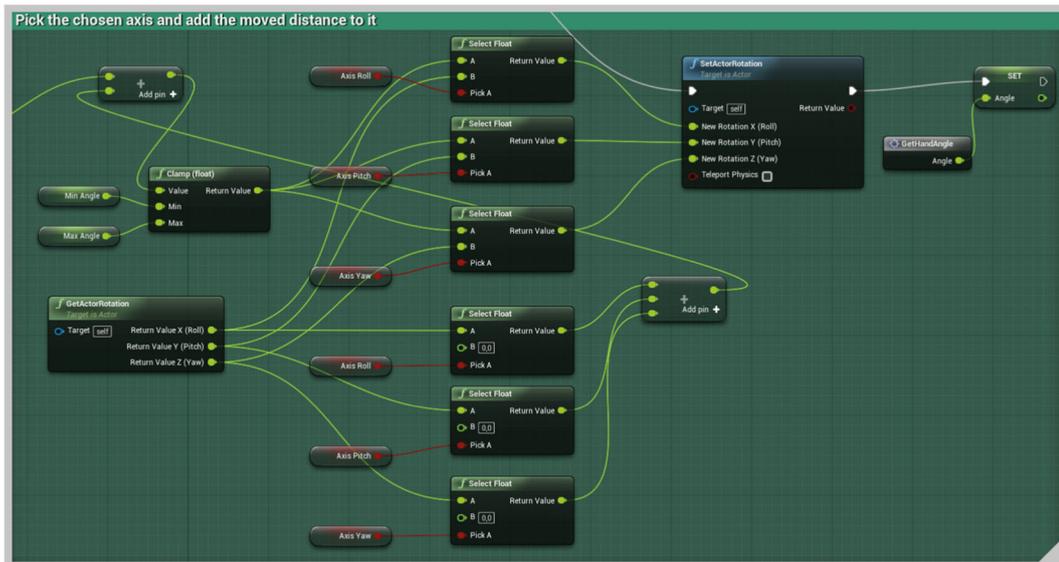


Abbildung 71: Blueprint zur Berechnung des neuen Drehwinkels (Screen Shot aus Unreal Engine 4)

6 Umsetzung / Usertest

Get Actor Rotation gibt die Rotation der Schranktür zurück (*target == self*) Er gibt den Wert der ausgewählten Achse zurück.

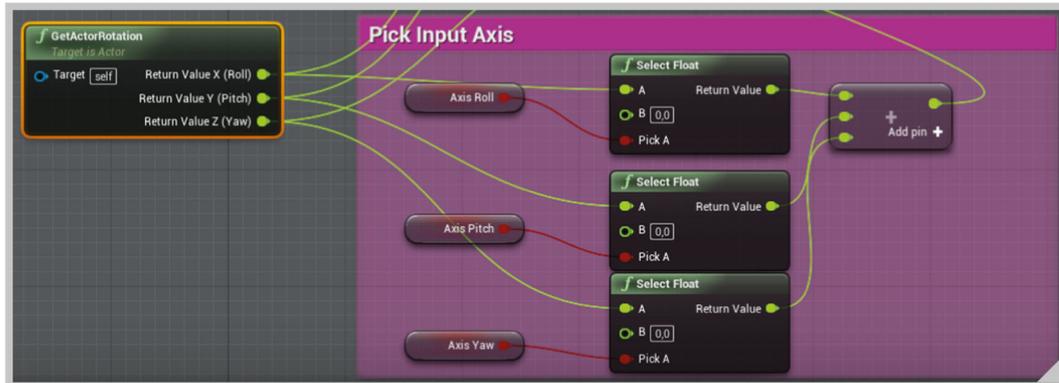


Abbildung 72: Blueprint zur Auswahl der Drehachse (Screen Shot aus Unreal Engine 4)

Dieser Wert wird mit dem Wert aus dem vorherigen Schritt addiert. Das Ergebnis dieser Addition wird in den erlaubten Bereich *geclamt*. (Dieser Bereich ist in der Engine manuell anzugeben).

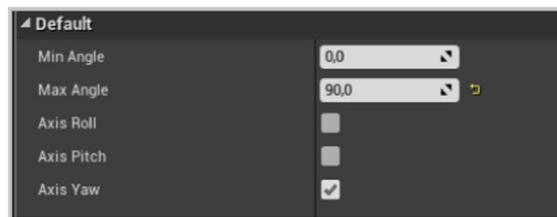


Abbildung 73: Editierbare Eigenschaften eines HingedDoor Blueprint (Screen Shot aus Unreal Engine 4)

An letzter Stelle muss die Rotation gesetzt werden. Das Ergebnis wird in die entsprechende Achse übernommen. Alle anderen Achsen behalten ihren Wert.

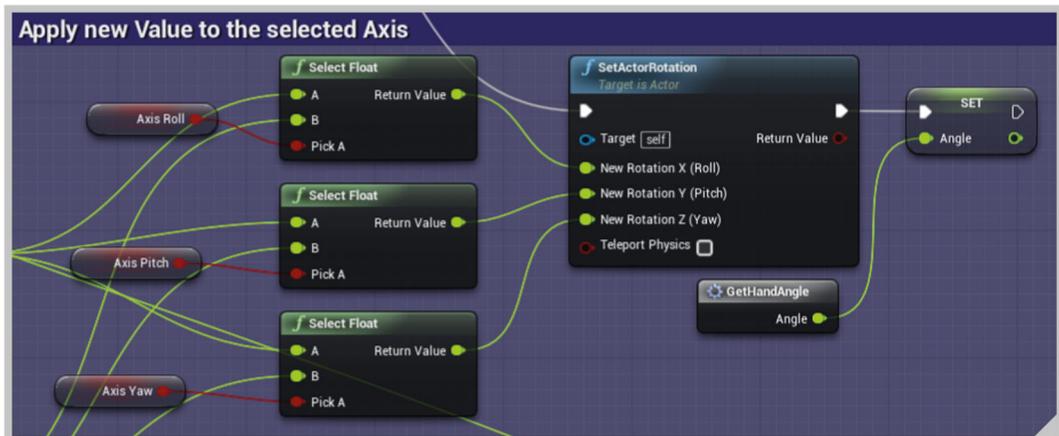


Abbildung 74: Blueprint zur Aktualisierung der Drehung entlang der ausgewählten Achse (Screen Shot aus Unreal Engine 4)

Ein Problem zeigte sich, nachdem die Gegenstände in der Küche aufeinander, nebeneinander und übereinandergestapelt wurden, um das Bild einer unaufgeräumten Küche wiederzugeben. Die Collisions von den heruntergeladenen Assets sind nicht ausreichend detailreich.

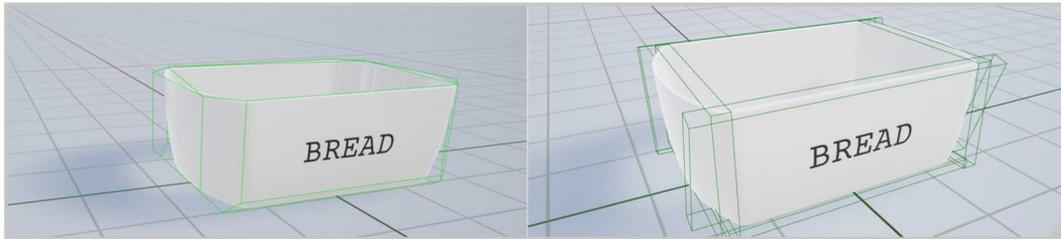


Abbildung 75: Darstellung der Collision Boxes, original (links), bearbeitet (rechts), (Screen Shot aus Unreal Engine 4)

Falls der Spieler mit Objekten interagieren soll, welche eine konkave oder konvexe Kollision simulieren, müssen diese überarbeitet werden. Das linke Bild bei Abbildung 72 zeigt die Standard Collision einer Brotdose. Mit den Collisions ist es nicht möglich ein Brot in diese Dose zu legen. Das rechte Bild entspricht der überarbeiteten Version. Dabei wurde zuerst die *Default Collision* entfernt und aus mehreren Collision Boxes eine konkave Form gebaut. Nun sollte der Spieler in der Lage sein mit den interagierbaren Objekten zu kommunizieren.

Woher weiß jedoch die Engine, mit welchem Objekt der Spieler interagieren wollte, wenn er zu dem Griff der Schublade greift? Aus Sicht des Users ist es offensichtlich, dass er zu dem Griff der Schublade fasst. Die Engine erkennt allerdings den Griff als auch das Besteck in der Schublade in unmittelbarer Nähe. Sobald der Spieler aus Versehen wenige Zentimeter in die Schublade greift, kann er das Besteck aus der geschlossenen Schublade herausziehen.



Abbildung 76: Darstellung der Entnahme eines Gegenstandes aus der geschlossenen Schublade (Screen Shot aus Unreal Engine 4)

6 Umsetzung / Usertest

Die erste Überlegung konzentrierte sich auf bewegbare Objekte, wie die Schublade und die Schranktür, sodass sie als präferierte Gegenstände in der Nähe der Hand erkannt werden. Diese Idee scheiterte jedoch daran, dass die Interaktion mit den Gegenständen in der Schublade sehr schwierig wurde. Die Schublade und die Türe wurden ständig als näheres Objekt erkannt und der Spieler bewegte sie öfter als dass er in der Lage war, den Inhalt in seine Hand zu nehmen.

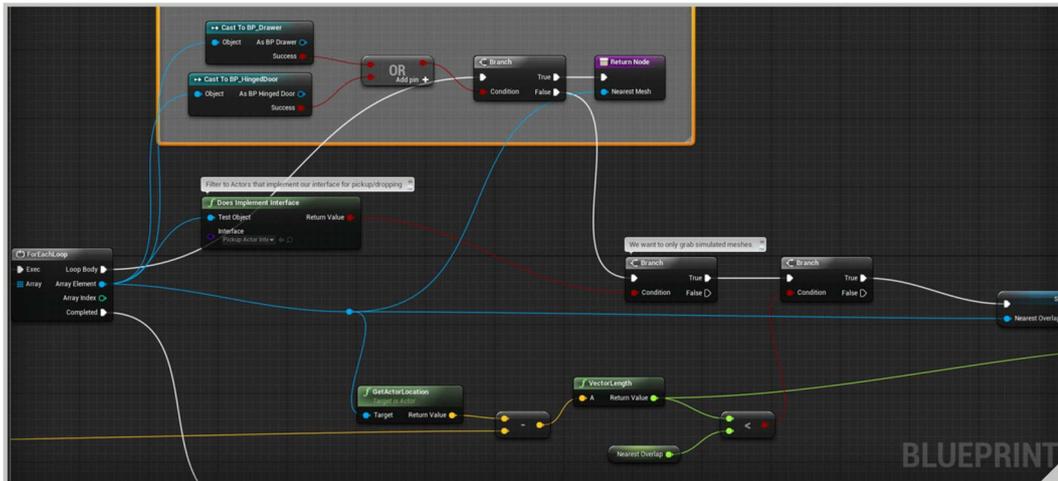


Abbildung 77: Anpassung des Vorlagenblueprints zur Präferenzierung bestimmter Objekttypen (Screen Shot aus Unreal Engine 4)

Letztendlich wurde daher ein zweites Objekt entwickelt, welches in diesem Fall eine kleine Kugel (Sphere) mit einem größeren Collisionsbereich ist. Diese Kugel folgt der Bewegung der Hand, wird jedoch von soliden Gegenständen aufgehalten. Die Idee dabei ist, dass diese Kugel hinter den Objekten, mit denen sie kollidiert, stehen bleibt und dafür sorgt, dass die *Grab Actor*-Funktion auf dieses Objekt reagiert und nicht auf die Objekte welche näher an dem Handmesh sind.

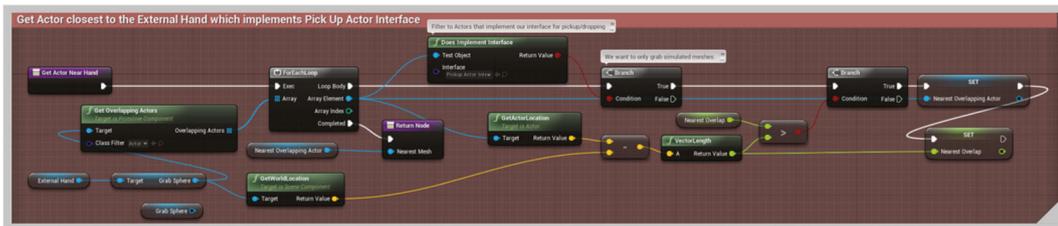


Abbildung 78: Blueprint zur Ermittlung des aufzunehmenden Objekts (Screen Shot aus Unreal Engine 4)

Da unter bestimmten Umständen die Kugel in der Szenerie stecken bleiben kann, wird sie ab einer Distanz von 20cm automatisch und unabhängig von den dazwischenliegenden Gegenständen nachgezogen. Dadurch wird es zwar

6 Umsetzung / Usertest

möglich sein Gegenstände durch tiefes Greifen in die geschlossene Schublade hinauszuziehen, jedoch passiert dies nicht mehr zufällig und aus versehen.

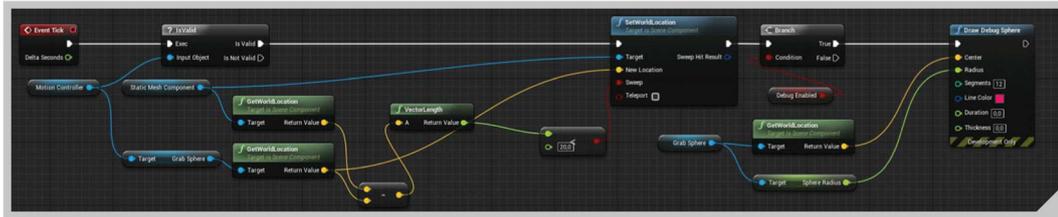


Abbildung 79: Blueprint um die Grab Sphere zu teleportieren, falls sie sich zu weit entfernt (Screen Shot aus Unreal Engine 4)



Abbildung 80: Visualisierung der Grab Sphere (Screen Shot aus Unreal Engine 4)

Diese Kugel und der Kollisionsbereich werden beim Starten auf *unsichtbar* (Hidden in Game) gesetzt. Vor der Finalisierung hat sich die Möglichkeit ergeben, den Test mit einigen unerfahreneren Spielern durchzuführen. Dabei wurde festgestellt, dass das oben gewünschte Ziel erreicht worden ist. Ein Gefühl der Präsenz und Immersion war soweit vorhanden, dass die Spieler mit beiden Controllern die Objekte bewegt haben. Zudem wurde offensichtlich, dass sie die Interaktion verstehen und sich auf ihr Ziel zubewegen konnten. Dabei fiel auf, dass sie lieber gegangen sind statt ihre Hände zum Teleportieren zu verwenden. Wenn die Benutzer ein Glas in der Hand halten, ist Teleportieren mit einer Hand eine sehr ungewöhnliche Aufgabe. Daher präferierten sie, dass sie zu dem Zielpunkt gehen, statt sich zu teleportieren. Bei der geschaffenen Umgebung ist es möglich, dass die Länge der Küche (der Bereich in welchem sich der Spieler zu den Objekten bewegen soll) an den physikalischen Bereich (der von den Lighthouses getrackt wird) angepasst wird. Es ist jedoch unbestreitbar, dass diese Option bei wenigen Umgebungen vorhanden ist.

6.3.2 Schlafzimmer

Der User beginnt das zweite Level in einem ebenso unaufgeräumten Zustand. Er interagiert in diesem Level mit Hilfe eines Zeigestabes. Um mit den Objekten interagieren zu können, wurde wieder ein Interface erstellt. Als *Pointable* bezeichnet man ein Interface, welches vier Funktionen beinhaltet:

- OnStartPointing
- OnEndPointing
- OnStartDragging
- OnEndDragging

Während des Zeigens auf ein Objekt überprüft der Player, ob dieses Objekt das Interface implementiert hat.

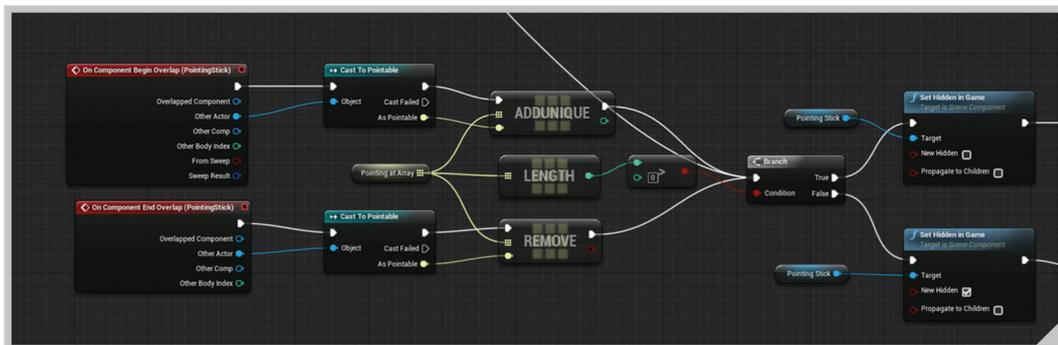


Abbildung 81: Blueprint zur Aktualisierung der Liste an Objekten, die vom Zeigestab überlappt werden. (Screen Shot aus Unreal Engine 4)

Diese Funktion setzt den Zeigestab auf sichtbar während er auf Objekte zeigt, welche *Pointable* implementieren.

Das Problem an diesem Level ist das Auswählen der Objekte. Grundsätzlich funktioniert das Auswählen so, dass man mit dem Zeigestab auf das Objekt zeigt und den *Trigger* betätigt. Dadurch aktiviert man die Funktion *OnStartDragging* und kann somit die Objekte bewegen. Angenommen es existieren zwei interagierbare Objekte, welche sich gleichzeitig mit dem Zeigestab überschneiden. Welches Objekt wird bewegt? Woher erfährt die Engine, welches Objekt gemeint ist?

Zur Lösung dieses Problems wurde die Funktion *GetClosestPointable* implementiert. *Line Trace for Objects* erzeugt eine *Trace*, die bei der Position des *Motion Controllers* (Die Hand des Spielers) beginnt und 10.000 Einheiten in die gezeigte Richtung läuft, um den Endpunkt der *Traces* zu setzen. *Motion Controller* als auch *Player* müssen ignoriert werden, damit die *Trace* überhaupt erzeugt

6 Umsetzung / Usertest

werden kann. Wenn diese *Trace* mit einem Objekt in Berührung kommt, setzt sie ihn als den Rückgabewert (*closest*) der Funktion.

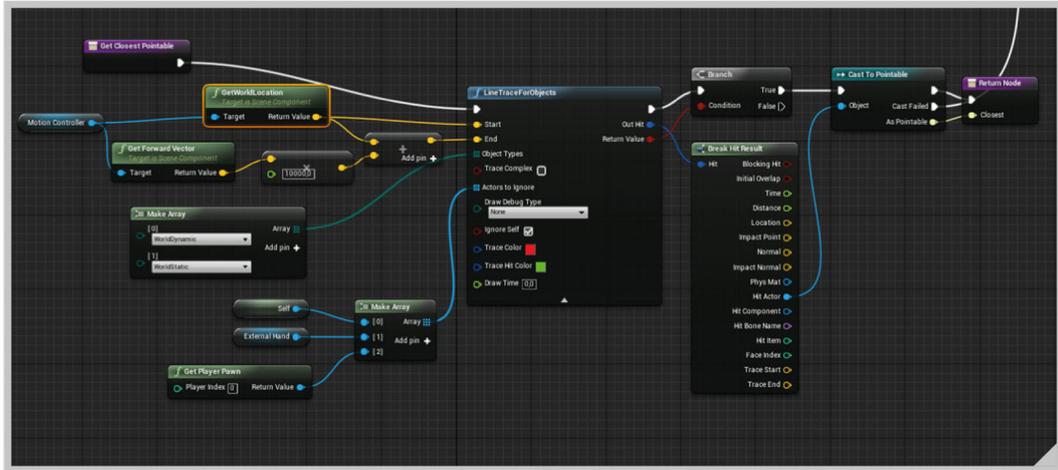


Abbildung 82: Blueprint zur Ermittlung des nächstgelegenen Objekts, auf das der Player zeigt (Screen Shot aus Unreal Engine 4)

Angenommen es stehen Objekte in unmittelbarer Nähe zueinander, sodass ein Objekt durch Zeigen *OnStartPointing* aktiviert wird, bevor die *OnEndPointing*-Funktion des anderen Objekts aktiviert wird. Es stellt sich die Frage, welches Objekt von der Engine gewählt wird. Um diese Überlappungsprobleme zu lösen, werden alle Objekte, welche mit dem Zeigestab überlappen, in dem Array *Pointing at* gespeichert. *ADDUNIQUE* fügt ein Element während des Zeigens dem Array hinzu, nachdem es überprüfen konnte, dass dieses Element in dem Array nicht bereits existiert. Beim Verlassen eines Objekts wird es aus dem Array mittels der *REMOVE*-Funktion wieder entfernt. Mit *LENGTH* erfolgt die Überprüfung, ob dieses Array überhaupt ein Element beinhaltet.

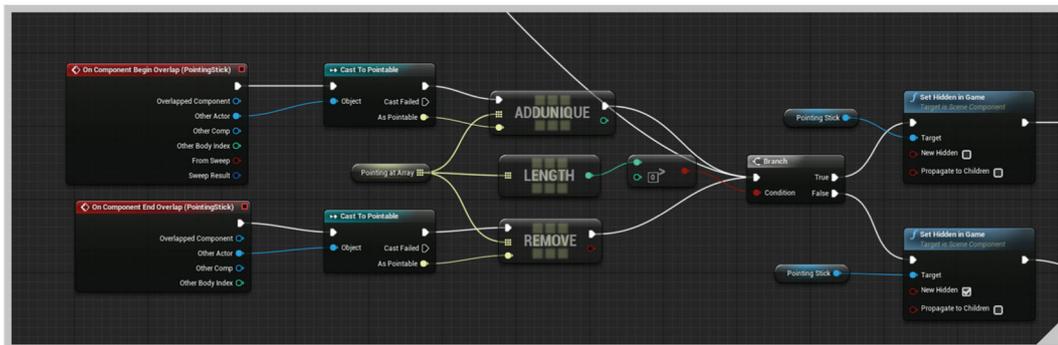


Abbildung 83: Blueprint zur Aktualisierung der Liste an Objekten, die vom Zeigestab überlappt werden. (Screen Shot aus Unreal Engine 4)

6 Umsetzung / Usertest

Zuerst war eine andere Funktion für die Bewegung zuständig. Diese Berechnung machte jedoch einen Fehler.

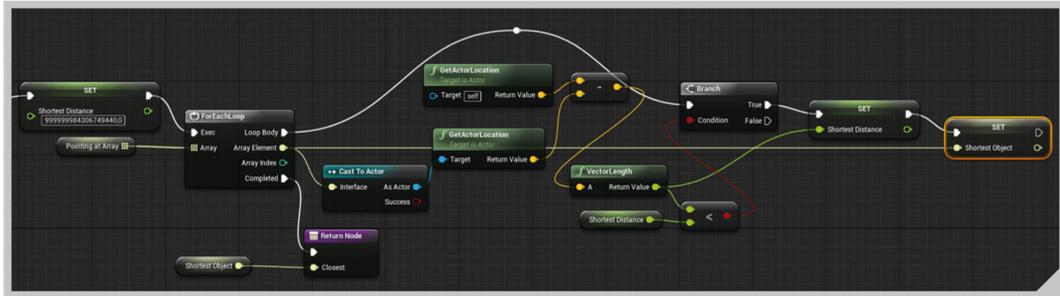


Abbildung 84: Blueprint zur Ermittlung des nächstgelegenen Objekts, auf das der Player zeigt, (erste Version) (Screen Shot aus Unreal Engine 4)

Get Actor Location berechnet die Position des *Pivot points* eines *Actors* in der Welt. Angenommen der Spieler zeigt mit dem Zeigestab auf einen Actor in der Welt, der neben einem anderen Actor steht. Aufgrund der Position des *Pivot points* der Objekte liegt aber das weiter entfernte Objekt näher *an* der Hand als das Anvisierte. Die Engine erkennt in diesem Fall das falsche Objekt und hängt es an den Zeigestab. Um diesen Fehler zu vermeiden, musste man auf eine Funktion zugreifen, die nicht von dem *Pivot point* eines Objekts ausgeht, sondern von jener Stelle, an welcher sich Objekt und Zeigestab überschneiden. Dies stellt die Aufgabe bzw. Funktionalität von einem *Trace* dar. (Abbildung 82)

Ein weiteres Problem besteht in dem Wunsch, die Objekte in der Art zu bewegen, sodass es scheint als würden sie auf dem Boden hin- und hergeschoben werden.

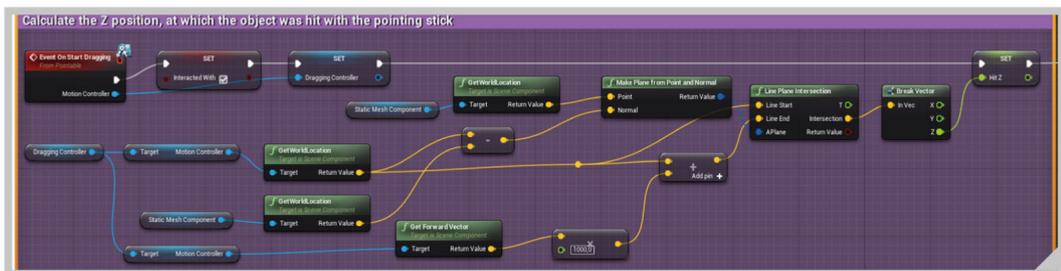


Abbildung 85: Blueprint zur Berechnung des Z-Werts des Treffpunktes (Screen Shot aus Unreal Engine 4)

Diese Funktion lokalisiert den Punkt, an welchem der Zeigestab das Objekt überschneidet. Dafür wird der Schnittpunkt des Zeigestabs mit einer im *Pivot point* des Zielobjekts liegenden und zum User ausgerichteten Ebene berechnet. Am Ende wird der Z-Wert dieses Punkts gespeichert.

6 Umsetzung / Usertest

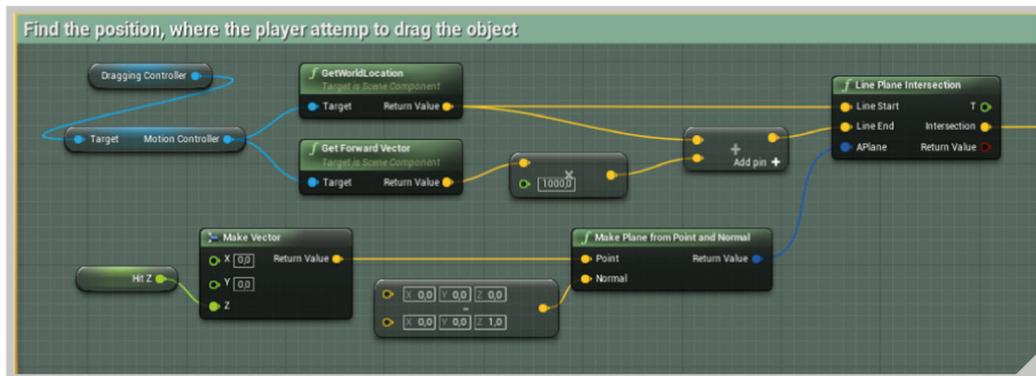


Abbildung 86: Blueprint zur Berechnung der Zielposition (Screen Shot aus Unreal Engine 4)

Nun kann man mit Hilfe von diesem Wert eine weitere Ebene erstellen, die genau diesen Z-Wert besitzt und mit dem *Normal* nach oben zeigt. Auf dieser Ebene bewegt sich nun jedes Objekt.

Diese Methode konnte gleichzeitig zwei Probleme lösen. Das erste Problem war das Hin- und Herschieben, was bereits erwähnt wurde. Das zweite Problem bezieht sich auf die Eigenschaft der Funktion *Get Actor Location*. Die Anwendung dieser Funktion führt dazu, dass die Objekte mit ihrem *Pivot point* an den Zeigestab *snappen*, sobald sie angehoben werden. Es verursachte einen Sprung am Anfang jeder Interaktion. Die Funktion konnte dieses Problem ebenfalls beheben.

Die Position des Objekts kann jetzt mit einer Transition errechnet werden.

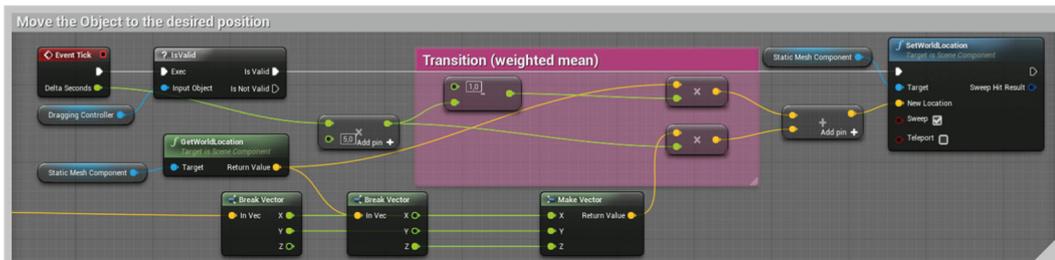


Abbildung 87: Blueprint zur Bewegung des Objekts zur Zielposition (Screen Shot aus Unreal Engine 4)

Da die Objekte einen *Collider* besitzen, können sie zuerst gar nicht bewegt werden, da sie mit dem Boden kollidieren sobald man sie mit dem Zeigestab in die Hand nimmt. Kollisionen müssen jedoch berechnet werden damit verhindert wird, dass Objekte durch Wände oder ineinandergeschoben werden können. Dafür werden die Objekte beim *OnStartDragging* um einen Zentimeter angehoben. *OnEndDragging* macht diese Funktion wiederum rückgängig.

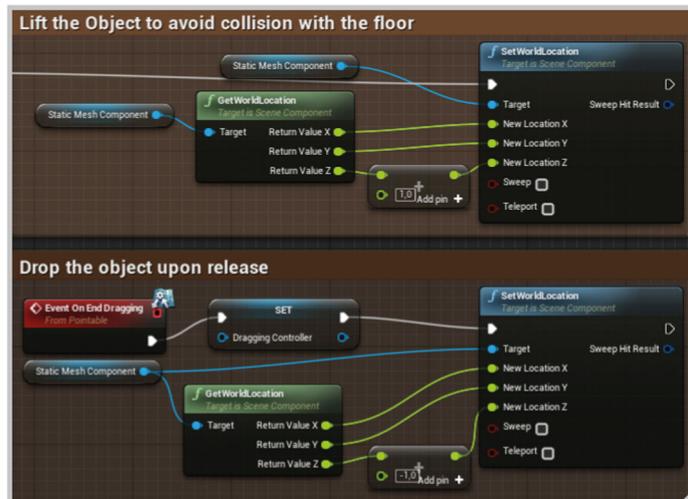


Abbildung 88: Blueprint zum temporären Anheben des gewählten Gegenstands (Screen Shot aus Unreal Engine 4)

Da die Möglichkeit gegeben sein muss, die Farben mancher Wände und das Material des Bodens zu verändern, sind diese ebenfalls *Pointable* und können auf den Zeigestab reagieren. Das alte *Material* wird mit dem neuen ersetzt, sobald der Player den Trigger betätigt.

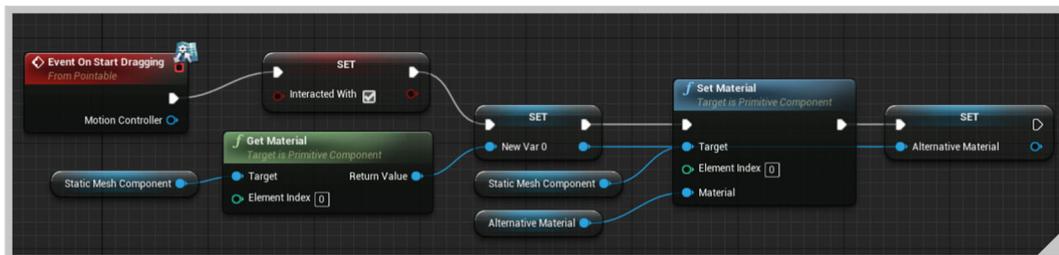


Abbildung 89: Blueprint zum Austausch des Materials eines Objekts (Screen Shot aus Unreal Engine 4)

Bei beiden Umgebungen handelt es sich um denselben Player. Der Zeigestab wird nur dann auf sichtbar gesetzt, wenn dieser mit Objekten in Berührung kommt, welche *Pointable* sind. Da im ersten Level keine Objekte als *Pointable* implementiert wurden, ist der Zeigestab dort nie sichtbar.

Wie bereits angesprochen, war ein Problem der Interaktion in diesem Level die Auswahl von Objekten, welche „geschoben“ bzw. „genommen“ werden konnten. Zu Beginn waren im Raum ebenfalls greifbare Bücher, Kerzen und einige weitere Objekte vorhanden. Während der Entwicklung stellte sich heraus, dass das Interagieren mit Objekten für einen User jedoch verwirrend ist. Geplant war, dass der Zeigestab nur bei den „schiebbaren“ Objekten sichtbar wird. Was passiert allerdings, wenn sich die Hand einem Objekt nähert, welches nicht verrückbar aber

wiederum so nah an einem schiebbaren Objekt ist, sodass der Player mit beiden in Berührung kommt? Lediglich einige Versuche waren nötig um zu erkennen, dass die Interaktion in solch einer Form zur Frustration führt. Daher wurden jene weiteren Objekte aus der Umgebung entfernt.

6.3.3 Gefängnis

In diesem Level wurden vom User vier hauptsächliche Interaktionen verlangt, welche für den Test relevant waren. Alle übrigen Interaktionen dienten dazu, dass die Umgebung begehbar wird und interessant wirkt. Die vier Interaktionen waren:

- 1: Die Wand musste zur Seite geschoben werden.
- 2: Die Decke musste hochgeschoben werden.
- 3: Eine Gittertür musste geöffnet werden.
- 4: Das Fenster in der Wand musste hinuntergezogen werden.

Die erste Wand hat grundsätzlich dieselbe Funktionalität wie die Schubladen des ersten Levels. Sie unterscheidet sich hinsichtlich der Achse, denn sie kann lediglich entlang der X-Achse bewegt werden. Sie wird beim Start auf eine Position gesetzt, die einen Spalt bildet. Dieser Spalt wurde mit einem kleinen Licht versehen, welches in die Zelle hineinscheint. Dieses Licht sollte den Blick des Spielers auf den Spalt lenken, sodass er auf die *merkwürdige* Position der Wand aufmerksam wird.

Nachdem der Spieler die Wand zur Seite geschoben hat, betritt er den nächsten Raum, in welchem er die Decke hochschieben muss. Jene Decke hat ebenfalls die gleiche Funktionalität wie die Erste, bewegt sich jedoch entlang der Z-Achse. Daraufhin öffnet man eine Gittertür, bei welcher es sich um eine *hinged-Door* handelt. Diese Tür verhält sich ähnlich wie die Schranktüren der Küche. Dafür musste nur ein Range gesetzt werden, in der sie sich bewegen darf.

Eine neue Funktionalität, welche in diesem Level hinzugefügt wurde, ist das schiebbare Fenster in der Wand. Es stellt die letzte Herausforderung für den User dar, um sich aus dem Gefängnis zu befreien.



Abbildung 90: Darstellung des bewegbaren Fensters (Screen Shot aus Unreal Engine 4)

Grundsätzlich muss das Fenster (Abbildung 90) entlang der Z-Achse nach unten gezogen werden. Seine Position auf der X-Achse bleibt erhalten. Mit der Bewegung muss sich die Öffnung in der Wand ebenfalls bewegen, ansonsten würde man nur die Mauer sehen. Um dieses Ziel zu erreichen, musste auf die Textur der Wand eine *Opacity Mask* gelegt werden, welche die Größe des Fensters hat. Die Position der Maske aktualisiert sich mit der Position des Fensters, sobald es vom Player bewegt wird. Dieser Bereich befindet sich zwischen zwei Werten. Der erste Wert ergibt sich aus einem Wertebereich zwischen R (Abbildung 91) und dem Wert 0.55. In diesem Bereich ist die Wand mit einer Sichtbarkeit von 1 versehen, ist also vollständig sichtbar.

$A = [0...1]; B = 0.55; (A > B? 0 : 1);$

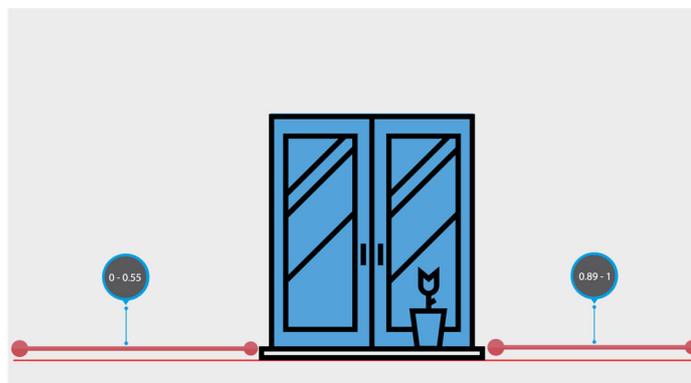


Abbildung 91: Darstellung des Bereichs, in dem sich das Fenster bewegen kann, (eigene Darstellung)

6 Umsetzung / Usertest

Das Ergebnis, welches aus dem ersten Node zurückgegeben wird, wird bei der zweiten IF-Node weiterverarbeitet. Dadurch wird jener, auf der rechten Seite des Fensters liegende, Bereich sichtbar.

$$A = [0..1]; B = 0.89; (A > B ? 1 : 0)$$

Abbildung 92 stelle die obige Rechnung dar.

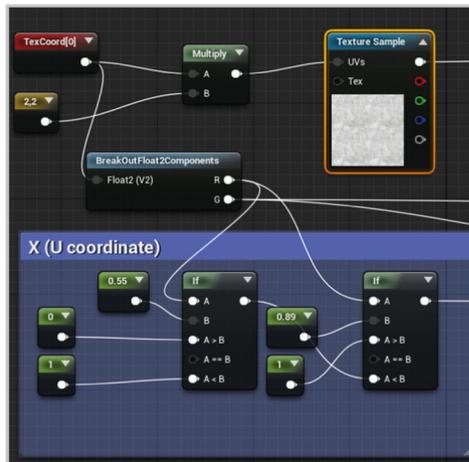


Abbildung 92: Ermittlung der Transparenz entlang der X-Achse, (Screen Shot aus Unreal Engine 4)

Der zurückgegebene Wert musste mit Werten der Y-Achse kombiniert werden.

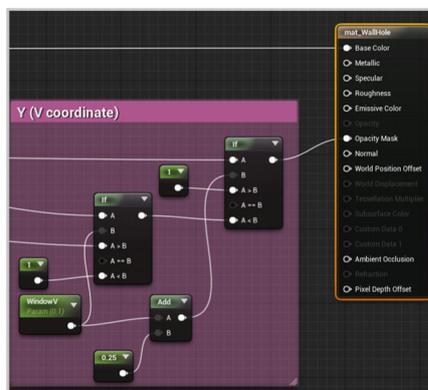


Abbildung 93: Ermittlung der Transparenz entlang der Y-Achse, (Screen Shot aus Unreal Engine 4)

$$A = [0..1]; B = WindowV; (A > B ? (U\text{-Coordinate}) : 1)$$

B ist hier der Wert, der sich aus der Addition von WindowV und 0.25 ergibt. 0.25 ist ebenfalls die Höhe des Fensters. Der zweite IF-Node überprüft folgendes:

Wenn A größer als B ist, sollte die Sichtbarkeit auf das Ergebnis der vorherigen Rechnung gesetzt werden. Wenn A kleiner ist als B, lautet das Ergebnis 1, d.h.

6 Umsetzung / Usertest

das Fenster ist komplett undurchsichtig. Zusammengefasst wird diese Rechnung an die *Opacity Mask* des Materials in Form einer Ternäroperation angegeben:

$$\text{Opacity Mask} = V > (\text{WindowV} + 0.25) ? (V > \text{WindowV} ? (U > 0.89 ? 1 : (U > 0.55 ? 0 : 1)) : 1) : 1$$

Dieses Material muss zu einem Dynamischen umgewandelt werden, damit man mit ihm interagieren kann:



Abbildung 94: Blueprint zur Erstellung eines dynamischen Materials für Interaktionsmöglichkeiten, (Screen Shot aus Unreal Engine 4)

Der Wert *WindowV*, mit welchem die Position des transparenten Bereichs der Textur aktualisiert wird, muss erst berechnet werden. Diese Berechnung erfolgt in zwei Schritten: Erstens wird die Position des Fensters berechnet. Aus der Position des Fensters wird der Wert für den Parameter *WindowV* abgeleitet und mit *Set Scalar Parameter Value* gesetzt.

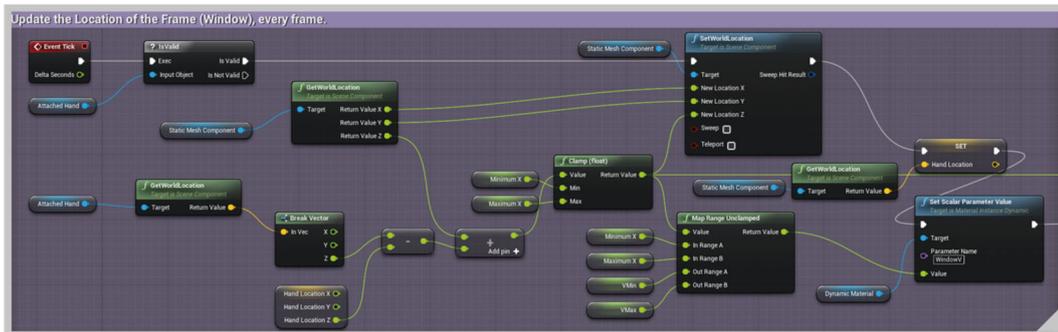


Abbildung 95: Blueprint zur laufenden Aktualisierung der Position des Fensterrahmens und der transparenten Öffnung, (Screen Shot aus Unreal Engine 4)

Sobald die Hand des Players das Fenster greift, wird die Position des Fensters permanent neu berechnet, dies bildet den zweiten Schritt. Jene Berechnung ergibt sich aus der Addition der Handposition, die bei jedem Frame aktualisiert wird, als auch der Fensterposition. Um diesen Wert im gewünschten Bereich zu erhalten, wird er zwischen *MaximumX* und *MinimumX* geclamt. Beide Werte werden im *Konstruktor* initialisiert. (Abbildung 96)

6 Umsetzung / Usertest

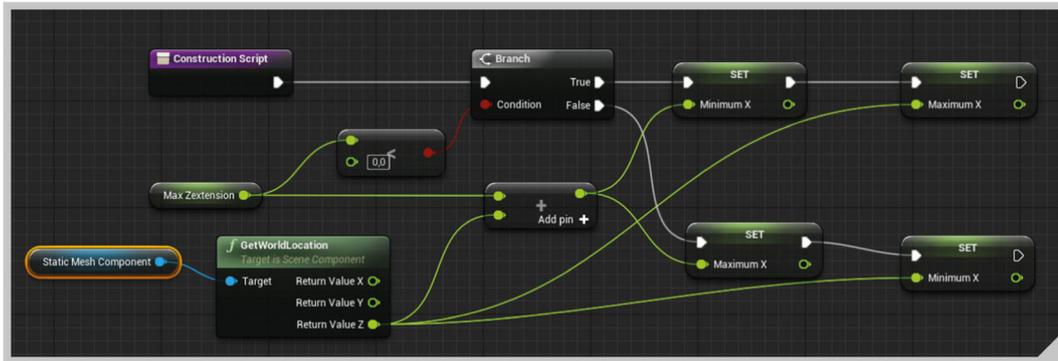


Abbildung 96: Blueprint zur Berechnung des Bewegungsbereichs entlang einer Achse, (Screen Shot aus Unreal Engine 4)

Zur Erinnerung: Die Aufgabe dieses Fensters besteht darin, dass der User die Schwelle erst überschreiten kann nachdem er das Fenster hinuntergezogen hat. Dies bedeutet, dass die Position des Fensters die *Collision* der Wand ebenfalls aktualisieren bzw. deaktivieren muss.

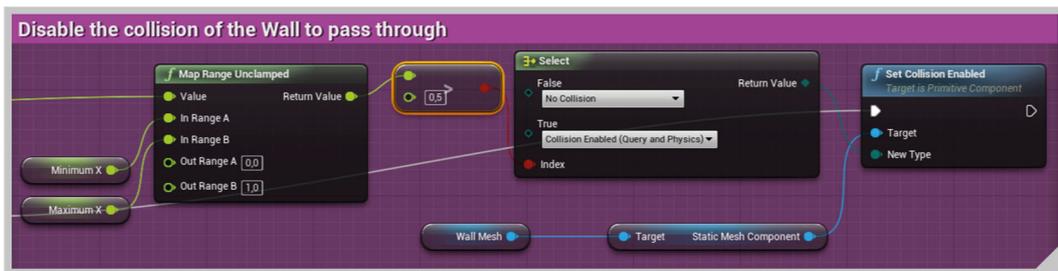


Abbildung 97: Blueprint zur Deaktivierung der Wandkollision, (Screen Shot aus Unreal Engine 4)

Sobald der User dieses Fenster über die Hälfte des Bereichs, in welchem sich das Fenster bewegen kann, hinunterzieht, wird die *Collision* deaktiviert.

6.3.4 Garage

Zu Beginn wurden die *Collisions* und *Physics* von allen Objekten dieser Umgebung entfernt. Der User kann nur mit der Wanduhr interagieren. Zwei wichtige Dinge müssen vorbereitet werden:

- Die Zeiger der Uhr
- Das Licht bzw. der simulierte Sonnenstand

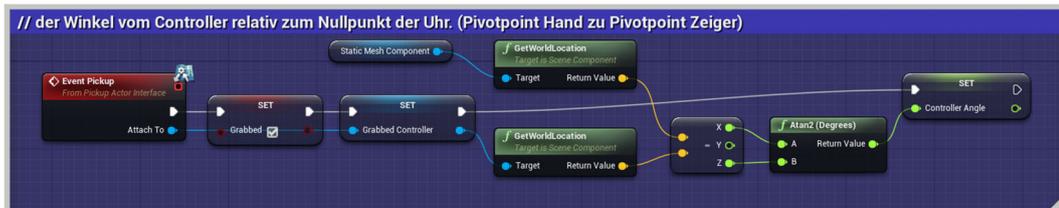


Abbildung 98: Blueprint zur Berechnung des Winkels zwischen Controller und dem Mittelpunkt der Wanduhr, (Screen Shot aus Unreal Engine 4)

Wenn der Spieler seine Hand zum Zeiger bewegt, um diesen zu greifen, wird an erster Stelle ein Winkel berechnet, welcher zwischen dem Winkelursprung und jener Linie liegt, die durch den *Pivot point* der Hand und des Zeigers gezeichnet wurde. Dadurch wird die Position des Controllers initialisiert.

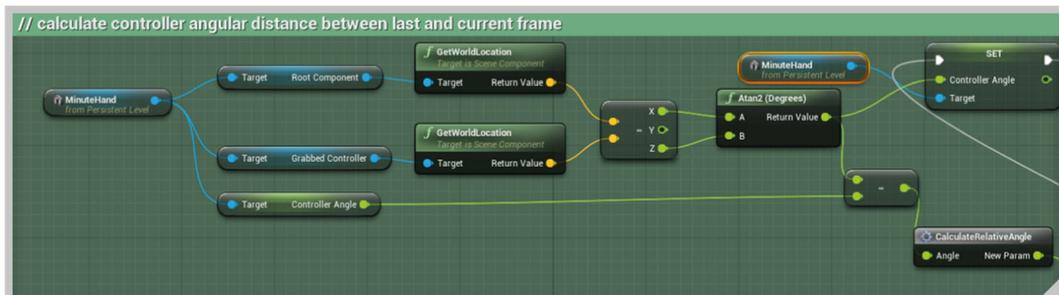


Abbildung 99: Blueprint zur Berechnung des relativen Winkelabstands, (Screen Shot aus Unreal Engine 4)

Controller Angle ist der Winkel, der bei jedem Frame aktualisiert wird.

6 Umsetzung / Usertest

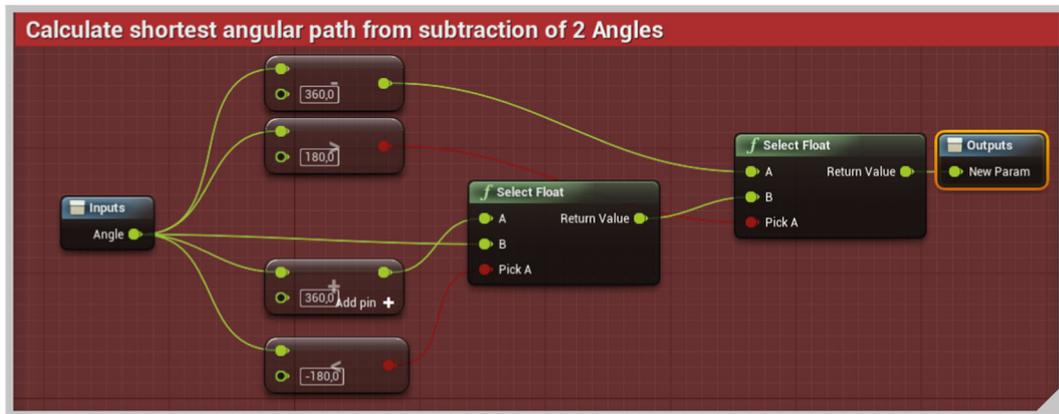


Abbildung 100: Blueprint zur Normalisierung einer Winkelangabe, (Screen Shot aus Unreal Engine 4)

Dieser Teil ist notwendig um einen Sprung des Zeigers bei 12 Uhr-Stellung zu vermeiden. Die *Atan*-Funktion der Unreal Engine gibt eine Zahl im Bereich zwischen -180 bis 180° zurück. Aus diesem Grund wird der Wert mit 180° verglichen. Dieser Teil sorgt dafür, dass die Bewegungen, welche größer sind als die Hälfte eines Kreises, automatisch auf die kürzere Bewegung in die entgegengesetzte Richtung abgebildet werden.

Die Bewegung der *Light source* ist, genau wie die Position des Zeigers, von der simulierten *InGame-Zeit (Current Time Minutes)* abhängig.

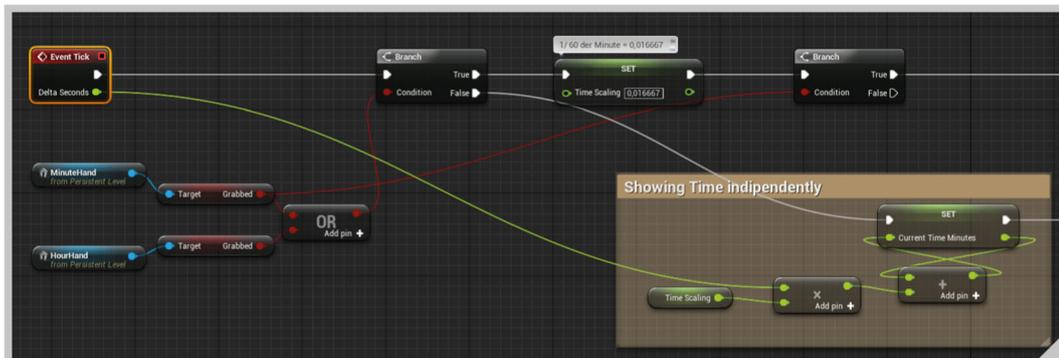


Abbildung 101: Blueprint zur Berechnung der InGame-Zeit, (Screen Shot aus Unreal Engine 4)

Es wird überprüft ob einer der beiden Zeiger momentan berührt wird. Ist dies **nicht** der Fall, bewegen sich Zeit und damit auch das Licht eigenständig weiter. *TimeScaling* wurde zu Beginn mit dem Wert 40 initialisiert und bedeutet, dass eine Sekunde in Echtzeit im Spiel wiederum 40 Minuten entspricht. *Delta Time in Seconds* beschreibt die Zeit, welche zwischen zwei Frames vergangen ist und wird mit *TimeScaling* multipliziert, um die *InGame-Zeit* voranschreiten zu lassen.

6 Umsetzung / Usertest

Wird ein Zeiger angefasst, muss die Bewegung der Uhr als auch die Zeit mit der Hand verbunden werden. *Time Scaling* wird an der ersten Stelle auf einen realistischen Wert gesetzt. 0,016667 entspricht der Anzahl an Minuten innerhalb einer Sekunde (1/60). Mit Anfassen des Minutenzeigers werden aktuelle Zeit und Position des Zeigers gesetzt.

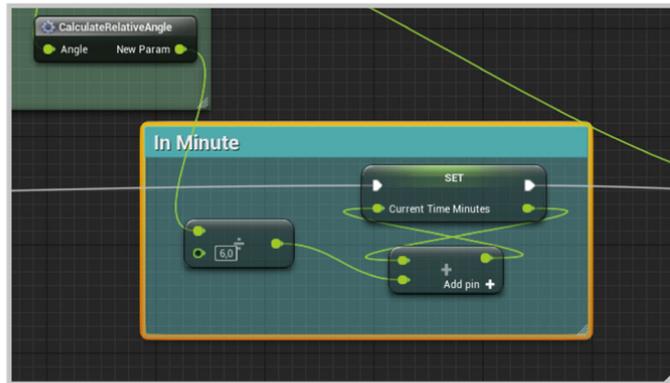


Abbildung 102: Blueprint zur Bewegung des Minutenzeigers, (Screen Shot aus Unreal Engine 4)

Für weitere Einstellungen ist der *Controller Angle* zuständig. Er ergibt sich aus der Winkeldifferenz zwischen dem aktuellen Winkel der Hand und deren Winkel im letzten Frame.

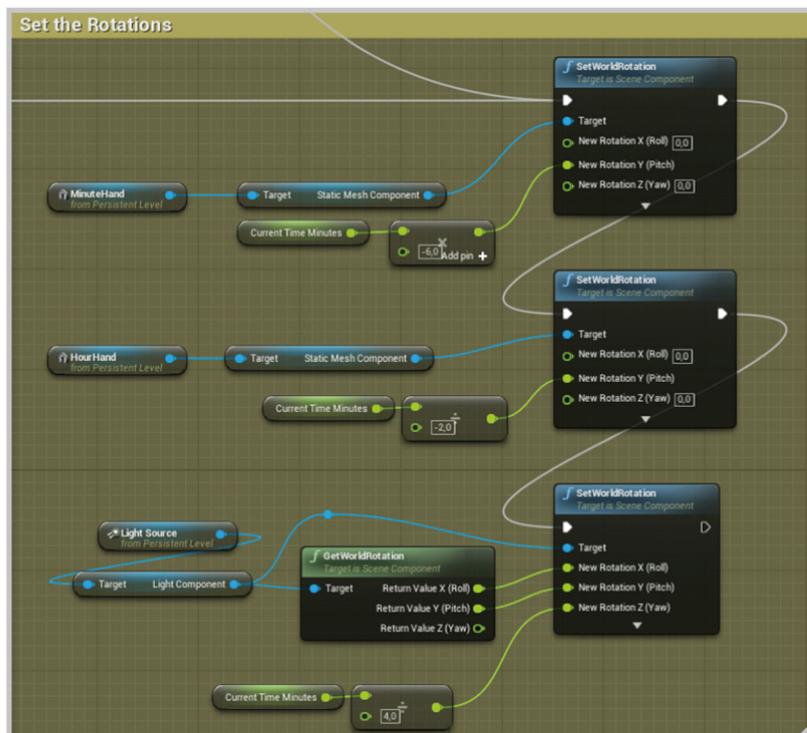


Abbildung 103: Blueprint zur Berechnung der Rotation der Zeiger und Lichtquelle, (Screen Shot aus Unreal Engine 4)

6 Umsetzung / Usertest

Hier werden die Winkel der Zeiger und der Lichtquelle eingestellt. Es wird für alle drei Elemente aus der aktuellen Uhrzeit ein Winkel berechnet.

```
// alpha = 360 * (Currenttime in Minutes / 60)
```

```
// beta = 360 * (Currenttime in Minutes / (60 *12))
```

```
// gamma = 360 * (Currenttime in Minutes / (60 *24))
```

Durch das Kürzen der Formeln ergeben sich die Multiplikations- bzw. Divisionsfaktoren mit denen die neuen Rotationswinkel berechnet wurden.

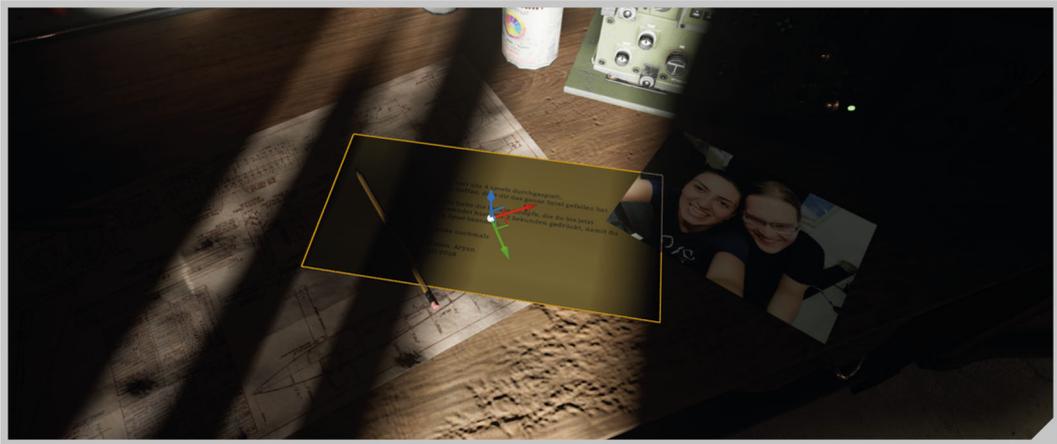


Abbildung 104: Darstellung der letzten Aufgabe des Usertests, (Screen Shot aus Unreal Engine 4)

Auf dem abgebildeten Zettel (Abbildung 104) steht eine Tastenkombination, welche das Level schließlich beendet (Abbildung 105). Das Material des Zettels ist so gewählt, dass sein Text nur unter bestimmten Lichtverhältnissen, also zu bestimmten Uhrzeiten, lesbar ist.

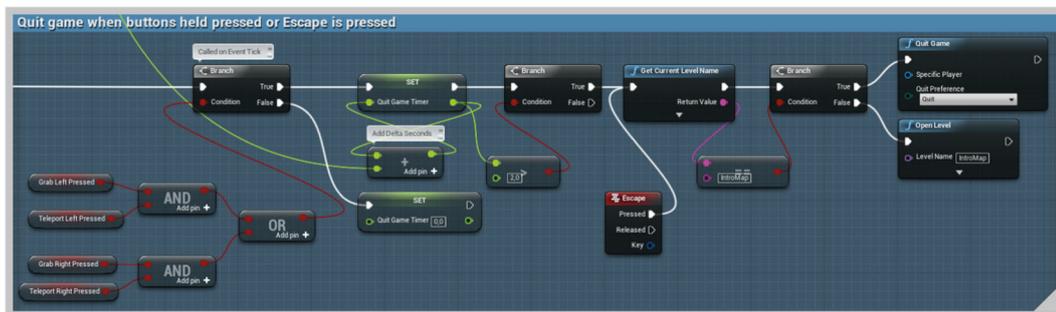


Abbildung 105: Blueprint zum Beenden der Anwendung mit den erwünschten Tastenkombinationen, (Screen Shot aus Unreal Engine 4)

6.4 Punktevergabe

Die Anzahl der Interaktionen des Players als auch dessen Interaktionszeit müssen zusammengezählt werden. Dafür wurde eine Funktion entwickelt, welche die Anzahl der Interaktion mit einem Objekt des Typs *BP_PickUp Item* hochzählt.

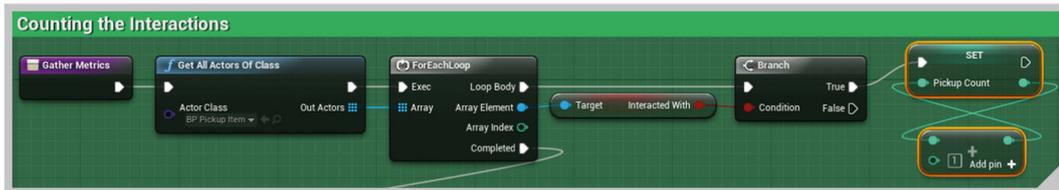


Abbildung 106: Blueprint zur Berechnung der Anzahl an Interaktionen, (Screen Shot aus Unreal Engine 4)

Die gleiche Funktion gilt für alle anderen *Types*.

Jene Werte, die in Variablen gespeichert sind, werden in *Metrics* Map der *Game Instance* eingetragen. Grund dafür ist eine Funktionalität der Unreal Engine, Werte und Funktionen existieren nur innerhalb eines Levels. Sobald der Spieler das Level verlässt, kann auf die Variablen und Werte nicht mehr zugegriffen werden.

Auf *Game Instance* kann man innerhalb des gesamten Projekts zugreifen.

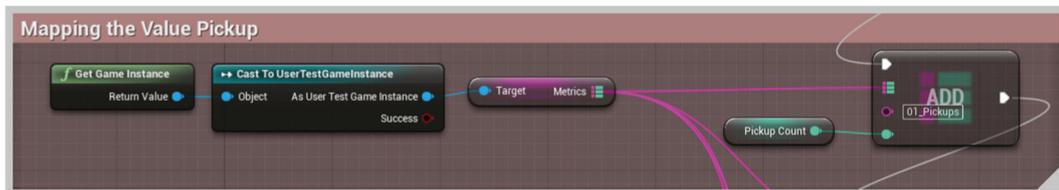


Abbildung 107: Blueprint zum Zugriff auf die in der *GameInstance* gespeicherten Daten, (Screen Shot aus Unreal Engine 4)

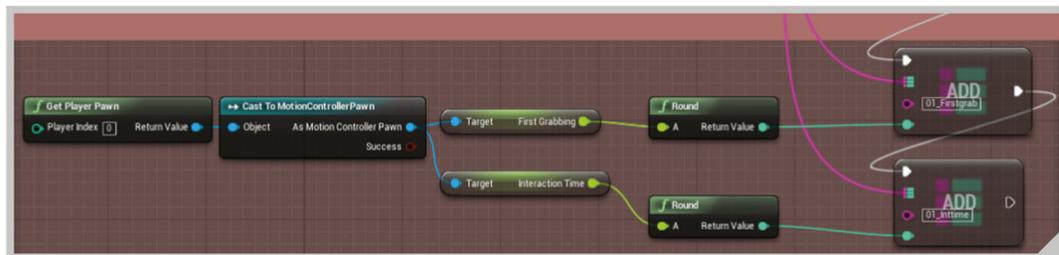


Abbildung 108: Blueprint zum Speichern von Zeitstempeln, (Screen Shot aus Unreal Engine 4)

6 Umsetzung / Usertest

Wichtig für die Evaluierung des Tests sind die Häufigkeit der Interaktion mit Objekten eines bestimmten *Types*, die Gesamtzeit und ebenfalls die allerersten Interaktionen mit einer Umgebung. Somit sind:

- *Types* Objekte, die eine bestimmte Art der Interaktion implementieren. Beispielsweise ist eine Tasse vom Typ *Grab Item*, Schränke sind vom Typ *Hinged-Door*.
- *Interaction Time* ist die Gesamtzeit, welche der Spieler in einem Level verbringt. Sie beginnt mit dem Ausblenden der Aufgabentafel und endet mit dem Verlassen des Levels.
- *First Grabbing* beschreibt jene Zeit, die ab dem Beginn des Levels bis zur ersten Interaktion gemessen wird.

Nachdem diese Variablen in den *Metrics* übertragen wurden, kann aus dem IntroMap Level darauf zugegriffen werden.

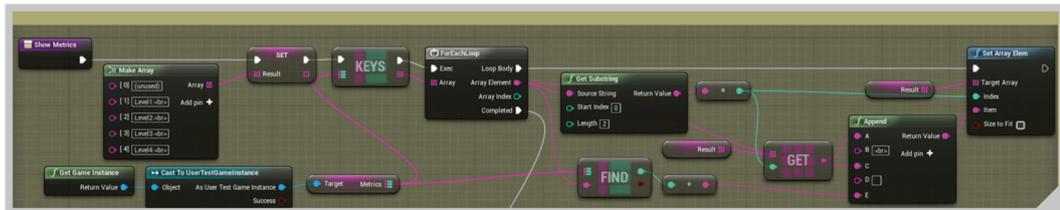


Abbildung 109: Blueprint zum Zusammenfügen der Textelemente für die finale Darstellung, (Screen Shot aus Unreal Engine 4)

In dieser Map werden die Werte aus den *Metrics* zu den passenden Beschreibungen hinzugefügt, um einen *Text* zu bauen, der letzten Endes im IntroMap Level angezeigt wird, sobald der Spieler ein Level verlässt.

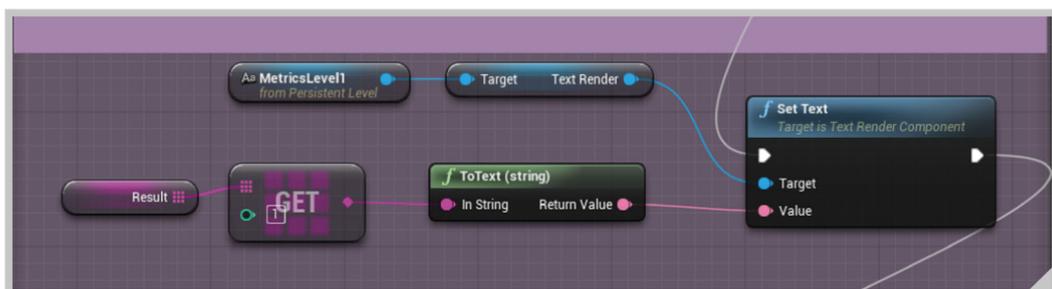


Abbildung 110: Blueprint zur Darstellung der Ergebnisse des ersten Levels, (Screen Shot aus Unreal Engine 4)

6.5 IntroMap

Dieses Level wurde zur Einführung entwickelt. Ein unerfahrener User sollte die Möglichkeit erhalten, sich mit der Funktionsweise der Tasten vertraut zu machen. Er nimmt die virtuelle Umgebung wahr und soll ebenso die Teleportation üben, gänzlich ohne Zeitdruck. IntroMap ist ein von der Unreal Engine erstelltes Wohnzimmer mit realistischen Lichtverhältnissen. Der Spieler beginnt den Test in diesem Level. Er setzt die VR-Brille auf, erkennt seine virtuellen Hände, nachdem er die Controller in die Hand bekommen hat und erhält die grundlegenden, notwendigen Hinweise über die Interaktion.



Abbildung 111: Das Einstiegslevel. Blick, wenn der Player den Usertest beginnt. (Screen Shot aus Unreal Engine 4)

An der rechten Wand des Zimmers hängen vier Bilder. Jedes Bild kann via Trigger ausgewählt werden und teleportiert den Spieler daraufhin in das entsprechende Level. Hierbei kommt der, aus der Beschreibung des Schlafzimmer-Levels bekannte, Zeigestab zum Einsatz. Das erste Bild verweist auf die Küche, das Zweite auf das Schlafzimmer usw.

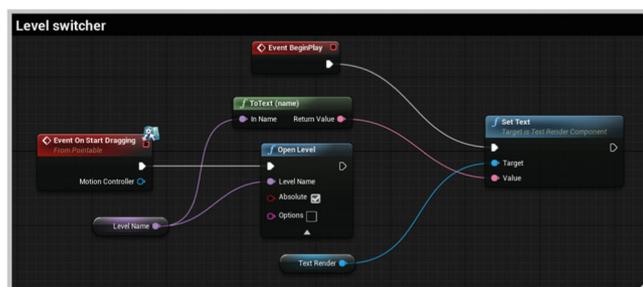


Abbildung 112: Blueprint des Umgebungswechslers, (Screen Shot aus Unreal Engine 4)

Der *Open Level Node* öffnet das Level, welches in der Variable *Level Name* gespeichert wurde.

6.6 Durchführung des Usertests

Die Art der Auswertung dieses Tests wurde bereits angedeutet und findet auf zwei Ebenen statt: Als erstes werden die Resultate tabellarisch überschaubar visualisiert, um ein anfänglichen Überblick zu erhalten. Daraufhin folgt die Auswertung aus Sicht des Beobachters während des Tests.

Grundsätzlich wurden vier zeitliche Messungen durchgeführt. An erster Stelle war es Intention dieses Tests, herauszufinden wie viel Zeit vergeht bis die Probanden ihre erste Interaktion mit einem Gegenstand unternehmen. Allerdings sollte das Umsehen zu Beginn jedes Levels keinen Einfluss auf diese Auswertung haben. Somit wurde die Zeit ab dem Ausblenden der Aufgabentafel bis zu der ersten Interaktion getrackt. Die Art der Interaktion spielte bei dieser Messung keine Rolle. Bei dem ersten Level handelte es sich um die Interaktion mit einem Gegenstand in der Küche, dem Öffnen einer Schranktür oder dem Anheben eines Löffels. Im zweiten Level wäre jene Interaktion das Verschieben von Möbeln oder das Ändern der Wandfarbe. Im dritten Level war diese Interaktion lediglich das Verschieben der Wand und im Vierten wurde die Zeit letztlich zwischen dem Ausblenden der Aufgabentafel und dem Anhalten des Uhrzeigers abgemessen. Im Laufe des Tests wurden die Probanden in allen Levels beobachtet und ihre Interaktion beeinflusst logischerweise das Ergebnis der zweiten Auswertung. Mit jedem Probanden wurde ein Abschlussgespräch geführt, in welchem acht Fragen gestellt wurden:

- 1: Hattest Du das Gefühl, dass sich der Schwierigkeitsgrad der Interaktion mit jedem Level erhöht?
- 2: Woher wusstest Du, wohin Du die Tassen in der Küche stellen solltest?
- 3: Wie viele Versuche waren nötig, um zu wissen, dass du die Schränke mit den roten Griffen nicht öffnen kannst?
- 4: Was war für dich in den Levels zu schwer?
- 5: Was hat dir an der Interaktion nicht gefallen?
- 6: Welche Interaktionen waren in deinen Augen im Vergleich zur Realität anders?
- 7: Konntest du dein Wissen in Umgang mit physischen Objekten in der realen Welt auf die Interaktion in der virtuellen Welt übertragen?
- 8: Was für eine Note würdest du für die Immersion in jedem Level geben?

6.7 Tabellarische Auswertung

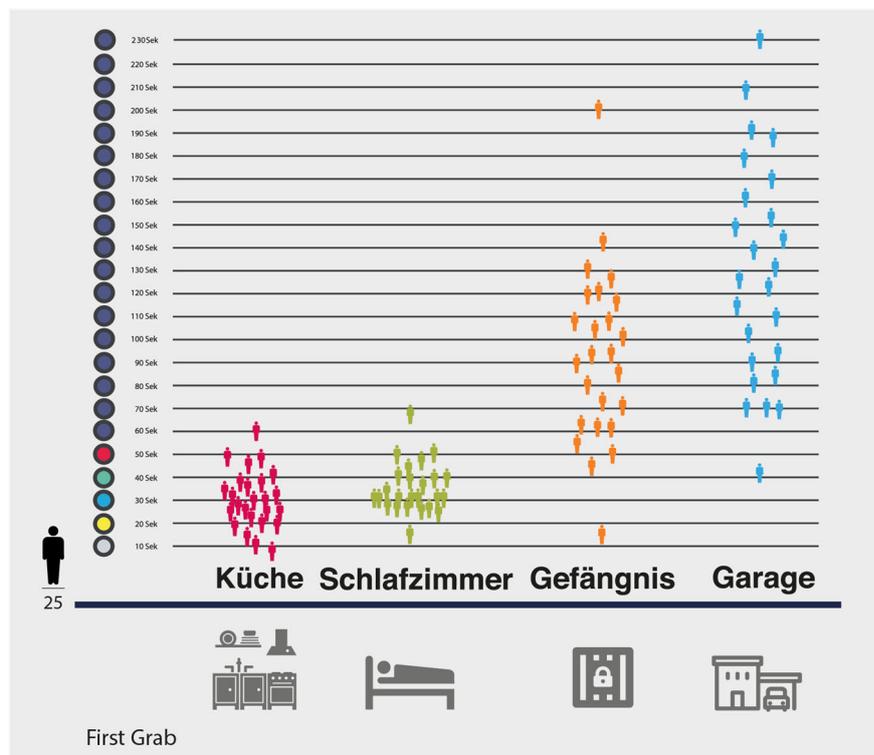


Abbildung 113: Darstellung der ersten Interaktion mit einem Objekt in den Umgebungen (in Sekunden),
(eigene Darstellung)

Dieses Diagramm zeigt die erste Interaktion mit einem Objekt im jeweiligen Level. Man erkennt, dass die Probanden tendenziell längere Zeit benötigen haben, um besagte erste Interaktion zu starten.

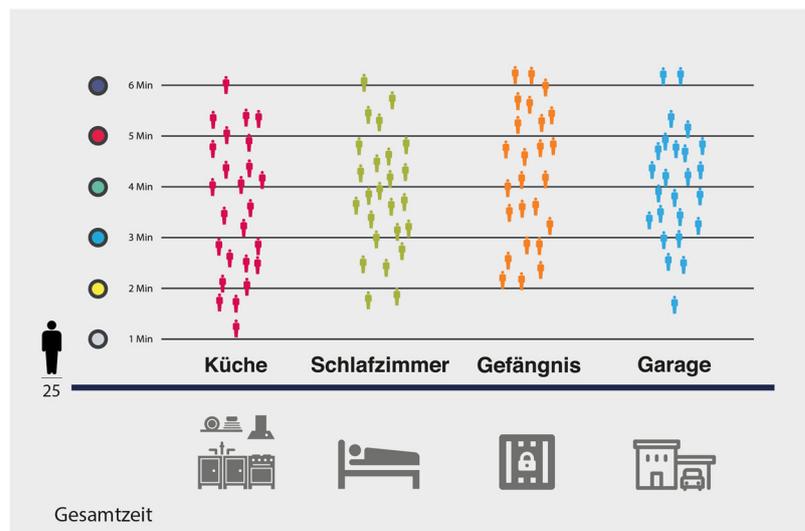


Abbildung 114: Darstellung der gesamten Zeit (in Minuten), die der Player in einer Umgebung verbracht hat, (eigene Darstellung)

Die Gesamtzeit zeigt die komplette Dauer an, welche die Probanden in einem Level verbracht haben. Sie beginnt mit dem Ausblenden der Aufgabentafel und endet mit dem Verlassen der Umgebung.

Immersion

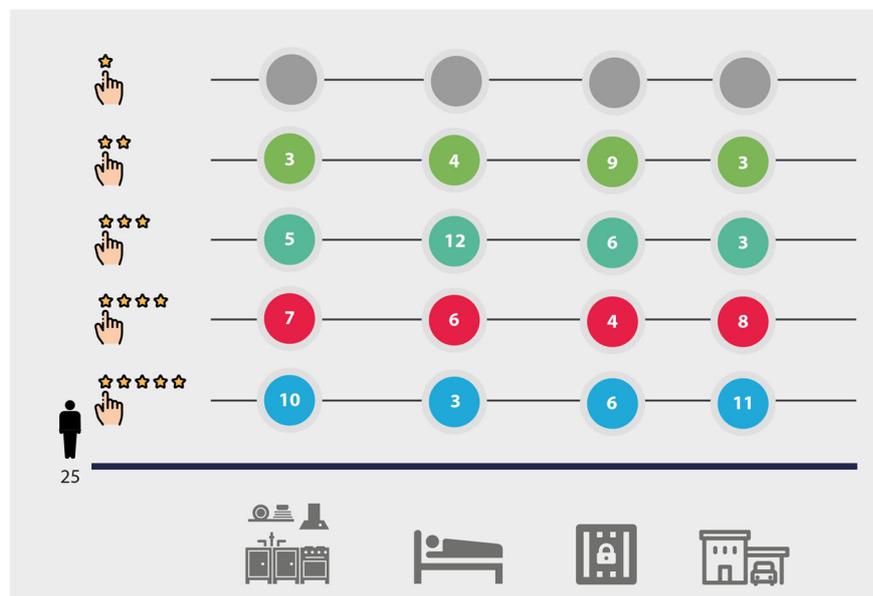


Abbildung 115: Darstellung der vergebenen Note hinsichtlich der Immersion, (eigene Darstellung)

Probanden wurden gebeten, jedem Level eine Schulnote hinsichtlich der Immersion zu geben.

Das erste Diagramm kann die Hypothese bestätigen: Je realistischer die Interaktion mit den Gegenständen entworfen ist, desto schneller wurde sie von den Probanden gefunden. Schon im zweiten Level konnte man Schwierigkeiten der User erkennen. Ihre erste Interaktion mit einem Gegenstand fand zwar schnell statt, war jedoch selten erfolgreich. Dies liegt daran, dass Sie versuchten die Gegenstände in die Hand zu nehmen und aktivierten dadurch ihren Zeigestab, woraufhin sie weitere Objekte durch Zufall bewegten. Vielen ist die Interaktion mit der Wandfarbe im Schlafzimmer durch diesen Zufall aufgefallen. Dies führte dazu, dass die Interaktion, obwohl sie nicht die Beabsichtigte war, trotzdem softwareseitig als die erste Interaktion registriert wurde. Somit stimmt die erste registrierte Interaktion von manchen Probanden mit dem, was sie tatsächlich als erste Interaktion vorgenommen haben, nicht überein.

Gleiches gilt ebenfalls für das Gefängnis. Der User startete dieses Level in einem Raum, in welchem keine weitere Aktion, außer dem Verschieben der vorgesehenen Wand, möglich war. Tendenziell benötigten sie etwas länger, um auf die Idee zu kommen, was mit diesem Spalt suggeriert werden sollte. Wenn die erste Aufgabe allerdings das Hochschieben der Decke gewesen wäre, könnte man behaupten, dass sich die Zeit zwischen Beginn des Levels und der ersten Interaktion verdoppelt hätte. Denn die meisten Probanden bückten sich, um darunter zu passen oder überlegten sehr lange, sodass ein externer Hinweis sie zu dem nächsten Schritt führen musste. Falls das Hinunterziehen des Fensters (letzte Aufgabe) wiederum die erste Aufgabe gewesen wäre, hätten viele der Probanden dieses Level abgebrochen und mehr als die Hälfte von ihnen hätten externe Hilfe benötigt.

Während des letzten Tests (Garage) stellte sich die Frage, ob das Anhalten des Uhrzeigers genügend Hinweise lieferte, um die Verbindung zu dem Licht herzustellen. Denn die Probanden haben, im Gegenteil zu dem, was in der Entwicklungsphase vermutet wurde, relativ schnell die Intention entwickelt, die Zeiger anzugreifen. Sie haben aber nicht bemerkt, dass sie damit die Bewegung des Lichts steuern. Dies legt wiederum folgenden Schluss nahe: Dass der User die Zeiger angegriffen hat, trifft keine Aussage darüber, dass sie das Licht damit zu steuern versuchten. Sie hielten die Zeit an und beobachteten ihre Umgebung. Wenn das Licht zufällig in ihrem Sichtfeld lag, konnte man davon ausgehen, dass sie eine Verbindung zwischen Licht und Zeiger hergestellt haben. Es war zwar interessant zu beobachten, dass die Probanden die Stundenzeiger angehalten und zu drehen versucht haben, manche fassten diese Reaktion jedoch falsch auf. Denn viele mussten an dieser Stelle darauf hingewiesen werden, dass sie weitermachen sollten, bzw. dass sie (wie es aus der realen Welt bekannt ist) die Minutenzeiger bewegen sollten.

6.8 Fazit des Usertests

Die in den Diagrammen dargestellten Werte liefern einen guten Eindruck über das Gesamtergebnis. Die Beobachtung der Probanden während des Tests hat jedoch viel mehr Aussagekraft. Entsprechende Informationen wurden während des Tests sorgfältig notiert. Zusätzlich wurde der Test von acht Experten durchgeführt, die aus eigener Erfahrung und aufgrund ihrer Kenntnisse den Test analysiert haben. Der folgende Abschnitt befasst sich daher mit diesen Informationen.

6.8.1 Allgemein

Im Allgemeinen kann man behaupten, dass Motion Sickness bei statischen Umgebungen kein Problem darstellt. Nur einer von 25 Probanden erwähnte ein seltsames Gefühl im Bauch und der Test dauerte insgesamt länger als gedacht, ohne dass er aus diesem Grund unterbrochen werden musste.

Die Interaktion mit den Controllern war bei vielen Probanden eine größere Herausforderung als bei der Entwicklung vermutet. Verwechseln und gleichzeitiges Betätigen der beiden Tasten löste bei vielen Probanden eine häufige Verwirrung aus. Es war ebenfalls zu beobachten, dass die Anspannung bei unerfahrenen Usern oft zu einem etwas gröberen Umgang mit den Tasten geführt hatte. Dies führte daraufhin dazu, dass Touchpad und Trigger zugleich betätigt wurden, was ebenfalls für Verwirrung sorgte.

Die Küche mit ihren kleinen Interaktionsmöglichkeiten diente als eine gute Übung, reichte vielen jedoch nicht aus. Dadurch wird verständlich, dass viele Entwickler, abhängig von ihrer Zielgruppe, alle Interaktionen nur mit Hilfe einer Taste zu ermöglichen versuchen. Bei Interaktionen und Aufgaben, welche der vorliegende Test stellt, ist dies jedoch nicht ratsam.

Der User hatte in allen Umgebungen einen relativ großen, begehbaren Bereich, in welchem er sich teleportieren konnte. Die Intention dahinter war, dass die Probanden sich in kleine Umgebungen nicht eingeschränkt fühlen sollten. Diese Bereiche führten bei vielen jedoch zur Verwirrung. Man kann behaupten, dass das Wohnzimmer im ersten Level, der Vorraum im zweiten, die mehreren Einzelzellen im dritten und der gesamte Bereich im vierten Level reduziert werden sollten. Durch eine passende Lichtsetzung und Farben wird der User kleinere Räume ebenfalls angenehmer empfinden, denn insbesondere die Größe der verwendeten Bereiche war keine Hilfe. Die Gesamtzeit, welche die Probanden in jedem Level verbracht haben, könnte ebenfalls von dieser Größe etwas beeinflusst sein, da sie oft an mehreren Stellen nach Lösungen gesucht haben. Interessant war die unvorhersehbare Beobachtung, dass viele User in dem Gefängnis einen Schlüssel

suchten, bevor sie versuchten eine Tür zu öffnen. Je unrealistischer die Erwartungen wurden, desto ungewöhnlicher wurden die geäußerten und versuchten Lösungsansätze der Probanden. Dies war vor allem im dritten Level besonders auffällig. Aus diesem Ergebnis kann man erkennen, wie einflussreich die Einschränkungen auch beim Design einer virtuellen Welt sind.

Es war bei vielen Probanden zu erkennen, dass der Übergang zu immer unrealistischeren Interaktionen zu schnell stattgefunden hat. Möglicherweise würde dieser Test andere Ergebnisse liefern, wenn sich dieser Schwierigkeitsgrad innerhalb einer höheren Anzahl an Level gesteigert hätte. Denn es darf nicht außer Acht gelassen werden, dass es sich um Probanden handelte, die keinerlei Erfahrung in Virtual Reality besaßen.

Eine der letzten Fragen bezog sich auf die Immersion. Die Probanden wurden befragt, ob sie jedem Level in Bezug auf die Immersion eine Schulnote geben.

Es wurde in der Entwicklungsphase vermutet, dass die Immersion mit jedem Level absteigen wird. **Immersion** kann als Eintauchen in eine virtuelle Welt verstanden werden. (Entertainment und Technik Magazin, 2015) Sie kann Gedanken und Gefühle zwar in eine gewollte Richtung lenken, jedoch nicht kontrollieren. (Jerald, 2015b, S. 80)

Präsenz ist das Gefühl des Daseins und kann als Erweiterung und als Ergebnis einer erfolgreichen Immersion verstanden werden. (Entertainment und Technik Magazin, 2015) Es lag die Vermutung vor, dass die Interaktion mit jedem Level schwieriger und realitätsfremder wird. Der User verliert immer mehr die Kontrolle über die Gegenstände. Ebenso verlängert sich auch die Zeit, welche er in einer Umgebung verbringt um eine Interaktion durchzuführen.

Die Frage nach Immersion wurde von den Probanden unterschiedlich aufgefasst. Grundsätzlich verbinden die VR-User Immersion mit zwei Faktoren: Sie referieren entweder auf die realistische Darstellung der Assets oder auf die Interaktionsmöglichkeiten, die sie in einem Raum haben. Somit beurteilten viele das dritte Level (Gefängnis) als das Level mit der höchsten Immersion, da sie die Lichtsetzung, die Assets und die Umgebung als sehr realistisch empfunden haben. Auf Probleme hinsichtlich der Interaktion sind sie jedoch nicht eingegangen. Manche Probanden verwechselten Immersion mit Interaktion und fühlten sich in dem ersten Level aufgrund der hohen Anzahl an Interaktionsmöglichkeiten am wohlsten. Interessant war zudem die Beobachtung, dass das vierte Level, wider zu erwarten, ebenso immersiver empfunden wurde. Diese Probanden brachten die Immersion mit der Begeisterung während des Lösen eines Rätsels in Verbindung.

Zusammengefasst kann die Auswertung hinsichtlich der Immersion kein plausibles Ergebnis liefern, da die Definition von Immersion bei den Probanden nicht einheitlich war. Für zuverlässige Ergebnisse muss zukünftig zuerst der Begriff „Immersion“ genauer definiert werden. Es muss dafür garantiert werden, dass alle User mit der gleichen Vorstellung dieses Begriffs die Frage beantworten. Es gäbe ebenfalls die Möglichkeit, die Frage nach der Immersion genauer auf bestimmte Elemente zu beziehen. Man könnte nach dem Gefühl der Präsenz in einer Szene bzw. nach der *subjektiv realistischsten Umgebung* fragen.

Manche haben ständig nachgefragt, ob sie in der physischen Welt bald gegen eine Wand laufen und manche haben diese Welt vollkommen vergessen. Ein User wollte am Ende des Tests die Controller auf den virtuellen Couchtisch legen.

Die Anstrengung während des Tests war ebenfalls interessant zu beobachten. Viele Probanden haben während des Tests geschwitzt und sie waren aufgeregt, aufmerksam und konzentriert. Anhand ihrer Kopfbewegungen und längerem Beobachten eines Objekts konnte man erkennen, dass sie manche Intentionen zum Teil begreifen. Aus unterschiedlichen Gründen, wie Unsicherheit, Verwirrung oder Diskrepanz zur Wirklichkeit, trauten sie sich nicht, eine Aktion durchzuführen. Dies war vor allem im vierten Level, in Bezug auf die Wanduhr und das bewegende Licht zu sehen. Manche User haben mehrmals zu der Uhr und daraufhin zu dem Licht hin und her geschaut. Es war ersichtlich, dass sie versuchten eine Verbindung herzustellen. Viele Probanden entschieden sich jedoch dagegen und verzichteten darauf, das Risiko einzugehen.

Es sei zu erwähnen, dass sich die Probanden insgesamt kaum auf das spielerische Experimentieren eingelassen haben. Es gab kaum Probanden, die den Test als ein *Spiel in einer virtuellen Welt, in der nichts Gefährliches passieren kann*, wahrgenommen haben.

6.8.2 Küche

Marc Gillies weist in seinem Coursera Online Course *Interaction in VR* auf Signale hin, welche ein Objekt aussendet. Jeder kennt diese Signale, wenn man beispielsweise versucht eine verschlossene Tür zu öffnen. Erst nach dem Versuch liest man die Information über der Türklinke, dass jene Tür nicht geöffnet werden kann. Dies liegt daran, dass Signale von Objekten stärker und aussagekräftiger empfangen werden, als ihre Farbe oder andere Eigenschaften. (Gillies & Xueni Pan, 2018, S. 6)

Zur Bestätigung dieser Aussage und um das Gesetz der Gestalttheorie zu überprüfen, wurden manche Schränke in der Küche mit einem roten Griff

versehen. Mit diesen Schränken konnte nicht interagiert werden. Mit den vorliegenden Ergebnissen kann die These von Marc Gillies bestätigt werden: Die Probanden haben einige Versuche benötigt, um zu begreifen, dass sie mit den rot markierten Schränken nicht interagieren können. Der Griff signalisierte somit eine offensichtliche Interaktion, die sich gegenüber der roten Farbe durchsetzen konnte.

Allerdings muss an dieser Stelle auf eine verständliche Kritik eingegangen werden. Rot ist eine Signalfarbe. („Warum gilt Rot als Signalfarbe?“, o. J.) Bewusst oder unbewusst beeinflusst diese Farbe die Reaktion der Menschen. Eine virtuelle Welt ist dabei keine Ausnahme. Es stellt sich die Frage, ob die Reaktion der Probanden die Gleiche gewesen wäre, wenn man sich für eine andere Farbe entschieden hätte. Daher war die Wahl der Farbe Rot in diesem Fall nicht gut.

Im Bezug auf die Gesetze der Gestalttheorie kann man jedoch von einem Erfolg sprechen. Es war bei allen Probanden ersichtlich, dass sie ohne Nachdenken und zögern gehandelt haben. Wenn sie die Besteckschublade geöffnet haben, legten sie die Messer neben die Messer, die Löffel neben die Löffel usw. Gleiches galt für die Tassen und Teller. Jeder stellte die Milchpackung in den Kühlschrank und suchte eine Schüssel mit Äpfeln, um auf der Arbeitsplatte verteilte Äpfel hinein zu legen. Dies zeigt, dass die Wahrnehmung und Intention der Menschen, was Ordnung und Zusammengehörigkeit in einer realen Welt angeht, gänzlich in einer virtuellen Umgebung wiedergefunden werden kann.

6.8.3 Schlafzimmer

Die Feedbacks der Experten und die Beobachtung der Probanden konzentrieren sich auf zwei unterschiedliche Bereiche. In diesem Level gab es sowohl technisch als auch inhaltlich kritische Stellen.

Es wurde bereits erwähnt, dass die Gegenstände zu Beginn einer Interaktion reagieren, indem sie 1 cm vom Boden angehoben werden. Diese Funktion wurde eingebaut, um das Problem der Kollision mit dem Boden zu lösen und wurde ebenfalls als ein angenehmes Feedback auf die Aktion des Users verstanden. 1 cm ist jedoch ein kaum wahrnehmbarer Wert. Dieser hätte vielleicht verdoppelt bzw. verdreifacht sein sollen. Dieses Problem lässt sich in Zukunft lösen, indem man die Zahlen an dieser Stelle des Codes überarbeitet.

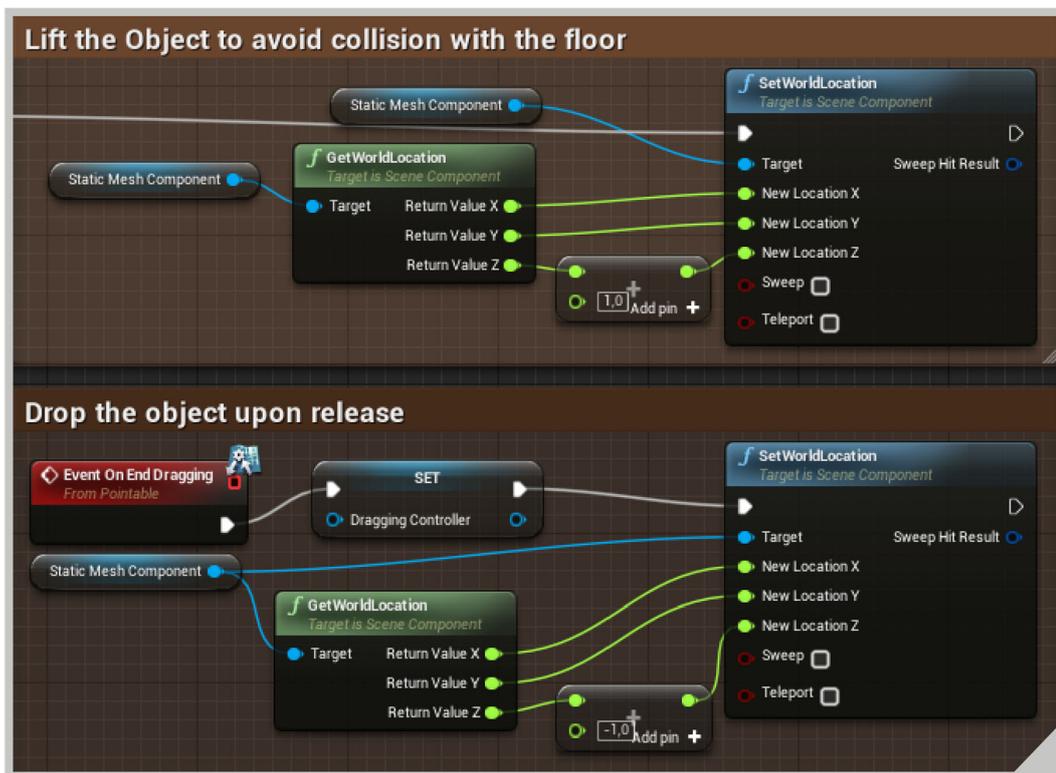


Abbildung 116: Blueprint zum temporären Anheben des gewählten Gegenstands (Screen Shot aus Unreal Engine 4)

Das größte Problem in diesem Level, welches als ein Fehler der Entwicklung angesehen werden kann, ist die Art der Interaktion. Das Interagieren mit Hilfe eines Zeigestabs gilt als Umschulung. Es handelt sich um eine neue Form der Interaktion und Änderungen solcher Art sind innerhalb einer Anwendung nicht ratsam. Dies stellte sich ebenfalls als eine große Herausforderung zu Beginn dieses Levels heraus. Die Probanden sind mit der gleichen Erwartungshaltung, die sie aus dem ersten Level gewohnt waren, in das Zweite eingestiegen. Dort war die erste Aktion das Greifen eines Gegenstands, indem sie ihre Hand zu einer Faust gemacht haben. Dies verursachte nun eine Interaktion an anderer Stelle, nämlich am Ende des Zeigestabs, führte zu Verwirrung bzw. enttäuschte die User. Interaktion fand also mit denselben Tasten statt, das hin- und herschieben sowie der Zeigestab fühlten sich, im Vergleich zum Greifen der Objekte, jedoch anders an, weshalb es als eine neue Art der Interaktion gesehen werden kann.

Das zweite Problem war die Platzierung des Bildes, auf welchem das eingerichtete Zimmer mit der gewünschten Wandfarbe abgebildet war. Es hängt an einer Wand, welche vom User aufgrund seiner Position meist nicht angesehen wird. Abgesehen davon ist das Bild klein und unauffällig. Aus diesem Grund wurde es lediglich von zwei Probanden entdeckt und entsprechend wahrgenommen. Die

Position des Users müsste bei Beginn dieses Levels auf eine bestimmte Stelle, genau vor diesem Bild, definiert werden, um dieses Problem in Zukunft zu vermeiden.

Axel Dietrich (Vrisch) schlug vor, dass die Aufgabe anders gestellt sein müsste: „Richte das Zimmer nach der Vorlage ein“. Auf eine ähnliche Art wurde die Aufgabe im vierten Level gestellt und dadurch kann sichergestellt werden, dass der User auf dieses Bild auf jeden Fall aufmerksam wird. Zusätzlich wurde das Wort *Einrichten* kritisiert, denn die Aufgabe des Levels lautet: *Richte bitte das Schlafzimmer ein*. Es drängt sich die Frage auf, was der User unter *Einrichten* versteht, denn *Einrichten* beinhaltet keine Änderung der Wandfarbe oder des Bodenmaterials. Die Probanden verstanden darunter lediglich das Bewegen der Möbel in einem Raum. Somit kann behauptet werden, dass die Aufgabenstellung in diesem Level das Ziel nicht gänzlich getroffen hat.

Bei der Entwicklung wurde entschieden, dass die Möbel wie ein Puzzle im Raum verteilt werden, weshalb sie nicht gedreht und skaliert werden konnten. Diese Absicht hat keiner der User wahrgenommen. Jeder versuchte mehrmals die Möbel zu drehen, was in den meisten Fällen zu Frustration führte. Allerdings stellte auch niemand die Frage, warum die Möglichkeit der Drehung nicht eingebaut wurde. Zusätzlich muss erwähnt werden, dass die Idee der Gewichtssimulation durch das Schieben und Ziehen mit einem Zeigestab keine erfolgreiche Idee darstellte.

6.8.4 Gefängnis

Die Probanden empfanden dieses Level in Bezug auf seine Aufgabenstellung als das Schwierigste. Sie waren jedoch der Meinung, dass dies beim vierten Level der Fall hätte sein sollen.

Obwohl die Art der Aufgabenstellung in allen Levels gleich war, wurde nicht darauf geachtet, dass die Aufgaben der dritten und vierten Level schrittweise gelöst werden mussten. Es ist für den User nicht möglich, die Decke hochzuschieben, bevor er die Wand zur Seite geschoben hat. In der Küche als auch im Schlafzimmer konnte er seiner Intuition folgen und direkt von seiner Ausgangsposition aus beginnen. Mit dieser Einstellung sind die User in das dritte Level eingestiegen und der objektlose Raum führte zu Verwirrung.

Mit den gewonnenen Erfahrungen dieses Levels kann man mit Sicherheit sagen, dass die Position eines Objekts nichts über die Affordanz aussagt. Eine etwas tieferstehende Zimmerdecke signalisiert keine Bewegungsmöglichkeit nach oben. Gleiches gilt für eine Wand, welche durch ihre Position einen Spalt verursacht.

Dieser Spalt zieht zwar die Aufmerksamkeit auf sich, führt jedoch nicht zu der Einsicht, dass man die entsprechende Wand weiter zur Seite schieben kann.

Die meisten User bückten sich daher, um unter dieser Decke zu stehen. Das Hochschieben ist meist zufällig passiert bzw. geschah nach einem Hinweis von außen. Nachdem der User nun die Wand zur Seite geschoben hatte, nahm er eine Position ein, durch die er mit seinem Kopf in der Decke stand. Dabei wurde für manche das Gefühl der Immersion zerstört. Aus physikalischer Sicht war es nicht möglich, dieses Verhalten zu vermeiden. Es müsste jedoch eine Idee entwickelt werden, wie man diesen Schritt in der Anwendung zukünftig verhindern kann.

Als nächstes muss das Problem in Bezug auf das schiebbare Fenster angesprochen werden. Dieses Fenster ist das Einzige ohne Gitter und ist ebenfalls etwas größer als alle Anderen der gesamten Umgebung. Diese Unterschiede sind jedoch keinem User aufgefallen. Im Gespräch mit Probanden und Experten wurde diese Aufgabe generell als die schwierigste Aufgabe empfunden. Viele haben erwähnt, dass man nicht auf die Idee kommt, besagtes Fenster hinunterzuziehen. Von 25 Probanden kamen drei ohne externe Hinweise bzw. Nachfragen auf die Lösung. Der Erste versuchte sich an dem Fenster festzuhalten um nach oben zu klettern. Unabsichtlich zog er dadurch das Fenster nach unten. Die anderen Zwei sind ebenfalls zufällig auf diese Lösung gekommen.

Marcel Nürnberg von SQUAREBYTES Interactive Media erwähnte einen auffälligen Punkt: Die verwendete Unreal Engine behandelt dynamische Objekte anders als statische, da beide Arten das Licht auf unterschiedliche Weise empfangen. Sie sind in den meisten Fällen heller oder dunkler. Dieses Problem wird zumeist mit der Texturfarbe gelöst. Nürnberg konnte bei dem Fenster als auch bei der Wand einen Farbunterschied erkennen. Er konnte trotz dieses erkennbaren Unterschieds die Funktionalität des Fensters nicht erkennen.

Die Interaktion mit dem Fenster begeisterte alle Benutzer, nachdem ihnen erklärt wurde, welche Möglichkeiten sich dahinter verbargen. Allerdings verwirrte und frustrierte es sie auch, da es offensichtlich als sehr schwierige Aufgabe empfunden wurde.

6.8.5 Garage

Die Reaktion der Probanden als auch der Experten verlief anders als gedacht. Es wurde vermutet, dass die User große Schwierigkeiten haben würden, ihre Aufgabe zu verstehen und die Gesamtzeit wurde viel länger eingeschätzt als es real erfolgte. Ebenso wurde beobachtet, dass die Probanden sich nach dem Ausblenden der Aufgabentafel zu dem angedeuteten Tisch bewegten.

Grundsätzlich konnte man ihre Reaktionen in zwei Kategorien unterteilen:

Wenn der User sich zu der Wanduhr bewegt hat und umgehend auf die Schnelle Bewegung der Zeiger aufmerksam wurde, versuchte er die Zeiger anzuhalten.

Wenn der User der Wanduhr allerdings keine große Aufmerksamkeit schenkte und sich direkt zu dem Tisch bewegte, versuchte er zuerst mit allen anderen Gegenständen zu interagieren. Auf dem Tisch befinden sich eine Karte, ein Foto, ein Radio, Werkzeuge und weitere Gegenstände. Da Interaktion mit den Objekten in dieser Umgebung nicht möglich war, wurde die einzige Interaktion mit der Wanduhr ebenfalls ausgeschlossen.

Doch die Interaktion mit den Gegenständen war nicht ohne Grund abgestellt. Denn kein User sollte die Möglichkeit haben, eine Lampe zu dem Zettel zu bringen bzw. den Zettel in die Hand zu nehmen und zum Licht zu gehen. Die Unmöglichkeit solcher Interaktion führt jedoch dazu, dass der User alle Objekte dieser Umgebung als nicht-interagierbare, statische Objekte wahrnimmt.

Interessanterweise griff jeder zu dem Stundenzeiger. Diese Aktion stoppt dessen Drehung, lässt aber keine weitere Kontrolle zu. Aus der Realität ist jedoch das Einstellen einer analogen Uhr mit Hilfe des Minutenzeigers(!) bekannt.

Wie erwähnt, bewegten sich das Licht als auch die Uhr 40 Mal schneller als in Echtzeit, wodurch beide Komponenten Aufmerksamkeit erzeugten. Nach dem Anhalten der Uhr musste die Bewegung des Lichts allerdings schneller als die Bewegung des Zeigers stattfinden. Das notwendige, häufige Drehen des Zeigers ermüdete den User bzw. erzeugte Zweifel, da das Ergebnis bzw. die Verbindung zwischen Zeiger und Lichtquelle nicht sofort ersichtlich waren.

Zudem wurde auf einen weiteren Fehler hingewiesen: Die Bewegung des Lichts sage nichts über die Tageszeiten aus, denn sie könnte auch das Vorbeifahren eines Autos oder die Nähe eines Leuchtturms suggerieren. Das bedeutet, es müsse nicht unbedingt der Schluss gezogen werden, dass sich die Tageszeit kontinuierlich ändert.

Erstaunlicherweise wurde dieses Level, entgegen der Erwartung der Entwickler, von vielen Usern als leichter und interessanter empfunden. Viele bezeichneten es als das Immersivste, obwohl fast keine Interaktion möglich war.

7 HOHO

Das folgende Kapitel handelt von der Umsetzung eines VR-Projekts, welches 2016 von der Firma Vrisch als Visualisierung eines Hotels angefertigt wurde. Da die vorliegende Arbeit mit Fokus auf Interaktion im Bereich der Architektur geschrieben wurde, konzentriert sich die Analyse des besagten Projekts HOHO (HolzHotel) mehr auf die Umsetzung der interaktiven als auf die konzeptionellen Inhalte.

Das Gebäude wurde in *Maya* erstellt und in *Unreal Engine* importiert. Bei den Assets wurde auf bereits bestehende Assets aus Onlinedatenbanken zurückgegriffen. Sie wurden passend zu der Innenarchitektur in den Räumlichkeiten platziert. Die Visualisierung von HOHO besteht aus drei unterschiedlichen Bereichen des Hotels:

Der Außenbereich



Abbildung 117: HOHO Außenbereich, (Screen Shot aus Unreal Engine 4)

7 HOHO

Die Lobby



Abbildung 118: HOHO Lobby, (Screen Shot aus Unreal Engine 4)

Das Office



Abbildung 119: HOHO Office, (Screen Shot aus Unreal Engine 4)

Teleportation

In diesem Hotel bewegt sich der User in allen Bereichen mittels Teleportation. Auffällige an diesem Projekt ist der Teleportationsbereich, womit der Bereich gemeint ist, in welchem sich der Spieler bewegen darf. Geöffnet und getestet wurde dieses Projekt mit HTC Vive. Bei dieser Plattform findet die Teleportation sowohl mit dem Touchpad als auch mittels Trigger statt. Mit Drücken des Touchpads oder Triggers aktiviert der User ebenfalls einen gepunkteten, leicht leuchtenden Bereich auf dem Boden. Dieser Bereich macht die Fläche sichtbar, auf welcher er sich teleportieren darf.

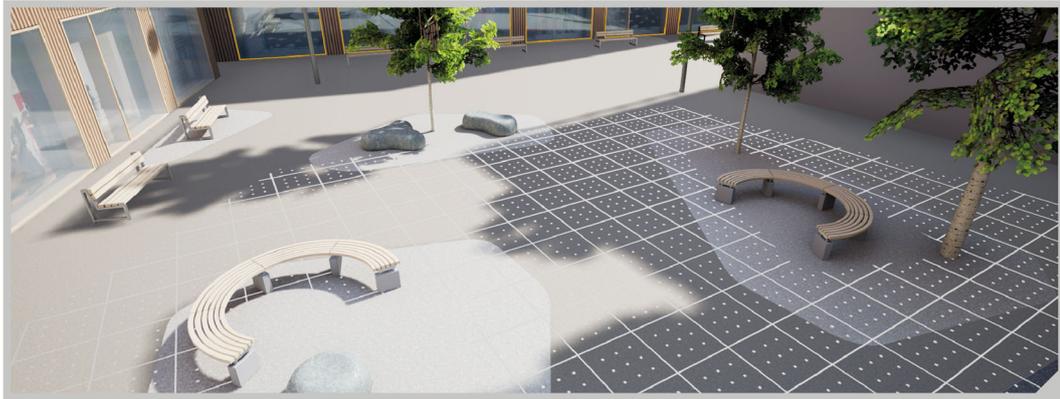


Abbildung 120: HOHO, Darstellung des sichtbaren Teleportationsbereichs, (Screen Shot aus Unreal Engine 4)

Das ist ein Feedback von der Anwendung, welche direkt auf die Aktion des Users reagiert. Die Visualisierung der Teleportation findet mit einer Parabel, welche an einem Kreis auf dem Boden endet, statt. Deutet der User auf eine Stelle, welche Teleportation verbietet, zeigt die Anwendung keine Reaktion.

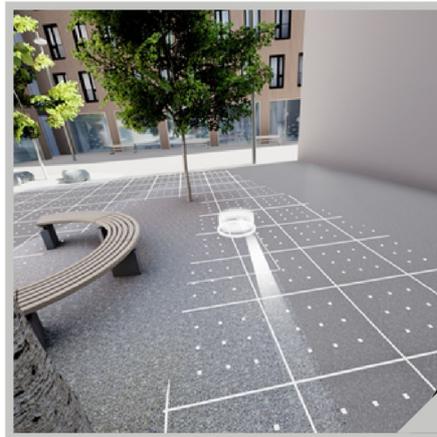


Abbildung 121: Visualisierung des Teleportationsziels, (Screen Shot aus Unreal Engine 4)

Die Bewegung des Controllers auf der Z-Achse (nach oben und unten) wird in diesem Fall auf eine Bewegung der X- und Y-Achse gemappt.

Bewegen entlang der Z-Achse ist für den User nicht möglich und bedeutet, dass der User immer auf dem vordefinierten Boden bleibt. Somit verliert die Z-Achse an Funktionalität. Sie sorgt jedoch dafür, dass das Teleportationsziel mittels einer Parabel mit der jetzigen Position verbunden wird. Mit der Bewegung seines Controllers entlang dieser Achse kann der Spieler das Teleportationsziel näher bzw. weiter positionieren.

Sobald der Spieler das Teleportationsziel auf eine, für ihn nicht vorgesehene, Stelle bringt, blendet der gepunktete Teleportationsbereich aus. Es ist somit für einen User leicht und schnell verständlich, wie weit und wohin er sich teleportieren darf. Beobachtet man die letzten zwei Bilder, bemerkt man, dass der Bereich, in welchem sich der Spieler fortbewegen darf, auffällig kleiner ist als die gesamte Fläche. Der Spieler könnte zuerst das Gefühl bekommen, dass er von Seiten der Entwickler etwas eingeschränkt wurde. Diese Constraints (Einschränkungen) haben jedoch in Bezug auf die Teleportation mehrere Vorteile:

- Die Entwickler können sich dadurch auf den Bereich konzentrieren der von ihnen visualisiert werden muss. D. h. sie müssen sich um den Teil der Umgebung kümmern, welche von dem User tatsächlich gesehen wird. Die Assets müssen entsprechend vorbereitet und ein Bereich für die Berechnung des Lichtes kann somit vordefiniert werden.
- Es verschafft dem User einen guten Überblick bzgl. der Größe und des zum Hotel gehörigen Bereichs.
- Es ist eine gute Möglichkeit die Fehlerwahrscheinlichkeiten aus Seiten des Users zu verringern.

Eines der Probleme, welches man in der Testphase mit den Probanden feststellen konnte, waren die angelegten Teleportationsziele nahe der Wände des Gefängnisses. Denn ein unerfahrener User landet dadurch entweder zu nah an einer Wand oder versucht durch diese Wände zu gehen. Dieser Umstand zerstört das Gefühl der Immersion und verfehlt somit das Ziel des Tests. Solche Probleme können im HolzHotel nicht vorkommen.

Vergleicht man beide Projekte miteinander, begreift man sofort, dass der Einsatz von das Norman's Prinzip, *die Constraints* hier einen auffällig erfolgreichen Einsatz gefunden hat. Die Einschränkungen für einen Spieler zeugen nicht unbedingt von Nachteilen, denn sie haben nachvollziehbare Gründe, die in diesem Fall, sowohl dem User als auch dem Entwickler, entgegenkommen.

Wie zu Beginn angedeutet, besteht HOHO aus drei begehbaren Räumlichkeiten, die sich jedoch auf unterschiedlichen Ebenen befinden. Die Treppen des Außenbereichs sind nicht für Teleportationen vorgesehen. Dies bedeutet, dass dem User eine andere Möglichkeit gegeben werden muss, diese Räumlichkeit betreten zu können.

Diese Möglichkeit wurde mit einem Switch zwischen den Räumen erreicht. Links von der Player Camera werden drei Icons auf *sichtbar* gesetzt sobald er seine Teleportationstaste betätigt.

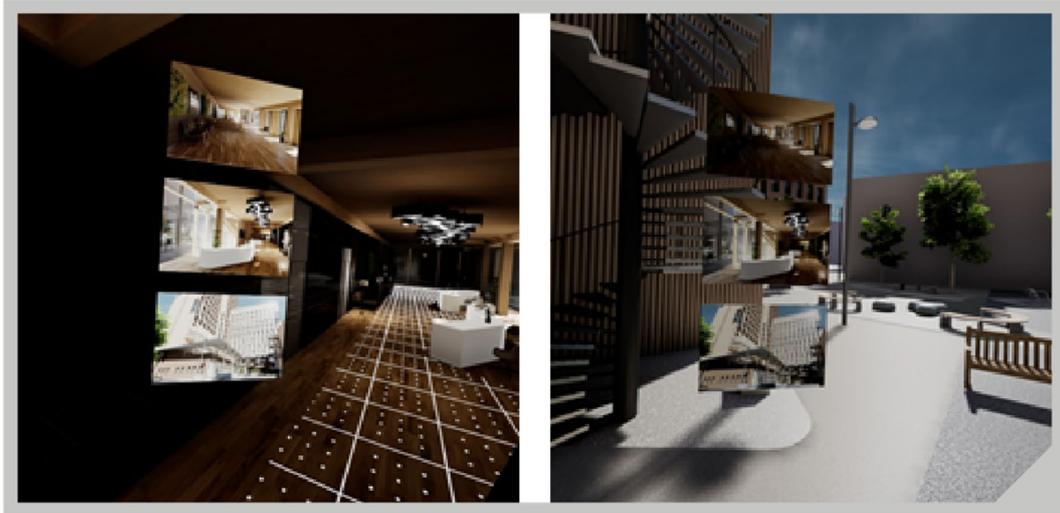


Abbildung 122: HOHO Umgebungswechsler, (Screen Shot aus Unreal Engine 4)

Das Auswählen der Icons findet mit derselben Taste statt mit welcher man sich fortbewegen kann. Dies bedeutet, dass sie jedes Mal beim Betätigen der Taste aufblenden. Die Position der Icons weist auf ein weiteres Prinzip von Norman hin: dass die Feedbacks nicht aufdringlich sind. Sie poppen auf der linken Seite des Sichtfeldes auf, bleiben jedoch auf derselben Stelle. Sie verändern ihre Position nicht, während der User seinen Kopf in eine andere Richtung dreht. Der User erhält mit dem Zielen auf diese Icons eine Vergrößerung der Bilder und ebenso ein Feedback, wenn er einen Raum auswählt.

Eine Frage, die sich ein User stellen könnte, wäre in Bezug auf den jetzigen Raum, in welchem er sich befindet. Abbildung 120 zeigt, dass sich der User an dieser Stelle im Außenbereich des Hotels befindet. Diese Information ist jedoch bei diesen drei Icons nicht visualisiert. Dieser Bereich, bzw. das Icon, welches für diesen Bereich zuständig ist, könnte beispielsweise eingerahmt sein um die jetzige Position zu verdeutlichen.

Als Kritik kann dies aber nicht angesehen werden. Die Icons sind jeweils mit einem Bild von der jeweiligen Umgebung versehen. Letztendlich handelt es sich dabei um drei Räumlichkeiten. Bei dieser geringen Anzahl kann man auf derartiges Feedback verzichten.

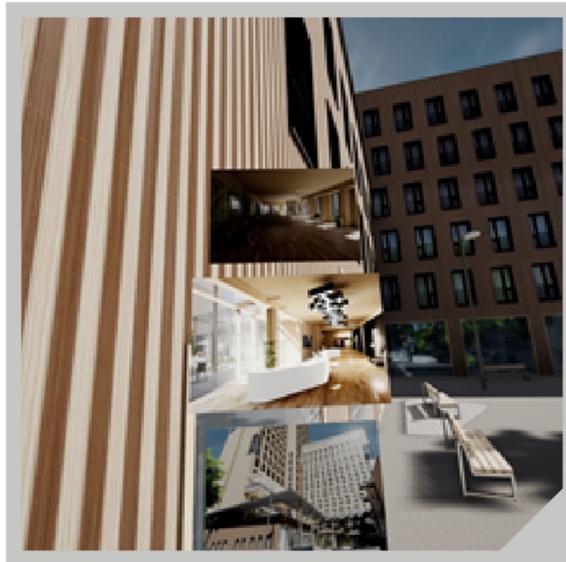


Abbildung 123: Feedback durch Vergrößerung bei der Auswahl, (Screen Shot aus Unreal Engine 4)

HOHO hat es geschafft, einen Rundgang mit nur einer einzigen Taste zu ermöglichen und dieser findet auf einer übersichtlichen Art und Weise statt. Die gesamte Visualisierung ist selbsterklärend. Der User kann seine Brille aufsetzen und ist auf keine Hilfe angewiesen um sich in dem Hotel bewegen zu können. Es besteht keine Möglichkeit der Interaktion mit Gegenständen wie Möbeln oder Äpfeln, welche sich auf dem Tisch des Office befinden. Es ist jedoch offensichtlich, dass Sinn und Zweck dieses Projekts nicht die Interaktion mit Objekten ist. Dieses Projekt konzentriert sich auf den Rundgang im Hotel.

8 Fazit

Anhand der hier dargestellten Methoden, Probleme und Lösungen ist ersichtlich, welche unterschiedlichen Aspekte der menschlichen Wahrnehmung bei der Entwicklung einer interaktiven VR-Anwendung beachtet werden müssen.

Nach den Recherchen und basierend auf den Antworten von Experten, Marcel Nürnberg, Axel Dietrich und Christian Friesenegger kann man hinsichtlich der ersten Forschungsfrage behaupten, dass sich die Interaktionsmethoden, welche sich nach der Realität gerichtet haben, am besten bewährt haben. Jedoch spielte das Ziel der Anwendung hinsichtlich der Freiheit, welche dem User gegeben wurde, eine wichtige Rolle. Zudem kann diese Aussage aus eigener Erfahrung aus dem Usertest bestätigt werden. Wird das Verlassen einer Umgebung (wie in dem dritten Level) für den User als Ziel gesetzt, ist es nicht zielführend und unbedingt immersiv, wenn alle Räumlichkeiten begehbar sind. Einschränkungen sind sogar hilfreich und dienen der Übersichtlichkeit. Es ist also lohnenswert sich über die psychologischen und physikalischen Aspekte zu informieren, sie gezielt einzusetzen und dem User bewusst durch einen Raum zu führen.

Die **Hypothese** dieser Arbeit besagt, dass es besser wäre, sich bei der Interaktion mit Umgebungen und dem Verhalten der Objekte nach der Wirklichkeit zu richten. Diese These kann nur bedingt bestätigt werden. Denn die Kommunikation mit der Umgebung basiert auf mentalen Modellen, welche der User im Laufe seines Lebens in Bezug auf seine Umgebung und den Signalen, welche er von den Objekten empfangen hat, bildet. Wenn es sich bei einer virtuellen Umgebung um eine realistische Wiedergabe der Welt handelt, kann man die These bestätigen. Sie beginnt jedoch an Plausibilität zu verlieren, je unrealistischer die Umgebungen werden.

Bezieht man sich auf das bereits bekannte Spektrum zwischen realistisch und unrealistisch als auch die Ergebnisse des Usertests, ist ersichtlich, dass es dem User bereits ohne Hilfsmittel und einem Hinweis im zweiten Level schwergefallen ist, die Möglichkeit zu entdecken, mit einem Klick die Wandfarbe ändern zu können. Diese Hilfsmittel, welche dem User aus der Realität offensichtlich nicht bekannt sind, bringen enorme Vorteile und verleihen der Anwendung Benutzerfreundlichkeit bzw. Attraktion. Diese Aussage kann mit einem Verweis auf die Wanduhr des vierten Levels erneut bestätigt werden. Versteht man jene Wanduhr als ein Interface, welches den User mit dem Licht in Kontakt bringt, bemerkt man die Notwendigkeit dieser Komponente in einer virtuellen Welt. Diese

8 Fazit

Erkenntnis dient als eine Überleitung um die zweite Forschungsfrage zu beantworten: Wie weist man den Benutzer darauf hin, welche Objekte auf eine Interaktion reagieren und welche nicht?

Richten sich die Objekte und ihre Affordanz auf die Wirklichkeit, müssen sie sich entsprechend verhalten und es ist davon auszugehen, dass sie in einer virtuellen Welt ident mit der realen Welt verstanden werden. Distanziert sich die Interaktion von der Wirklichkeit, ist auf die Kenntnisse aus der realen Welt kein Verlass mehr. Hinweise, die einen Unterschied zwischen interagierbaren und nicht-interagierbaren Objekten darstellen (oder Hilfsmittel welche diese Interaktion erklären), sind nicht nur wichtig, sondern auch notwendig. Auf eine bestbewährte Methode konnte jedoch nicht hingewiesen werden.

Dies bietet eine gute Grundlage für eine weiterführende Forschungsfrage in Bezug auf das Entwerfen von GUIs, welche in einer VR-Anwendung eingesetzt werden. Dieses Thema wurde im Laufe dieser Arbeit nur kurz behandelt. Das liegt daran, dass die GUIs als Hilfsmittel in den Vordergrund rücken, sodass die Affordanz eines Objekts allein und ohne sie weniger ins Auge fällt. Der Usertest hat jedoch gezeigt, dass die GUIs für eine benutzerfreundliche, immersive VR-Anwendung essentiell sind.

“If a user can interact using natural skills, then the application can take advantage of the fact that the user already has a great deal of knowledge about the world.”
(LaViola, Bowman, Kruijff, Poupyrev, & McMahan, 2017, S. 4)

Am Ende dieser Arbeit kann diese Aussage nur bedingt bestätigt werden. Sie stimmt nur, wenn Virtual Reality die Wiedergabe der Realität beabsichtigt. Dies ist zwar möglich, oft und kontextabhängig sogar erwünscht, sie ist jedoch nicht das Ziel von Virtual Reality.

„Some consider real reality the gold standard of what we are trying to achieve with VR whereas others consider reality a goal to surpass—for if we can only get to the point of matching reality, then what is the point?“ (Jerald, 2015b, S. 85)

Literaturverzeichnis

- 3D Interaction Design in Virtual Reality. (2018, Mai 3). Abgerufen 3. Mai 2018, von <https://www.coursera.org/learn/3d-interaction-design-virtual-reality>
- A quick guide to Degrees of Freedom in Virtual Reality. (2018, Februar 12). Abgerufen 11. August 2018, von <https://kei-studios.wtf/quick-guide-degrees-of-freedom-virtual-reality-vr/>
- Akenine-Möller, T., Haines, E., & Hoffman, N. (2008). Real-time Rendering (3. Auflage). Wellesley, Mass: Taylor & Francis Ltd.
- All you need to know about SteamVR Tracking 2.0... will it be the foundation of Vive 2? (2017, Juni 7). Abgerufen 18. August 2018, von <https://skarredghost.com/2017/06/07/need-know-steamvr-tracking-2-0-will-foundation-vive-2/>
- Allgemeine, A. (2017). Demenz: Können VR-Brillen den Krankheitsverlauf verlangsamen? Abgerufen 10. August 2018, von <https://www.augsburger-allgemeine.de/wissenschaft/Demenz-Koennen-VR-Brillen-den-Krankheitsverlauf-verlangsamen-id42666111.html>
- Archinteriors for UE vol. 3 - Evermotion. (o. J.). Abgerufen 5. August 2018, von https://evermotion.org/shop/show_product/archinteriors-for-ue-vol-3/13665
- Archinteriors for UE vol. 4 - Evermotion. (o. J.). Abgerufen 5. August 2018, von https://evermotion.org/shop/show_product/archinteriors-for-ue-vol-4/14406
- Architectural Visualization. (2018, Mai 27). Abgerufen 27. Mai 2018, von <https://www.unrealengine.com/marketplace/content-cat/assets/archvis>
- Archives | MicroFilmmaker Magazine. (o. J.). Abgerufen 28. August 2018, von <http://www.microfilmmaker.com/2012/05/>
- Azmandian, M., Grechkin, T., & Rosenberg, E. S. (2017). An evaluation of strategies for two-user redirected walking in shared physical spaces (S. 91–98). IEEE. <https://doi.org/10.1109/VR.2017.7892235>
- Bastian, M. (2016, August 1). Virtual Reality: Wissenschaftler zeigen Redirected-Walking-Trick. Abgerufen 24. Juni 2018, von <https://vrodo.de/virtual-reality-wissenschaftler-zeigen-redirected-walking-trick/>
- Berner, W. (2015). Finalität: Menschliches Handeln ist immer zielgerichtet. Abgerufen 16. Juli 2018, von <https://www.umsetzungsberatung.de/psychologie/finalitaet.php>
- Biglari, E., & Feng, Y. (2014). Interactive Virtual-Reality Driven Learning Framework for Engineering and Science Education, 11.

- Biocca, S. (2017). Die verschiedenen Einsatzgebiete für Virtual Reality (VR). Abgerufen 8. August 2018, von <https://www.androidpit.de/die-verschiedenen-einsatzgebiete-fuer-virtual-reality>
- Bowman, D. A., Kruijff, E., LaViola, J. J., & Poupyrev, I. (o. J.). „Navigation „ Wayfinding „ Travel, 19.
- Bragull, J. (o. J.). Gestalttheorie.pdf.
- Braining, A. (2018). Immersionserweiterung durch akustisches, visuelles und haptisches Feedback, 61.
- Brown, M. (2017, Januar 28). How to customize the HTC Vive's Chaperone with SteamVR. Abgerufen 24. Juni 2018, von <https://www.vrheads.com/how-customize-htc-vives-chaperone-steamvr>
- Buckley, S. (o. J.). This Is How Valve's Amazing Lighthouse Tracking Technology Works. Abgerufen 18. August 2018, von <https://gizmodo.com/this-is-how-valve-s-amazing-lighthouse-tracking-technol-1705356768>
- Budget Cuts Demo bei Steam. (2016). Abgerufen 26. August 2018, von https://store.steampowered.com/app/459860/Budget_Cuts_Demo/
- Bumble. (o. J.). Locomotion in VR: Overview of different locomotion methods on HTC Vive. Abgerufen von <https://www.youtube.com/watch?v=p0YxzgQG2-E>
- Burmester, D. M. (2017). Was bedeutet intuitive Nutzung? | Einfach nutzbare UIs gestalten. Abgerufen 3. August 2018, von <https://www.uid.com/de/aktuelles/intuitive-nutzung-ohne-anleitung-zu-positiven-nutzererlebnissen>
- Carbotte, K. (2016). VR Locomotion Is A Problem That Has Many Half-Solutions. Abgerufen 8. Juni 2018, von <https://www.tomshardware.com/news/vr-locomotion-developer-solution-roundup,33108.html>
- Carbotte, K. (2018, März 10). Do the Locomotion: The 19 Ways You Walk and Run in VR Games. Abgerufen 7. August 2018, von <https://www.tomshardware.com/picturestory/807-virtual-reality-games-locomotion-methods.html>
- Charlie, C. (1925). The Gold Rush | Silent Film Festival. Abgerufen 8. September 2018, von <http://silentfilm.org/archive/the-gold-rush-1925>
- Cherry, K. (2018). Change Blindness Is How We Miss the Big Changes Around Us. Abgerufen 8. August 2018, von <https://www.verywellmind.com/what-is-change-blindness-2795010>
- christian friesenegger.wav. (o. J.).
- Conceptual Integrity. (o. J.).
- Cosky, E. (2018, Mai 27). Object Interaction Part 1: Common Interaction Patterns. Abgerufen 27. Mai 2018, von <https://developer.oculus.com/blog/object-interaction-part-1-common-interaction-patterns/>

- daviddewhirst | VR Movement System. (o. J.). Abgerufen 26. August 2018, von <https://www.daviddewhirst.co.uk/vr-locomotion>
- Dawn of the New Everything, A Journey Through Virtual Reality by Jaron Lanier. (2018, Mai 6). Abgerufen 6. Mai 2018, von <https://www.penguin.co.uk/books/1109194/dawn-of-the-new-everything/>
- Degrees of freedom | Google VR. (o. J.). Abgerufen 11. August 2018, von <https://developers.google.com/vr/discover/degrees-of-freedom>
- DeMichele, T. (2016). The Sensorama Was the First VR Device - Fact or Myth? Abgerufen 19. August 2018, von <http://factmyth.com/factoids/the-sensorama-was-the-first-vr-device/>
- Developer, 11 Incredible VR Challenges for Developers-AT&T. (2018, Mai 5). 11 Incredible VR Challenges for Developers. Abgerufen 5. Mai 2018, von <https://developer.att.com/blog/virtual-reality-developer-challenges>
- DevTon Studio. (2017). DevTon Studio Prison. Abgerufen 5. August 2018, von <https://www.unrealengine.com/marketplace/devton-abandoned-prison>
- Dietrich, A. (2018). Interview mit Axel Dietrich, Firma Vrisch.
- Dix, A., Finlay, J., Abowd, G. D., & Beale, R. (2003). Human Computer Interaction (3. A). Harlow, England ; New York: Prentice Hall.
- Dörner, R., & Steinicke, F. (2013). Wahrnehmungsaspekte von VR. In R. Dörner, W. Broll, P. Grimm, & B. Jung (Hrsg.), Virtual und Augmented Reality (VR / AR) (S. 33–63). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-28903-3_2
- Dr. Andrew Rosen. (2016, Dezember 17). Virtual Reality Therapy for Phobias. Abgerufen 8. August 2018, von <http://centerforanxietydisorders.com/virtual-reality-therapy-for-phobias/>
- Duden. (o. J.). Duden | vir-tu-ell | Rechtschreibung, Bedeutung, Definition, Synonyme, Herkunft. Abgerufen 19. August 2018, von <https://www.duden.de/rechtschreibung/virtuell>
- Duden | Hap-tik | Bedeutung, Definition, Herkunft. (o. J.). Abgerufen 11. August 2018, von <https://www.duden.de/rechtschreibung/Haptik>
- Eichenberg, C., & Wolters, C. (2012, August 10). Virtuelle Realitäten in der Behandlung psychischer Störungen: Eintauchen ins virtuelle System. Abgerufen 8. August 2018, von <https://www.aerzteblatt.de/archiv/128395/Virtuelle-Realitaeten-in-der-Behandlung-psychischer-Stoerungen-Eintauchen-ins-virtuelle-System>
- Empathy and virtual reality: interconnection and impact on society. (2017, Juni 13). Abgerufen 6. Mai 2018, von <https://teslasuit.io/blog/virtual-reality/empathy-virtual-reality>
- Entertainment und Technik Magazin. (2015). Immersion und Präsenz. Abgerufen 16. August 2018, von <http://www.onlyvr.de/virtual-reality/immersion>
- Ergonomie in der virtuellen Realität. (2016, Dezember 14). Abgerufen 2. Mai 2018, von <https://www.zuehlke.com/blog/ergonomie-in-der-virtuellen-realitaet/>

- Ergonomie und Mensch-Maschine Systeme.pdf. (o. J.).
- Eriksson, M. (o. J.-a). Reaching out to grasp in Virtual Reality, 57.
- Eriksson, M. (o. J.-b). Sträck ut och ta tag i virtuell verklighet, 57.
- Evermotion Kitchen. (o. J.). scene 1 AI4 for Unreal Engine Archinteriors - Evermotion. Abgerufen 7. Juli 2018, von https://evermotion.org/shop/show_product/scene-1-ai4-for-unreal-engine-archinteriors/14694
- February 12, C. B. S., & World, 2016 3D. (o. J.). 10 VR tips for Unreal Engine. Abgerufen 7. Juli 2018, von <https://www.creativebloq.com/3d/10-vr-tips-unreal-engine-21619215>
- Feltham. (2015). WizDish Announces Price for ROVR Development Kit. Abgerufen 8. September 2018, von <https://www.vrfocus.com/2015/02/wizdish-announces-price-rovr-development-kit/>
- Force Feedback Definition & Erklärung | Informatik Lexikon. (o. J.). Abgerufen 11. August 2018, von <http://www.informatik-verstehen.de/lexikon/force-feedback/>
- Fourth, B. (2017, März 27). Passive and Interactive Storytelling in VR. Abgerufen 6. Mai 2018, von <https://virtualrealitypop.com/passive-and-interactive-storytelling-in-vr-eea98b2f76c5>
- Friesenegger, C. (2018). Interview mit Christian Friesenegger.
- Game Cooker US Rocks VR. (2016). VR zGame -- New Game First Gameplay Early Access. Abgerufen von <https://www.youtube.com/watch?v=5K1TM9JclAk>
- Garage Workshop by Gabro Media in Environments - UE4 Marketplace. (o. J.). Abgerufen 5. August 2018, von <https://www.unrealengine.com/marketplace/residential-garage>
- Garg, R. (2013). Optical Navigation Systems: The foundation of modern pointing devices. Abgerufen 18. August 2018, von <https://www.edn.com/design/sensors/4419587/Optical-Navigation-Systems--The-foundation-of-modern-pointing-devices>
- Gesetz der guten Gestalt und Webdesign. (o. J.). Abgerufen 12. August 2018, von <https://www.html-seminar.de/webdesign-gesetz-der-guten-gestalt.htm>
- Gestalt Theory of Visual Perception. (o. J.). Abgerufen 8. Juni 2018, von <http://www.users.totalise.co.uk/~kbroom/Lectures/gestalt.htm>
- Gillies, D. M., & Xueni Pan, D. S. (2018, Mai 3). Coursera | Online Courses From Top Universities. Join for Free. Abgerufen 3. Mai 2018, von <https://www.coursera.org/learn/3d-interaction-design-virtual-reality/home/welcome>
- Google Cardboard now available in Google Play in UK, Canada, Germany and France - Android Authority. (2016, Mai). Abgerufen 2. September 2018, von <https://www.androidauthority.com/google-cardboard-available-google-play-uk-canada-germany-france-691926/>

- GPU Profiling. (o. J.). Abgerufen 22. August 2018, von <https://docs.unrealengine.com/en-us/Engine/Performance/GPU>
- Grundlagen_der_Gestaltpsychologie. (o. J.).
- Harding, C. (2013). L is for... London Stereoscopic Company: The home of 100,000 views. Abgerufen 2. September 2018, von <https://blog.scienceandmediamuseum.org.uk/a-z-photography-l-is-for-london-stereoscopic-company/>
- Hills-Duty, R. (2018). Researchers Say VR Systems Can Be Hacked To Change What You See – VRFocus. Abgerufen 8. September 2018, von <https://www.vrfocus.com/2018/04/researchers-say-vr-systems-can-be-hacked-to-change-what-you-see/>
- HoHo Vienna. (2018, Mai 4). Abgerufen 4. Mai 2018, von <https://www.vrisch.at/projects/hoho-vienna-2/>
- Hopp, J. (o. J.). Figurenkonzeption. Abgerufen 27. Juli 2018, von https://www.uni-potsdam.de/u/slavistik/vc/filmanalyse/arb_stud/hopp_stolz/docs/figurenkonzeption.htm
- HTC_Vive_Tracker_Developer_Guidelines_v1.3.pdf. (o. J.). Abgerufen von https://dl.vive.com/Tracker/Guideline/HTC_Vive_Tracker_Developer_Guidelines_v1.3.pdf
- Huang, X. (o. J.). VIRTUAL REALITY/ AUGMENTED REALITY TECHNOLOGY: THE NEXT CHAPTER OF HUMAN-COMPUTER INTERACTION, 59.
- Hubbold, R., Murta, A., West, A., & Howard, T. (1993). Design Issues for Virtual Reality Systems. In Presented at the First Eurographics Workshop on Virtual Environments (S. 224–236). Springer.
- Hubbold, R., Murta, A., West, A., & Howard, T. (1995). Design Issues for Virtual Reality Systems. In M. Göbel (Hrsg.), Virtual Environments '95 (S. 224–236). Vienna: Springer Vienna. https://doi.org/10.1007/978-3-7091-9433-1_19
- Immersionsstrategien in den Computerspielreihen Dragon Age und Mass Effect.pdf. (o. J.).
- Institut für Qualität und Wirtschaftlichkeit im Gesundheitswesen. (2017). Wie funktioniert der Gleichgewichtssinn? - gesundheitsinformation.de. Abgerufen 10. August 2018, von <https://www.gesundheitsinformation.de/wie-funktioniert-der-gleichgewichtssinn.2253.de.html>
- Interaction Design - Beyond Human-Computer Interaction, 4th Edition.epub. (o. J.).
- Interactivity Element. (2018, Mai 6). Abgerufen 6. Mai 2018, von <https://www.vrs.org.uk/virtual-reality/interactivity-element.html>
- Interaktion. (o. J.). Abgerufen 3. August 2018, von <https://www.sign-lang.uni-hamburg.de/projekte/slex/seitendvd/konzepte/152/15214.htm>

- James, P. (2016, Juli 31). New Video Shows Crazy „Infinite“ Redirected Walking in VR. Abgerufen 8. September 2018, von <https://www.roadtovr.com/new-unlimited-corridor-video-shows-mind-bending-redirected-walking/>
- Jan, S. (2009). Warum Menschen immer im Kreis laufen - wissenschaft.de. Abgerufen 12. Juli 2018, von <https://www.wissenschaft.de/umwelt-natur/warum-menschen-immer-im-kreis-laufen/>
- Jerald, J. (2015). The VR Book: Human-Centered Design for Virtual Reality, 685.
- Kaulich, C. (2015). Masterarbeit Immersion_ Interaktion in Virtual Reality Anwendungen.
- Kinästhetik Definition und Erklärung. (o. J.). Abgerufen 11. August 2018, von <https://www.kinaesthetics.at/was-ist-kinaesthetics.cfm>
- Klickkomplizen. (2010, Februar 8). Gesetz der Geschlossenheit. Abgerufen 12. August 2018, von <https://www.klickkomplizen.de/blog/print-design/gesetz-der-geschlossenheit/>
- Kościelnicki, M. (o. J.). nVidia Hardware Documentation, 661.
- Lang, B. (2016, Januar 31). „Budget Cuts“ Locomotion is a Lesson for VR Developers. Abgerufen 8. September 2018, von <https://www.roadtovr.com/hands-on-budget-cuts-inventive-locomotion-is-a-lesson-for-vr-developers/>
- Langbehn, E., Lubos, P., & Steinicke, F. (2018). Evaluation of Locomotion Techniques for Room-Scale VR: Joystick, Teleportation, and Redirected Walking, 9.
- LaViola, J. J., Bowman, D. A., Kruijff, E., Poupyrev, I., & McMahan, R. P. (2017). 3D User Interfaces: Theory and Practice. Boston: Pearson Education.
- Leave The Nest bei Steam. (2016). Abgerufen 22. August 2018, von https://store.steampowered.com/app/471270/Leave_The_Nest/
- Lighthouse von VALVE erklärt – HTC Vive – VR JUMP. (2018, Mai 14). Abgerufen 14. Mai 2018, von <http://vrjump.de/lighthouse-erklart>
- Linowes, J. (2015). Unity virtual reality projects: explore the world of virtual reality by building immersive and fun VR projects using Unity 3D. Birmingham Mumbai: Packt Publishing.
- mentale Modelle.pdf. (o. J.).
- Methode_Gestaltgesetze. (o. J.).
- Milde, T. (2017a). (10) Unreal 4 Lighting Academy - Session 3.1 - YouTube. Abgerufen 26. August 2018, von <https://www.youtube.com/watch?v=plbpHCtmDjo&t=451s>
- Milde, T. (2017b). Unreal 4 Lighting Academy - Session 3.1.
- Mohammady, F. K. (2018, April 23). So revolutioniert Virtual Reality Ausbildung und Studium. Abgerufen 3. August 2018, von <https://www.morgenpost.de/wirtschaft/article214096329/So-revolutioniert-Virtual-Reality-Ausbildung-und-Studium.html>

- Nabiyouni, M. (2015). How Does Interaction Fidelity Influence User Experience in VR Locomotion?, 158.
- Natalie, W. (2011). Why Do Humans Walk in Circles? Abgerufen 8. August 2018, von <https://www.livescience.com/33431-why-humans-walk-circles.html>
- Nescher, T., Ying-Yin Huang, & Kunz, A. (2014). Planning redirection techniques for optimal free walking experience using model predictive control (S. 111–118). IEEE. <https://doi.org/10.1109/3DUI.2014.6798851>
- Neues Verfahren für Redirected Walking. (2018). Abgerufen 12. Juli 2018, von <https://www.virtual-reality-magazin.de/neues-verfahren-fuer-redirected-walking>
- News, von C. S. in, & Zelected. (2018, März 29). Sakkaden für Redirected Walking in VR nutzen. Abgerufen 12. Juli 2018, von <https://www.vrnerds.de/sakkaden-fuer-redirected-walking-in-vr-nutzen/>
- Norman, D. A. (2013). The design of everyday things (Revised and expanded edition). New York, New York: Basic Books.
- Nürnberg, M. (2018). Interview mit Marcel Nürnberg,.
- okreylos. (2016, Mai 25). Lighthouse tracking examined. Abgerufen 18. August 2018, von <http://doc-ok.org/?p=1478>
- Okyeonjunga VR Museum Gear VR Experience - VR Pill. (2018, Mai 6). Abgerufen 6. Mai 2018, von <http://www.vrpill.com/recent-news/okyeonjunga-vr-museum-gear-vr-experience/>
- Owen, D. (o. J.). Colour Temperature Chart. Abgerufen 8. September 2018, von <https://www.mediacollege.com/lighting/colour/colour-temperature.html>
- Paintey bei Steam. (2016). Abgerufen 22. August 2018, von <https://store.steampowered.com/app/445050/Paintey/>
- Paolis, L. T. D. (o. J.). Virtual and Augmented Reality Applications, 23.
- Parisi, T. (o. J.). Learning Virtual Reality, 128.
- Pausch, R., Snoddy, J., Taylor, R., Watson, S., & Haseltine, E. (1996). Disney's Aladdin: first steps toward storytelling in virtual reality (S. 193–203). ACM Press. <https://doi.org/10.1145/237170.237257>
- Perez Ariza, V. Z., & Santís-Chaves, M. (2017). INTERFACES HÁPTICAS: SISTEMAS CINESTÉSICOS Vs. SISTEMAS TÁCTILES. Revista EIA, 13(26). <https://doi.org/10.24050/reia.v13i26.1065>
- Pinker, A. (2018, Januar 26). VR Healthcare: Virtual Reality in der Medizin. Abgerufen 3. August 2018, von <https://medialist.info/2018/01/26/vr-healthcare-virtual-reality-in-der-medizin/>
- Playstation VR: „Ein Headset ist keine Brille und kein Hut“ - Golem.de. (2018, Mai 5). Abgerufen 5. Mai 2018, von <https://www.golem.de/news/playstation-vr-ein-headset-ist-keine-brille-und-kein-hut-1604-120327.html>
- Project DREAMSPACE. (o. J.). Abgerufen 1. Mai 2018, von <https://www.dreamspaceproject.eu/>

- PropriozeptiveWahrnehmung. (o. J.).
- Protagonist | Bedeutung, Funktion und Beispiele. (o. J.-a). Abgerufen 17. Juli 2018, von <https://wortwuchs.net/protagonist/>
- „Pygmalion’s Spectacles,“ Probably the First Comprehensive and Specific Fictional Model for Virtual Reality (1935) : HistoryofInformation.com. (o. J.). Abgerufen 19. August 2018, von <http://www.historyofinformation.com/expanded.php?id=4543>
- Razzaque, S., Swapp, D., Slater, M., Whitton, M. C., & Steed, A. (o. J.). Redirected Walking in Place, 8.
- Redirected_Walking. (o. J.).
- Rekhi, S. (2017). Don Norman’s Principles of Interaction Design. Abgerufen 11. August 2018, von <http://www.sachinrekhi.com/don-norman-principles-of-interaction-design>
- Rendering Faster and Better with VRWorks in UE4. (o. J.).
- RIPMotion. (2016, April 17). Abgerufen 26. August 2018, von <http://smirkingcat.software/ripmotion/>
- Roberts, J. C. (2001). Redirected Walking.
- Robertson, A. (2016, August 1). Here’s how you could walk forever in an infinite VR hallway. Abgerufen 9. August 2018, von <https://www.theverge.com/circuitbreaker/2016/8/1/12344702/siggraph-virtual-reality-redirected-walking-unlimited-corridor>
- Rogers, P. (o. J.). Interaction Design, 474.
- Rogers, Y., Sharp, H., & Preece, J. (2015). Interaction Design: Beyond Human-Computer Interaction (4. Auflage). Chichester: Wiley John + Sons.
- Ropinski, T., Steinicke, F., & Hinrichs, K. (o. J.). Reflecting on the Potential of VR for the Future of HCI, 4.
- Rossmann, C. (o. J.). Grundlagen des sozialwissenschaftlichen Experiments.
- rvm88. (o. J.). How the Vive Lighthouse Works. Abgerufen von <https://www.youtube.com/watch?v=75ZytcYANTA&t=1031s>
- r/Vive - Vive lighthouse explained. (o. J.). Abgerufen 18. August 2018, von https://www.reddit.com/r/Vive/comments/40877n/vive_lighthouse_explained/
- r/Vive - WalkAbout - a new locomotion system for VR games. Presented in a teaser for future horror game called The Dark Within. (o. J.). Abgerufen 26. August 2018, von https://www.reddit.com/r/Vive/comments/4cclot/walkabout_a_new_locomotion_system_for_vr_games/
- Saft, D. (2013). Gleichgewichtssinn - vestibuläre Wahrnehmung. Abgerufen 9. August 2018, von <https://www.heilpaedagogik-info.de/entwicklungsfoerderung-2/292-gleichgewichtssinn-vestibulaere-wahrnehmung.html>

- Sapolsky, R. (2010, November 14). This Is Your Brain on Metaphors. Abgerufen 8. Juni 2018, von <https://opinionator.blogs.nytimes.com/2010/11/14/this-is-your-brain-on-metaphors/>
- Sears, A., & Jacko, J. A. (Hrsg.). (2009). Human-computer interaction. Fundamentals. Boca Raton: CRC Press.
- Sherman, W. R., & Craig, A. B. (2002). Understanding Virtual Reality - 1st Edition. Abgerufen 19. August 2018, von <https://www.elsevier.com/books/understanding-virtual-reality/sherman/978-1-55860-353-0>
- Sherman, W. R., & Craig, A. B. (2018). Understanding Virtual Reality: Interface, Application, and Design (2. Auflage). Cambridge, MA: Morgan Kaufmann.
- Siegel, Z. D. (2015). Improving distance perception in virtual reality, 65.
- Snyder, C. (2016). Intro to VR: Degrees of Freedom. Abgerufen 11. August 2018, von <http://www.leadingones.com/articles/intro-to-vr-4.html>
- Spektrum Akademischer Verlag. (2000a). Affordanz. Abgerufen 3. August 2018, von <https://www.spektrum.de/lexikon/psychologie/affordanz/320>
- Spektrum Akademischer Verlag. (2000b). mentale Modelle. Abgerufen 3. August 2018, von <https://www.spektrum.de/lexikon/psychologie/mentale-modelle/9560>
- Spell Fighter VR bei Steam. (2016). Abgerufen 25. August 2018, von https://store.steampowered.com/app/455440/Spell_Fighter_VR/
- Spells „n“ Stuff bei Steam. (2016). Abgerufen 25. August 2018, von https://store.steampowered.com/app/448880/Spells_n_Stuff/
- Springer, christoph. (2016). Cyberith Virtualizer verschiebt sich auf unbestimmte Zeit. Abgerufen 8. September 2018, von <https://www.vrnerds.de/cyberith-virtualizer-verschiebt-sich-auf-unbestimmte-zeit/>
- Springer, A. (2016, Februar 21). Bewegungskrankheit: Warum uns auf Reisen übel wird. DIE WELT. Abgerufen von <https://www.welt.de/wissenschaft/article152315539/Warum-uns-auf-Reisen-uebel-wird.html>
- Springer, C. (2016a). Kein Room Scale VR mit Rift oder Playstation VR. Abgerufen 8. September 2018, von <https://www.vrnerds.de/kein-room-scale-vr-mit-rift-oder-playstation-vr/>
- Springer, C. (2016b). Neue Version von Infinadeck auf der CES 2016 - VR-Nerds. Abgerufen 8. September 2018, von <https://www.vrnerds.de/neue-version-von-infinadeck-auf-der-ces-2016/>
- Statista. (2014). Active virtual reality users worldwide 2014-2018 | Statistic. Abgerufen 5. August 2018, von <https://www.statista.com/statistics/426469/active-virtual-reality-users-worldwide/>
- Stelzl, E. (2004). Kinästhetik – Gesundheitsentwicklung in der Hauskrankenpflege.pdf.

- Stone, P. (2015). Valve / HTC Vive VR Render Pipeline & Architecture Dive-Down. Abgerufen 8. September 2018, von <https://www.gamersnexus.net/guides/2068-htc-vive-vr-render-pipeline-and-architecture>
- Suma, E. (2016). Making Small Spaces Feel Large: Practical Illusion in Virtual Reality.
- Takala, T. (o. J.). LECTURE 5: USER INTERACTION IN VR, 95.
- Tanja. (2013, August 16). Wahrnehmung: Gesetz der guten Fortsetzung. Abgerufen 12. August 2018, von <https://usabilitytanja.wordpress.com/2013/08/16/wahrnehmung-gesetz-der-guten-fortsetzung/>
- The challenges of creating a VR application | MiddleVR. (2018, Mai 2). Abgerufen 2. Mai 2018, von <http://www.middlevr.com/resources/the-challenges-of-creating-a-vr-application/>
- The Gallery - Episode 1: Call of the Starseed bei Steam. (2016). Abgerufen 25. August 2018, von https://store.steampowered.com/app/270130/The_Gallery__Episode_1_Call_of_the_Starseed/
- The Haunted Swing. (o. J.). The Haunted Swing illusion. Abgerufen 19. August 2018, von <http://www.deceptology.com/2010/09/haunted-swing-illusion.html>
- The importance of Conceptual Integrity. (2011). Abgerufen 12. August 2018, von http://architecture.typepad.com/architecture_blog/2011/10/the-importance-of-conceptual-integrity.html
- The term „affordance“ in UI design (Philipp Holzer). (o. J.). Abgerufen 29. Mai 2018, von http://www.medien.ifi.lmu.de/fileadmin/mimuc/mmi_ws0506/essays/uebun_g2-holzer.html
- To, S. (2016, Dezember 1). Virtual Reality development with Unreal Engine 4. Abgerufen 7. Juli 2018, von <https://www.brightdevelopers.com/virtual-reality-development-unreal-engine-4/>
- Unreal Engine | Blog. (o. J.). Abgerufen 5. August 2018, von <https://www.unrealengine.com/en-US/blog>
- Unreal Engine 4. (o. J.). Reflections. Abgerufen 8. September 2018, von <https://docs.unrealengine.com/en-us/Resources/Showcases/Reflections>
- Unseen Diplomacy bei Steam. (2016). Abgerufen 20. August 2018, von https://store.steampowered.com/app/429830/Unseen_Diplomacy/
- Van de Kerckhove, E. (2016). HTC Vive Tutorial for Unity | Ray Wenderlich. Abgerufen 8. September 2018, von <https://www.raywenderlich.com/792-htc-vive-tutorial-for-unity>
- Virtual Reality for Human Computer Interaction.pdf. (o. J.).
- Virtual Reality : Introduction. (2018, Mai 4). Abgerufen 4. Mai 2018, von <http://web.tecnico.ulisboa.pt/ist188480/cmul/introduction.html>

- Virtual Reality : Issues and Challenges. (2018, Juni 8). Abgerufen 8. Juni 2018, von <http://web.tecnico.ulisboa.pt/ist188480/cmul/issues.html>
- Virtual Reality Society. (2017). History Of Virtual Reality. Abgerufen 19. August 2018, von <https://www.vrs.org.uk/virtual-reality/history.html>
- Virtual Reality-App. (2017). Virtual-Reality-App für Handrehabilitation nach einem Schlaganfall.
- Virtual-Reality-App für Handrehabilitation nach einem Schlaganfall.pdf. (o. J.).
- Virtuelle Realität. (2018). In Wikipedia. Abgerufen von https://de.wikipedia.org/w/index.php?title=Virtuelle_Realit%C3%A4t&oldid=176794958
- Virtuelle Realität – Wikipedia. (2018, Mai 4). Abgerufen 4. Mai 2018, von https://de.wikipedia.org/wiki/Virtuelle_Realit%C3%A4t
- Virtuo-VR | Kat Walk. (o. J.). Abgerufen 8. September 2018, von <https://www.virtuo-vr.com/products/kat-walk/>
- VR Research. (o. J.). Pinterest. Abgerufen 2. September 2018, von <https://www.pinterest.at/pin/21532904447424423/>
- VRZ: Torment bei Steam. (o. J.). Abgerufen 20. August 2018, von https://store.steampowered.com/app/484350/VRZ_Torment/
- Ware, C., & Balakrishnan, R. (1994). Reaching for objects in VR displays: lag and frame rate. *ACM Transactions on Computer-Human Interaction*, 1(4), 331–356. <https://doi.org/10.1145/198425.198426>
- Warum gilt Rot als Signalfarbe? (o. J.-b). Abgerufen von <https://www.kids-and-science.de/kinderfragen/detailansicht/datum/2009/10/08/warum-gilt-rot-als-signalfarbe.html>
- Was ist Intuition oder intuitiv? Definition | Wendezeit - Informationen zum Leben. (o. J.). Abgerufen 3. August 2018, von <http://wendezeit.ch/was-ist-intuition-intuitiv-definition>
- Weitzman, E. (2016). Why Interaction Must Come Before Language. Abgerufen 3. August 2018, von <http://www.hanen.org/Helpful-Info/Articles/Why-Interaction-Must-Come-Before-Language.aspx>
- Welche Experimente gibt es. (o. J.). Abgerufen 28. Juli 2018, von <http://www.stangl-taller.at/TESTEXPERIMENT/experimentarten.html>
- What is user interface? definition and meaning. (o. J.). Abgerufen 13. August 2018, von <http://www.businessdictionary.com/definition/user-interface.html>
- Why Virtual Reality Is Lot More Important Than You Think. (2016, Mai 23). Abgerufen 3. August 2018, von <http://surevin.com/virtual-reality-lot-important-think/>
- Widrich, L. (2012). The Science of Storytelling: Why Telling a Story is the Most Powerful Way to Activate Our Brains. Abgerufen 8. Juni 2018, von <https://lifehacker.com/5965703/the-science-of-storytelling-why-telling-a-story-is-the-most-powerful-way-to-activate-our-brains>

- Williams, S. P. P. (2017, November 10). Is VR really more immersive? Abgerufen 5. August 2018, von <https://medium.com/@simonppwilliams/is-vr-really-more-immersive-9f885c5795e>
- Windlands bei Steam. (2016). Abgerufen 22. August 2018, von <https://store.steampowered.com/app/428370/Windlands/>
- Wishin, I. (o. J.). Think-negative Technik.
- World of Level Design. (2015). UE4: Beginner Step-by-Step to Creating Your First Level/Map in Unreal Engine 4. Abgerufen 8. September 2018, von <https://www.worldofleveldesign.com/categories/ue4/ue4-step-by-step-first-simple-level.php.Draw+Your+Mind+MapLet>
- Xinxing, T. (Hrsg.). (2012). Virtual Reality - Human Computer Interaction. InTech. <https://doi.org/10.5772/3333>
- Yates, A. (2016). Alan Yates on the Impossible Task of Making Valve's VR Work.
- Zabulis, X., Baltzakis, H., & Argyros, A. (2009). Vision-Based Hand Gesture Recognition for Human-Computer Interaction. In C. Stephanidis (Hrsg.), *The Universal Access Handbook* (Bd. 20091047, S. 1–30). CRC Press. <https://doi.org/10.1201/9781420064995-c34>
- Zyda, M., & DeFanti, T. (2003). Praise for Understanding Virtual Reality: Interface, Application, and Design. In *Understanding Virtual Reality* (S. i). Elsevier. <https://doi.org/10.1016/B978-1-55860-353-0.50019-7>

Abbildungsverzeichnis

Abbildung 1: Anzahl der aktiven VR User, (Statista, 2014).....	1
Abbildung 2: Google Cardboard (links), Brewster's Stereoskop (rechts), (Harding, 2013).....	9
Abbildung 3: The Haunted Swing Illusion, (The Haunted Swing, o. J.).....	10
Abbildung 4: Cycling Sytem, (VR Research, o. J.).....	20
Abbildung 5: Kat Walk VR („Virtuo-VR Kat Walk“, o. J.).....	21
Abbildung 6: Virtualizer (christoph Springer, 2016).....	21
Abbildung 7: WizDish (Feltham, 2015).....	22
Abbildung 8: Infinadeck Treadmill (C. Springer, 2016b).....	22
Abbildung 9: VR Chaperone (Hills-Duty, 2018).....	23
Abbildung 10: HTC VIVE Trackingbereich (C. Springer, 2016a).....	24
Abbildung 11: Rotation Gain- <i>Startposition</i> , (eigene Darstellung), (vgl. Suma, 2016)	26
Abbildung 12: Rotation Gain- <i>Drehung</i> , (eigene Darstellung), (vgl. Suma, 2016)	26
Abbildung 13: Simulation des Curvature Gain, (eigene Darstellung), (vgl. Suma, 2016).....	27
Abbildung 14: Die virtuelle Umgebung des Unlimited Corridor (James, 2016) ...	27
Abbildung 15: Die physische Umgebung bei Unlimited Corridor (James, 2016).	27
Abbildung 16: Translation Gain, (eigene Darstellung), (vgl. Suma, 2016)	28
Abbildung 17: Vergleich zwischen dem virtuellen und physischen weg bei ein einem redirected Walking versucht (Roberts, 2001)	29
Abbildung 18: Impossible Spaces, Änderung der Ausgangsposition, (eigene Darstellung), (vgl. Suma, 2016)	30
Abbildung 19: Darstellung des unendlichen Kreises, (eigene Darstellung), (vgl. Suma, 2016).....	31
Abbildung 20: Impossible Spaces, (eigene Darstellung), (vgl. Suma, 2016).....	32

Abbildung 21: Tunneling, Screen Shot aus dem Spiel „Informer podcast“.....	33
Abbildung 22: Teleportation in „The Gallery“ (Carbotte, 2018).....	34
Abbildung 23: Teleportation, Screen Shots aus dem Spiel „Spells ‘n’ Stuff“	34
Abbildung 24: Auswahl der Teleportationsziel bei „Budget Cuts“ (Lang, 2016) ..	35
Abbildung 25: Visualisierung der Umgebung an Teleportationsziel bei „Budget Cuts“ (Lang, 2016)	35
Abbildung 26: HTC VIVE Head Mounted Display (Stone, 2015).....	40
Abbildung 27: Einstellung der Befehle, (Screen Shot aus Unreal Engine 4).....	40
Abbildung 28: Einstellung der Bewegungsachsen, (Screen Shot aus Unreal Engine 4).....	41
Abbildung 29: Einstellung der Befehleingaben, (Screen Shot aus Unreal Engine 4)	41
Abbildung 30: Einstellung der Fortbewegung, (Screen Shot aus Unreal Engine 4)	42
Abbildung 31: Vergleich zwischen gutem und schlechtem Mapping (Rekhi, 2017)	48
Abbildung 32: Darstellung des Fortsetzungsprinzip (Jerald, 2015a, S. 272).....	52
Abbildung 33: Shader Complexity, (Screen Shot aus Unreal Engine 4), (DevTon Studio, 2017).....	57
Abbildung 34: DevTon Prison, (Screen Shot aus Unreal Engine 4),(DevTon Studio, 2017)	57
Abbildung 35: Darstellung der Decals, (Screen Shot aus Unreal Engine 4),(DevTon Studio, 2017).....	57
Abbildung 36: Decals und ihre Einstellungen, (Screen Shot aus Unreal Engine 4), (DevTon Studio, 2017)	58
Abbildung 37: Project „Reflections“ (Unreal Engine 4, o. J.)	59
Abbildung 38: GPU-Visualizer für einen Frame, Project „Reflections“ (Unreal Engine 4, o. J.)	59
Abbildung 39: GPU-Visualizer für einen Frame, (Screen Shot aus Unreal Engine 4), DevTon Prison (DevTon Studio, 2017)	60
Abbildung 40: Reaktion der dynamischen Objekte (Screen Shot aus Unreal Engine 4), DevTon Prison (DevTon Studio, 2017)	61

Abbildung 41: Farbtemperaturen in Kelvin, Referenz für Helligkeitseinstellung (Owen, o. J.).....	62
Abbildung 42: Screen Shot aus Lighting Academy – Session 3.1 (Milde, 2017).	63
Abbildung 43: Darstellung des Lightmaps, Screen Shot aus Lighting Academy – Session 3.1 (Milde, 2017)	63
Abbildung 44: Finale Lichtberechnung (<i>baking</i>), Screen Shot aus Lighting Academy – Session 3.1, (Milde, 2017)	64
Abbildung 45: Charlie Chaplin in <i>The Gold Rush</i> (Charlie, 1925).....	72
Abbildung 46: Reihenfolge der Umgebungen des Usertests, (eigene Darstellung)	73
Abbildung 47: Tastenbelegung der Motion Controller (Van de Kerckhove, 2016)	74
Abbildung 48: Visualisierung der Teleportation, Screen Shot aus Unreal Engine (Evermotion Kitchen, o. J.).....	75
Abbildung 49: Einstellung der Eingabetasten, (Screen Shot aus Unreal Engine 4)	77
Abbildung 50: Darstellung des NavMeshVolume, (Screen Shot aus Unreal Engine 4), (Evermotion Kitchen, o. J.).....	77
Abbildung 51: Anpassung der Raumgröße, (Screen Shot aus Unreal Engine 4), (Evermotion Kitchen, o. J.).....	78
Abbildung 52: Größenvergleich für realistische Skalierung (World of Level Design, 2015).....	79
Abbildung 53: Anpassung der Schrankgrößen an Character (Evermotion Kitchen, o. J.)	79
Abbildung 54: Einstellung der Player-position in der Umgebung, (Screen Shot aus Unreal Engine 4), (Evermotion Kitchen, o. J.).....	80
Abbildung 55: Fixierung der Playerhöhe in VR, (Screen Shot aus Unreal Engine 4)	80
Abbildung 56: Aufgabentafel in der ersten Umgebung des Usertests, (eigene Darstellung).....	81
Abbildung 57: Der Zustand der ersten Umgebung, wenn der Spieler in das Level einsteigt. (Screen Shot aus Unreal Engine 4), (Evermotion Kitchen, o. J.)..	82

Abbildung 58: Aufgabentafel in der zweiten Umgebung des Usertests, (eigene Darstellung).....	84
Abbildung 59: Darstellung des erwünschten Zustands des Zimmers, (Screen Shot aus Unreal Engine 4)	86
Abbildung 60: Aufgabentafel in der dritten Umgebung des Usertests, (eigene Darstellung).....	89
Abbildung 61: Aufgabentafel in der vierten Umgebung des Usertests, (eigene Darstellung).....	91
Abbildung 62: Blueprint zum Deaktivieren der Physiksimulation eines Objekts beim Aufheben, (Screen Shot aus Unreal Engine 4)	94
Abbildung 63: Blueprint zum Reaktivieren der Physiksimulation eines Objekts beim Ablegen, (Screen Shot aus Unreal Engine 4)	94
Abbildung 64: Blueprint zur Ermittlung des Bewegungsbereichs entlang der gewählten Achse (Screen Shot aus Unreal Engine 4)	95
Abbildung 65: Blueprint zur Ermittlung der ausgewählten Achse (Screen Shot aus Unreal Engine 4)	96
Abbildung 66: Blueprint zur laufenden Aktualisierung der Position der Lade, (Screen Shot aus Unreal Engine 4)	96
Abbildung 67: Blueprint zur Berechnung des Nick-Winkels zwischen Hand und Objekt (Screen Shot aus Unreal Engine 4)	97
Abbildung 68: Blueprint zur Berechnung des Gier-Winkels zwischen Hand und Objekt (Screen Shot aus Unreal Engine 4)	97
Abbildung 69: Blueprint um den Initialwinkel zu speichern (Screen Shot aus Unreal Engine 4).....	98
Abbildung 70: Blueprint zur Berechnung der kürzesten Distanz zwischen zwei Winkeln (Screen Shot aus Unreal Engine 4).....	98
Abbildung 71: Blueprint zur Berechnung des neuen Drehwinkels (Screen Shot aus Unreal Engine 4)	98
Abbildung 72: Blueprint zur Auswahl der Drehachse (Screen Shot aus Unreal Engine 4).....	99
Abbildung 73: Editierbare Eigenschaften eines HingedDoor Blueprint (Screen Shot aus Unreal Engine 4)	99

Abbildung 74: Blueprint zur Aktualisierung der Drehung entlang der ausgewählten Achse (Screen Shot aus Unreal Engine 4).....	99
Abbildung 75: Darstellung der Collision Boxes, original (links), bearbeitet (rechts), (Screen Shot aus Unreal Engine 4)	100
Abbildung 76: Darstellung der Entnahme eines Gegenstandes aus der geschlossenen Schublade (Screen Shot aus Unreal Engine 4).....	100
Abbildung 77: Anpassung des Vorlagenblueprints zur Präferenzierung bestimmter Objekttypen (Screen Shot aus Unreal Engine 4)	101
Abbildung 78: Blueprint zur Ermittlung des aufzunehmenden Objekts (Screen Shot aus Unreal Engine 4)	101
Abbildung 79: Blueprint um die Grab Sphere zu teleportieren, falls sie sich zu weit entfernt (Screen Shot aus Unreal Engine 4)	102
Abbildung 80: Visualisierung der Grab Sphere (Screen Shot aus Unreal Engine 4)	102
Abbildung 81: Blueprint zur Aktualisierung der Liste an Objekten, die vom Zeigestab überlappt werden. (Screen Shot aus Unreal Engine 4).....	103
Abbildung 82: Blueprint zur Ermittlung des nächstgelegenen Objekts, auf das der Player zeigt (Screen Shot aus Unreal Engine 4).....	104
Abbildung 83: Blueprint zur Aktualisierung der Liste an Objekten, die vom Zeigestab überlappt werden. (Screen Shot aus Unreal Engine 4).....	104
Abbildung 84: Blueprint zur Ermittlung des nächstgelegenen Objekts, auf das der Player zeigt, (<i>erste Version</i>) (Screen Shot aus Unreal Engine 4)	105
Abbildung 85: Blueprint zur Berechnung des Z-Werts des Treffpunktes (Screen Shot aus Unreal Engine 4)	105
Abbildung 86: Blueprint zur Berechnung der Zielposition (Screen Shot aus Unreal Engine 4).....	106
Abbildung 87: Blueprint zur Bewegung des Objekts zur Zielposition (Screen Shot aus Unreal Engine 4)	106
Abbildung 88: Blueprint zum temporären Anheben des gewählten Gegenstands (Screen Shot aus Unreal Engine 4)	107
Abbildung 89: Blueprint zum Austausch des Materials eines Objekts (Screen Shot aus Unreal Engine 4)	107

Abbildung 90: Darstellung des bewegbaren Fensters (Screen Shot aus Unreal Engine 4).....	109
Abbildung 91: Darstellung des Bereichs, in dem sich das Fenster bewegen kann, (eigene Darstellung).....	109
Abbildung 92: Ermittlung der Transparenz entlang der X-Achse, (Screen Shot aus Unreal Engine 4)	110
Abbildung 93: Ermittlung der Transparenz entlang der Y-Achse, (Screen Shot aus Unreal Engine 4)	110
Abbildung 94: Blueprint zur Erstellung eines dynamischen Materials für Interaktionsmöglichkeiten, (Screen Shot aus Unreal Engine 4).....	111
Abbildung 95: Blueprint zur laufenden Aktualisierung der Position des Fensterrahmens und der transparenten Öffnung, (Screen Shot aus Unreal Engine 4).....	111
Abbildung 96: Blueprint zur Berechnung des Bewegungsbereichs entlang einer Achse, (Screen Shot aus Unreal Engine 4)	112
Abbildung 97: Blueprint zur Deaktivierung der Wandkollision, (Screen Shot aus Unreal Engine 4)	112
Abbildung 98: Blueprint zur Berechnung des Winkels zwischen Controller und dem Mittelpunkt der Wanduhr, (Screen Shot aus Unreal Engine 4)	113
Abbildung 99: Blueprint zur Berechnung des relativen Winkelabstands, (Screen Shot aus Unreal Engine 4)	113
Abbildung 100: Blueprint zur Normalisierung einer Winkelangabe, (Screen Shot aus Unreal Engine 4)	114
Abbildung 101: Blueprint zur Berechnung der InGame-Zeit, (Screen Shot aus Unreal Engine 4)	114
Abbildung 102: Blueprint zur Bewegung des Minutenzeigers, (Screen Shot aus Unreal Engine 4)	115
Abbildung 103: Blueprint zur Berechnung der Rotation der Zeiger und Lichtquelle, (Screen Shot aus Unreal Engine 4)	115
Abbildung 104: Darstellung der letzten Aufgabe des Usertests, (Screen Shot aus Unreal Engine 4)	116
Abbildung 105: Blueprint zum Beenden der Anwendung mit den erwünschten Tastenkombinationen, (Screen Shot aus Unreal Engine 4)	116

Abbildung 106: Blueprint zur Berechnung der Anzahl an Interaktionen, (Screen Shot aus Unreal Engine 4)	117
Abbildung 107: Blueprint zum Zugriff auf die in der <i>GameInstance</i> gespeicherten Daten, (Screen Shot aus Unreal Engine 4).....	117
Abbildung 108: Blueprint zum Speichern von Zeitstempeln, (Screen Shot aus Unreal Engine 4)	117
Abbildung 109: Blueprint zum Zusammenfügen der Textelemente für die finale Darstellung, (Screen Shot aus Unreal Engine 4)	118
Abbildung 110: Blueprint zur Darstellung der Ergebnisse des ersten Levels, (Screen Shot aus Unreal Engine 4)	118
Abbildung 111: Das Einstiegslevel. Blick, wenn der Player den Usertest beginnt. (Screen Shot aus Unreal Engine 4)	119
Abbildung 112: Blueprint des Umgebungswechslers, (Screen Shot aus Unreal Engine 4).....	119
Abbildung 113: Darstellung der ersten Interaktion mit einem Objekt in den Umgebungen (in Sekunden), (eigene Darstellung).....	121
Abbildung 114: Darstellung der gesamten Zeit (in Minuten), die der Player in einer Umgebung verbracht hat, (eigene Darstellung)	122
Abbildung 115: Darstellung der vergebenen Note hinsichtlich der Immersion, (eigene Darstellung).....	122
Abbildung 116: Blueprint zum temporären Anheben des gewählten Gegenstands (Screen Shot aus Unreal Engine 4)	128
Abbildung 117: HOHO Außenbereich, (Screen Shot aus Unreal Engine 4).....	132
Abbildung 118: HOHO Lobby, (Screen Shot aus Unreal Engine 4)	133
Abbildung 119: HOHO Office, (Screen Shot aus Unreal Engine 4).....	133
Abbildung 120: HOHO, Darstellung des sichtbaren Teleportationsbereichs, (Screen Shot aus Unreal Engine 4)	134
Abbildung 121: Visualisierung des Teleportationsziels, (Screen Shot aus Unreal Engine 4).....	134
Abbildung 122: HOHO Umgebungswechsler, (Screen Shot aus Unreal Engine 4)	136
Abbildung 123: Feedback durch Vergrößerung bei der Auswahl, (Screen Shot aus Unreal Engine 4)	137

Anhang

A. Interview mit Marcel Nürnberg SQUARE BYTES Interactive Media

Arian: Ich habe gesehen, dass ihr zur zweit seid. Ich habe aber ehrlich gesagt nicht viele Videos von Euren interaktiven Sachen sehen können. Macht ihr es in Unreal Engine? Nehme ich an, weil ich diese Pisslinie nicht gesehen habe...

Marcel: (lachen). Also grundsätzlich ja, wir arbeiten mit Unity, wir arbeiten mit Unreal Engine, also je nachdem, was der Bedarf ist. Und nochmals auf deine Affordanzgeschichte zu kommen, wir, vom Team her, sind wir ja im Prinzip, also ich bin ja der ursprüngliche Gründer, mein Background ist Computer graphic und Softwareentwicklung und mein Geschäftspartner, der kommt aus der Finanz und Vertriebsbereich und Immobilien.

Arian: Sehr typische Kombi, ja

Marcel: Ja, Wir haben hier so zu sagen, 2 Software Developer und kaufen und Graphikleistungen einfach zu. So Freelancer, Agenturen usw. Das heißt, wir sind eigentlich keine Agentur, die Graphikleistungen anbietet.

Arian: Das heißt, wenn ihr jetzt sage ich mal grob gesagt, Assets braucht, kauft ihr die von ihnen?

Marcel: Ja, also es gibt entweder so Stores wo man sie kaufen kann. Man kann auch für Kunden, wenn sie spezielle Möbel haben möchte, die lassen wir dann produzieren und im Prinzip alles für den Echtzeitgebrauch.

Arian: Das heisst, nicht so highpoly usw.

Marcel: Das machen wir auch. Wir sind im Prinzip eine Agentur, die alle möglichen Vertriebstools von einer Immobilienbranche für ein Immobilienprojekt anbieten. Das fängt an von einer Website, Folder, auch Renderings, Werbefilme, VR, AR, Experiences, Im Prinzip das volle Programm. Das sind wir eigentlich.

Arian: Okay, das heißt, ihr sein nicht speziell auf VR oder... ok...

Marcel: Also das VR Know How und die Softwareentwicklung ist hier. Das Herzstück im Prinzip und der Rest sind dann immer Kooperationen mit anderen Agenturen.

Arian: Ok. Verstanden.

Marcel: Und... Nochmals auf diese Affordanzfrage zurück zu kommen. Also wir haben unseren Prozess so gebaut, dass wir so Rapid Prototyping betreiben. Im Prinzip denken wir uns kleine Interaktionen aus und sie werden dann schnell programmiert und dann in ein, zwei, drei Tagen versuchen wir die und holen uns das Feedback und versuchen es dann zu verfeinern. Und auf deine Frage zu kommen, also um dir ein Beispiel zu geben, wir haben versucht etwas zu bauen, wo sich einer der die Brille aufsetzt, sofort weiß, was er tun muss, ohne dass er irgendwo eine Introduction kriegt. Dass es einfach so intuitiv ist, dass er weiß, was er tun muss und daran arbeiten wir im Prinzip. Dass er Weiß, okay ich kann jetzt teleportieren, dass er weiß, okay, ich kann da drücken... All das versuchen wir jetzt gerade im Design darzustellen... und das lustige ist eigentlich, dass wir von Design keine Ahnung haben. Wir haben keine Ahnung, wie solche Interaktionen passieren, sondern wir versuchen eben durch Rapid Prototyping drauf zu kommen.

Arian: Wenn ich es richtig verstanden habe ist Rapid Prototyping einfach testen, testen

Marcel: Schnell testen, mobil und agil zu agieren und möglichst schnell viel Erfahrung zu sammeln.

Arian: Okay, also ich möchte jetzt nicht klugscheissen, korrigier mich, wenn ich falsch bin. Ich bin selber auch beim Schreiben...Aber ein Teil von dem Ganzen ist wirklich Design und der andere Teil ist... so wie ich es verstanden habe, man geht einfach von der psychologische Wahrnehmungsaspekte von Menschen aus. Wie denken Menschen? Das geht Richtung Psychotherapie, Verhaltenstherapie und schieß mich tot... Dass warum suchen sich Menschen Lösungen, die so sind wie sie sind. Das beste Beispiel, dass sie halt immer haben ist, dass mit der chinesischen Schere. Wie die Schere bei ihnen aussieht. Unsere Schere sieht so aus und deren Schere sieht so aus. Und wenn wir diese Schere sehen, kommen wir nicht eigentlich sofort drauf, warum ist es so. Später kommst du drauf, dass es viel logischer ist, weil das eine größere Ähnlichkeit zu der Funktionalität hat. Also es ist auch kulturell bedingt usw. Also man fragt sich zuerst wie denken Menschen und warum denken sie so wie sie denken und dann setzt man sich hin und denkt man über Form, Farben usw. nach... so, dass halt um meinen Senf dazu gegeben zu haben. (Lachen)

Marcel: Ne super, also wir haben da keinen akademischen Zugang, wir machen es halt einfach intuitiv und was uns am meisten, also ziemlich geile Experience war... Also bei den Applikationen hast du halt immer so einen geraden Strahl, wo du halt was auswählen kannst. Und das war uns halt nicht intuitiv genug. Das hat keiner verstanden. Keiner hat erstens den Strahl gecheckt und keiner hat gecheckt, dass er auf den Knopf drücken muss. So wir haben uns hingesezt und haben gesagt, okay, was machen wir da. Wir haben uns von Unity, da gibt es auch

ein Strahl, aber der ist nicht gerade... der wabbelt so im Prinzip mit. Der zieht so bisschen nach... und in Kombination mit Snapping, sind wir mit Zufall draufgekommen, dass, wenn es irgendwelche interessante Objekte gibt, erzeugt du damit eine implizite Aufmerksamkeit. Also er wabbelt und wenn er bei dem Sofa ankommt, dann snappt er hin und wenn ich weitergehe, dann bleibt er trotzdem dort hängen und da denkt man sich, okay, da muss irgendwas Interessantes sein. So haben wir gecheckt, ah so können wir zum Beispiel Aufmerksamkeit schaffen, zu Gegenständen.

Arian: Ich sage mal so, das ist ein Hinweis. Ein hint.

Marcel: Richtig. Aber ohne, dass irgendeine Schrift dort auftaucht, also dadurch dass sich das Verhalten von einem Objekt ändert erzeuge ich damit Aufmerksamkeit. Jetzt müssen wir die Leute dazu bringen, dass, wie schaffen wir es auf den Knopf zu drücken?

Dass er einfach jetzt den Knopf drückt...

Intuitiv, Wie schaffen wir das? Das haben wir noch nicht rausgefunden, da sind wir auch am Prototyping. Versuchen alles damit aus.

Arian: Aber schau mal, wenn man jetzt wie ein Programmierer vorgeht und alles sehr genau analysiert, ist das kein Signal von diesem Objekt sondern eine Reaktion von dem Interface. Also dieser Strahl halt, und der geht hin und snappt da. Das Objekt ist in dem Fall ja immer noch passiv. Der macht nichts. Jetzt frage ich mich, wie würde man es schaffen, dass dieses Objekt dieses Signal sendet, „hey mich kann man anfassen“. Zum Beispiel: Du siehst, eine komplette Wand aus Glas. Die ganzen Glasplatten sehen alle gleich aus und nur an einer Stelle hast du einen Griff, eine Klinke, du gehst automatisch dorthin und greifst dahin. Das ist also das einzige, was dir ein Signal sendet. Also das ist etwas was du nicht von dem Interface bekommst, sondern von dieser Tür. Ich glaube eigentlich, dass ich mit Affordanz das meinte. Das andere Signal geht Richtung User Interfaces, oder?

Marcel: Ja, stimmt. Um dir ein Beispiel zu geben, wir haben zum Beispiel immer unseren Griller draußen stehen gehabt. Und er war immer geschlossen gewesen. Und 80 % von den Leuten wollten diesen Griller aufmachen. Das war für uns ein Zeichen, wir müssen diesen Griller einfach offenbar machen, dass da drin wirklich Essen serviert ist. Jeder hat erwartet, dass er sich diesen Griller aufmachen kann. Von dem User Interface usw. hat er kein Signal bekommen, das war intuitiv. Jeder Mensch will einen Griller aufmachen. Oder Klodeckel. Jeder Mensch will einen Klodeckel aufmachen. So dass sind Experiences, die wir durch probieren mit verschiedenen Gegenständen sammeln konnten, dass Menschen sich so verhalten, wenn sie es sehen. Bei den Türen lustigerweise, sie checken es nicht. Sie checken nicht, dass sie die Tür aufmachen können. Das hat mich so geflasht,

weil ich mir gedacht habe, ist doch logisch, eine Tür öffnet man doch. Macht keiner, jeder steht davor und denkt, okay, was mach ich jetzt.

Arian: [...] zu dem Thema Controller zurück zu kommen,

Marcel: Das wollte ich noch sagen. Wir glauben, dass die Leute, die Türen nicht aufmachen, weil sie einen Controller in der Hand haben. Wenn die Leute nur ihre Hände hätten, natürliches Interface, dann würden sie es machen. Das ist glaub ich das Problem.

Arian: meinst du jetzt halt, Controller oder...

Marcel: kennst du die HTV VIVE Controller? Also ich glaube, wenn die Menschen nur ihre Hände frei hätten, würden sie sich wieder anders verhalten.

Arian: kennst du die Leap Motion? [...] Meine Erfahrung damit war auch, dass es eigentlich eine sehr gute Idee ist, aber irgendwie... doch noch nicht...

Marcel: Ja weil man wahrscheinlich diesen Force Feedback dazu braucht. Wenn man eine Türklinke wirklich mit der Hand greift, dann will man eigentlich wirklich etwas in der Hand haben und das ist verwirrend. Das ist glaub ich diese Uncanny Valley...

Arian: ja, genau. Deshalb denke ich dass Schiebetüren besser funktionieren würden, wobei man auch irgendwie signalisieren musst, dass das Zum Beispiel eine Schiebetür ist.

Marcel: Genau, und ein blickendes Icon funktioniert nicht!

Arian: Ja,

Marcel: Ich kann es dir zeigen. Komm mal mit.

(das gesamte Experiment wurde mir gezeigt und das Gespräch wurde währenddessen nicht aufgenommen.)

Arian: Wenn man über VR-Themen liest, es bewegt sich alles in einer Spanne zwischen so realistisch wie möglich, Objekte hin und herschieben... wie macht man das... „Do, what you can't“, also mach in VR alles, was in der realen Welt nicht möglich ist. Wir nehmen diese Spanne von 0 bis 100 und versuchen Interaktionen zu schaffen, bzw. Affordanz von Objekten zu testen, die sich genau in dieser Spanne bewegen. Das erste ist Bewegen von Objekten, wie es in der Realität auch ist. Das nächste ist das ändern der Farbe und Material. Es ist zwar realistisch aber nicht auf Knopfdruck und so leicht. Das dritte ist das bewegen von Objekten, die noch ein bisschen weiterweg gehen, wie Bäume, Wände, Fenster usw. und das vierte ist das Verändern der Zeit. Hier geht es darum, wie könnte ich ohne Hinweise einen Menschen dazu zu bringen, dass er etwas tut, was er

wahrscheinlich sein ganzes Leben gewünscht hat, wie die Zeit zu versetzen. Hey, das kannst du hier schon machen, aber wie sag ich ihm das?

So das ist eigentlich mein User Test um die Frage zu beantworten, dass die Leute mit der Erfahrung die sie aus dem Leben mitbringen haben sie versucht Objekte zu greifen, etwas zu bewegen, dies und das zu machen aber es wird immer schwieriger, weil sie ja keine Erfahrung haben wie sie die Zeit verändern können. Deshalb kommen sie auch nicht drauf. Es gibt eine Annahme, die besagt: wenn die Leute, die Skills von Interaktion die sie von der realen Welt haben, mitnehmen können in VR, ist es für uns das allerbeste, weil dann jeder Mensch der einsteigt ganz viel Wissen über diese Welt hat, wie alles funktioniert und du musst ihm nicht alles von Vorne beibringen und das ist der Grund, warum man sich nach Wirklichkeit richtet. Aber das Problem ist eben, dass wir halt nicht immer alles realistisch machen wollen. Wir wollen ja eigentlich weggehen.

Marcel: und dann wird es schwierig so zu sagen.

Arian: Das heißt, du musst in der Lage sein, es ziemlich schnell, also innerhalb von wenigen Minuten zu erweitern. Und ja... Wie macht man das? Meine Erfahrung und meine Vermutung ist, dass die Leute in VR vielen nicht mit Hilfe von ihrem Wissen tun, sondern sie testen einfach herum. Sie fassen alles an und schauen was möglich ist und was nicht. Aber um es zu beweisen muss ich diesen Test machen.

Marcel: Also die Annahme, okay, sagen wir so... Bei uns funktioniert es relativ gut, weil wir ja eigentlich schon die Realität abbilden. Für Immobilien ist es auch... ja... Man soll den Kunden auch nicht tricksen und sagen, das ist eine super tolle Villa und es ist ein kleines Loch, was Scheiße aussieht. Das soll eben nicht passieren. Deshalb können wir ja eigentlich schon ansetzen, dass die Leute ja in der ersten Phase dieses Vorwissen haben. Aber sogar da differenziert sich, wie große dieses Vorwissen ist. Da hast Leute, die nicht einmal wissen, wie sie einen Knopf drücken, bis hin zu Leuten wie Dich zum Beispiel die voll drin sind und die damit fast aufwachsen. Auch da kämpfen wir schon. Das habe ich dir auch erzählt, mit dem Strahl usw. Also ich habe mir gedacht... Wie war das mit dem Handy. Grundsätzlich hat keiner gewusst, wie man es einschalten kann am Anfang. Und Keiner hat gewusst, dass ich jetzt zoomen kann. Aber anscheinend war das eine intuitive Geste, die die Leute relativ schnell gelernt haben und jetzt in ihrem Körper haben und alle Applikationen jetzt darauf aufbauen. Und meine Frage ist: Wird sich das in VR auch irgendwie durchsetzen? Dass die Leute wissen, ich kann einfach Gegenstände aufheben und das ist eh logisch. Das kann ich in der Realität auch, mit der Zeit, wenn sie die Technologie lernen. Also je weiter und je fortgeschrittener wir in der Zeit sind, desto mehr verschiebt sich dieses Anforderungsfenster. Und irgendwann wird dann die Zeit kommen, dass man gelernt hat, okay, ich weiß, irgendwo wird ein 3D Slider sein, der herumschwirrt und irgendwo, als Beispiel.

Und ich weiß, wenn ich es sehe, dann kann ich damit keine Ahnung die Zeit verändern oder die Lichtsituation, aber ohne das Vorwissen des Sliders, nein, es geht einfach nicht. Und, die Erwartungshaltung ist auch eine wichtige. Die Erwartungshaltung, die der Kunde hat. Wir haben den Leuten einfach die Brille aufgesetzt und haben gesagt, ja mach mal! Bis hin zu... Okay du bekommst einen Controller in der Hand und du hast nur den One-Button Solution. Das heißt du hast nur den einen Daumen und mit dem kannst du machen was du willst. Allein diese Information bringt so viel. Wir haben auch immer gedacht, wie können wir Makler einschulen, damit sie ihre Kunden die besten Informationen so schnell wie möglich geben virtuelle. Und das war einfach nur dieser eine simple Satz. Einen Knopf und damit kannst du alles machen, was du willst und das hat extrem geholfen, die Vorinformation und auch die Erwartungshaltung. Das heißt, wenn du die Brille aufsetzt, bist dann in einer Immobilie. Wahrscheinlich in der, wo der Makler schon den Vorschlag gemacht hat. Das ist wie in den Games, dass du startest und am Anfang nicht weißt wo du dich befindest, was du eigentlich für einen Charakter bist usw. das weißt du eben nicht. Also es besteht aus Vorwissen, Erwartungshaltung und die Information, die du vorher bekommst. Aus diesen drei Dingen, glaube ich, wirst du dann wissen, wie sich diese virtuelle Welt dann verhalten wird.

Arian: Ich bin nicht 100 % damit einverstanden. Aber offensichtlich ist es immer noch so, also nicht, mit dem was du gesagt hast, sondern mit der Aussage, dass viele sagen, VR ist immer noch für die Menschen eine blinde Erfahrung. Sie sind in VR immer noch blind. Das was du sagst, mit dem Handy zum Beispiel... Ich sehe da ja meine Hand und sie bewege ich dann. Wenn ich meine Hand nicht sehe, dann komme ich auch nicht auf die Idee zu zoomen.

Marcel: Also, weil der eigene Körper nicht drin ist.

Arian: genau. Und das wird dann auch solan g so bleiben, bis haptisch auch alles funktioniert. Ichvergleich das auch immer mit der Maus. Es kommt auch auf die Degrees of Freedom an, DOF. Je mehr DOFs ein Device hat, desto verwirrender ist es für die Menschen. Bei einem Handy war es gut, weil, egal, was du gemacht hättest, war es bei einem Handy halt 2D. Wie eine Maus oder Keyboard. Deshalb hat man auch nicht wirklich eine Anleitung gebraucht. Man muss sich ja nur um 2 Achsen kümmern. Es ist aber in VR nicht so. Obwohl wir Menschen eigentlich 3D denken und leben, aber viele von uns haben mit dieser dritten Dimension schon bisschen ein Problem. Warum das so ist, kann ich dir nicht erklären, weiß ich selber auch nicht. Aber das ist auch der Grund warum Computer so 2D und so gut funktionieren.

Marcel: Aber in VR. Nimm dir mal das optimale Interface her. Das bist du selber und die Umgebung. Das ist das beste aber dann müssen wir da bei Realität bleiben und können nicht abstrakter werden. Für mich gibt es 2 Grenzen, die zu Gefahr werden können. Einerseits, wenn es zu abstrakt ist. Das typische Beispiel ist die

Lokomotion, also die Fortbewegung. Also je weiter du dich von der realen Bewegung entfernst, desto unwohler fühlst du dich dabei.

Und auf der anderen Seite hast du den Uncanny Valley. Je realistischer etwas wird, desto sensibler werde ich auf kleine Diskrepanzen, die nicht funktionieren und irgendwo dazwischen funktionieren es ziemlich gut, sind wir draufgekommen. Das beste Beispiel ist das Teleportieren. Es funktioniert ziemlich gut, weil das genau dazwischen liegt. Es ist nicht so realistisch aber auch nicht so abstrakt.

Oder wir schaffen es irgendwann, dass wir alle Sinne perfekt und richtig ansprechen können oder sie zumindest glaubhaft manipulieren können.

Dass sich alle Anwendungen und Möglichkeiten sich irgendwo genau in der Mitte bewegen ist der Trend.

Arian: So, um die Fragen ganz konkret zu stellen. Die erste Frage lautet: Welcher Methoden der Interaktion mit der Innenarchitektur in VR haben sich bis jetzt am besten bewährt und warum sind sie sinnvoller einzusetzen als andere Methoden?

Marcel: An das haben wir gar nicht gedacht. Wir haben versucht so abzubilden wie sie sind. Am Anfang war alles beweglich. Man konnte alles Umwerfen unso. Das wollen die Auftraggeber aber nicht, weil sie einen perfekten Stand der Dinge visualisieren wollen. Sie wollen nicht, dass du den Tisch und Stuhl verrückst, weil das ein Teil von dem Interior Design ist. Also es geht darum einen Eindruck zu schaffen und kein Chaos. Das war einfach der Anspruch, warum wir alles so gemacht haben.

Arian: Es ist für die Applikation in VR von Vorteil, wenn der Benutzer auf die gleiche Art und Weise mit der Umgebung interagieren kann, wie er mit der realen Welt interagiert. Es ist somit davon auszugehen, dass der Benutzer mit einem großen Vorwissen an Möglichkeiten der Interaktion in die virtuelle Welt einsteigt. Hypothese A: Es ist immer besser oder richtig, sich bei der Interaktion mit der Umgebung und dem Verhalten der Objekte nach der Wirklichkeit zu richten.

Marcel: Also ich würde nicht generalisieren. Wenn ich jetzt zum Beispiel das Sofa hernehme. Nein, angenommen, wir bauen jetzt ein Tool, wo ich dieses Zimmer einrichten kann und ich hätte schon diese Gegenstände hier stehen, dann gibt es halt mehrere Möglichkeiten dieses Sofa zu bewegen. Entweder grabbing und ich kann damit machen, was ich will, oder ich bau absichtlich Limitierungen, weil ich weiß, dass dieses Sofa eigentlich immer auf die 4 Füßen stehen wird. Deshalb limitiere ich die Bewegung nur auf dieser Ebene. Weil... Naja, was bringt mir das, wenn ich die Möglichkeit habe das Sofa auf dem Tisch zu stellen. Also vielleicht... das Nutzen des Möbels heranzuziehen als Limitierung und somit als Hilfe zur Einrichtung, so würde ich das sehen. Also es hängt davon ab, was willst du damit machen.

Arian: Ja, aber kann man sagen, dass du damit für den User im Nachhinein bisschen vordenkst?

Marcel: Ja, klar, also ich möchte im Prinzip. Also man muss immer denken, so was ist das Ziel, was will ich damit machen? Ich will eine realistische Möblierung hier darstellen. Und da ist es nicht wirklich realistisch, wenn das Sofa dann auf dem Tisch ist. Natürlich kann ich dem Benutzer die Freiheit geben, aber dann ist es halt ein lustiges Spiel. Bauen Sie sich ein Sesselturm oder so, aber das verfehlt komplett das Ziel, dass ich diesen Raum komplett einrichten kann.

Arian: Wie weist man den Benutzer in VR darauf hin, welche Objekte auf eine Interaktion reagieren und welche nicht? Was sind die Ansätze zur Lösung ähnlicher Art festzulegen?

Marcel: Also ich habe jetzt auch nicht die Antwort, aber ich glaube es ist falsch, die 2D Screens in 3D abzubilden. Es ist falsch Sliders oder 2D Buttons zu verwenden.

Arian: Du findest auch falsch, dass man so Inventory hat und Objekte auswählen kann, wie Menüs usw.

Marcel: Wieder Zielsetzung. Unser Anspruch ist, mit möglichst wenig Interaktion und möglichst intuitiv, etwas zu bewirken in der Welt. Und durch die Menüs, wird es plötzlich zu einem Experten Tool. Durch die Menüs musst du plötzlich nicht... Also stell dir vor, du bist bei der Küche, hast einen Strahl und dann drückst du und dann kommt ein Menü und dann kannst du verschiedene Materialien auswählen und dann machst du das Menü wieder zu oder wie auch immer. Ja? Das sind gewesen, Hinzeigen, Drücken, Menü erscheint, ich wähle aus und schliesse wieder. Das waren jetzt 4 oder 5 Aktionen. Warum nicht, hinzeigen und drücken? 2 Aktionen. Und ein User Interface schaffen, dass wie diesen Wabbel-strahl die Aufmerksamkeit dorthin richtet, und dann der 2. Punkt, den wir noch nicht wissen, dass er den Knop drückt. Und ich habe plötzlich genau dieselbe Funktionalität aber nur mit 2 Interaktionen geschafft. Um das geht es. Das ist eben die Zielsetzung. Es ist nicht falsch aber für unsere Zielsetzung halt nicht relevant. Wir können jetzt ein Einrichtungstool bauen für Experten mit 10000 Menüs mit einem Inventory von 5000 Möbeln und einem Searching tool und Sprachsteuerung, der auch filtern kann und was auch immer. Das können wir alles bauen, aber das ist nicht unser Anspruch.

Arian: die letzte Frage: Ist es lohnenswert, sich über die physikalischen, psychologischen und ergonomischen Aspekte Gedanken zu machen, oder sind die Entscheidungen in den meisten Fällen eine Auswahl von bestehenden Möglichkeiten, die sich bereits bewährt haben oder letztendlich eine „Geschmacksfrage“?

Marcel: Bei uns ist es grundsätzlich so: wir haben zwar Ideen, aber solange wir sie nicht in einer größeren Runde probieren, sind sie irrelevant. Weil wir ja die Dinge nicht für uns bauen, sondern für die Masse eigentlich. Also ja natürlich sind psychologische Komponenten und auch technische Limitierungen ein Thema. Natürlich würden wir gerne die Realität perfekt abbilden aber wir können es halt noch nicht und sind da auch limitiert. Und Geschmacksfrage: gut, eine Idee kommt ja von einem bestimmten Geschmack. Man denkt doch, ja das wäre cool, wenn man so was einbauen könnte. Aber entscheidet tut quasi der Markt wirklich.

Habe ich dir irgendwie helfen können?

Arian: Du?? Du hast mir die Hälfte der Diplomarbeit geschrieben. (lachen)

Vielen Danke nochmals. Es war wirklich gut.

Marcel: Gerne.

B. Interview mit Axel Dietrich, Vrisch

Arian: Das Thema dieser Arbeit ist Architekturvisualisierung und Interaktion mit dem Raum in VR. Ich gehe einfach davon aus, dass man das ganze Modellieren von Assets erstmal hinter sich gebracht hat. Das heißt, man ist jetzt mit der Visualisierung fertig und jetzt geht es um die Interaktion.

Wenn ihr eine Anfrage von den Kunden bekommt, wie ist es für Euch? Richtet ihr euch an die Realität oder was ist Interaktion für euch wirklich? Zum Beispiel, Wände kann man nicht hin und her schieben in der Realität. Aber denkst du in VR sollte man es können?

Axel: Ganz prinzipiell muss man überlegen, wer schaut sich das an, mit welchem Ziel und was für ein Erlebnis will die Person überhaupt haben. Ist das eine Präsentation von einem fertigen Gebäude oder Raum wo eigentlich keine Interaktion notwendig ist und wo es eigentlich nur darum geht diesen Raum selbst zu erleben.

Arian: Also wie es einfach aussieht.

Axel: Ja genau. Ja so wie wir es bei unseren ersten Visualisierungen hatten. Da war es so, dass der Gerät selber und die Installation selber ist ohne Betreuung gewesen. Also die Anforderung war, dass sich das Jeder, von der 80 jährigen Oma bis zu einem 12 jährigen Kind sich einfach die Brille aufsetzen können und verstehen worum es geht. Und da war die Anforderung und die Idee, die Interaktion so simple wie möglich zu halten. Und da ging es auch nur darum, sich in dem Raum zu bewegen.

Wenn man jetzt einen Architekten hat, der mit diesem Raum experimentieren möchte und verschiedene Varianten ausprobieren möchte ist natürlich genau das andere Extreme. Da geht man sowieso davon aus, dass sie die Steuerung in und auswendig kennen und wissen wo jeder einzelne Knopf ist... da könnte man dann... ja... alles Mögliche an Interaktion mit einbauen, angefangen bei Wände verschieben und Fenster auf und zu machen.

Arian: Aber das ist etwas, dass von ihnen als Wunsch kommt und ihr es umsetzen musst.

Axel: Ja, oder man erarbeitet es mit dem Kunden gemeinsam. Ja... was jetzt wirklich die Anforderungen sind... dementsprechend gibt es verschiedene Tiefen an Interaktionen und je tiefer das wird desto komplexer wird dann natürlich die ganze Steuerung.

Arian: Wie macht man es dann, wenn man vor den technischen Schwierigkeiten steht, die die Umsetzung von manchen Sachen nicht möglich machen. Zum Beispiel wie macht man den Step Forward, wenn es einem in VR eigentlich schlecht wird... wie löst du es, damit es niemandem auffällt...

Axel: Es hängt einerseits davon ab, wie viel Erfahrung die Leute schon mit VR haben. Das macht natürlich auch einen Unterschied. Es gibt mittlerweile auch schon Erfahrungsberichte. Ich mag jetzt nicht Studien sagen, das wäre jetzt zu weit geführt, aber Erfahrungsberichte halt von den Leuten, die es viel nutzen und sich halt diese Bewegungen... äh... Motion Sickness dass sie es überwinden mit der Zeit.

Das sich einfach das Gehirn und der Körper daran gewöhnt und man härtet halt ab in dem Effekt.

Sprich, wenn man mit solchen Leuten arbeitet, kann man die Steuerung ganz anders bauen als mit jemandem der das vielleicht zum ersten Mal macht. Auch Alter dürfte da einen Unterschied machen, ältere Leute tun sich etwas schwerer mit Motion sickness als junge Leute usw.

Arian: Aber wie ist denn das wenn du ein Projekt hast, wie Hoho? Das weißt du ja nicht.

Axel: Nein, bei Hoho weiß man es nicht. Das heißt, man muss es für alle erträglich machen. Das heißt man nimmt die aller erträglichste Variante, das heißt keine Beschleunigungen keine Bewegungen in dem Sinne. Wir haben es in dem Fall über Teleportation gelöst. Wo wir wissen, dass es funktioniert, wenn nicht einer irgendwie alle halbe Sekunde auf Teleport drückt, dann ist das was jeder aushält eigentlich. Alles andere was an Bewegung passiert ist halt die direkte Übersetzung von meinen Bewegungen, das funktioniert aus meiner Erfahrung alle. Solang sich

halt der Raum, in dem ich bin nicht bewegt, usw. Also solange jede Bewegung aus realer Welt in die virtuelle Welt übertragen wird und nichts zusätzliches passiert.

Arian: Gibt es irgendwelche Erfahrungsberichte, die besagen, dass man manche Bewegungen in VR nicht machen sollte, also dass man dem Benutzer die Freiheit nehmen sollte?

Axel: Jein, prinzipiell sage ich so... Beschleunigungen sind immer schwierig. Schnelle Kurven, alles was halt irgendwie sich davon entfernt, was ich jetzt in Wirklichkeit mache. Man kann gegensteuern mit Motion seats, wenn die gut gemacht sind dann kann man da bisschen helfen, dass das besser funktioniert aber es sind Bewegungen, die sind halt einfach nicht gut für uns.

Also wenn man denkt in Vrei gibt es ja diese Motion Seats für die Autorennen... es funktioniert super in den Kurven und wenn man halt gegen eine Wand fährt dann überträgt es dem Motion seat auch aber es wird einem auch trotzdem schlecht, weil ja... gegen die Wand fahren ist halt ein Erlebnis, dass man halt ja... (Lachen)

Aber es gibt dann wieder... kann ich jetzt aus Berichten von Spielen, hab ich selber nicht erfahren aber aus Berichten von Spielen, wo man jetzt bei schnelleren Bewegungen das Sichtfeld einschränkt. Das heißt, man hat jetzt nicht mehr so einen Oculus 110 Grad Sichtfeld, sondern es wird eingeschränkt, so dass man mehr das Gefühl hat, man schaut auf einem Bildschirm. Dadurch ist dann das Gefühl von Motion Sickness auch bisschen eingeschränkt. Also da gibt es wieder Tricks mit denen man arbeiten kann um dann wieder doch diese Art von Bewegung so zu kontrollieren.

Arian: hast du mir mal gesagt, dass, wenn man Seitenbewegungen macht, wird es einem schlecht.

Axel: Bei dem BUWOG-projekt hat man nicht mit 3D Objekten gearbeitet, sondern mit Projektionen auf Kugeln und das funktioniert gut, wenn man sich halt um die eigene Achse dreht aber sobald man anfängt sich auf die Seite zu Lähnen, dann bewegt sich halt der Raum mit einem mit. Sprich die Bewegung, die ich im realen mache, wird nicht übersetzt in das was ich sehe und ja... das ist halt ein Problem.

Arian: Wie ist denn das? Macht es in VR von der Benutzerfreundlichkeit einen Unterschied, wenn man zum Beispiel, sagen wir mal Objekte bewegen will, verschiebe ich es einfach auf dem Boden oder habe ich irgendwie so eine Suchleiste und ich kann es mit Drag and Drop irgendwo hinziehen oder ich mache einfach ein Fade und es ist auf einmal von hier, dann dort oder so. Macht es einen Unterschied oder geht man einfach nach dem Geschmack und wie sieht es besser aus? Und so wünsche ich es mir einfach irgendwie bei diesem Projekt.

Axel: Habe ich wenig Erfahrung leider. Aber bei unseren Applikationen, die wir bis jetzt gemacht haben, sie sind eigentlich da sehr simpel und insofern jetzt ... genau... wie man jetzt auf Datenbanken zugreift und Objekte mit einfügt in einer Szene.

Arian: Bei Hoho zum Beispiel, sieht man, dass man den Stuhl zum Beispiel ziehen kann aber heben, kann man den nicht. Die Achse funktioniert zum Beispiel nicht

Axel: Ich kann halt auch nur sagen, was ich kenne aus... Oculus Home, das gibt es auch seit ein paar Monaten, der Home Screen ist jetzt tatsächlich ein Raum, den ich selber verändern kann. Musst du dir mal anschauen. Die haben zum Beispiel auch das Problem, dass ich aus meinem Menü oder Library quasi Objekte einführen kann, Tische hinstellen kann und... weiß nicht, was gibt es noch, Sessel, Kerzen und Bilder an die Wand hängen und alles Mögliche ja... da gibt es Zwei Interaktionen. Eine ist halt, ich kann Objekte, also ich kann mich hinbewegen zu einem Objekt und es angreifen und dann es so wie in der realen Welt machen würde und dann irgendwo hinstellen und dann da wirkt dann auch die Physics, also man hat dann Physics, also wenn ich die Kerze schief hinstelle, dann fällt sie um und liegt halt da. Was natürlich eine sehr intuitive natürliche Art und Weise ist die Objekte zu bewegen, nur kompliziert wird es halt, wenn man einen größeren Raum hat und dann ein Sofa verschieben möchte der halt in einem 10 mal 10 Meter Raum ist und ich möchte es von einer Ecke zum anderen Schieben. Das heißt, wenn ich jetzt hingehere und es händisch mache dann ist es halt ein Hin und her zwischen, ich greif das Sofa an, teleportier mich wieder einen Meter, dann zieh ich es mit. dann wird es halt zu so einer ... (lachen) Arbeit, wie es halt im realen Leben wäre. Aber das ist dann wahrscheinlich auch nicht zielführend und auch nicht zeiteffizient. Gut, sie haben das jetzt so gelöst, dass man das Sofa zum Beispiel wählen kann, das man mit Zeigefinger hindeuten kann, dann gibt es über diesem Joystick eine Möglichkeit, dass man das Objekt entlang der Bodenebene verschiebe, über dem Boden schiebe. Das ist bei weitem nicht so exakt, wie als wenn ich das mit der Hand ein paar Centimeter zur Seite schiebe und man muss halt dann mit Beschleunigungen arbeiten usw. Aber es ist gut, dass es die beiden Variationen gibt. Perfekt ist es wahrscheinlich auch nicht, also das kann man sich halt noch ausarbeiten.

Aber ja, die 2 Ansätze kenne ich schonmal und das macht auch Sinn, dass es zum Beispiel auf Ebenen wie den Boden oder Wände Snappen auch.

Arian: Naja, das sind halt Sachen, die an der Realität nicht angepasst sind aber in VR einfach angenehmer und einfacher und schneller zu verstehen sind. Auch sogar für jemand der das exakt an der Realität angepasst machen will. So würde es einem auch viel schneller langweilig werden, wenn man es wie in der Realität alles hin und her schieben würdest. Also ja es ist eine Entscheidungsfrage, ja da möchte ich es exakt wie in der Realität machen und da nicht.

Gibt es eigentlich immer noch dieses riesengroße Motionsickness Problem oder ist das Problem fast gelöst mit den Templates, die es schon gibt in allen Game Engines, die die versuchen das Problem zu lösen? Ja, weil man nimmt immer ein Template und arbeitet von dort weiter oder?

Axel: Ja, das vermeidet sicher mal einen Haufen Fehler, die man machen kann, wenn man dieses Grundgerüst falsch aufbaut.

In dem Effekt, auch wenn das steht, auch wenn man jetzt auf eine Erfahrung baut, die einfach zu wild ist und zu sehr von den realen Eindrücken abweicht und dann halt auch für das falsche Publikum, also vom falschem Publikum angewendet wird, dann ja, gibts das Thema natürlich immer noch. Wie schon vorher erwähnt es ist schon so, dass man sich daran gewöhnen kann und Motionsickness zu einem gewissen Grad auch überwinden kann, aber , ja... derzeit sehe ich das jetzt nicht, dass komplett verschwinden wird. Entwickler werden sicher besser und aus Erfahrungswerten oder wenn sie sich einlesen in das Thema, wissen, was sie ihrem Publikum zumuten können und damit wird es glaube ich immer weniger... sagen wir mal so... Fettnäpfchen geben... wo die Leute einsteigen, aber... ja... es wird auf jeden Fall immer was sein, was man mit beachten muss.

Arian: Nochmal, die letzte Frage ist, das was ich eigentlich auch immer fragen muss ist halt immer, es gibt immer in einem Bereich ein Software oder ein Workflow, was halt optimal ist: 100 % gibt es nicht aber es gibt halt immer etwas, was sich einfach besser bewiesen hat. Gibt es für dich sowas, so was wie sage ich mal so: Ich arbeite lieber in Unreal Engine oder lieber in Unity, oder nein um Gottes Willen, nie in Unreal Engine oder sowas. Ich rede jetzt nur von Architekturvisualisierung.

Axel: Also ich persönlich, ich habe mit Unreal angefangen. Es ist für mich interessant zu sehen, dass viele Module, so wie die Shader und auch das ganze Blueprintssystem, dass das Ganze eine Geschichte ist, dass ich als nicht Programmierer, nicht Developer, auch im Großen und Ganzen, ganz gut verstehe und das hilft mir, selber schnell mal einen Prototypen zu machen oder auch im Gespräch dann mit einem Developer zu sehen, was vielleicht eine Sachen ist, die man... also wie man Sachen anpassen kann ... um einfach irgendwie eine gleiche Sprache zu sprechen, als sagen wir mal, wenn man jetzt eine „line Coding“ macht.

Aber im Prinzip, nein, glaube ich nicht, dass es ein Tool gibt, dass DAS Ding ist.

Arian: Das heißt im Prinzip, es gibt auch nicht wirklich ein so DER Workflow, der beste, der immer funktioniert, bei Architekturvisualisierung in dem Fall.

Axel: Es ist auch so, wir sind auch nicht auf Archtekturvisualisierung spezialisiert. Insofern kann ich es nicht sagen. Wir haben keine Fixfertige Pipeline aufgebaut für das. Wir haben Prototype gemacht, natürlich einiges gelernt, aber ich mute mir

jetzt nicht zu, Maße mir nicht an zu sagen, ich weiß jetzt wie der perfekte Workflow auszusehen hat.

Arian: Der Grund warum ich frage... Es gibt zum Beispiel nicht selten Tutorials, die man halt schaut und die Leute, die dann sagen, Ich komm nicht mal auf die Idee Assets von Maya in Unreal Engine also für Unreal zu machen, weil 3DS-Max macht es halt irrsinnig einfacher. Du hast einfach Plug ins die, die dafür gemacht sind, dass die Assets so angepasst werden, dass du die in unreal importieren kannst, oder so... Ich kann es nicht wirklich bestätigen. Ich kenn mich mit 3DsMax überhaupt nicht aus

Axel: Ja, das kann ich dir auch nicht sagen.

Arian: Ja, das ist halt so ein Ding, wo du dir denkst, dass musst es wohl ein Workflow geben. Wenn man an Architekturvisualisierung denkt, denkt man okay, ich würde jetzt nicht auf die Idee kommen, Maya jetzt rein zu bringen ins Spiel.

Axel: Was ich halt aus Gesprächen schon rausgehört habe ist, dass Max halt doch mehr einen Fokus auf Architektur hat und Maya halt mehr in Gaming und Animationbereich angewendet wird. Aber ja, wie gesagt, das sind halt 3D Tools, mit denen man sehr wohl in beiden, auch in Cinema kann man alles machen, auch in Blender. Die Frage ist halt, wie wohl fühlt sich jeder Artist mit dem Programm und mit den einzelnen Workflows.

Arian: Vielen lieben Dank für deine Zeit.

Axel: Gerne. Ich hoffe, ich konnte helfen

C. Interview mit Christian Friesenegger, 3Motion Virtual GmbH

Christian: Ich bin der Gesellschafter und Geschäftsführer von 3Motion Virtual GmbH. Dieses Unternehmen, dass aus 2D Plänen 3D Modelle macht und verschiedene Nutzungsmöglichkeiten daraus generiert, virtuelle Fotos, Videos 360 Grad Rundgänge und interaktive 3D Rundgänge.

Wir nutzen Grundrisse und aus den Grundrissen generieren wir die 3D-Modelle. Also aus dem Grundriss und Schnitt, aus dem bauen wir ein Modell und richten es danach ein.

Arian: Also es ist nicht nur so ein Interior Archviz, sondern wirklich Wände, Türen usw.

Christian: Nicht nur Interior. Es ist wirklich eine Mischung zwischen, wenn man es mit der realen Welt vergleicht, von einem konstruktiven Bau und dem Interior Design, dass danach kommt.

Arian: Kann ich es mir so vorstellen, dass jemand mit einem Plan herkommt und sagt, ich möchte gerne die Wand dorthin haben und man schaut sich an, wie es aussieht.

Christian: Das geht grundsätzlich schon, wenn man die Wände vorher als beweglich vordefiniert hat. Also wenn sie keine tragenden Wände sind. Tragend ist vielleicht der reale Begriff dafür. Man muss immer vordefinieren welche Teile im Modell statisch und welche beweglich sein sollen.

Arian: Jetzt komme ich zu der Forschungsfrage. Welche Methoden der Interaktion haben sich für Menschen, die nie etwas mit VR am Hut hatten haben sich am besten bewährt?

Christian: Aus meiner Erfahrung sehr uneinheitlich. Es ist schon so, dass man merkt, dass die Leute eine gewisse Angewöhnungszeit brauchen, bevor sie wissen wie sie was bedienen.

Ich glaube bei der Interaktion auch, dass der reale Vergleich das entscheidende ist. Man denkt sich, okay, ein Bett ist schwer, das muss ich nicht unbedingt anheben können. Aber bei einer Tasse sollte es schon funktionieren.

Arian: Okay, kann man es so formulieren, dass man aus der Erfahrung von der realen Welt dann ein Bild und eine Vorstellung von Gewicht hat und auch wie es in VR funktionieren sollte. Aber wie simuliert man in VR eigentlich Gewicht?

Christian: Gar nicht momentan. In dem Anwendungsbereich, wo wir tätig sind, ist Gewicht keine relevante Größe. Da geht es darum, ob ich es verschieben kann... Ob es sich schwer anfühlt oder nicht, das ist weniger wichtig. Was für uns viel wichtiger ist, ist zu überlegen, wie kann ich die realen Bedingungen, und das reale Verhalten virtuell abbilden.

Das heißt, zum Beispiel, wenn ich mir vornehme die Textur von diesem Tisch zu ändern, dann sollte jetzt kein Menü aufpoppen und mich aussuchen lassen. Das würde die Immersion wieder zerstören, sondern ich habe vielleicht in der einen Hand ein Tablet und da sehe ich die Farben und ändere sie beim Klicken.

Ein Tablet verstehe ich von der realen Welt und da weiß ich was ich tun soll. Das entspricht mittlerweile dem Verhalten der realen Welt. Und ich glaube, dass gerade bei Interaktionen solche Überlegungen das Kernthema sein werden. Wir überlegen auch, dass wir halt, dass wir halt an der Wand ein Monitor aufhängen, wo man halt was auswählen kann. Weil... Touch Screen Monitore kennt man.

Arian: In VR sagt man, dass... wir haben ein Spektrum zwischen sehr realistisch und sehr abstrakt und irgendwo dazwischen bewegen sich alle Projekte. Sie überlegen und versuchen bei den Projekten mehr Richtung realistisch zu gehen, weil Sie der Meinung sind, dass die Leute es besser verstehen.

Christian: Ja, schon. Also Anwender, die bei uns die Brille aufsetzen, denen muss ein möglichst hohe Immersion gegeben werden. Und je vergleichbare es mit der Natur ist, desto leichter fallen sie rein. Wenn ich jetzt das Ganze aus einer abstrakten Arbeitssichtweise vielleicht sehe. Das heißt, ich möchte in VR etwas arbeiten, ist es besser, wenn ich die abstrakten Möglichkeiten, die es gibt, besser nutze. Ich glaube, dass es einerseits auf die Anwendung ankommt, wer ist meine Zielgruppe und wenn man es bisschen besser an jetzt anpasst, wie gewohnt sind wir zu den VR-Inhalten.

Wenn wir in Zehn Jahren über VR reden, werden viele Leute es ganz gewöhnlich nutzen, und da wird sich die Realität sicher ändern. Da wissen die Leute alles, in VR habe ich keinen Bildschirm, da verschiebe ich einfach alles. Das wirkt real für sie schon. Jetzt sind wir in der Phase wo alles so neu ist. Das ist diese Phase der Realitätsnähe und die Vergleichbarkeit ist das, womit sich die Leute leichter tun. Leute gehen bei uns nicht auf Wände. Sie gehen immer herum, obwohl es eigentlich egal ist. Zumindest 90 Prozent der Leute kommen nicht auf die Idee über die Terrasse raus zu gehen. Auch dieses Gefühl, ist das, was für uns wichtig ist, weil gerade in dem Bereich wo wir tätig sind, dass wir einen emotionalen Eindruck von der Immobilie zu geben, im Vergleich zu einem Plan lesen und sich vorstellen, wie es ist... Für uns ist diese Immersion ein wichtiges Thema. Ich glaube, daran unterscheidet man welchen Plan man hat. Ich habe es gesehen, ein Mitbewerber von uns hat eine menschliche Figur in die Welt reingestellt. Die Figur war auch nicht so fotorealistisch... Man hat gemerkt, nein, es funktioniert nicht.

Arian: Es gibt die Annahme, dass die Interaktion für die Benutzer in VR leichter ist, wenn sie eine größere Ähnlichkeit zu der Realität haben. Also sie Bestätigen die Aussage?

Christian: Ja, Wo auch immer wir sind. Du musst in der virtuellen Realität immer möglichst genau bauen. Das ist bei interaktiven Themen manchmal schwierig oder komplex. Wenn ich zum Beispiel eine Tasse nehmen möchte und ich in die Tasse greife, ist es komisch. Die Kollisionen dürfen auch nicht zu weit weg sein, wenn ich greife, denn dass ist es wieder komisch. Ein spannendes Tool wäre jonglieren. Wenn ich einen Ball hochwerfe, wann habe ich ihn denn gefangen? Wie löst man hier das Problem des Greifens. Das da wo ich sage, dass Interaktion mit der realen Welt zusammenpassen wird. Mich interessiert diese lebensnahe Definition von Objekten. Diese Lebensnahe Erfahrung ist das Kernelement und auch wichtig für die Immersion.

Arian: Wie weist man den Benutzer in VR darauf hin, mit welchen Objekten er interagieren kann und mit welchen nicht?

Christian: Also wir machen es aktuell mit... mit Highlighten... also mit leicht leuchtenden Kanten. Das heißt, dass man sie bewegen kann.

Arian: Es ist aber ein Hinweis. Es ist halt eine grafische Änderung.

Christian: Ja es ist ein Hinweis. Was mir aber persönlich zum Beispiel gar nicht gefällt sind da wo so Texte auf die Objekte stehen. Sie sind Immersionsbrechend für mich. Also Bilder, die mit normalen Bildern nachempfunden sind. Also alles was ich mir real auch vorstellen könnte... das ist gut...

Also wie man die Leute auf Interaktionsmöglichkeiten hinweisen kann... Ich glaube, es gibt keine Patentlösung dazu, aber ich möchte einen Vergleich dazu anstellen, wie wissen Sie wo sie draufklicken?

Arian: Erfahrung?

Christian: Exakt. Das wird in VR auch werden. Es wird eine gewisse Zeit brauchen, bis die Leute soweit kommen. Es werden sich dann auch Regeln und Systematiken herausbilden, wie etwas funktioniert. Leute werden in zehn Jahren wissen, wenn sie einen roten Kleber irgendwo sehen, was das bedeutet... Ich glaube nicht, dass es etwas ist, was von heute auf morgen funktionieren wird, weil ich intuitiv gewisse Dinge nicht machen werde. Intuitiv werde ich keine Wand verschieben. Außer ich habe eine Notwendigkeit. Also ich glaube, dass alles wo ich eine Notwendigkeit drin erkenne, werde ich versuchen zu machen.

Arian: Das heißt also, dass ich dann immer ein Ziel brauche. Wenn ich weiß, was ich erreichen möchte, dann werde ich versuchen alles zu tun.

Christian: Und ich glaube, dass Dinge, die wir in der realen Welt nicht gewohnt sind zu tun, Wände verschieben als Beispiel... dafür werden Systematiken geben, die man über die Jahre und Erfahrung lernt, dass das geht und wie es geht.

Vor 15 Jahren hat man mit einem Tablet in der Hand auch nicht viel machen können. Ich glaube, dass diese Lernphase auch hier nötig sein wird.

Arian: Das heißt eigentlich, wie die Leute sich in VR fühlen, wird in zehn Jahren ganz anders sein. Es ist viel zu neu für sie.

Christian: Ja, genau. Was ich mir gut vorstellen kann, ist das man etwas in einer Umgebung hat, was natürlich scheint, was dir Hinweise gibt... Und das zeigt dir, so und so kannst du die Wand verschieben. Ich glaube, dass diese Prozesse die ersten sein werden.

Arian: Wie ist es bei Ihren Applikationen? Machen Sie es mit Teleportation?

Christian: Teleportation. Die Räume, die wir abstecken können, sind für die Wohnung die wir herzeigen meistens zu klein. Also wenn wir 100 qm haben, können wir sie nicht normal begehen. Das heißt wir müssen eine Lösung finden, wie wir uns zwischen den Räumen fortbewegen können.

Arian: Und redirected Walking wäre viel zu aufwändig zu programmieren. Oder?

Christian: Das wird die Zukunft sein glaube ich. Ja. Ich habe auch vor kurzem gelesen, dass sie es total optimiert haben. Ein Engländer glaube ich war das. Wenn das perfektioniert ist, wird das ein riesiger Meilenstein sein. Wir machen das mit Eye Tracking, das was ich gelesen habe. Die verschieben nicht automatisch, die gehen nach Eye Tracking und verschieben das nach Eye Tracking.

Arian: Aber dann freies Gehen ist immer noch nicht möglich, sondern man denkt, dass man es tut.

Christian: Genau.

Arian: benutzen Sie unreal Engine für ihre Applikationen?

Christian: Als wir angefangen haben, war die Darstellungsqualität so viel schlechter. Das war vor 3 Jahren.

Arian: So, ich habe jetzt alle Fragen durch. Das war sehr hilfreich. Danke sehr. Gibt es etwas, was Sie mir sagen können? Etwas, was mich retten kann?

Christian: Ich glaube, dass es gerade der Bereich Innenarchitektur ein riesigen Thema sein wird. Gerade der Beginn. Ich glaube auch, dass in der Zukunft Möbelhäuser unnötig werden. Die Haptik könnte vielleicht ein Thema sein. Interaktion... Aus meiner Sicht hat es momentan die Schwäche, dass es viele Unterschiede zur realen Wahrnehmung gibt.

Arian: Vielen lieben Dank.

Christian: Gerne.