



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Steffen Schweneker

*Storytelling in VR – Analyse der Wirkung von Interaktion und
Perspektive*

Steffen Schwenecker

Storytelling in VR – Analyse der Wirkung von Interaktion und Perspektive

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung

im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Philipp Jenke
Zweitgutachter : Prof. Dr.-Ing. Andreas Meisel

Abgegeben am 28.05.2018

Steffen Schwenecker

Thema der Bachelorarbeit

Storytelling in VR – Analyse der Wirkung von Interaktion und Perspektive

Stichworte

Virtuelle Realität, Immersion, 3D Grafik, Storytelling

Kurzzusammenfassung

Dieser Bachelorarbeit untersucht verschiedene Ansätze, eine Geschichte in einer virtuellen Umgebung darzustellen. Um dieses Ziel zu erreichen wird ein Szenario beschrieben, welches den Grundstein legt, der in vier verschiedenen Fällen aufgegriffen wird. Diese Fälle setzen sich aus zwei Arten der Perspektive sowie zwei Interaktionsstufen zusammen. Es wird die First-Person Perspektive sowie die Vogel-Perspektive (eine Art der Third-Person Perspektive) behandelt. Diese Perspektiven werden gepaart in einer interaktiven sowie in einer nicht interaktiven Version des Szenarios und bilden somit die vier verschiedenen Fälle. Anschließend werden mittels einer Nutzerstudie die Auswirkungen dieser Fälle analysiert.

Steffen Schwenecker

Title of the paper

Storytelling in VR – Impact analysis off interaction and perspective

Keywords

Virtual reality, immersion, 3D graphics, storytelling

Abstract

This bachelor thesis examined different approaches of storytelling in a virtual environment. To achieve this goal, a scenario will be described to lay the foundation, which will be picked up by four different cases. The cases combine two kinds of perspectives together with two grades of interactions. The first-person perspective and the birds-view perspective (a variant of the third-person perspective) will be discussed. These perspectives will be accompanied with an interactive and a non-interactive version of the scenario to form the four cases. Finally, there will be a user study to analyze the effect of those cases.

Inhaltsverzeichnis

1	Einleitung	6
1.1	Motivation.....	7
1.2	Ziel	7
1.3	Herausforderung.....	7
1.4	Aufbau	8
1.5	Anwendungsbeispiele	8
1.5.1	Unterhaltung.....	9
1.5.2	Bildung	10
1.5.3	Marketing.....	11
1.6	Geschichtsüberblick: VR Head Mounted Display.....	12
2	Grundlagen	15
2.1	Stand der Technik.....	15
2.2	Storytelling in VR.....	16
2.3	Wahrnehmung in VR: Immersion, Präsenz und Kohärenz.....	17
2.4	Simulator Sickness.....	19
2.5	Vier-Sichten-Modell	20
3	Konzept.....	22
3.1	Story	22
3.1.1	Perspektiven.....	23
3.1.2	Interaktion.....	25
3.2	Grafik: Designentscheidung	25

3.3	Steuerung	26
3.4	Szenenablauf	27
4	Implementierung.....	30
4.1	Verwendete Technologien	30
4.2	Grafik: Erstellung der Assets	31
	Auswahl des Motives	31
4.3	Skripte und Komponenten	35
4.3.1	Komponenten des Autos.....	35
4.3.2	Steuerung und Interaktion.....	40
4.4	Spielablauf.....	43
4.4.1	Der Anfang	43
4.4.2	Die Mitte	45
4.4.3	Das Ende.....	48
5	Nutzerstudie.....	50
5.1	Versuchsbeschreibung	50
5.2	Ziel	50
5.3	Auswertung	51
5.4	Zusammenfassung	57
6	Abschluss	60
6.1	Fazit	60
6.2	Ausblick	60
7	Literaturverzeichnis	61
8	Anhang.....	63

1 Einleitung

Virtuelle Realität ist ein Begriff, der sich mit der Zeit stark gewandelt hat. Wo vor vielen Jahren noch Gemälde oder bewegte Bilder den Eindruck von Realität erzeugt haben [1] sprechen wir heute von virtueller Realität, wenn wir in eine virtuelle Welt eintauchen können, in der wir uns umsehen, interagieren und mittlerweile sogar fühlen und riechen können¹.

Dieser Wandel bietet nicht nur mehr Möglichkeiten, sondern auch viel mehr Dinge, die bedacht werden müssen. Daher ist es besonders wichtig sich die Frage zu stellen welche Auswirkungen Konzepte und Funktionen, einer virtuellen Welt, auf die Anwender haben.

Lange Zeit war VR nur eine experimentelle Idee, die viele Schritte durchlebt hat und größten Teils nur für Entwickler zugänglich war.

Mit dem Erscheinen von konsumentenfreundlichen VR-Brillen, wie die Oculus Rift, HTC Vive oder PSVR, haben auch Nicht-Entwickler Zugang zur virtuellen Welt und der Bedarf an VR-Software steigt seitdem stetig an².

Darum ist es besonders wichtig, dass Produkte den Kunden ein möglichst angenehmes und immersives Erlebnis bieten.

Da jeder VR anders wahrnimmt ist es sinnvoll eine Szene in einer virtuellen Welt aus unterschiedlichen Perspektiven darzustellen um sich deren Auswirkung klar zu machen. Um dieses Ziel zu erreichen wird sich diese Arbeit an ein Konzept von Devon Dolan anlehnen, welches ein Szenario in vier verschiedene Sichten aufteilt³.

¹ Quelle: <https://feelreal.com>

² Quelle: <https://www.canalys.com/newsroom/media-alert-virtual-reality-headset-shipments-top-1-million-first-time>

³ Quelle: <https://medium.com/@devon.michael/redefining-the-axiom-of-story-the-vr-and-360-video-complex-bee3c20d69df>

1.1 Motivation

Virtual Reality findet in unterschiedlichsten Bereichen Anwendung. Da nicht jedes Anwendungsfeld den Inhalt der VR-Anwendung gleich darstellen kann oder will, müssen verschiedene Ansätze gefunden werden um eine virtuelle Welt zu präsentieren. Auch innerhalb eines Anwendungsfeldes können VR-Anwendung auf verschiedenste Arten dargestellt werden. Dabei stechen zwei Punkte deutlich heraus auf die eingegangen werden müssen: Perspektive und Interaktion. „Aus welcher Sicht soll der Anwender meine virtuelle Welt betrachten?“ oder „Wie viel Interaktion ist für meine Anwendung nötig?“ sind Fragen die man sich beim Entwickeln einer VR-Anwendung stellen muss. Bei Spielen stellt Interaktion eine wichtige Komponente dar. Doch wie ist es beispielsweise bei Filmen in einer virtuellen Umgebung?

Ich möchte mit dieser Arbeit untersuchen, welche Auswirkung das Zusammenspiel von Perspektive und Interaktion auf verschiedene Anwender hat.

1.2 Ziel

Das Ziel dieser Arbeit ist eine Anwendung zu entwickeln, die eine virtuelle Umgebung auf vier verschiedene Arten darstellt. Dafür werden zwei unterschiedliche Perspektiven (First-Person und Third-Person) implementiert. Zusätzlich werden diese Perspektiven in jeweils einem nicht interaktiven und einem interaktiven Szenario aufgeteilt. Anschließend sollen diese Szenarien mithilfe einer Nutzerstudie untersucht werden, um festzustellen, welche Auswirkung diese auf die Anwender haben und in welchen Anwendungsbereichen diese zum Einsatz kommen könnten.

1.3 Herausforderung

Um das Ziel zu erreichen muss zunächst ein Szenario entwickelt werden, in dem die Perspektiven und Interaktionen sinnvoll integriert werden können. Dazu muss sich darüber Gedanken gemacht werden, wie verschiedenen Perspektiven dargestellt werden und welche Interaktionen dem Spieler geboten werden können. Da die Anwendung für jeden zugänglich sein soll, dürfen die Interaktionen nicht zu komplex ausfallen und sollten offensichtlich dargestellt sein. Auch muss geklärt werden, wie der Spieler auf Schlüsselemente aufmerksam gemacht wird.

1.4 Aufbau

Die Arbeit setzt sich aus sechs Kapiteln zusammen. Das erste Kapitel, die Einleitung, dient dem Leser einen Einstieg in das Thema dieser Arbeit. Im zweiten Teil, den Grundlagenteil, wird der Leser an Themen herangeführt, die zum verstehen dieser Arbeit wichtig sind und zusätzlich einen tieferen Einblick in das Themengebiet bietet. Im darauffolgenden Kapitel wird das Konzept der Arbeit beschrieben. Hier werden die Herangehensweisen und der Entwurf des Projekts dargestellt. Das vierte Kapitel beschäftigt sich mit der Implementation des Konzepts und zeigt die verschiedenen Arbeitsschritte, die nötig waren um das Projekt umzusetzen. Im fünften Kapitel wird die Durchführung und Auswertung einer Nutzerstudie beschrieben, die im Laufe der Arbeit stattgefunden hat. Im letzten Kapitel wird ein Fazit zu der Arbeit gezogen und ein Ausblick auf die zukünftige Entwicklung von VR und dem damit zusammenhängenden Storytelling gegeben.

1.5 Anwendungsbeispiele

Virtual Reality wird mittlerweile in vielen Bereichen eingesetzt. Ein großer Bereich ist die Unterhaltungsbranche. Doch auch in den Bereichen Medizin, Militär, Bildung wird VR schon länger angewendet [2]. Da diese Arbeit im Bereich des Storytellings angesiedelt ist, schränke ich die Anwendung auf die für dieses Thema am meisten relevanten Bereich ein:

- Unterhaltung
- Bildung
- Marketing

1.5.1 Unterhaltung

Gerade in den letzten Jahren, mit dem Erscheinen der Oculus Rift und HTC Vive, ist die Anzahl und Qualität der Unterhaltungsapplikationen stark gestiegen. Immer mehr Entwickler steigen in die VR-Entwicklung ein. Auch große Entwickler wie Bethesda Game Studios arbeiten an VR Umsetzungen für AAA-Titel wie Fallout 4 (Siehe Abb. 2.1 a) oder Skyrim.

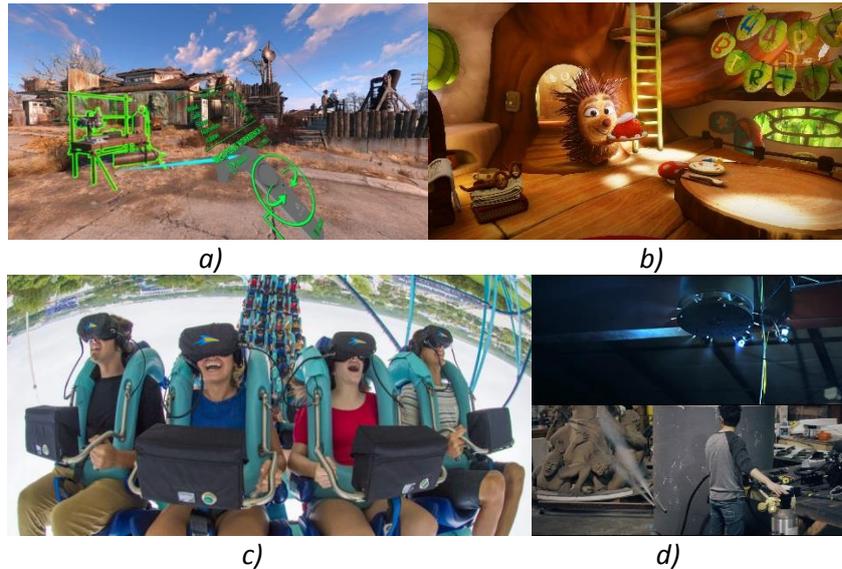


Abbildung 2.1: **a)** Baumenü in Fallout 4 VR mit der HTC Vive

Bildquelle: <http://www.pcgames.de/Fallout-4-Spiel-18293/News/virtual-reality-htc-vive-bundle-rollenspiel-endzeit-1240218/galerie/2749456/>,

b) Henry Kurzfilm produziert von Oculus Story Studio

Bildquelle: <https://www.oculus.com/story-studio/films/henry/>,

c) VR Achterbahn Unleash the Kraken im Seaworld Orlando

Bildquelle: <http://www.insidethemagic.net/2017/06/seaworld-orlandos-new-kraken-unleashed-vr-experience-enhances-old-thrills-high-tech-visuals/>

d) Mixed Reality Technik im VR-Themenpark The Void

Bildquellen: <https://www.youtube.com/watch?v=cML814JD09g>, <https://www.thevoid.com>

Nicht nur in der Spieleindustrie schreitet der Einsatz von VR voran. Auch Animationsfilme, wie Henry (siehe Abb. 2.1 b), wurden mit großem Erfolg produziert. Mit dem langsamen aber stetig anschwellenden Markt werden wir in Zukunft vielleicht sogar Filme in voller Länge in VR genießen können.

In einigen Freizeitparks wird VR mittlerweile als zusätzliche Spannungskomponente zu den Fahrattraktionen genutzt. Die Achterbahnfahrt wird mit einer Fahrt durch eine virtuelle Welt mittels HMDs erweitert (siehe Abb. 2.1 c)⁴.

Mixed Reality ist die Verbindung zwischen realer Welt und virtueller Welt. Dies wird durch ein Kontinuum nach Paul Milgram verdeutlicht [Abb. 2.2]. Wie in dieser Abbildung zu sehen befindet sich die Mixed Reality zwischen einer echten und einer virtuellen Umgebung und nutzt Elemente aus beiden Umgebungen.

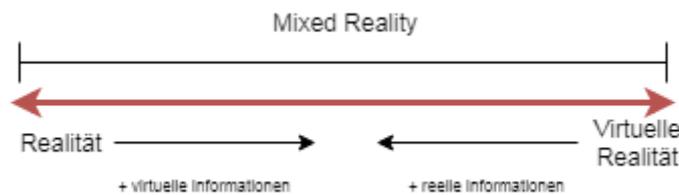


Abbildung 2.2: Reality-Virtuality Continuum

Originale Bildquelle: <http://media2mult.uos.de/pmwiki/fields/cg-II-09/index.php?n=MobileGraphicsII.RealityVirtualityContinuum>

Mit diesem Konzept hat „The Void“, 2017 ein modernen VR-Themenpark eröffnet. Hier wird eine physische Umgebung mittels VR Leben eingehaucht. Es werden interessante Techniken verwendet, um die Immersion zu steigern, wie Wassersprinkler, um Regeneffekte zu erzeugen [siehe Abb. 2.1 d].

1.5.2 Bildung

Ein weiterer großer Teil der VR ist der Bereich Training und Aus/Weiterbildung. Im Militär werden Simulationsübungen mit Hilfe von VR-Ausrüstung durchgeführt, um einem echten Einsatzzenario möglichst nahe zu kommen und Soldaten auf mögliche Situationen vorzubereiten. Aber nicht nur im Militär kann und wird VR zu Bildungszwecken genutzt. Unter anderem auch im Pilotentraining und in der Medizin.

Training in VR bietet Möglichkeiten, die in einem realen Training nicht realisierbar sind, da sie zu gefährlich oder aufwendig sind. So können Operation in einer simulierten virtuellen Umgebung geübt werden, die sonst üblicherweise nur durch Lehrvideos, Plastikmodellen oder Patienten trainiert werden konnten [3].

⁴ Quelle: <https://vrscout.com/news/seaworld-kraken-vr-roller-coaster/>

Reales Training Vs. Virtuelles Training

- **Risiko**
Reales Training ist oft mit einem Risikofaktor verbunden. Das virtuelle Training entfernt diese Komponente komplett. So können auch riskante Praktiken, wie schwierige Operationen oder Flugmanöver, risikofrei geübt werden. Zwar ist das virtuelle Training kein kompletter Ersatz, aber es senkt den Bedarf und das Risiko an realen Übungen, durch Erfahrung im virtuellen Training.
- **Kosten**
Kosten von realen Übung sind meisten sehr hoch. Bei virtuellen Simulationen sind die Kosten einer Trainingsstunde vernachlässigbar.
- **Ausbilder**
Der Bedarf an Ausbildungskräften kann beim virtuellen Training gesenkt werden da, durch interaktive Lehrübungen der Nutzer ein direktes Feedback bekommt was er zu machen hat.

1.5.3 Marketing

Auch im Marketingbereich kann VR sinnvoll genutzt werden. So kann beispielweise ein potentieller Käufer eine Yacht von zuhause aus über VR betreten und bekommt ein Eindruck von dem Produkt in voller Größe [4]. 2017 startete Audi das erste vollwertige Virtual Reality Projekt für Autohäuser. Die mit VR-Brillen ausgestatteten Autohäuser ermöglichen es dem Kunden sein individuell konfiguriertes Auto im Detail zu betrachten⁵.

Auch kann spielerisch ein Produkt beworben werden. Disney-Pixar erstellte für ihren neusten Film Coco eine VR Applikation, die durch Konzeptgrafiken und Filmausschnitten bei den Spielern Interesse an dem Film wecken soll⁶.

⁵ Quelle: <https://www.audi-mediacycenter.com/de/pressemitteilungen/audi-startet-virtual-reality-im-autohaus-9270>

⁶ Quelle: <https://www.vrnerds.de/pixars-soziale-vr-app-coco-vr-fuer-oculus-rift-ist-da/>

1.6 Geschichtsüberblick: VR Head Mounted Display

1986: Sword of Damocles

Das erste VR Head Mounted Display (HMD), Sword of Damocles, wurde im Jahre 1968 von Ivan Sutherland entwickelt. Das Positionstracking des Kopfes wurde mit zwei unterschiedlichen Sensoren realisiert. Ein mechanischer Sensor, welcher an der Decke montiert wurde und direkte Verbindung zum HMD hatte und ein Ultraschall Sensor, der mittels drei Transmitter die Position des Kopfes misst. Durch den mechanischen Sensor war die freie Bewegung jedoch auf 6 Fuß (ca. 1.8 Meter) in der Diagonalen eingeschränkt [5].

1980: Virtual Interface Environment Workstation

Mitte der 1980er wurde die Virtual Interface Environment Workstation (kurz VIEW) im Ames Research Center der NASA entwickelt. Dieses System konnte zum einen, über eine integrierte Spracherkennungssoftware via Sprachbefehle gesteuert werden, zum anderen über Datenhandschuhe. Die Datenhandschuhe konnten mittels Lichtwellenleiter an den Fingergelenken die Fingerposition erfassen. Zusätzlich war das HMD mit elektromagnetischen Erfassungssensoren ausgestattet. Diese wurden genutzt um die Position der Hände relativ zum Host System zu erfassen [6].

1995: Forte VFX1

Mit dem Forte VFX1 wurde, 1994 (erschieden 1995) auf der Consumer Electronics Show in Chicago, das erste VR HMD vorgestellt, das für den Endverbraucher erschwinglich sein sollte. Mit \$1000 lag der Preis zwar hoch, jedoch nicht außer Reichweite. Das Headset war mit zwei Epson LCD Monitoren ausgestattet und hatte 3 Achsen Tracking zum Erkennen von horizontale Drehbewegung, vertikale Neigbewegung und seitliche Rollbewegung^{7 8}.

2014: Google Cardboard

Das Google Cardboard wurde 2014 auf der Google I/O vorgestellt und ist im selben Jahr auf dem Markt erschienen. Das Google Cardboard ist ein, zum größten Teil, aus Pappe bestehender Bausatz, der zu einer Handyhalterung zusammengebaut wird. Zwei in der Halterung integrierte Linsen ermöglichen das stereoskopische Darstellen von VR Apps auf dem eingesetzten Handy. Auch das Headtracking wird über das Handy gesteuert. Dadurch ist das Google Cardboard frei von technischen Zusätzen.

⁷ Quelle: <http://www.ibiblio.org/GameBytes/issue21/flooks/vfx1.html>

⁸ Quelle: <http://www.mindflux.com.au/products/iis/vfx1-2.html>

Durch die dadurch sehr geringen anfallenden Materialkosten hat das Google Cardboard einen Kaufpreis von Rund \$10-\$15 und ist somit der günstigste Einstieg in die VR⁹ (Kosten der benötigten Hardware nicht in Betracht gezogen).

2013 – 2016: Oculus Rift

Der Erfinder Palmer Lucky stellte 2012 auf der Electronics Entertainment Expo zum ersten Mal einen Prototyp der Oculus Rift vor¹⁰. Das Projekt wurde ab 2012 über die Seite Kickstarter via Crowdfunding finanziert und erreichte damit mehr als 2.4 Millionen US Dollar von über 9500 Unterstützern¹¹.

2013 wurde die erste Iteration der Oculus Rift, das Development Kit 1 (DK1) veröffentlicht. Der Hintergedanke, der Oculus Rift, war es, ein VR-Gerät zu entwickeln, das zum Großteil der Spieleindustrie dienen sollte¹². Mit einer relativ niedrigen Auflösung von 640 x 800 pro Auge und ohne Positionstracking diente das DK1 als erster Prototyp für Entwickler.

2014 wurde die zweite Iteration, das Development Kit 2 (DK2) veröffentlicht und verbesserte einige Mängel des DK1, in Form von einem hochauflösenden Full HD OLED Bildschirm mit 960 x 1080 Pixel pro Auge, und Positionstracking. [7]. Im gleichen Jahr wurde Oculus vom Großkonzern Facebook für einen Betrag von 2 Milliarden US Dollar aufgekauft¹³.

2016 wurde mit der dritten und aktuellsten Iteration die Customer Version 1 (CV1) veröffentlicht, welche zum ersten Mal auf den Endnutzer zugeschnitten war. Die CV1 hatte eine verbesserte Auflösung zum DK2, mit 1080 x 1200 Pixel pro Auge.

Das Tracking erfolgt über 1 bis 3 Sensoren, die Infrarot LEDs auf dem HMD erkennen und die Daten an den PC weitergeben. Mit diesem Verfahren kann die 6 DOF (Degrees of Freedom) Bewegung erfasst werden.

Zusätzlich kamen nach dem Veröffentlichungs-Termin die „Touch-Controller“, für die CV1. Mit den Controllern kann man Handbewegung zur Softwarebedienung übertragen.

⁹ Quelle: <https://vr.google.com/cardboard/>

¹⁰ Quelle: https://de.wikipedia.org/wiki/Oculus_Rift

¹¹ Quelle: https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game?ref=most_funded&ref=discovery

¹² Quelle: <http://www.monkeyinthecage.com/2012/12/25/interview-with-oculus-rift-creator-palmer-lucky-part-1/>

¹³ Quelle <https://www.forbes.com/sites/briansolomon/2014/03/25/facebook-buys-oculus-virtual-reality-gaming-startup-for-2-billion/>

2016: HTC Vive

Die HTC Vive wurde in Kooperation mit Valve produziert und ist im Jahre 2016 erschienen. Die HTC Vive nutzt, genau wie die Oculus Rift, zwei 1080 x 1200 OLED Bildschirme. Der größte Unterschied zur Oculus Rift sind die Sensoren. HTC Vive nutzt zwei Infrarot Basisstationen, die Lichtsignale an die Brille weitergeben. Die Brille gibt die Signale an den PC weiter, somit müssen die Basisstationen, im Gegensatz zu Oculus Rift, nicht mit dem PC verbunden sein. Auch mit diesem Verfahren lässt sich die 6 DOF Bewegung erfassen¹⁴. 2018 veröffentlichte HTC eine verbesserte Version der Vive, die Vive Pro. Die Vive Pro wurde mit zwei 1440 x 1600 Bildschirmen und integrierten Kopfhörern verbessert¹⁵. Zusätzlich sollen dieses Jahr verbesserte Basisstationen für das Tracking erscheinen¹⁶.

2016: PSVR

Die PSVR wurde Ende 2016 veröffentlicht und ist exklusiv für die Playstation 4 zu erwerben. Die PSVR ist zurzeit die einzige für Konsole verfügbare VR Brille und steht somit weniger in Konkurrenz zu der Oculus Rift und HTC Vive. Die PSVR ist mit zwei 960 x 1080 OLED Bildschirmen ausgestattet und nutzt ein 6 DOF Bewegungssensor System¹⁷. Durch die geringe Hardware Anforderung und dem niedrigsten Preis hat sich die PSVR, im dritten Quartal 2017, fast 500.000-mal verkauft, was 49% der Verkäufe von VR Geräten ausmachte¹⁸.

2018: Pimax

Die Pimax VR ist ein weiteres Kickstarter Projekt und wurde 2017 von 5900 Unterstützern mit 4.2 Millionen US Dollar finanziert. Pimax setzt auf weitaus bessere Auflösung und ein größeres Sichtfeld als bisherige VR Systeme.

Zwei HMDs sollen, nach einer Verzögerung, 2018 erscheinen. Das Pimax 5k mit einer Auflösung von 2 mal 2560 x 1440 Pixel und 200 Grad Sichtwinkel und das Pimax 8k mit einer Auflösung von 2 mal 3840 x 2160 Pixel und 200 Grad Sichtwinkel. Das Pimax nutzt das gleiche Tracking System wie die HTC Vive und ist mit der HTC Vive Hardware kompatibel¹⁹.

¹⁴ Quelle: <https://www.vive.com/de/product/>

¹⁵ Quelle: <https://www.vive.com/de/product/vive-pro/>

¹⁶ Quelle: <https://vr-world.com/ces-2018-steamvr-tracking-2-0/>

¹⁷ Quelle: <https://www.playstation.com/en-us/explore/playstation-vr/tech-specs/>

¹⁸ Quelle: <https://www.canalys.com/newsroom/media-alert-virtual-reality-headset-shipments-top-1-million-first-time>

¹⁹ Quelle: <https://www.kickstarter.com/projects/pimax8kvr/pimax-the-worlds-first-8k-vr-headset/description>

2 Grundlagen

2.1 Stand der Technik

In diesem Abschnitt möchte ich eine Arbeit vorstellen, die sich auf das Thema Storytelling in VR bezieht, sowie eine Arbeit, die Unterschiede zwischen First-Person und Third-Person aufzeigt. Meine Arbeit verknüpft diese Themen und erweitert diese um die Komponente der verschiedenen Interaktionsstufen.

Im Jahr 1995 wurde im EPCOT Center ein Versuch durchgeführt, der es Gästen erlaubte einen fliegenden Teppich durch die Welt von Aladdin (ein Zeichentrickfilm von Disney) zu fliegen. Das Ziel dieses Experimentes war es Autoren und Produzenten auf die VR Technologie aufmerksam zu machen. Außerdem zeigten sich einige Herausforderungen, die beim Storytelling in VR auftreten können. Die Anwendung beinhaltete einen Teil der Welt und mehrere Charaktere aus dem Aladdin Universum. Die Gäste konnten den fliegenden Teppich mit einem Controller selbst steuern und die Umgebung eigenständig erkunden. Dies stellte ein Problem für die Entwickler dar, da der Ablauf der Geschichte dem Nutzer übergeben wurde. Das Problem wurde gelöst, indem für Schlüsselszenen Eingänge implementiert wurden, die den Nutzer in eine Initialposition drängten. So konnte der Ablauf der Geschichte kontrolliert werden. Dies zeigte den Konflikt zwischen einer vordefinierten Geschichte und der nutzergesteuerten Erkundung in einer virtuellen Umgebung. Die Ergebnisse des Versuchs zeigten, dass die Nutzer am besten in der virtuellen Umgebung zurechtkamen, wenn sie eine Hintergrundgeschichte, sowie ein Ziel kannten. Außerdem wurde die Erkenntnis gezogen, dass es den Nutzern hilft, wenn sie kurz und einfach gehalten wird [8].

In einer Arbeit von Patrick Salamin, Daniel Thalmann und Frédéric Vexo wurde untersucht, welche Vorteile die Third-Person Perspektive in Virtual und Augmented Reality hat. Es wurde ein Versuch dargestellt, in dem Teilnehmer einige Aufgaben in First-Person und Third-Person Sicht bewältigen mussten. Zu den Aufgaben gehörten unter anderem ein Hindernisparcours, sowie einige Interaktionen mit Objekten. Die Ergebnisse zeigten, dass die Teilnehmer beim Hindernisparcours die Third-Person Perspektive, jedoch bei dem Interagieren mit Objekten die First-Person Perspektive bevorzugten [9].

2.2 Storytelling in VR

Virtual Reality ist weniger eine neue Art des Geschichtenerzählens, als vielmehr eine neue Art eine Geschichte zu erleben [10].

Skriptgesteuert vs. Nutzergesteuert

Das Eintauchen in eine virtuelle Welt eröffnet die Möglichkeit, dem Publikum die Steuer in die Hand zu geben. Beim Erstellen von Geschichten muss der Erzähler wählen, ob er selbst entscheidet, wie und in welchem Tempo die Geschichte erzählt wird oder ob der Nutzer die Kontrolle darüber hat. Beides hat Vor- und Nachteile. Skriptgesteuertes Storytelling hat den Vorteil, den Zuschauer zu zwingen, an für die Geschichte, relevanten Orten zu sein. Das verhindert, dass der Zuschauer die Geschichte verpasst. Dieses Verfahren kann allerdings frustrierend auf den Zuschauer wirken, wenn dieser mit der Welt mehr interagieren möchte [10]. Dieses Problem wird mit dem Ansatz des nutzergesteuerten Storytellings entgegengewirkt. Der Zuschauer kontrolliert die Kamera oder sogar einen Teilnehmer in der Geschichte. Er bestimmt somit die Geschwindigkeit der Geschichte und kann diese selbst erkunden. Dadurch tritt das Problem auf, dass der Zuschauer der Geschichte entgehen kann, als würde er das Kino verlassen bevor der Film vorbei ist [8]. Eine interessante Lösung ist z.B. Gravitation, die den Zuschauer in die richtige Richtung lenkt und ihn an wichtigen Orten festhält [11]. Auch kann der Zuschauer räumlich beschränkt werden, indem man ihm die Möglichkeit nimmt, zurück oder zu weit vor zu gehen [8].

In dieser Arbeit nutze ich einen skriptgesteuerten Ablauf der Geschichte, erlaube dem Nutzer jedoch in einigen Instanzen mit Objekten zu interagieren. Es wird in zwei der vier Szenen die Möglichkeit gegeben, den Ausgang der Geschichte zu ändern. Somit stelle ich eine interaktive und eine nicht interaktive Erzählung der Geschichte gegenüber.

Interaktiv vs. Nicht interaktiv

Durch die interaktive Natur von VR bietet sich interaktives Storytelling besonders gut an. Interaktives Storytelling bedeutet, dass das Publikum durch handeln und Entscheidungen Einfluss auf den Geschichtsverlauf nehmen kann. Zudem können Charaktere der Geschichte auf Taten des Spielers reagieren. Dadurch wird das Publikum mehr in die Geschichte involviert. Jeder erlebt die Geschichte anders.

Im Gegensatz dazu steht das nicht interaktive Storytelling, das den Zuschauer auch als Zuschauer behandelt und eine vorgeschriebene, nicht änderbare Story erzählt. Der Erzähler hat die Kontrolle darüber, wie das Publikum die Geschichte erleben soll²⁰.

Das Projekt dieser Arbeit behandelt eine reduzierte Version des interaktiven Storytellings, das den Spieler lediglich ermöglicht, auf das Geschichtsende Einfluss zu nehmen.

Es gibt auch einige Probleme zu lösen, die die virtuelle Welt mit sich bringt. Der Rundumblick in einer virtuellen Umgebung birgt noch einige Probleme. Thomas Wallner stellte in einem Beitrag zur IFA+ Summit Messe in Berlin ein wichtiges Problem dar: Wie lenke ich die Blicke des Zuschauers auf wichtige Orte?²¹

Dazu zeigte er ein paar Lösungsansätze. Der einfachste Weg ist es, vor einer Schlüsselszene den Blick des Zuschauers automatisch an die gewünschte Position zu lenken. Das ist kein sehr schöner Ansatz und desorientiert den Zuschauer möglicherweise. Eine bessere Lösung ist es, den Spieler auf den Point Of Interest aufmerksam zu machen, beispielsweise über eine Soundkulisse, die den Zuschauer in die richtige Richtung lenkt.

2.3 Wahrnehmung in VR: Immersion, Präsenz und Kohärenz

Immersion:

Die Immersion bezieht sich auf den technologischen Aspekt und beschreibt wie stark man in eine fiktive Welt integriert wird. Gerade hier wird durch VR ein neues Ausmaß an Immersion geschaffen. Dies liegt dem zu Grunde, dass die für VR genutzten Head Mounted Displays, Stereo Vision zur tiefen Wahrnehmung und Headtracking ermöglichen. Diese charakteristischen Eigenschaften sorgen für ein hohes Level an Immersion, im Gegensatz zu normalen Bildschirmen. Hinzu kommen diverse Tracking und Haptik Module, die die Immersion weiter beeinflussen. Die Stärke der Immersion hängt von vielen Faktoren ab [12]:

- Ausmaß des Trackings (Wieviel des Körpers wird getrackt)
- Field of View
- Auflösung der Bildschirme
- Qualität der Grafik
- Haptik
- Latenz (Wie genau werden Bewegung und Haptik übertragen)

²⁰ Quelle: <https://virtualrealitypop.com/passive-and-interactive-storytelling-in-vr-eea98b2f76c5>

²¹ Quelle: <https://www.youtube.com/watch?v=GqSwzpkp2h4>

Präsenz:

Das Wort Präsenz im Bezug auf virtuelle Realität beschreibt das Eintauchen in ein Medium. Präsenz kann unterschiedlich stark auf eine Person wirken. Das hängt einerseits von der Person an sich ab, andererseits auch wie glaubwürdig und konsistent ein Medium dargestellt wird und wie gut die Technik es schafft, reale Stimulation der Sinne zu simulieren. Je besser die Immersion ist, desto einfacher ist es für jemanden sich in einem Medium präsent zu fühlen.

Allerdings kann Präsenz auch mit einem geringen Maß an Immersion funktionieren. Ein Buch beispielsweise trägt nur sehr gering zur Immersion bei, doch kann man sich so darin vertiefen, als würde man die Geschichte vor Ort miterleben [12, 13].

Kohärenz:

Kohärenz ist ein Begriff, der die Glaubwürdigkeit und Konsistenz eines Mediums beschreibt. Das bedeutet in welchem Grad eine Welt in sich schlüssig ist.

Ein Beispiel:

Beim Schauen eines Cartoons erwartet man, dass die Welt und die Charaktere gezeichnet sind. Taucht in einem Cartoon ein realer Schauspieler auf, wird die Kohärenz gebrochen. In dem Film *SpaceJam* [siehe Abb. 2.2] wurde dieser Kohärenzbruch bewusst als Stilmittel benutzt aus dem Kontext gerissen wirkt diese Abbildung unglaubwürdig.



Abbildung 2.2 Ausschnitt aus dem Film *SpaceJam*

Bildquelle: <http://comicbook.com/2016/02/25/space-jam-air-jordan-11-returning-late-2016/>

Das selbe Phänomen kann man auch in Spielfilmen mit schlechter oder veralteter Computer Generated Imagery (CGI), also mithilfe 3D-Computergrafik erstellter Bilder, beobachten. In solchen Filmen haben die CGI Effekte einen hohen Kontrast zu den realen Drehorten und Schauspielern und lassen die Welt unglaubwürdig erscheinen. Kohärenz ist nicht nur wichtig im Visuellen, sondern auch im Verhalten von Charakteren oder Gegenständen.

2.4 Simulator Sickness

Wer sich mit Virtual Reality auseinandersetzt, kommt an dem Thema Simulator Sickness nicht vorbei. Simulator Sickness entsteht, wenn das Auge eine Bewegung wahrnimmt, die für den Körper aber gar nicht stattfindet. So nimmt das Auge einer virtuellen Welt, in der man sich bewegen kann, eine klare Bewegung wahr; der Körper bleibt aber auf der Stelle stehen/sitzen. Das kann dazu führen, dass Symptome wie Schwindel, Übelkeit, Schweißausbrüche und Orientierungslosigkeit auftreten [14].

Für den Entwickler von VR-Anwendungen ist es daher wichtig herauszufinden, was bei den Anwendern Simulator Sickness auslöst und diese Elemente so gut wie möglich zu vermeiden oder zu verbessern. Viele VR Spiele setzen auf die Fortbewegung per Teleportation, da der Körper sich in der virtuellen Welt nicht direkt bewegt und so das Auftreten von Simulator Sickness geringgehalten wird. Teleportation als Bewegung ist aber oft nicht die optimale Technik, da sie nicht immer logisch in den Kontext der virtuellen Welt integriert werden kann. In diesen Fällen bricht es die Präsenz und kann sich negativ auf das Erlebnis des Spielers auswirken.

Auch mit flüssiger Bewegung kann man den Simulator Sickness Effekt niedrig halten. Die folgenden Punkte beeinflussen stark, wie schwer die Simulator Sickness einsetzt:

Beschleunigung: Beschleunigung und Schnelligkeit sind Einflussfaktoren für Simulator Sickness. Langsamere Bewegung nehmen Spieler angenehmer wahr, als schnelle Bewegungen [15]. Als Lösung muss hier die richtige Bewegungsgeschwindigkeit gewählt werden, die einerseits angenehm, als auch praktikabel für die Anwendung ist.

Grad der Kontrolle: Eine unkontrollierte Bewegung in VR hat hohes Potential Simulation Sickness hervor zu rufen [16]. Kamerafahrten oder ähnliche Stilmittel, die dem Spieler die Kontrolle entreißen sind daher mit Bedacht einzusetzen oder zu meiden.

Länge der Nutzung: Die Länge einer Spielsitzung hat auch Einfluss auf das Wohlbefinden. Je länger man ein VR-System nutzt, desto höher die Anfälligkeit auf Simulation Sickness. VR-Anwendungen sollten daher die Möglichkeiten bieten Pausen einzulegen [16].

Field of View: Der FoV des HMD und der gerenderten Umgebung haben auch Einfluss auf das Auftreten von Simulator Sickness [17]. Ein beliebtes Verfahren ist es daher, dass der FoV, gerade in Bewegung, künstlich gesenkt wird. Das Verfahren hat den Nachteil, dass die Immersion gesenkt wird.

Sprungbewegung: Ähnlich wie beim flüssigen Laufen führen Sprungbewegung zum verstärkten Unwohlsein, da hier eine extreme Bewegung wahrgenommen wird, die physisch

nicht stattfindet. Viele VR-Spiele verzichten komplett auf das Sprungelement, da es nur schwierig ist eine gute Lösung zu finden, die für den Spieler angenehm ist und sich gut in die virtuelle Welt einfügt.

Latenz: Je höher die Latenz, desto mehr scheint die Welt hinter der Kopfbewegung hinterher zu hängen. Da dieser Effekt zu Simulator Sickness führen kann, sollte die Latenz so gering wie möglich gehalten werden. Es gibt einige Methoden, die das „Hinterhängen“ des Bildes reduzieren. Bei einer konsistenten Latenz kann man auf die Methode predictive tracking zurückgreifen. Beim predictive tracking wird voraus berechnet wo sich die Position des Kopfes befinden könnte, wenn das Bild dargestellt wird, um somit das richtige Bild zu rendern [18].

2.5 Vier-Sichten-Modell

Das Vier-Sichten-Modell ist ein Konzept von Devon Dolan²². Dieses beschreibt die folgende Abbildung (siehe Abb. 2.4). Das Vier-Sichten-Modell beschreibt das Zusammenspiel von Interaktion und Perspektive in einer Geschichte, wie sie im Laufe dieser Arbeit umgesetzt werden soll.

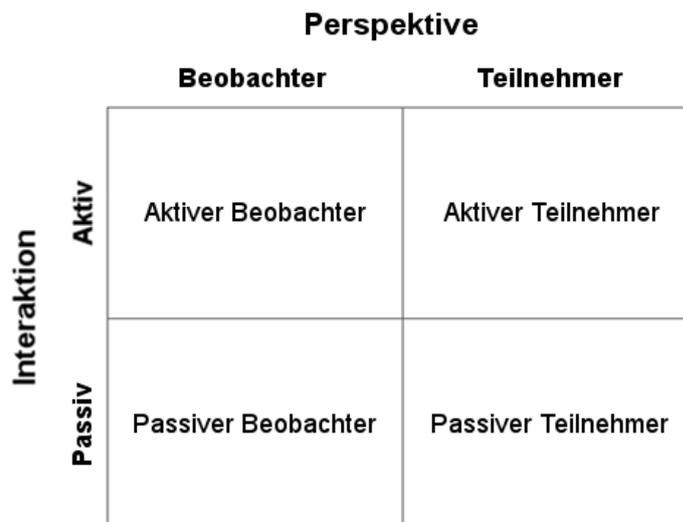


Abbildung 2.4) Vier-Sichten-Modell

Originale Bildquelle: <https://medium.com/@devon.michael/redefining-the-axiom-of-story-the-vr-and-360-video-complex-bee3c20d69df>

²² Quelle: <https://medium.com/@devon.michael/redefining-the-axiom-of-story-the-vr-and-360-video-complex-bee3c20d69df>

Dieses Modell setzt sich zum einen aus der Perspektive zusammen, welche nochmals in Beobachter und Teilnehmer unterteilt ist. Beobachter wird hier gewertet, als jemand der in der Szene nicht direkt involviert ist (praktisch wie ein Geist oder Zuschauer im Theater), der Teilnehmer hingegen ist in der Szene als Akteur integriert.

Zum anderen setzt sich das Modell aus der Interaktion, welche aus Passiv und Aktiv besteht. Passiv beinhaltet keinerlei Interaktion, während Aktiv die Möglichkeit zum Handeln bietet.

Dieses Schema wird im Verlauf dieser Arbeit genutzt, um einen Prototyp eines Spiels zu entwickeln der diese Perspektiven und Interaktionsstufen beinhaltet.

3 Konzept

In diesem Kapitel wird das Konzept für eine Anwendung beschrieben, mit deren Hilfe man verschiedene Erzählweisen eines virtuellen Szenarios vergleichen kann.

3.1 Story

Die Geschichte soll auf das Konzept des Vier-Sichten-Modell passen. Sie soll jeden Quadranten des Vier-Sichten-Modell repräsentieren können und somit aus zwei Perspektiven mit zwei Interaktionsstufen erzählt werden können. Eine weitere Herausforderung ist die Länge der Geschichte, da diese für einen Versuch, der bis zu vier Mal von derselben Person erlebt werden kann, nicht zu lang sein darf. Dies kommt auch der Simulator Sickness zugute, die bei längeren VR-Sitzungen und gerade bei Neueinsteiger zunehmen kann.

Um die Situation nicht vorher erklären zu müssen, ist die Wahl auf ein eher realistisches Szenario gefallen, in dem sich der Nutzer direkt zurechtfinden soll.

Die Geschichte setzt sich aus einem Anfang, einem Mittelteil und einem Schluss zusammen.

Der Anfang

Ein Auto fährt auf einer Straße durch den Wald.

Der ruhige Anfang dient als Einfindungsphase für den Spieler. Hier gibt es keine Interaktion und der Spieler kann sich erst einmal orientieren, wo er sich gerade befindet, was gerade passiert und wo er gerade ist. Die Szene des aktiven Teilnehmers (siehe 2.4) unterscheidet sich an dieser Stelle von den anderen Szenen, da der Spieler hier mit dem Auto interagieren muss. Der Spieler hat keine zeitliche Vorgabe das Ziel zu erreichen und kann sich auch in dieser Szene zuerst zurechtfinden.

Die Mitte

Das Auto überquert einen Bahnübergang. In der Mitte des Bahnübergangs geht der Motor aus. Die Schranken schließen sich und ein Zug kündigt sich an. Die Zugeräusche sollen hierbei gleichzeitig die Aufmerksamkeit des Nutzers auf den Point Of Interest lenken.

Nach der Anfangsphase wird der Spieler direkt in eine aufregende Situation geworfen.

Das Ende

Hier gibt es zwei Enden:

Das Auto hat die Situation erfolgreich überstanden und ist aus der Gefahrenlage entkommen. Das Auto schafft es nicht rechtzeitig aus der Gefahrenlage und wird von der Bahn erfasst.

Die Geschichte wird nach dem Vier-Sichten-Model in zwei verschiedenen Perspektiven und Interaktionsstufen dargestellt.

3.1.1 Perspektiven

Die Perspektive ist in zwei Teile aufgeteilt: Beobachter und Teilnehmer. Beide Teile repräsentieren jeweils eine andere Perspektive.

- Teilnehmer: First-Person Perspektive (FPP)
- Beobachter: Third-Person Perspektive (TPP)

Der Spieler soll beide Perspektiven erleben und miteinander vergleichen können. Dazu wird er im Wechsel ein Szenario in den unterschiedlichen Perspektiven durchlaufen.

First-Person

In VR Anwendung wird zum größten Teil die First-Person (auch Ego-Perspektive genannt) verwendet. Das macht Sinn, da man in eine virtuelle Welt direkt eintaucht und eine Rolle in der Umgebung einnimmt. Die Präsenz ist hierbei besonders stark, da es nur natürlich ist durch die Augen einer Person zu sehen.

In der Applikation übernimmt der Spieler die Sicht eines Autofahrers (siehe Abb. 3.1).



Abbildung 3.1 Screenshot der First-Person Sicht

Die Third-Person Perspektive wird über eine stationäre Top-Down Perspektive realisiert, in der der Spieler sich in einer außenstehenden Rolle wiederfindet (siehe Abb. 3.1). Hier spielt der Abstand der Kamera zur Szene eine große Rolle. Je weiter die Kamera sich vom zu betrachtenden Objekt befindet, desto kleiner nimmt der Spieler das Objekt wahr. In der Abbildung 3.1 befindet sich die Kamera im Auto. So nimmt der Spieler die Umgebung in einer realistischen Größe wahr. In Abbildung 3.2 ist die Kamera weiter außerhalb. Der Spieler nimmt alles in Spielzeuggröße wahr, man kennt diesen Effekt, wenn man aus einem Flugzeug auf eine Stadt guckt.



Abbildung 3.2 Screenshot der Third-Person Sicht

3.1.2 Interaktion

Zusätzlich zu den verschiedenen Perspektiven besteht das Projekt aus unterschiedlichen Interaktionsstufen (siehe Abb. 2.4). Die Interaktionsstufen setzen sich, wie auch schon die Perspektive, aus zwei Bereichen zusammen.

- Interaktionsmöglichkeit
- Keine Interaktionsmöglichkeit

Der Spieler soll in einigen Szenen die Möglichkeit bekommen mit der Umgebung zu interagieren. Hier ist die Art der Interaktion unterschiedlich, je nachdem in welcher Perspektive sich der Spieler befindet. Aus der Third-Person Perspektive soll der Spieler die Möglichkeit haben Gegenstände, wie das Auto, aufzuheben und zu bewegen. In der First-Person-Perspektive soll der Spieler die Möglichkeit bekommen das Auto selbst zu steuern. In anderen Szenen steht dem Spieler keine Interaktion, außerhalb der Rundumsicht, zur Verfügung.

3.2 Grafik: Designentscheidung

Um eine visuell konsistente Welt zu schaffen, müssen alle Assets einem vorher definierten Stil entsprechen. Mehrere Faktoren können darüber entscheiden, welcher Grafikstil am sinnvollsten und geeignetsten ist. Folgende Punkte sind für diese Arbeit zu berücksichtigen:

- Aufwand
- Performance
- Anwendungszweck
- Eigene Präferenzen
- Fähigkeiten
- 2D/3D

Für die Erstellung der Applikation, im Rahmen dieser Arbeit, sind die Faktoren Aufwand und Fähigkeiten am kritischsten. Auch Performance ist für VR Applikationen ein wichtiger Aspekt, der allerdings bei dem Umfang der Applikation keine große Rolle spielt, dennoch in der Entscheidung berücksichtigt wird. Die Entscheidung 2D oder 3D wird durch die Tatsache entschieden, dass die Arbeit sich um eine VR Applikation handelt, die eine 3D Welt voraussetzt.

Um all diese Faktoren zu berücksichtigen bietet sich ein Low Poly Stil an. Der Low Poly Stil wird dadurch definiert, dass dieser mit einer relativ kleinen Anzahl an Polygonen auskommt.

3D Modelle setzen sich aus mehreren Polygonen zusammen (auch Polygonnetz genannt). Polygone sind geometrische Figuren, die mindestens drei Punkte miteinander verbinden. In der Computergrafik werden meistens Dreiecke oder Vierecke verwendet.

Durch die simplen Modelle dieses Stils können Assets mit relativ wenig Aufwand und geringem Modellierungsaufwand erstellt werden. Auch auf die Performance der Applikation wirkt sich dieser Stil positiv aus, da die Low Poly Assets mit wenig Ressourcen gerendert werden können.

3.3 Steuerung

In einem VR Projekt ist es für die Immersion wichtig, sich für die richtigen Steuereingabegeräte zu entscheiden. Mittlerweile gibt es diverse Eingabegeräte, um ein möglichst realitätsnahes Erlebnis zu ermöglichen. Die am häufigsten verwendeten Geräte sind Maus und Tastatur, Gamepads oder bewegungserfassende Controller.

Maus und Tastatur bzw. Gamepads sind immersionsbrechend und können bei einigen Spielern verstärkt zu Simulator Sickness führen. Das hat den Grund, dass die Steuerung unnatürlich ist und die Bewegung in der VR nicht mit der körperlichen Bewegung übereinstimmt [19].

Bewegungserfassende Controller erlauben es dem Spieler die Bewegung von Händen und Fingern in die virtuelle Welt zu übertragen. Durch die natürliche Bewegung wird das Gefühl der Immersion verstärkt und ermöglicht einzigartige Interaktionen in der virtuellen Umgebung.

In diesem Projekt wird die Oculus Rift verwendet. Diese ermöglicht die Steuerung, der Kamera/Sicht des Spielers, über das Headtracking des HMDs.

Für Interaktionen in der Spielwelt werden die Oculus Touch Controller verwendet, welche Hand und Fingerbewegung ins Spiel übertragen können. Im Zusammenspiel mit dem Headtracking der Oculus Rift kann so ein hohes Maß an Immersion erzeugt werden.

3.4 Szenenablauf

Wie oben beschrieben setzt sich das Spiel aus insgesamt vier Szenen zusammen, die durch die verschiedenen Perspektiven und Interaktionsstufen definiert werden. Dazu kommt ein Startbildschirm, von dem aus der Spieler das Spiel eigenständig starten kann, sobald er sich bereit fühlt. In beiden *aktiven* Szenen (Beobachter Aktiv und Teilnehmer Aktiv) kann der Spieler scheitern. In diesem Fall bekommt der Spieler einen „Game Over“ Bildschirm und kann das Level erneut versuchen, bis es ihm gelingt oder diese Szene überspringen.

Im Szenendiagramm (siehe Abbildung 3.3) wird dargestellt, wie die einzelnen Szenen miteinander verbunden sind und wie der Verlauf des Spiels aussieht.

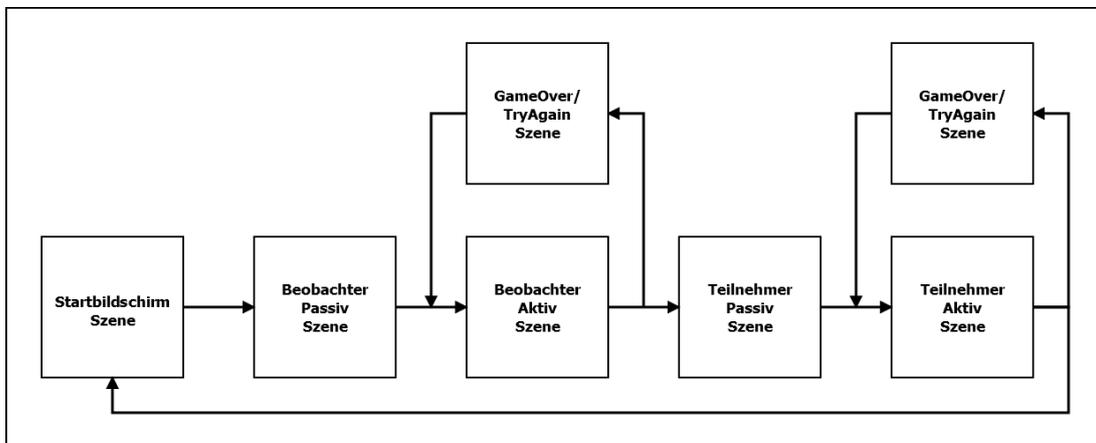


Abbildung 3.3

Die folgenden Zustandsdiagramme zeigen den Ablauf in den einzelnen Szenen.

Das in Abbildung 3.4 gezeigte Diagramm fasst die beiden passiven Szenen (Beobachter und Teilnehmer Passiv) zusammen, da dort der Ablauf gleich ist. Die Szenen nehmen einen linearen Ablauf durch die Geschichte. Der Ablauf dieser Szene ist automatisch.

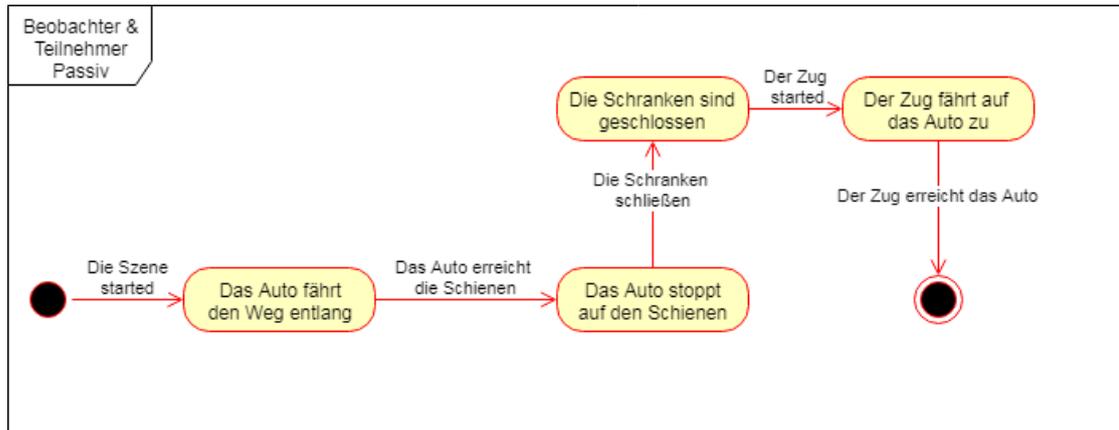


Abbildung 3.4

Das in Abbildung 3.5 gezeigte Diagramm stellt den Szenenablauf in der aktiven Beobachterszene dar. Zu sehen ist, dass der Anfang der Szene gleich verläuft. Ab dem Zustand „Das Auto ist in Gefahr“ treten die Entscheidungsmöglichkeiten in Kraft. Der Spieler kann sich entscheiden das Auto oder den Zug von den Gleisen zu nehmen.

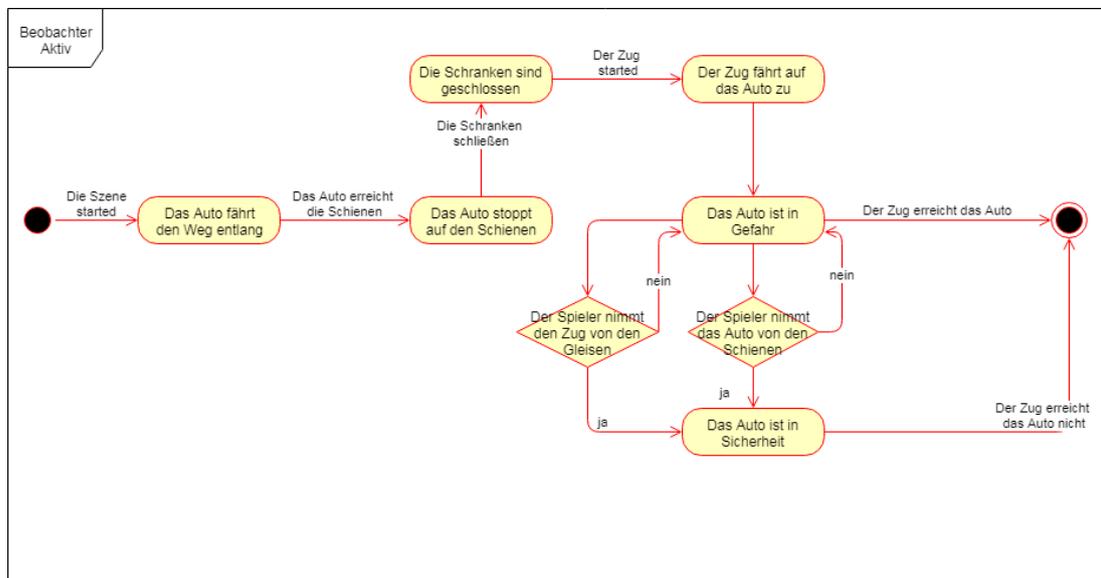


Abbildung 3.5

Abbildung 3.6 zeigt den Ablauf der aktiven Teilnehmerszene. Der Spieler kann das Tempo der Szene selbst bestimmen. Ist das Auto im „Das Auto ist in Gefahr“ Zustand hat der Spieler die Möglichkeit das Auto von den Schienen (in Sicherheit) zu fahren.

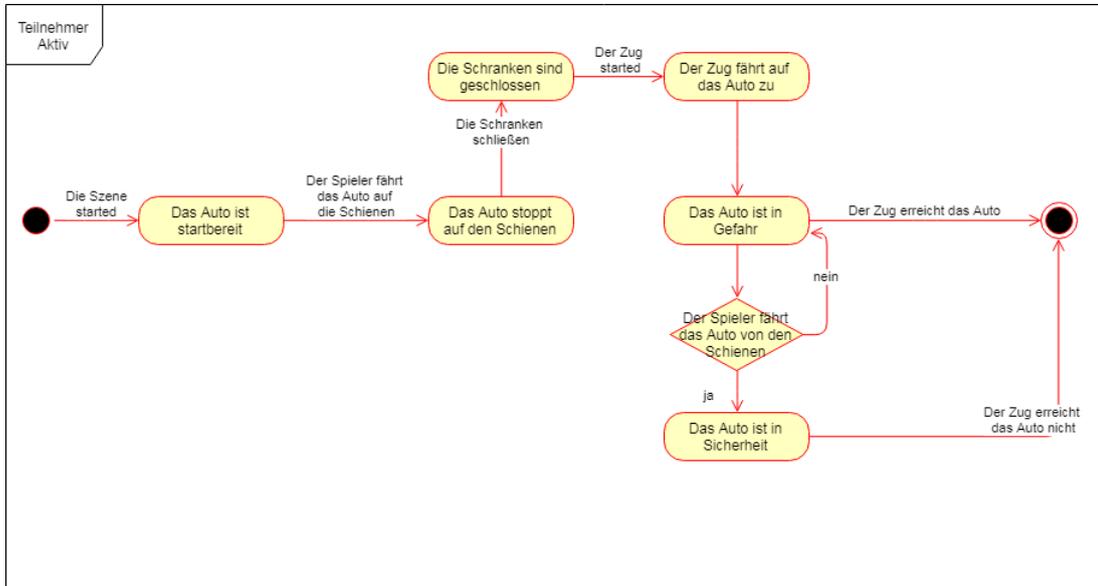


Abbildung 3.6

4 Implementierung

In diesem Kapitel wird beschrieben, wie das Projekt umgesetzt wurde. Dazu wird zunächst aufgelistet, welche Technologien verwendet wurden und warum. Danach gibt es eine Beschreibung des Workflows zum Erstellen eines Grafik Assets anhand eines Beispiels.

Zuletzt wird beschrieben, wie der Spielablauf der verschiedenen Szenen umgesetzt wurde und wie der Spieler jeweils agieren kann.

4.1 Verwendete Technologien

Unity

Ich nutze für dieses Projekt die Game Engine Unity in der Version 2017.3. Da dies mein erstes Projekt dieser Art ist, habe ich mich aus folgenden Gründen für Unity aus folgenden Gründen entschieden:

- Einfacher Einstieg
- Gute Dokumentation
- VR Support

Programmiersprache

Unity unterstützt die Programmiersprachen C# und JavaScript. Da es sich um eine von Unity modifizierte Version von JavaScript handelt, spricht man auch von UnityScript. Für kleinere Projekte macht es keinen Unterschied für welche Sprache man sich entscheidet und hängt von der Präferenz des Programmierers ab.

Ich habe mich in diesem Projekt für C# entschieden, da diese Sprache Java ähnelt, in der ich durchs Studium schon Erfahrung gesammelt habe.

Blender

Zum Erstellen der Grafik-Assets verwende ich Blender²³ in der Version 2.79. Blender ist für den Einstieg eine recht komplexe Grafiksoftware, die sich aber durch die freie Lizenz für nicht finanzierte Projekte gut eignet. Aufgrund der Langlebigkeit der Software ist diese gut dokumentiert und bietet trotz der Komplexität eine für dieses Projekt angemessene Lernkurve.

Oculus Rift

Die Oculus Rift ist eines der beiden PC Orientierten VR-Geräte, die bewegungssensitive Controller beinhalten. Die Auswahl des VR-Gerätes fiel anhand des verfügbaren Gerätes.

4.2 Grafik: Erstellung der Assets

Im Folgenden wird der Workflow des Erstellens eines Assets beschrieben. Als Beispiel dient hier das Motiv des Autos und zeigt die Schritte von der Motivauswahl bis hin zum fertig importierten Objekt in Unity.

Auswahl des Motives

Bei der Auswahl des Motives kann man sich gerade bei realitätsnahen Objekten gut an der echten Welt orientieren. Zur Inspiration helfen hier beispielsweise Fotos von ähnlichen Objekten.



Abbildung 4.1²⁴

²³ URL: <https://www.blender.org/>

²⁴ Bildquelle: <https://www.freeimages.com/photo/red-car-1448997>

Das Auto in diesem Projekt ist im Fokus des Spielflusses. Deswegen ist es sinnvoll, ein Auto mit einer Signalfarbe zu wählen. In diesem Fall habe ich mich für Rot entschieden (siehe Abb. 4.1). Da das Projekt im Low Poly Stil gehalten wird, dient das Bild lediglich als Inspiration als eine strikte Vorlage.

Skizzierung und Vereinfachung des Motives

Um einen Low Poly Stil zu erreichen, ist es sinnvoll, für das Motiv eine reduzierte Zeichnung anzufertigen. Im Low Poly Stil werden oft unwichtige Details weggelassen und die Objekte auf die wichtigen Merkmale reduziert, wie in der Skizze zu sehen (Abb. 4.2).

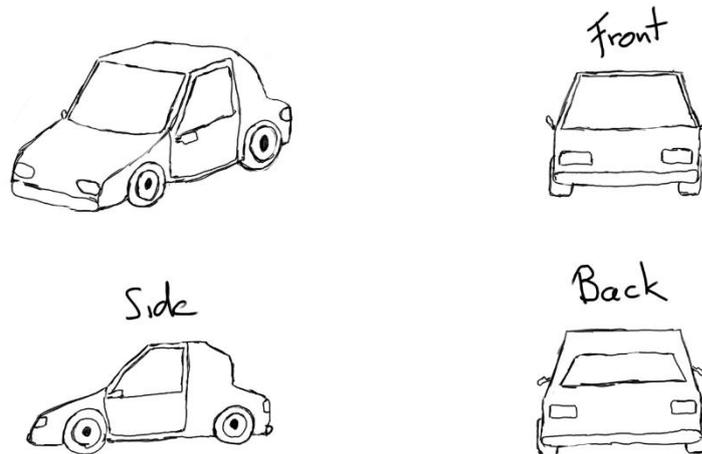


Abbildung 4.2

Das Auto besitzt nun kaum noch Details und unterscheidet sich von der echten Vorlage. Da alle Assets auf die wichtigsten Merkmale reduziert werden und sie so aufeinander abgestimmt sind, ist die Kohärenz der virtuellen Welt gegeben.

Umsetzung in Blender

Die oben gezeigte Skizze dient nun als Vorlage für die 3D Modellierung. Blender ermöglicht es, das Bild als Hintergrundgrafik zu verwenden und dadurch den Modellierungsprozess deutlich zu vereinfachen.



Abbildung 4.3 Screenshot aus Blender

Import in Unity

Das fertiggestellte Grafikmodell kann in Unity über ein FBX Format importiert werden. Das FBX Format speichert neben dem Modell unter anderem auch die Materials, Texturen und Animationen. Eine andere Möglichkeit ist das von Blender gestellte .blend Format. Dieses benötigt im Gegensatz zu FBX, die installierte Blender Software für den Import in Unity²⁵.

Beim Import von in Blender erzeugten Modellen muss bedacht werden, dass Blender ein linkshändiges Koordinatensystem verwendet, während Unity ein rechtshändiges Koordinatensystem benutzt.

²⁵ Quelle: <https://docs.unity3d.com/2017.4/Documentation/Manual/HOWTO-ImportObjectBlender.html>

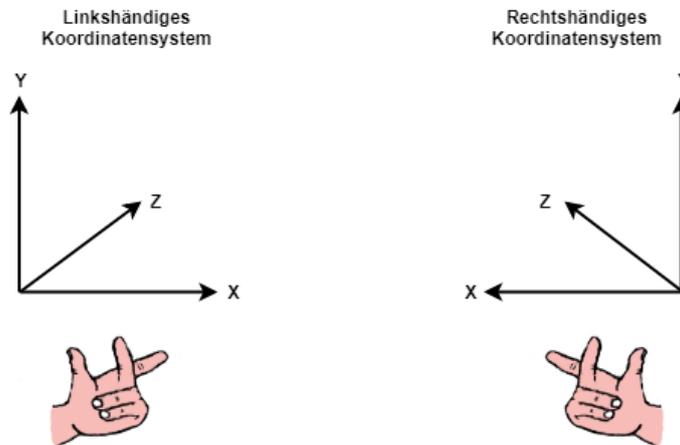


Abbildung 4.4 Rechts- und Linkshändiges Koordinatensystem

Das führt dazu, dass die importierten Objekte in der Rotation angepasst werden müssen. Das kann entweder direkt in Blender oder nachträglich in Unity vorgenommen werden.

Skalierung

Skalierung ist gerade in VR Anwendungen wichtig, da man einen besseren Eindruck von Größen bekommt. Die meisten Objekte sollten eine annähernd realistische Größe haben, damit sie nicht merkwürdig groß oder klein erscheinen. Ausnahmen wären Objekte zum Interagieren, die in der Realität sehr klein sind beispielsweise ein Schlüssel oder eine Keycard um das Interagieren zu erleichtern.

Um Objekte einheitlich und realistisch in Relation zueinander zu skalieren, ist es sinnvoll sich ein Vergleichsobjekt zu schaffen.

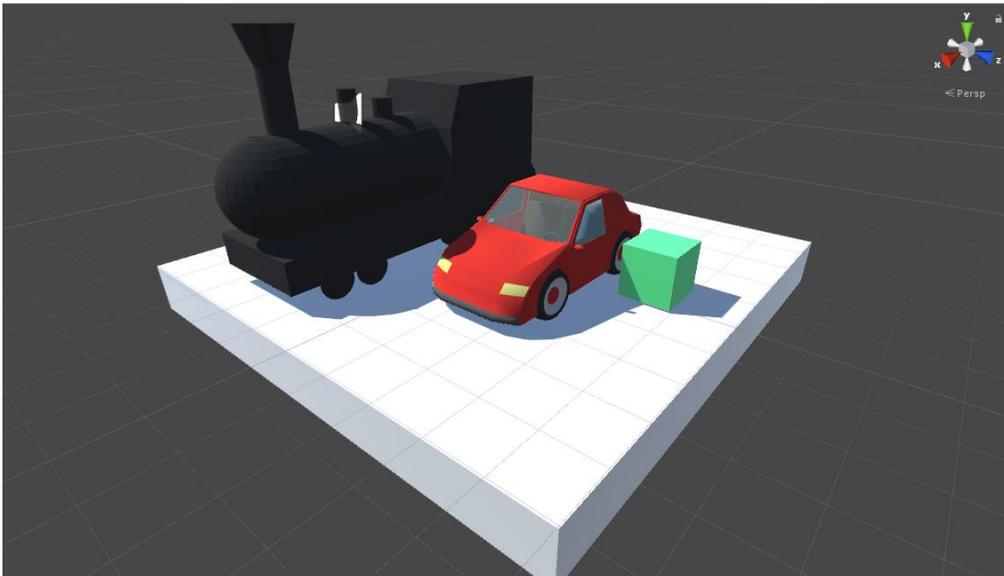


Abbildung 4.5: Vergleichswürfel in grün zum Abgleichen anderer Objekte

Die einfachste Möglichkeit ist es, einen Standard-Würfel aus Unity zu verwenden. Der Würfel hat eine Kantenlänge von 1. Diese kann als 1 Meter interpretiert werden. Um die richtige Skalierung zu finden, muss also nur noch der Würfel mit dem zu skalierenden Objekt abgeglichen werden (siehe Abb. 4.5).

4.3 Skripte und Komponenten

In diesem Kapitel werden die Umsetzung und der Zusammenhang wichtiger Skripte und Komponenten erläutert.

4.3.1 Komponenten des Autos

Rigidbody

Die Rigidbody-Komponente in Unity ermöglicht das physische Verhalten von Objekten. Objekte mit einer Rigidbody-Komponente werden von Gravitation beeinflusst. Gravitation in Unity ist ein vordefinierter Vektor, der eine Kraft in Abhängigkeit des Vektors auf alle Objekte mit einer Rigidbody-Komponente ausübt. Zudem können Objekte durch eigen definierte Kraft oder (wenn mit einem Collider ausgestattet) durch Kollision anderer Objekte bewegt werden.

Da das Auto ein von der Physik beeinflusstes Objekt sein soll, ist eine Rigidbody Komponente vonnöten. Die Masse (Mass) gibt das Gewicht des Autos an und ist wichtig, damit Kollision und Fahrverhalten realistisch simuliert werden können. Die Checkbox „Use Gravity“ ist im Normalfall aktiviert und bedeutet, dass das Objekt von der Gravitation angezogen wird. „Is Kinematic“ kommt bei der interaktiven Third-Person Szene zum Tragen und muss aktiviert werden, wenn das Auto von dem Spieler bewegt werden soll.

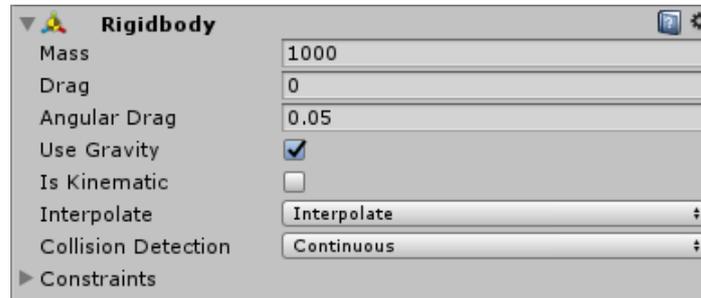


Abbildung 4.6: Rigidbody Komponente des Autos in Unity

Collider

Collider sorgen dafür, dass das Auto von Kollisionsabfragen beeinflusst wird. Das ist nicht nur wichtig um nicht durch Objekte zu fahren, sondern auch, damit das Auto von Triggern erfasst werden kann. Das Auto besitzt ein Mesh Collider, welcher eine konvexe Abbildung des Autos darstellt und eine ziemlich genaue Kollisionsabfrage ermöglicht.

Außerdem besitzt das Auto vier Wheel Collider, diese simulieren die Reifen und werden im späteren Verlauf dieser Arbeit noch einmal angesprochen.

Skripte

Skripte werden in Unity verwendet um eigene Funktionalitäten zu schaffen. Skripte sind eigene Komponenten und können Informationen von anderen Komponenten anfragen, verändern oder neue Komponenten erzeugen. Wie in den meisten GameEngines stellen Skripte die Kernfunktionen der Anwendung dar.

Das Hauptskript des Autos ist das CarController Skript, welches die Funktionalitäten des Autos implementiert. Zum Steuern des Autos stellt das Skript einige Methoden zur Verfügung, wie beispielsweise Beschleunigen, Bremsen oder Starten des Motors. Diese Methoden werden in Szenen mit einem automatischen Ablauf von dem AICar Skript genutzt, welches das Auto automatisch steuert. Die Skripte KeyController, SteeringWheelController und ClutchController werden genutzt, um einzelne Funktionen zu abstrahieren und dienen zur Steuerung des Autos über den Spieler.

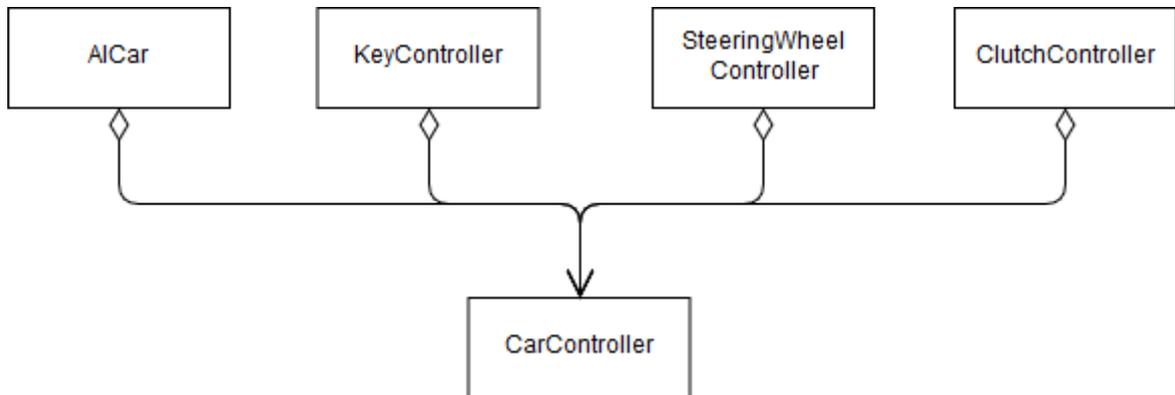
Funktionalität des Autos:

Abbildung 4.7: Steuerungszugriffe des CarController Skripts in UML

Um dem Auto Funktionalität zu geben, wurde ein Skript verwendet. Das CarController Skript implementiert die Hauptfunktionen des Autos, wie Gas geben, Bremsen und Lenken und stellt diese anderen Skripten zur Verfügung. Das wird genutzt, um das Auto von verschiedenen Skripten anzusprechen und zu steuern.

Das Skript benötigt die vier Wheel Collider für die einzelnen Reifen. Zwei Wheel Collider werden jeweils zu einer Achse zusammengefasst. Die Achsen bekommen zusätzlich Informationen über Checkboxes, ob die Achse lenkbar, angetrieben oder bremsbar ist. Außerdem wird festgelegt, wie stark das Auto maximal beschleunigen, lenken und bremsen kann. Die Wheel Collider sind nicht als Reifen in der Szene sichtbar, deswegen müssen visuelle Reifenobjekte hinzugefügt werden, die sich abhängig vom Wheel Collider bewegen.

Damit passend zum Status des Autos Sounds abgespielt werden können, werden diese dem Skript hinzugefügt.

Um das Auto von außen zu steuern, stellt das Skript für jede Aktion eine Methode zur Verfügung. Mit diesen Methoden lässt sich die Beschleunigung, Bremskraft und Lenkung beeinflussen

KI Skript des Autos:

Das AICar Skript steuert das Auto automatisch, sobald das Skript dem Auto Objekt hinzugefügt wird. Das Skript sorgt dafür, dass das Auto einen vorgegebenen Weg abfährt.

Der Weg wird in Wegpunkten (WayPoints) angegeben. Wegpunkte sind leere Objekte, die in der Szene nicht zu sehen sind. Die Wegpunkte werden der Reihenfolge nach, in der das Auto sie abfahren soll, hinzugefügt.

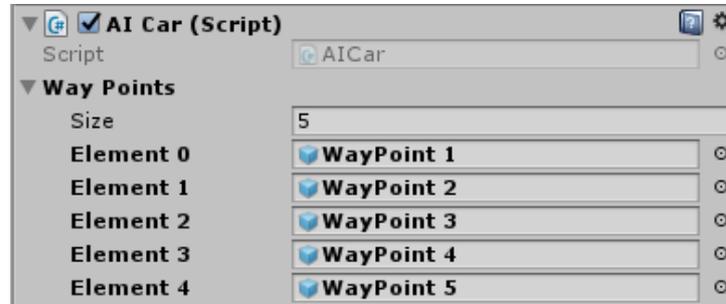


Abbildung 4.8: Screenshot des AICar Skript in Unity

Das Skript berechnet den Winkel zu dem nächsten Wegpunkt, um die Stärke der Lenkung festzulegen. Da Unitys Winkelfunktion nur einen positiven Winkel zurück gibt, muss dieser umgerechnet werden, um es möglich zu machen, zwischen Links- und Rechtslenken zu unterscheiden.

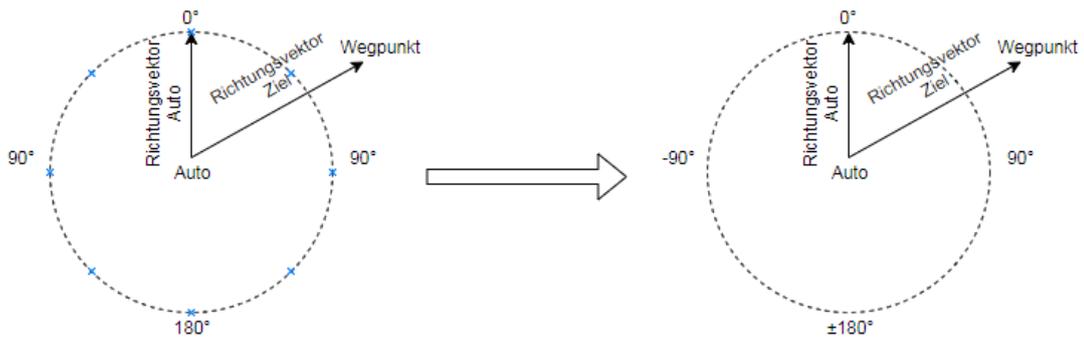


Abbildung 4.9: Grafische Darstellung der Richtungsbestimmung

Für die Berechnung wird ein Referenzvektor gebildet, der von der Blickrichtung um 90° nach rechts zeigt. Der Referenzvektor wird aus dem Kreuzprodukt aus dem Vektor in Blickrichtung (Vektor a) und einem Vektor senkrecht nach oben (Vektor b) berechnet.

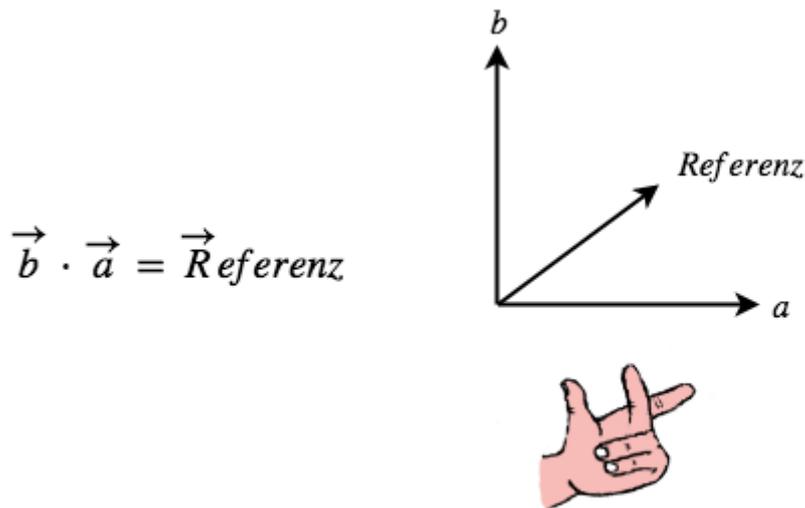


Abbildung 4.10 grafische Darstellung des Kreuzproduktes

```
//Vektor in Blickrichtung des Autos
Vector3 referenceForward = transform.forward;
//Vektor 90° nach Rechts zeigend
Vector3 referenceRight = Vector3.Cross(Vector3.up, referenceForward);
```

Der Referenzvektor wird mit dem Richtungsvektor zum Wegpunkt über das Skalarprodukt abgeglichen. Das Skalarprodukt gibt, bei Normalvektoren, als Ergebnis einen Wert zwischen -1 und 1 zurück. Dieser Wert entspricht 1, wenn die Vektoren die exakt gleiche Richtung haben und „-1“ bei exakt entgegengesetzter Richtung.

```
//Winkel zwischen Blickrichtung des Autos und Richtung des Ziels (0°..180°)
float directionToTarget = Vector3.Angle(headingToTarget, referenceForward);

//Berechne das Vorzeichen des Winkels: Rechts der Blickrichtung positiv, Links der
Blickrichtung negativ
float sign = Mathf.Sign(Vector3.Dot(headingToTarget, referenceRight));
float angleToTarget = sign * directionToTarget;
```

Das Vorzeichen gibt uns die Information, ob der angesteuerte Wegpunkt rechts oder links liegt und wird genutzt um das Vorzeichen des berechneten Winkels zwischen Blickrichtung und Ziel zu setzen.

Das Auto fährt mit voller Beschleunigung bis zu einer festgelegten Distanz zum letzten Wegpunkt und soll auf dem Bahnübergang zum Stehen kommen. Dazu wird die Entfernung zum letzten Wegpunkt berechnet und ein Beschleunigungsfaktor aus der Entfernung zum letzten Wegpunkt berechnet. Dieser wird an das CarController Skript übergeben.

```
//Vektor zum letzten Wegpunkt
Vector3 directionToLastWayPoint = waypoints[waypoints.Length - 1].transform.position -
transform.position;
//Entfernung zum letzten Wegpunkt
float distanceToLastWayPoint = directionToLastWayPoint.magnitude;

//Verringere die Beschleunigung des Autos in Abhängigkeit
//der Entfernung zum letzten Wegpunkt
float acclFactor = distanceToLastWayPoint / stallDistance;
currentCar.Accelerate(acclFactor);
```

4.3.2 Steuerung und Interaktion

Steuerung und Interaktion unterscheiden sich je nach Szene. In der interaktiven Third-Person Szene können Objekte angehoben werden.

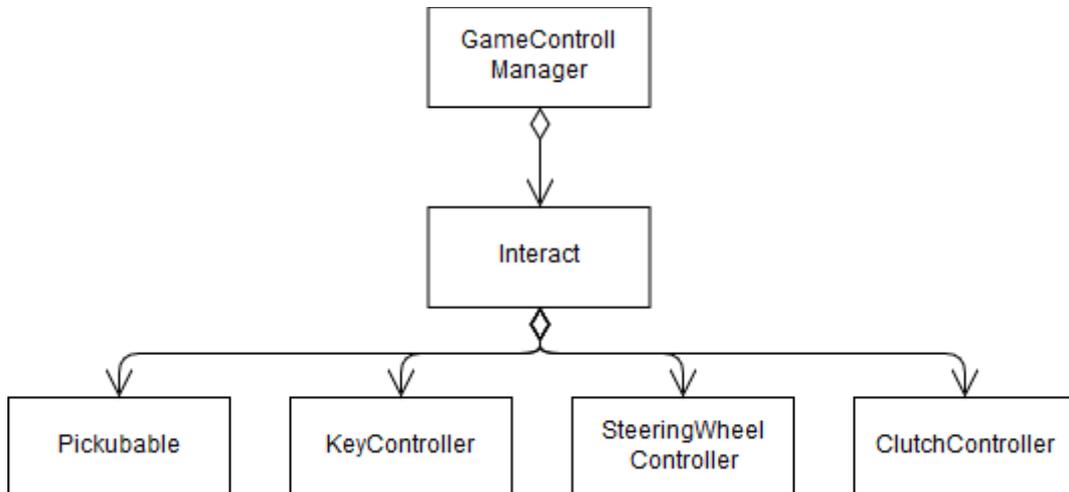


Abbildung 4.11: Interaktionen mit dem Controller als UML

Interaktion mit Objekten:

Das Interact Skript wird dem Oculus Hand Anker hinzugefügt. Dieser symbolisiert die Position eines Oculus Rift Controllers. Beim Drücken des Trigger wird in einem sphärischen Bereich nach Collidern gesucht. Dafür wird die Funktion `Physics.OverlapSphere(Vector3 position, float radius)` genutzt, die ein Array mit allen Collidern die sich in der Sphere befinden zurück gibt.

Befindet sich in der Sphäre ein gültiges Objekt zum Interagieren werden je nach Art des Objekts Aktionen durchgeführt.

Aufnehmbare Objekte:

Aufnehmbare Objekte sind Objekte, die der Spieler aufheben und bewegen kann. Diese haben ein Pickupable-Skript, welches definiert, dass sie aufgenommen werden können. Damit das Objekt dem Controller folgt, wird es dem Controller als Kind-Objekt zugeordnet.

Das Pickupable-Skript dient hauptsächlich als Klassifizierung, um ein Objekt als aufhebbar zu markieren. Es ist möglich, dem Skript eine Relative Position und Rotation anzugeben. Beim Aufheben wird das Objekt auf diese Position/Rotation in Abhängigkeit des Controllers gesetzt.



Abbildung 4.12: Screenshot des Pickupable Script in Unity

Lenkrad:

Damit der Spieler das Lenkrad greifen und lenken kann, gibt es ein Lenkrad Objekt, welches mit einem `SteeringWheelController` Skript ausgestattet ist.

Das `SteeringWheelController`-Skript berechnet den Winkel zwischen der relativen Position der Hand zur Lenkradmitte und der X-Achse des Lenkrads.

```
private float CalculateAngle()
{
    //Relative Handposition im bezug auf das Lenkrad
    Vector3 relativeHandPosition =
    steeringWheelBase.transform.InverseTransformPoint(hand.transform.position);
    //Winkel in Grad der Handposition ausgehend von der Mitte des Lenkrads
    float value = (Mathf.Atan2(relativeHandPosition.y, relativeHandPosition.x))
    *Mathf.Rad2Deg;

    return value;
}
```

Danach wird berechnet, um welchen Winkel sich das Lenkrad/die Hand in dem aktuellen Frame bewegt hat. Dazu wird der Winkel des letzten Frames gesichert und mit der Funktion `Mathf.DeltaAngle(currentAngle, lastAngle)` mit dem aktuellen Winkel verrechnet. Die Funktion gibt die kleinste Differenz der beiden Winkel zurück. Das wird benötigt, um Probleme beim Überlauf von -179 auf +179 (zu sehen in Abbildung 4.10) und umgekehrt zu verhindern.

Die Differenzen aller Frames werden zusammenaddiert um festzustellen, wie weit das Lenkrad/die Hand gedreht wurde.

Um die Drehung auch visuell auf das Lenkrad zu übertragen, wird mit dem berechneten Winkel noch die Rotation des Lenkrad Objektes angepasst.

Schlüssel:

Das Schlüssel Objekt dient dazu, das Auto zu starten. Der Spieler kann das Objekt greifen und drehen, dafür ist dieses mit einem KeyController Skript ausgestattet.

Zum Drehen des Schlüssels wird die Rotation des Oculus Touch Controllers genutzt. Um eine Änderung in der Rotation festzustellen, wird die Differenz aus dem aktuellen Winkel der Rotation und dem Winkel des letzten Frames berechnet.

Die Rotation des Schlüssel Objektes wird entsprechend der Differenz angepasst. Sobald der Schlüssel mehr als 60° in der Z-Achsenrotation aufweist, wird der Motor des Autos gestartet.

Gashebel:

Um das Auto zu beschleunigen, bremsen und rückwärtsfahren zu lassen, befindet sich ein Gashebel in der Mitte des Autos. Dieser ist mit „D“ für Drive zum Beschleunigen und „B“ für Brake zum Bremsen gekennzeichnet. Je mehr der Hebel nach vorne/hinten gelegt wird, desto stärker beschleunigt bzw. bremst das Auto. Wird der Hebel ganz nach hinten gezogen, fährt das Auto in einer gesetzten Geschwindigkeit rückwärts.

Um das zu erreichen, bestimmt das ClutchController Skript die Position des Controllers und berechnet, ähnlich wie im KeyController und SteeringWheelController Skript, die Änderung zum letzten Frame, um die Bewegung des Hebels zu erfassen.

Da die Bewegung auf die Rotation des Hebels addiert wird, muss die Positionsänderung angepasst werden, damit die Bewegung des Hebels mit der Hand übereinstimmt.

Aus dem Winkel des Hebels und dem maximalen oder minimalen Winkel wird der Faktor zum Beschleunigen und Bremsen berechnet und an das CarController Skript weitergegeben.

```
float acclFactor = currentAngleX / maxClutchAngle;  
carController.Accelerate(acclFactor);
```

4.4 Spielablauf

Der Spielablauf ist abhängig davon, ob die Szene interaktiv ist oder nicht. Nicht interaktive Szenen haben einen automatischen Spielablauf, während der Spielablauf von interaktiven Szenen von Aktionen des Spielers beeinflusst werden können.

Die Umsetzung der Szenen wurden möglichst selbsterklärend gestaltet und bieten Raum, damit der Spieler ein wenig rumprobieren kann.

Um dies zu erreichen, wurden Objekte und Werkzeuge zum Interagieren zu bestimmten Zeitpunkten hinzugefügt. Durch diese Änderungen soll der Spieler auf neue Objekte aufmerksam gemacht werden und motiviert, damit zu interagieren und Neues auszuprobieren.

Im Laufe des Kapitels wird auf dieses Prinzip genauer eingegangen.

4.4.1 Der Anfang

Der Anfang der Szene ist die Fahrt des Autos zum Bahnübergang. In drei der vier Szenen wird diese Fahrt automatisch ausgeführt und ist dementsprechend gleich umgesetzt. In der vierten Szene, der aktiven First-Person Szene, wird das Auto vom Spieler selbst gesteuert und unterscheidet sich hier von den anderen Szenen.

In den Third-Person Szenen, sowie in der passiven First-Person Szene, wird der Anfang durch das oben beschriebene AICar-Skript gesteuert. Zu Beginn der Szene wird die Start-Methode des Skriptes aufgerufen, welche das Auto startet.

Danach bestimmt das Skript in jedem Frame, wie schnell das Auto in welche Richtung fährt (siehe Kapitel 4.6.1 Komponenten - Skripte). Das Auto fährt automatisch eine vorgegebene Route. Wenn das Auto den Bahnübergang erreicht hat beginnt der Mittelteil.

In der aktiven First-Person Szene kann der Spieler selber Einfluss auf den Verlauf des Anfangs nehmen. Im Gegensatz zur passiven Szene werden dem Autoinnenraum Objekte zum Interagieren hinzugefügt.

Um mit dem Auto fahren zu können, muss mit Objekten in bestimmter Reihenfolge interagiert werden. Wie in einem echten Auto muss zunächst der Motor gestartet werden. Der Spieler soll mithilfe eines Schlüssels den Motor starten (siehe Kapitel 4.6.1 Komponenten – Steuerung und Interaktion). Auch ist es möglich, den Motor wieder zu stoppen indem man

den Schlüssel wieder zurückdreht. Das hat keinen direkten Nutzen für die Szene und dient lediglich dazu, dem Spieler ein realistischeres Gefühl zu geben.

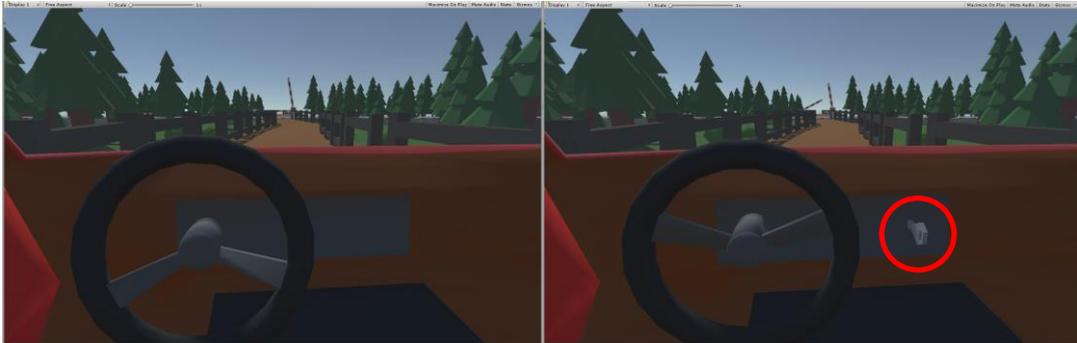


Abbildung 4.13: Screenshot der First-Person Szenen: links passiv, rechts aktiv mit Schlüssel

Durch das Fehlen des Schlüssels in der passiven Szene, soll der Spieler auf das neue interaktive Objekt aufmerksam gemacht werden.

Wenn das Auto gestartet ist, kann der Spieler mithilfe eines Gashebels das Auto zum Beschleunigen oder Bremsen bringen. Der Gashebel befindet sich zwischen Fahrer- und Beifahrersitz. Der Hebel hat keine Funktion, wenn der Motor ausgeschaltet ist.

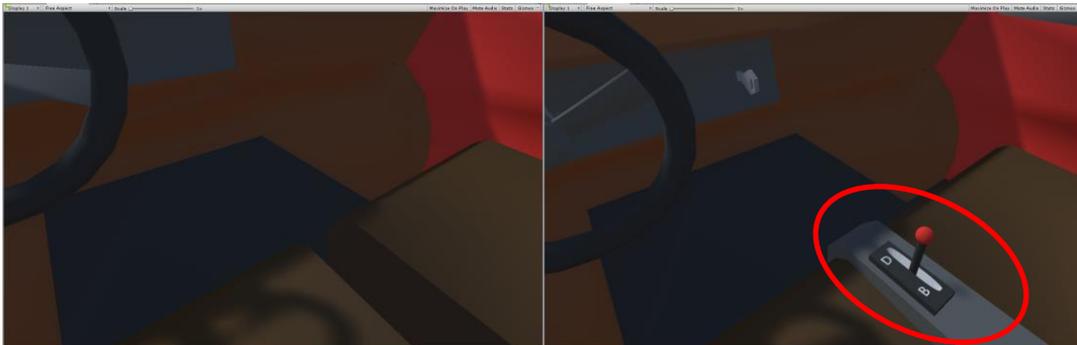


Abbildung 4.14: Screenshot der First-Person Szenen: links passiv, rechts aktiv mit Gashebel

Auch der Hebel wird in der aktiven Szene hinzugefügt und zusätzlich mit Rot als Signalfarbe verdeutlicht, da der Hebel nicht direkt im Sichtfeld des Spielers liegt und durch Umschauen entdeckt werden muss. Der Hebel ist mit „D“ für Drive und „B“ für Brake markiert, um den Spieler einen Hinweis darauf zu geben, in welche Richtung der Hebel zum Beschleunigen oder Bremsen gekippt werden muss. Beim Greifen des Hebels wird das Skript *ClutchController* (siehe Kapitel 4.6.1 Komponenten – Steuerung und Interaktion) angesprochen und regelt je nach Hebelstellung, wie sich das Auto verhält.

4.4.2 Die Mitte

Der Mittelteil startet, sobald das Auto den StallEventTrigger erreicht. Dieser befindet sich auf den Schienen und startet eine Reihe von Eventroutinen.

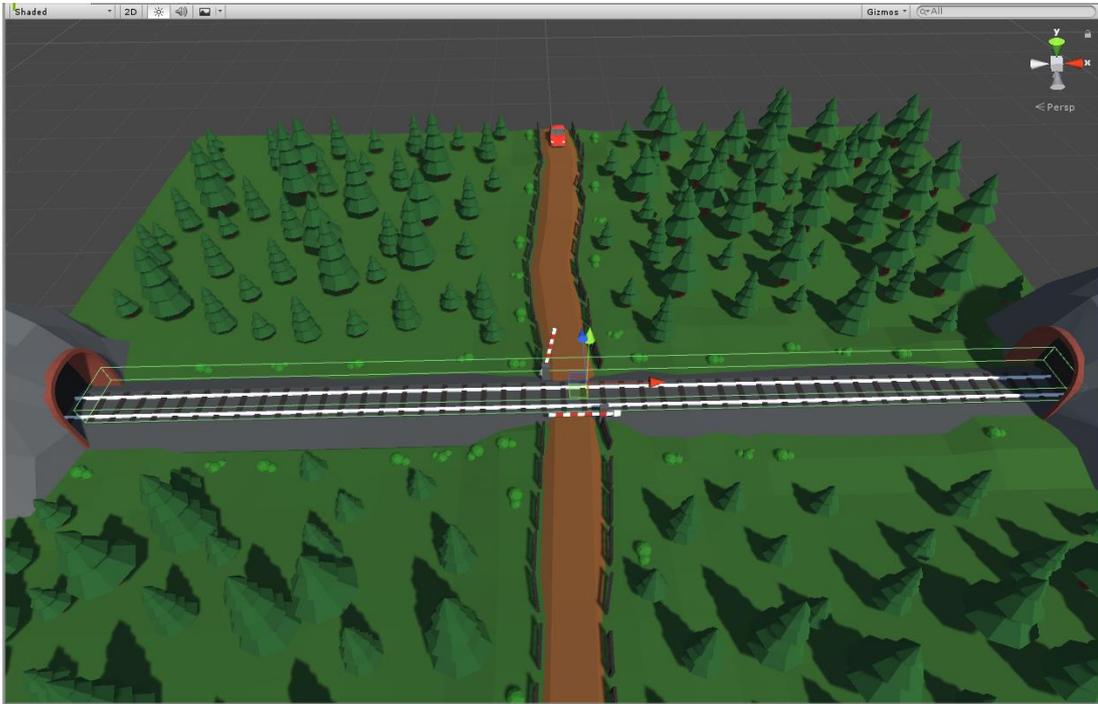


Abbildung 4.15: Screenshot des StallEventTriggers (Grüne Box)

Der StallEventTrigger ist gekoppelt mit dem StallEvent Skript, welches angesprochen wird, sobald das Auto den StallEventTrigger betritt.

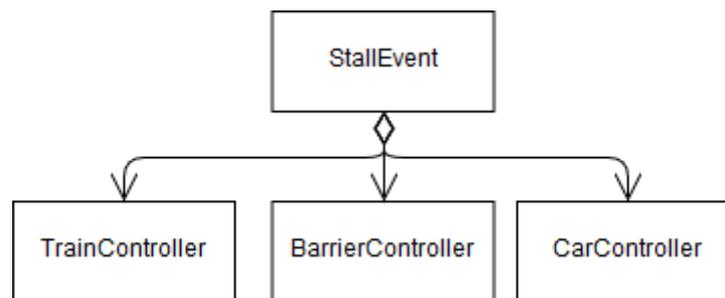


Abbildung 4.16: Aggregation des StallEvent in UML

Zunächst prüft das Skript, welche Szene gerade aktiv ist, um zu entscheiden, welche Events stattfinden müssen.

In den passiven Szenen wird das CarController-Skript angesprochen und die StallMotor-Methode aufgerufen. Die StallMotor-Methode startet die StallMotor-Routine. Diese stoppt den Motor und spielt, nachdem der Motor gestoppt ist, den Audio Clip zum Starten des Motors ab. Die Motorstart Clip wird in einer Schleife abgespielt um zu symbolisieren, dass das Auto Probleme beim Starten hat und Hilfe benötigt. Außerdem soll damit zusätzlich Spannung erzeugt werden.

Der Mittelteil der aktiven Third-Person Szene ist von außen betrachtet identisch mit den passiven Szenen. Hier wird allerdings beim Betreten des Triggers dem Auto zusätzlich das Pickupable-Skript (siehe Kapitel 4.6.1 Komponenten – Steuerung und Interaktion) hinzugefügt, welches dem Spieler das Heben des Autos ermöglicht.

Beim Betreten des StallEventTriggers in der aktiven First-Person Szene wird der Motor gestoppt und das Auto abgebremst, da dieses Mal der Spieler selbst versuchen soll, den Motor wieder zu starten und nicht wie zuvor ein Skript.

Nachdem das Auto gebremst wird, werden die Schranken geschlossen. Auch hier wird eine Routine aufgerufen, die je nach Szene agiert. Die Schranken verhalten sich nur in der aktiven First-Person Szene anders, daher wird diese gesondert behandelt. In allen anderen Szenen werden beide Schranken geschlossen, sodass das Auto festsetzt. Hierfür werden Animationen genutzt, die über einen Animator angesprochen werden.

Der Animator hat drei Animationen zwischen denen gewechselt werden kann. Der Default Status ist *BarrierOpen*. In diesem Status befinden sich die Schranken beim Start aller Szenen. Von diesem Status kann entweder zur *BarrierClosing* Animation oder *BarrierBroken* Animation gewechselt werden. Beide Schranken können unabhängig voneinander animiert werden.

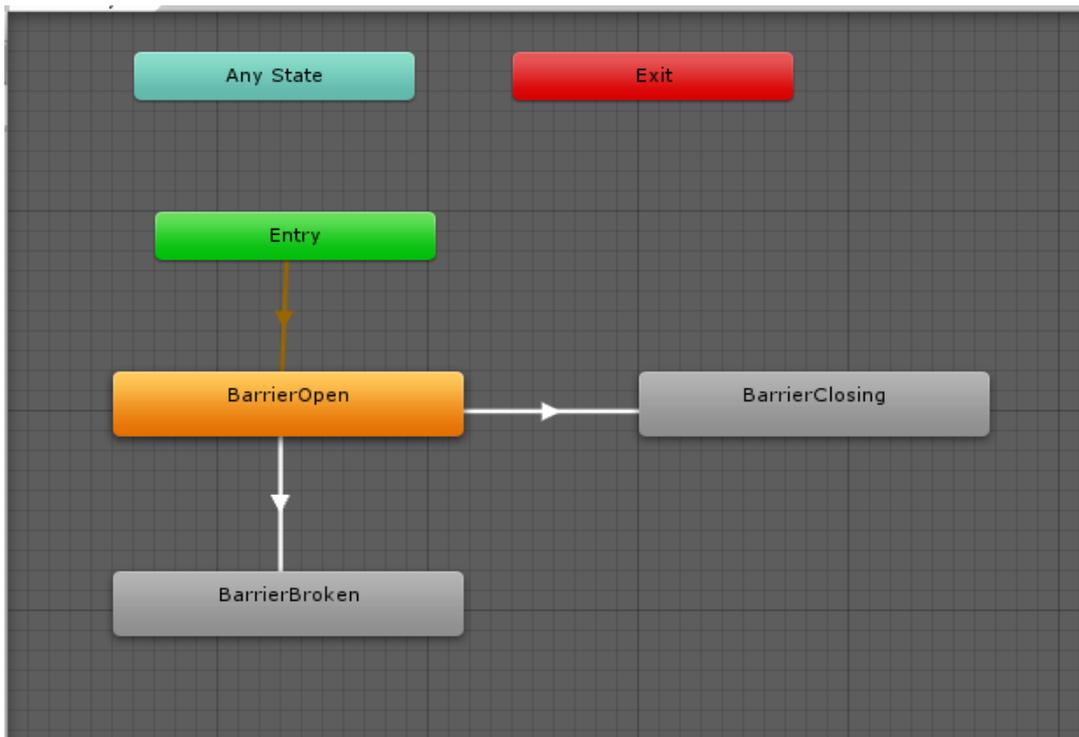


Abbildung 4.17: Screenshot vom Animator mit den drei Animationen

Die Schranken in der First-Person Szene schließen sich nicht komplett, damit der Spieler eine Chance hat, von den Schienen zu fahren. Die Schiene in Blickrichtung des Autos ist defekt und stoppt nach kurzem Schließen, sodass das Auto noch Platz hat, die Schienen zu verlassen. Dies wird wieder mit einer Animation realisiert, die die Schranke zum Teil schließt und anschließend eine Partikel Animation aktiviert, die Rauchschwaden erzeugt, um anzudeuten, dass die Schranke defekt ist (siehe Abbildung 4.19).



Abbildung 4.18: Defekte Schranke

Zum Schluss wird der Zug nach einem Delay gestartet, der den Schranken Zeit gibt, zu schließen. An dieser Stelle soll die Aufmerksamkeit auf den nahenden Zug gelenkt werden. Dies wurde mit einer räumlichen Soundquelle (spatial sound) gelöst. Die räumliche Soundquelle ändert je nach Entfernung zum Spieler die Lautstärke, sodass der Zug beim Annähern lauter wird. Die Lautstärke wird für jeden Lautsprecher unabhängig angepasst, sodass der Spieler die Richtung der Soundquelle feststellen kann. Diese Funktion wird in Unitys Audio Source Komponente unterstützt.

4.4.3 Das Ende

Die passiven Szenen enden automatisch kurz bevor der Zug das Auto erfasst. Dazu wird mit einem BoxCollider vor dem Zug geprüft, ob der Zug kurz vor dem Aufprall ist.

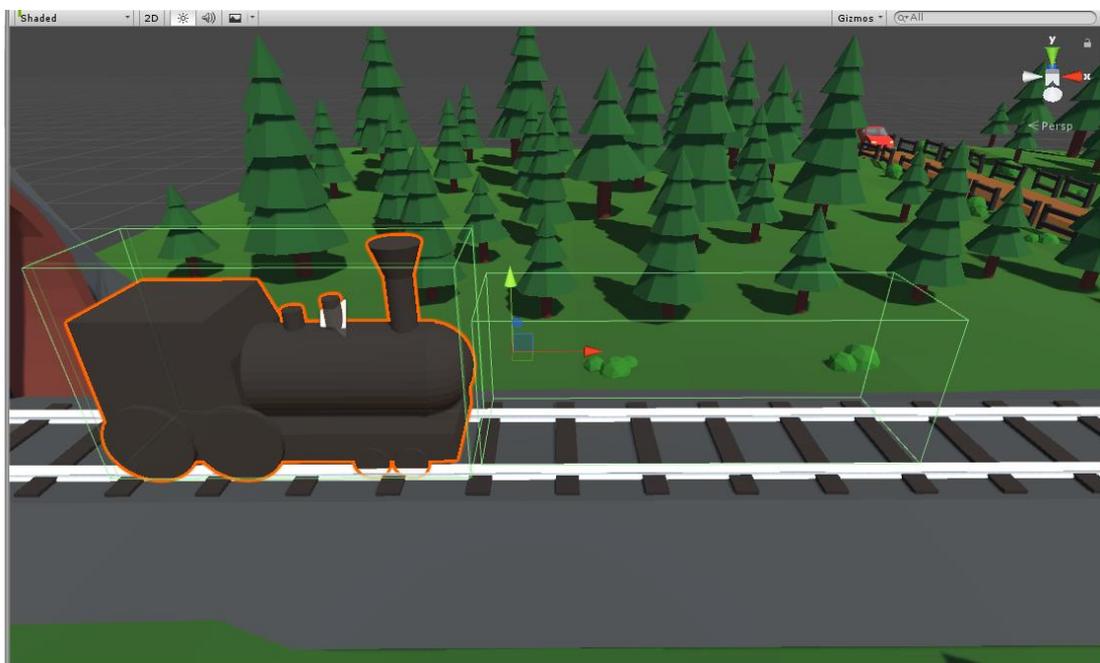


Abbildung 4.19: Screenshot vom BoxCollider (Grüner Kasten vor dem Zug)

Der BoxCollider wird als Trigger gesetzt, um eine Kollision zu vermeiden. Damit kann geprüft werden, ob sich das Auto in diesem Bereich befindet und dementsprechend agiert werden. Dieser Trigger wird ebenfalls genutzt, um in den aktiven Szenen zu bestimmen, dass der Spieler den Unfall nicht rechtzeitig verhindert hat.

Für die aktive Third-Person gibt es zwei Möglichkeiten die Szene erfolgreich abzuschließen: Die TrainWinGameRoutine und die CarWinGameRoutine.

Die TrainWinGameRoutine startet sobald das Auto den StallEventTrigger erreicht hat und lädt nach Ablauf eines Delays das nächste Level. Der Delay ist lang genug, damit diese Bedingung nur eintreten kann, wenn der Spieler den Zug von den Gleisen gehoben hat. Ansonsten tritt vorher die andere Siegbedingung oder die *GameOver*-Bedingung ein. Die zweite Möglichkeit ist es, das Auto von den Schienen zu heben. Sobald das Auto den StallEventTrigger verlässt, wird die CarWinGame-Routine gestartet. Auch diese wechselt bei Erfolg mit einem Delay zur nächsten Szene.

Wird das Auto vor dem Szenenwechsel auf die Schienen gesetzt und in Gefahr gebracht, wird der Szenenwechsel verhindert und die Routine wird beim erneuten Hochheben wieder gestartet.

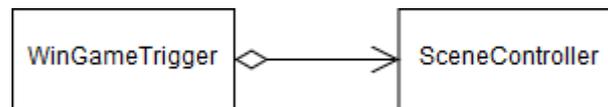


Abbildung 4.20: UML des WinGameTrigger Skripts

Die aktive First-Person Szene entscheidet das WinGameTrigger Skript. Dieses führt einen Szenenwechsel zur End Szene aus, wenn der Spieler das Auto erfolgreich von den Gleisen gefahren hat und wird durch einen weiteren Trigger festgestellt (siehe Abbildung 4.22). Da der Motor des Autos, wie im Mittelteil beschrieben, ausgeht, wenn das Auto auf die Schienen fährt, muss der Spieler das Auto erneut starten, um von den Schienen fahren zu können. Da der Schlüssel sich beim Ausgehen des Motors nicht automatisch zurückdreht, muss der Spieler den Schlüssel zunächst in die Ausgangsposition drehen, um danach den Motor zu starten. Das sorgt für Spannung und erschwert dem Spieler das Starten des Motors.

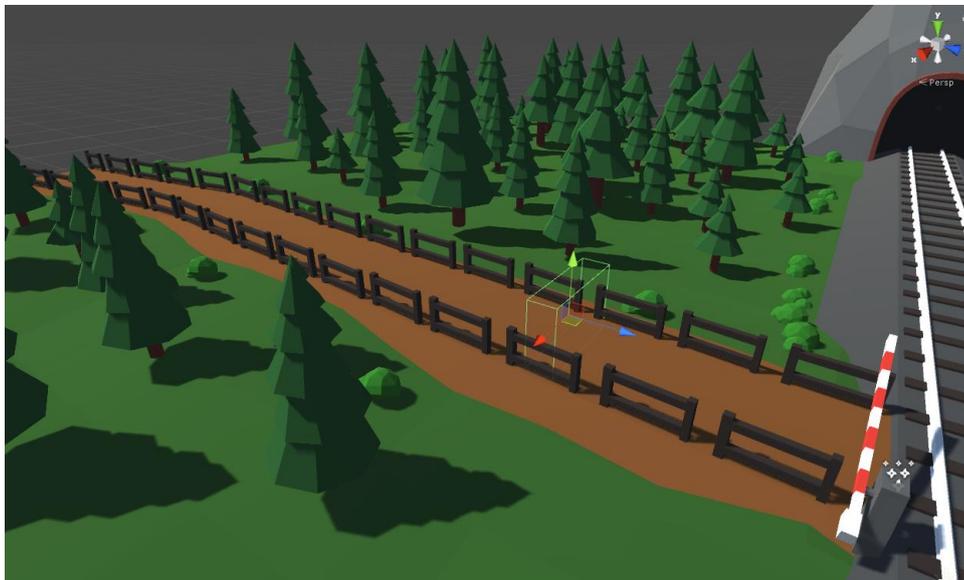


Abbildung 4.21: WinGameTrigger

5 Nutzerstudie

In diesem Kapitel wird die Nutzerstudie beschrieben, die mittels verschiedener Probanden durchgeführt wurde. Die Nutzerstudie umfasst Daten, die mittels eines Fragebogens von den Probanden erhoben werden.

5.1 Versuchsbeschreibung

Der Versuch wurde von insgesamt 13 Probanden durchgeführt. Die Probanden waren im Alter zwischen 25 und 70 Jahren und wurden aus einem Bereich von unerfahrenen Personen im Hinblick auf Videospiele und VR, sowie Personen, die das Medium Videospiele regelmäßig nutzen, ausgewählt. Die Aufteilung der Probanden in diesen Bereichen war ausgeglichen. Jeder der Probanden hat alle Szenarien mindestens einmal durchlaufen, bevor sie den Fragebogen ausgefüllt haben. Die Dauer eines Durchlaufs betrug im Schnitt 10 Minuten. Die Probanden kannten die Szenen vor dem Test nicht.

5.2 Ziel

Das Ziel dieser Studie ist es heraus zu finden, welche Auswirkungen die verschiedenen Szenen auf das Spielerlebnis haben. Daraus lässt sich anschließend ermitteln, für welche Anwendungszwecke diese gut geeignet sein können. Außerdem lässt sich durch die Studie herausfinden, welche Szenen wie stark Simulator Sickness auslösen oder sogar vermeiden.

5.3 Auswertung

In diesem Abschnitt wird der Fragebogen sowie die Auswertung der Daten erläutert. In den dazu gezeigten Diagrammen werden aus Platzgründen Abkürzung für die einzelnen Szenen verwendet. Die Abkürzung setzen sich wie folgt zusammen und werden in allen Diagrammen verwendet:

P1P – Perspektive 1 Passiv - beschreibt die Third-Person Perspektive ohne Interaktionen

P1A – Perspektive 1 Aktiv - beschreibt die Third-Person Perspektive mit Interaktion

P2P – Perspektive 2 Passiv - beschreibt die First-Person Perspektive ohne Interaktion

P2A – Perspektive 2 Aktiv - beschreibt die First-Person Perspektive mit Interaktion

Der Fragebogen ist in zwei Teile unterteilt. Jeder Punkt hat folgende vier Antwortmöglichkeiten.

- Trifft nicht zu
- Trifft eher nicht zu
- Trifft eher zu
- Trifft voll zu

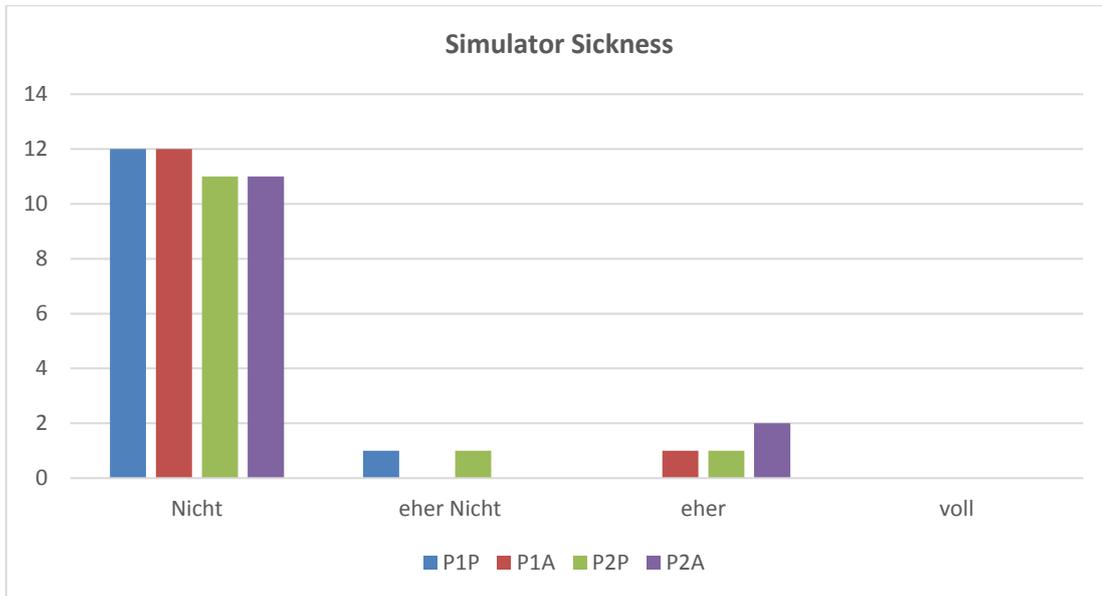
Die Probanden mussten sich für eine negative oder positive Antwort entscheiden und erhielten somit keine Möglichkeit eine neutrale Antwort zu geben. Zudem bekommt man über die Abstufungen der negativen und positiven Antworten ein genaueres Bild, wie stark sich der jeweilige Punkt auf den Probanden auswirkt, ohne zu ausschweifend zu werden.

Der erste Teil besteht aus Punkten, die sich auf das Wohlbefinden der Spieler beziehen.

- Diese Szene hat Übelkeit verursacht
- Diese Szene hat Desorientierung verursacht
- Diese Szene hat Schwindel verursacht

Diese Punkte sind interessant, um sich ein Bild über Simulator Sickness innerhalb der verschiedenen Perspektiven und Interaktionen zu machen.

Im folgenden Vergleich werden die einzelnen Kategorien für jede Szene verglichen, um Informationen darüber zu erhalten, welche Auswirkung diese haben.



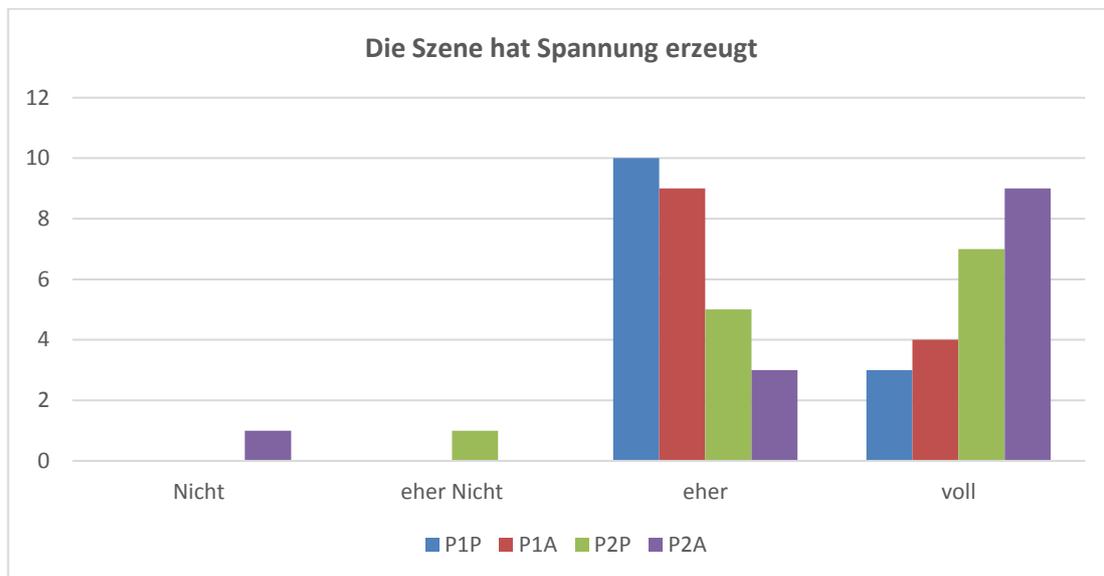
Dieses Diagramm repräsentiert alle folgenden Punkte: Übelkeit, Desorientierung und Schwindel.

Simulator Sickness setzte bei den Probanden nur sehr wenig ein. Die Mehrheit verspürte weder Übelkeit noch Schwindel oder Desorientierung. Die wenigen Abweichung die sich zeigen, sind auf einzelne Personen zurückzuführen. Anhand dieses Ergebnisses kann man erkennen, dass Simulator Sickness eine subjektive Messung ist. Die Gründe der abweichenden Werte könnten erhöhte Anfälligkeit auf Simulator Sickness, Länge des Nutzertests bei unerfahrenen Probanden und gesundheitliche Verfassung sein. Die größten Probleme zeigten sich in der aktiven First-Person Szene. Die passive First-Person Szene bereitete etwas weniger Probleme, während die beiden Third-Person Szenen keine Probleme bereiteten. Es zeigt sich, dass die First-Person Perspektive einen größeren Einfluss auf Simulator Sickness haben kann, als eine Third-Person Perspektive.

Der zweite Teil des Fragebogens bezieht sich direkt auf das Spiel und wie die Szenen auf die Probanden gewirkt haben. Hier hatten die Probanden vier Punkte zu beantworten.

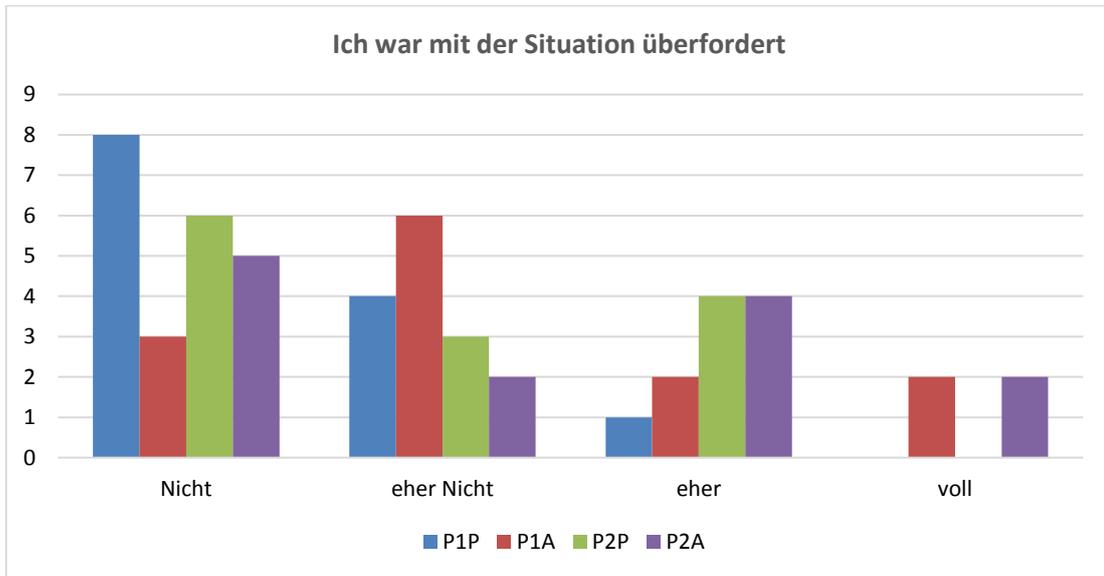
- Die Szene hat Spannung erzeugt
- Ich habe mich hilflos gefühlt
- Ich war mit der Situation überfordert
- Ich könnte mir eine VR Applikation in dieser Perspektive vorstellen

Anhand dieser Punkte soll herausgefunden werden, welche Auswirkung die verschiedenen Szenen auf das Storytelling haben und welche Einsatzmöglichkeiten sich eignen.



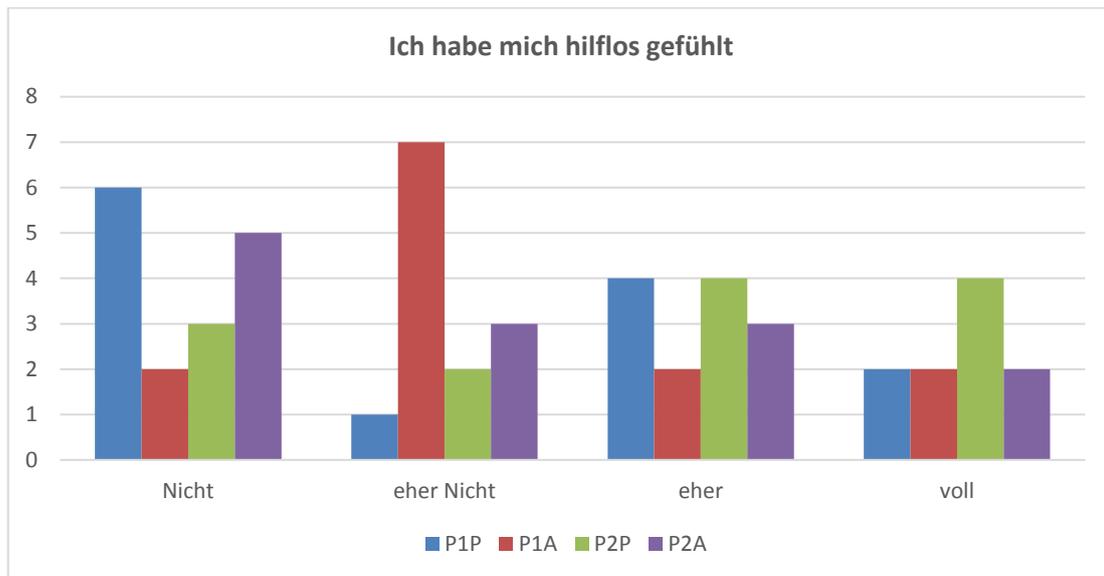
Wichtig für diese Szenen ist die Spannung, da diese ein Kernelement der Geschichte ist. In diesem Diagramm sieht man, wie sich die Spannung auf die Probanden ausgewirkt hat. Klar zu sehen ist, dass sich die First-Person Szenen deutlich von den Third-Person Szenen absetzen nur zwei Stimmen wurden für wenig bis gar keine Spannung abgegeben. Besonders in der aktiven First-Person Szene haben 9 von 13 Probanden die Spannung mit voller Zustimmung bewertet. Es wurde erwartet, dass die Spannung sich in der letzten Szene nochmal anhebt, da der Spieler nun den Druck hatte, sich selbst von den Schienen zu befreien. Das fügte ein zusätzliches Spannungsmoment hinzu.

In den Third-Person Szenen haben die meisten Probanden mittelmäßige Spannung verspürt. Das war auch das erwartete Ergebnis, da der Spieler die Geschichte von außen betrachtet und mehr Distanz zu dem Szenario hat als bei den First-Person Szenen.



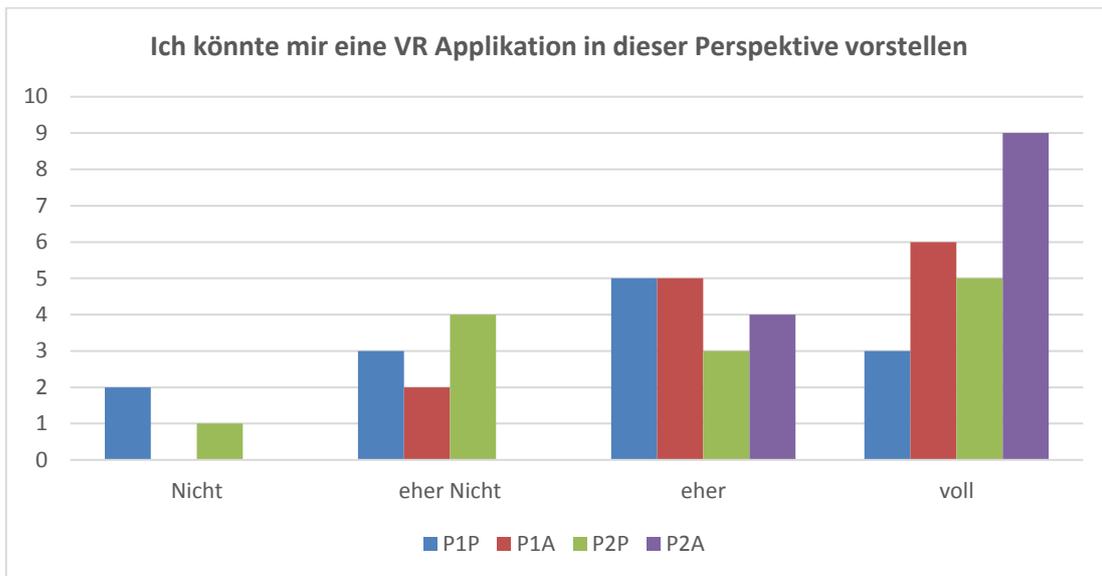
In diesem Diagramm wird gezeigt, ob die Probanden mit der Situation überfordert waren. Hier wurden sehr unterschiedliche Meinungen abgegeben. Die passive Third-Person Szene tendiert hier zu wenig bis keiner Überforderung. Dieses Ergebnis wurde erwartet, da der Spieler hier das Szenario nur von außen betrachten könnte.

In der aktiven Third-Person Szene verspürte die Mehrheit der Probanden leichte Überforderung. Das zeigt, dass die Spieler erst mit der Steuerung und den Interaktionsmöglichkeiten zurechtfinden mussten oder sogar auf Hilfe angewiesen waren. Die Ergebnisse für die First-Person Szenen zeigen, dass die Probanden etwas stärker überfordert waren. Das könnte die fehlende Erfahrung in einer VR Umgebung widerspiegeln, da nun der Fokus nicht mehr, wie in den Third-Person Szenen, direkt vor den Spielern lag, sondern es zu allen Seiten was zu entdecken gab. Des Weiteren gab es in der aktiven First-Person Szene mehr zum Interagieren, was auch zur Überforderung führen kann.



Hier sehen wir, wie stark sich bei den Probanden Hilflosigkeit gezeigt hat. Die Ergebnisse sind stark gemischt und zeigen wie unterschiedlich die Probanden auf die Szene reagiert haben. Die Hälfte der Probanden haben in der passiven Third-Person Szene keine, sowie in der aktiven Third-Person Szene, wenig Probleme gezeigt. Der Grund warum die aktive Szene mehr Probleme bereitet hat, lässt sich darauf schließen, dass die Probanden sich mit der zusätzlichen interaktiven Ebene erst zurechtfinden mussten.

Die First-Person Szenen zeigen ein recht durchwachsendes Bild, wobei die aktive Szene weniger Probleme bei den Probanden bereitet hat. Es zeigt sich, dass besonders die passive First-Person Szene Hilflosigkeit auslöst, da die Probanden hier den Verlauf der Geschichte nicht verändern können und der Szene ausgesetzt sind.



Dieses Diagramm zeigt, ob das Projekt Interesse bei den Probanden geweckt hat. Es ist klar zu sehen, dass die aktive First-Person Szene mit Abstand am besten bei den Probanden ankam. Der größte Teil stimmte für volle Zustimmung.

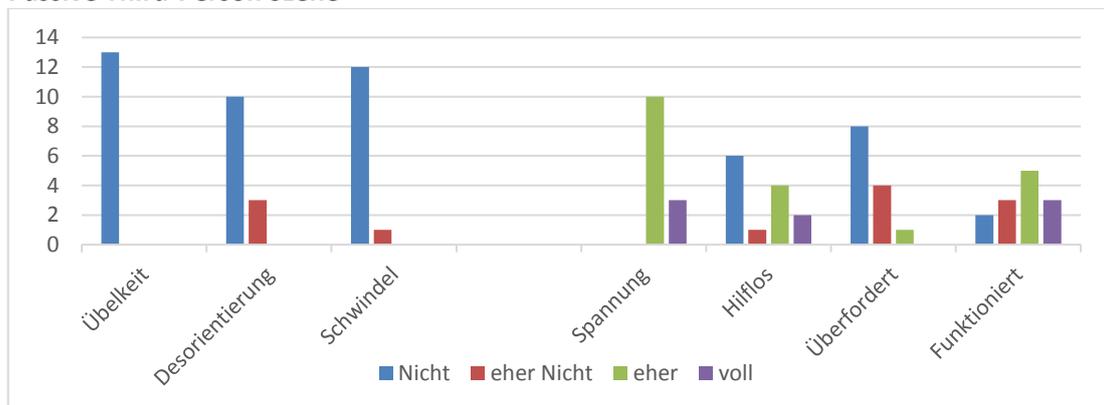
Die aktive Third-Person Szene ist ebenfalls gut bei den Probanden angekommen und zeigt ein recht ausgeglichenes Bild zwischen mittlerer und voller Zustimmung. Die beiden passiven Szenen haben ein ähnliches Muster und schneiden etwas schlechter ab als die aktiven Szenen. Einigen Probanden sagten die Szenen wenig bis gar nicht zu.

Man kann hier recht gut erkennen, dass die Probanden auf eine virtuelle Welt besser reagieren, wenn sie dort mit Objekten interagieren können, sind aber nicht abgeneigt, eine virtuelle Welt als passiver Zuschauer auf sich wirken zu lassen.

5.4 Zusammenfassung

In diesem Abschnitt wird über jede Szene eine kurze Zusammenfassung der ausgewerteten Daten gegeben und mögliche Anwendungszwecke sowie gewonnene Informationen. Die gezeigten Diagramme zeigen eine Übersicht über alle Punkte zu der jeweiligen Szene.

Passive Third-Person Szene

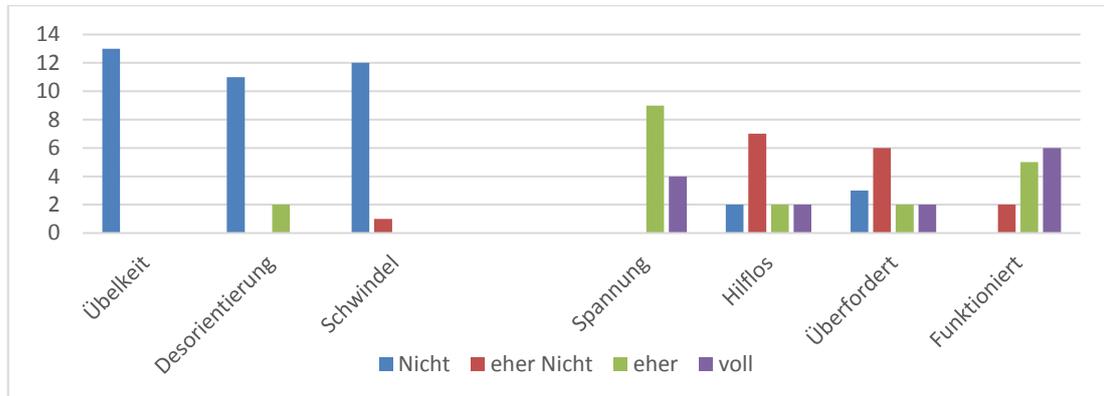


Die Passive Third-Person Szene hat in Punkten Übelkeit, Schwindel und Desorientierung am besten abgeschnitten und bei den Probanden keine Probleme ausgelöst. Daraus lässt sich schließen, dass diese Art von Perspektive sich gut für Einsteiger oder Anwender eignet, die sensible auf Simulator Sickness reagieren.

Diese Szene eignet sich im Vergleich zu der First-Person Szene jedoch weniger gut, um Spannung aufzubauen, da die Nutzer weniger auf die Spannungsmomente angesprochen haben.

Der Nutzer fühlte sich im Allgemeinen wohl beim Ablauf der Szene und verspürte wenig bis gar keine Desorientierung und nur geringe Hilflösigkeit. Besonderes Merkmal an dieser Szene ist, dass der Nutzer keine Möglichkeit zu interagieren hat und auf Grunde dessen keine Probleme mit Steuereingaben hat. Aus diesem Ergebnis lässt sich herleiten, dass diese Art von Szenendarstellung für sachliche Zwecke wie Veranschaulichungen (z.B. Darstellung eines Produktes auf einer Messe), in dem der Nutzer sich sofort zurechtfinden soll, besonders gut geeignet ist.

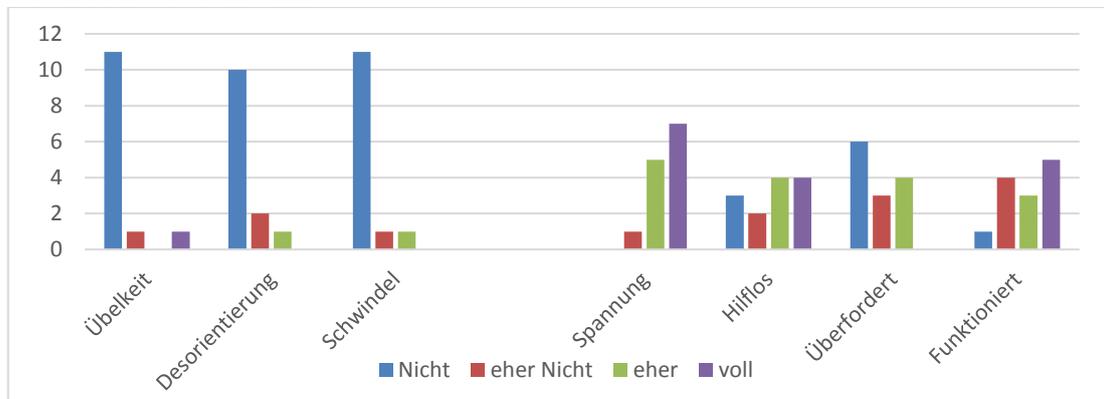
Aktive Third-Person Szene



Die aktive Third-Person Szene hat in Punkten Übelkeit, Schwindel und Desorientierung am zweitbesten abgeschnitten und bereitete nur bei Desorientierung etwas mehr Probleme bei den Probanden.

Genau wie in der passiven Third-Person Szene lässt sich sagen, dass die Art von Szenendarstellung sich gut für Einsteiger eignet ist, um deren Wohlbefinden möglichst hoch zu halten. Durch das Hinzufügen von Interaktionen ändert sich kaum was am Empfinden für Spannung bei den Probanden, jedoch fühlten sich die Nutzer stärker überfordert und hilfloser. Durch die, im Vergleich zu der First-Person Variante, reduzierte Hilflosigkeit und Desorientierung könnte diese Art von Umsetzung gut für Rätsel- und Jump N Run-Spiele geeignet sein, bei denen der schlechtere Spannungsaufbau nicht ins Gewicht fällt.

Passive First-Person Szene



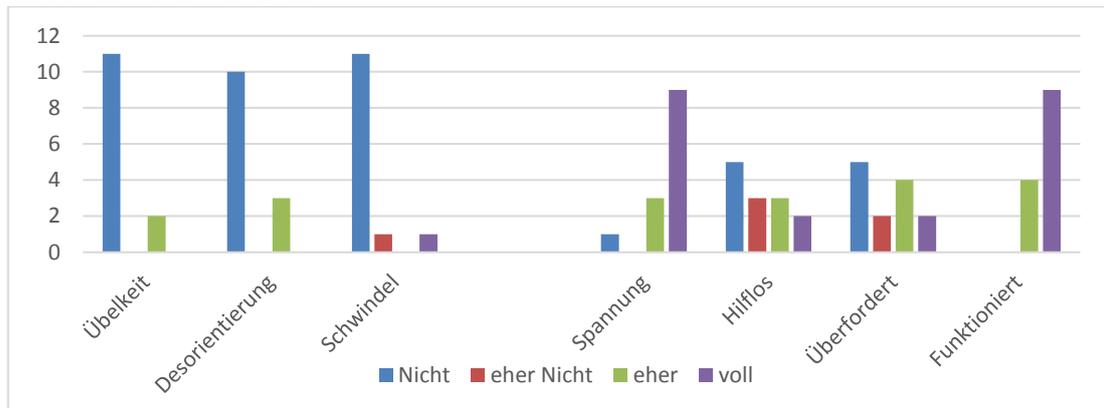
Die passive First-Person Szene hat in Punkten Übelkeit, Schwindel und Desorientierung am drittbesten abgeschnitten und machte nur wenig Probleme bei den Probanden. Die Szene war sehr bewegungsschwach, was den oben genannten Punkten zugutekommt. Es ist anzunehmen, dass sich die Probleme bei schnelleren Szenen verstärken. Der Spannungsbogen hat sich entgegen dem Third-Person-Äquivalent, merklich verbessert und

zeigt, dass sich die Probanden besser in die Szene hineinversetzen konnten. Die Probanden haben starke Hilflosigkeit verspürt und fühlten sich leicht überfordert.

Auch hier kann die Szene gut für Produktvorstellungen oder Führungen (Museum, Firmenvorstellung etc.) genutzt werden, da der Nutzer nicht auf Steuereingaben angewiesen ist und schnell Zugang findet.

Außerdem ist diese Umsetzung gut in kurzen Filmen zu verwenden, bei denen die Spannungsmomente zum Tragen kommen (z.B. in Krimigeschichten oder Horrorfilmen). Diese Filmarten können gut in einem ruhigeren Szenario eingesetzt werden, um die Probleme der Simulator Sickness einzugrenzen und dennoch die verstärkte Spannung und Hilflosigkeit auszunutzen.

Aktive First-Person Szene



Die aktive First-Person Szene hat in Punkten Übelkeit, Schwindel und Desorientierung am schlechtesten abgeschnitten. Die Probanden fanden die Szene sehr spannend und gaben fast alle volle Zustimmung. Man merkt, dass das Erzeugen von Spannung hier voll funktioniert hat. Die Hilflosigkeit hielt sich in Grenzen und beruhte, genau wie bei der Überforderung, zum Großteil darauf, dass die Probanden nicht mit der Steuerung vertraut waren oder zurechtkamen. Die Probanden sprachen sehr positiv auf diese Szene an.

Für spannende Szenarien funktioniert diese Art von Szene besonders gut. Die Geschichte hat hier ihre volle Wirkung gezeigt. Im Vergleich zu den anderen Szenen hat diese am besten funktioniert. Dies war das erwartete Ergebnis, da sich bereits viele Spiele einer interaktiven First-Person Perspektive bedienen. Simulator Sickness könnte bei schnelleren Szenen zum Problem werden, allerdings lässt sich hier mit Verfahren gegenwirken. Diese Szene funktioniert am besten für Nutzer mit VR Erfahrung und sollte bei Neueinsteigern am besten in vereinfachter Form oder mit ausführlichem Tutorial angewandt werden.

6 Abschluss

6.1 Fazit

Durch die Auswertung der Nutzerstudie hat sich gut gezeigt, dass die vier Szenen unterschiedliche Wirkungen auf die Nutzer hatten. Auch zeigte sich, dass die Auswirkung sich zwischen verschiedenen Nutzern unterschied. Die aktive First-Person Szene hat am besten in der Studie abgeschnitten. Gerade bei VR Neueinsteigern führte die Durchführung der Studie zu großer Begeisterung und hat Interesse geweckt.

Allerdings ist diese Studie auf Grund der kleinen Fallzahl nicht als repräsentativ anzusehen. Im Allgemeinen denk ich, dass das Projekt erfolgreich war und meinen Erwartung entsprach. Mit mehr Zeit und einem größeren Projekt könnte man jedoch bessere und genauere Ergebnisse erzielen. Dies war leider im Rahmen dieser Arbeit nicht umzusetzen.

6.2 Ausblick

Mit den steigenden Verkaufszahlen von VR-Geräten und der sich stetig verbessernden Qualität wird VR wohl auch in der Zukunft ein Thema bleiben²⁶. Die hohen Kosten der VR-Geräte und die hohe Hardware Anforderung ist zurzeit noch ein Faktor, der das Wachstum einschränkt. Speziell VR-Geräte für den PC sind im Preis schon gesenkt worden und mit weiteren Versionen ist davon auszugehen, dass die Preislage, gerade für ältere Modelle, noch weiter sinkt.

Die Ergebnisse dieser Arbeit haben gezeigt, dass die unterschiedlichen Perspektiven und Interaktionsstufen starke Auswirkung auf das Erlebnis des Anwenders haben. Diese Erkenntnis lässt viel Raum für weitere Experimente. Eine Idee zur Erweiterung des Projekts wäre, eine Szene zu entwickeln, die je nach Situation zwischen verschiedenen Perspektiven schalten oder dem Spieler die Wahl der Perspektive lässt. Dieses Prinzip lässt sich auch auf die Interaktionsmöglichkeiten anwenden. Dem Nutzer werden je nach Situation mehr oder weniger Möglichkeiten gegeben auf die Umgebung einzugehen. Die Möglichkeit, Perspektive und Interaktion als Stilmittel einzusetzen, ist noch längst nicht erschöpft und lässt hoffen auch für zukünftige Projekte mehr Ideen hervorzubringen.

²⁶ Quelle: <https://arstechnica.com/gaming/2017/11/more-than-a-fad-vr-headset-sales-are-slowly-creeping-higher/>

7 Literaturverzeichnis

- [1] STEVEN M. LAVALLE, „*Virtual Reality*“, Cambridge University Press, November 2016
- [2] FREDERICK P. BROOKS, „*What’s real about virtual reality*“, in IEEE Computer Graphics and Applications, Band 19, Ausgabe 6, Seite 16-27, Nov-Dec 1990
- [3] RONGHAI WANG, JUNGFENG YAO, LIN WANG, XIAOHAN LIU, HONGWEI WANG, LILING ZHENG, „*A surgical training system for four medical punctures based on virtual reality and haptic feedback*“, in 2017 IEEE Symposium on 3D User Interfaces, Seite 215-216, März 2017
- [4] STEFAN MARKS, JAVIER E. ESTEVEZ, ANDY M. CONNOR, „Towards the Holodeck: Fully Immersive Virtual Reality Visualisation of Scientific and Engineering Data“, Proceedings of the 29th International Conference on Image and Vision Computing New Zealand, 2014
- [5] IVAN E. SUTHERLAND, „*A head-mounted three dimensional display*“, In AFIPS Fall Joint Computer Conference Teil 1, 1968, Seite 757-764.
- [6] S.S. FISHER, E.M. WENZEL, C. COLER, M.W. MCGREEVY, „*VIRTUAL INTERFACE ENVIRONMENT WORKSTATIONS*“, PROCEEDINGS OF THE HUMAN FACTORS SOCIETY, Seite 92-95, 1988
- [7] PARTH RAJESH DESA, POOJA NIKHIL DESAI, KOMAL DEEPAK AJMERA, KHUSHBU MEHTA, „*A Review Paper on Oculus Rift-A Virtual Reality Headset*“, in International Journal of Engineering Trends and Technology (IJETT), Band 13, Seite 175-179, Juli 2014
- [8] RANDY PAUSCH, JON SNODDY, ROBERT TAYLOR, SCOTT WATSON, ERIC HASELTINE, „*Disney’s Aladdin: first steps towards storytelling in virtual reality*“, In SIGGRAPH '96 Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ACM, Seite 193-203, 1996
- [9] PATRICK SALAMIN, DANIEL THALMANN, FRÉDÉRIC VEXO, „*The benefits of third-person perspective in virtual and augmented reality*“, In VRST '06: Proceedings of the ACM symposium on Virtual reality software and technology, ACM, Seite 27–30, November 2006

- [10] JOHN BUCHER, „*Storytelling for Virtual Reality: Methods and Principles for Crafting Immersive Narratives*“, Taylor & Francis Group, 2017
- [11] ALEJANDRO GUARIN DAVILA, „*Virtual Reality Storytelling*“ Master Thesis, Auckland University of Technology, 2017
- [12] MEL SLATER, SYLVIA WILBUR, „*A framework for immersive virtual environments five: Speculations on the role of presence in virtual environments*“, In *Presence: Teleoperators and Virtual Environments*, Band 6, Ausgabe 6, Seite 603-616, Dezember 1997
- [13] M. SLATER, „*A Note on Presence Terminology*,“ *PresenceConnect* 3, Januar 2003
- [14] EUGENIA M. KOLASINSKI, „*Simulator Sickness in Virtual Enviroments*“, Technical Report, Band 1027, Mai 1995
- [15] RICHARD H. Y. SO, W. T. LO, ANDY T. K. HO, „*Effects of Navigation Speed on Motion Sickness Caused by an Immersive Virtual Environment*“, in *Human Factors* Band 43, Ausgabe 3, Seite 452-491, 2001
- [16] THIAGO M. PORCINO, ESTEBAN CLUA, DANIELA TREVISAN, CRISTINA N. VASCONCELOS, LUIS VALENTE, „*Minimizing cyber sickness in head mounted display systems: design guidelines and applications*“, Fluminense Federal University Institute of Computing, 2017
- [17] MARK H. DRAPER, ERIK S. VIIRRE, THOMAS A. FURNESS, VALERIE J. GAWRON, „*Effects of Image Scale and System Time Delay on Simulator Sickness within Head-Coupled Virtual Environments*“, in *Human Factors*, Band 43, Ausgabe 1, Seite 129-146, 2001
- [18] M. OLANO, J. COHEN, M. MINE, AND G. BISHOP, "Combating rendering latency," in *Proceedings of the 1995 symposium on Interactive 3D graphics*, 1995, Seite 19-24
- [19] CASSANDRA N. ALDABA, PAUL J. WHITE, AHMAD BYAGOWI, „*Virtual reality body motion induced navigational controllers and their effects on simulator sickness and pathfinding*“, in *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE*, Seite 4175-4178, Juli 2017

8 Anhang

Fragebogen zur Bachelorarbeit

Storytelling in VR – Analyse der Wirkung von Interaktion und Perspektive

Allgemeines

Geschlecht	<input type="checkbox"/> Männlich	<input type="checkbox"/> Weiblich	
Alter	<input type="checkbox"/> < 25 Jahre	<input type="checkbox"/> zwischen 25 – 50 Jahren	<input type="checkbox"/> > 50 Jahre
VR Erfahrungen	<input type="checkbox"/> Keine	<input type="checkbox"/> Wenig	<input type="checkbox"/> Viel

Perspektive 1 ohne Interaktion

	Trifft nicht zu	Trifft eher nicht zu	Trifft eher zu	Trifft voll zu
Diese Szene hat Übelkeit verursacht	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diese Szene hat Desorientierung verursacht	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diese Szene hat Schwindel verursacht	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die Szene hat Spannung erzeugt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich habe mich hilflos gefühlt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich war mit der Situation überfordert	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich könnte mir ein VR Film in dieser Perspektive vorstellen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Perspektive 1 mit Interaktion

	Trifft nicht zu	Trifft eher nicht zu	Trifft eher zu	Trifft voll zu
Diese Szene hat Übelkeit verursacht	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diese Szene hat Desorientierung verursacht	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Diese Szene hat Schwindel verursacht	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die Szene hat Spannung erzeugt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich habe mich hilflos gefühlt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich war mit der Situation überfordert	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich könnte mir ein VR Spiel in dieser Perspektive vorstellen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Perspektive 2 ohne Interaktion

	Trifft nicht zu	Trifft eher nicht zu	Trifft eher zu	Trifft voll zu
Diese Szene hat Übelkeit verursacht	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diese Szene hat Desorientierung verursacht	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diese Szene hat Schwindel verursacht	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die Szene hat Spannung erzeugt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich habe mich hilflos gefühlt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich war mit der Situation überfordert	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich könnte mir ein VR Film in dieser Perspektive vorstellen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Perspektive 2 mit Interaktion

	Trifft nicht zu	Trifft eher nicht zu	Trifft eher zu	Trifft voll zu
Diese Szene hat Übelkeit verursacht	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diese Szene hat Desorientierung verursacht	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diese Szene hat Schwindel verursacht	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die Szene hat Spannung erzeugt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich habe mich hilflos gefühlt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich war mit der Situation überfordert	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich könnte mir ein VR Spiel in dieser Perspektive vorstellen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den _____