

Bearbeitungsbeginn: 01.03.2017

Vorgelegt am: 26.09.2017

Thesis

zur Erlangung des Grades

Bachelor of Science

im Studiengang Medieninformatik

an der Fakultät Digitale Medien

Maximilian Kanisch Matrikelnummer: 245750

Markus Weiß Matrikelnummer: 247652

Integration realer Objekte in die virtuelle Welt

Eine Untersuchung der Auswirkungen von visuellem Feedback, haptischem Feedback und der Latenzzeit auf die Immersion

Erstbetreuer: Prof. Christoph Müller

Zweitbetreuer: Prof. Dr. Matthias Wölfel

Inhalt

Abstract	4
Einleitung	5
Was ist VR?	6
Seit wann gibt es VR und wie hat es sich entwickelt?	8
Anwendungsbereiche	9
VR-Systeme.....	10
Hardware.....	12
Oculus Rift	14
HTC Vive	16
Peripherie	17
Leap Motion	18
Immersion und Präsenz	21
Was ist Immersion?	21
Wie lässt sich Immersion aufbauen?	22
Was stört die Immersion?.....	23
Haptisches Feedback	24
Wie wird haptisches Feedback realisiert?	25
Latenzzeit.....	26
Vuforia.....	29
Voruntersuchung.....	36
Resultat der Voruntersuchung.....	36
Kombinieren der Devices.....	37
Erstellung eines Vuforia Multitargets	42
Demografischer Überblick der Testprobanden	43
Untersuchung der Immersion durch ein VR-Video.....	45
Problemstellung	46
Versuchsaufbau	47
Versuchsablauf	48
Versuchsauswertung.....	50
Untersuchung der Immersion in einer VR-Anwendung mit visuellem Feedback	53
Problemstellung	54
Versuchsaufbau	55

Versuchsablauf	57
Versuchsauswertung	58
Untersuchung der Immersion in einer VR-Anwendung mit haptischem Feedback	61
Problemstellung	62
Versuchsaufbau	63
Versuchsablauf	64
Versuchsauswertung	65
Untersuchung der Immersion in einer VR-Anwendung mit erhöhter Latenzzeit	67
Problemstellung	68
Versuchsaufbau	69
Versuchsablauf	72
Versuchsauswertung Latenzversuch	74
Messung mit 50 Bildern pro Sekunde	75
Messung mit 40 Bildern pro Sekunde	78
Messung mit 30 Bildern pro Sekunde	80
Messung mit 20 Bildern pro Sekunde	81
Messung mit 10 Bildern pro Sekunde	82
Gesamtauswertung des Fragebogens – Gegenüberstellung und Auflistung	83
Resümee und Ausblick	92
Fazit	94
Literaturverzeichnis	96
Onlinequellen	96
Abbildungsverzeichnis	97
Tabellenverzeichnis	99
Eidesstattliche Erklärung	100
Arbeitsaufteilung	100
Anhang	101
Latenzversuch Auswertung	101
Messung mit 50 Bildern pro Sekunde	101
Messung mit 40 Bildern pro Sekunde	102

Abstract

In VR-Anwendungen¹ spielt neben der visuellen Präsentation das haptische Feedback eine wichtige Rolle für den Grad der Immersion und damit für das Eintauchen in die virtuelle Welt. Uns interessiert aufgrund dessen, wie durch den momentanen Stand der Technik dieses immersive Erlebnis verbessert werden kann.

Das Ziel einer ersten Untersuchung war es, zu erforschen, welche Möglichkeiten bestehen, um eine verbesserte Immersion durch haptisches Feedback in VR-Räumen zu ermöglichen.

Hohe Latenzzeiten haben große Auswirkungen auf den Grad der Präsenz und Immersion in virtuellen Räumen. Daher war das Ziel der zweiten Untersuchung, festzustellen welche Übertragungszeiten gewährleistet sein müssen, um ein natürliches, immersives Spielerlebnis zu gewährleisten.

Zunächst wird die Vorauswahl der technischen Systeme im Theorieteil dargestellt und begründet. Diese misst sich an ihrer Eignung für die Integration von haptischem Feedback. Die Eignung wird einerseits durch Eigenuntersuchung, wie auch Literaturrecherche verifiziert. Das praktische Vorgehen besteht darin, VR-Systeme mit einer Spiele-Engine zu verbinden und anhand von vorab festgelegten Kriterien ihre Eignung abzuwägen.

Nach Abschließen dieser Vorauswahl, wird mit der empirischen Untersuchung zum haptischen Feedback begonnen. Das heißt, es wird eine VR-Umgebung erstellt, die eine reale Oberfläche nutzt um dem User haptisches Feedback im VR-Raum zu geben.

Im Anschluss wurde eine VR-Umgebung erstellt, in der sich unterschiedliche Latenzzeiten durch das Festlegen der Bildwiederholfrequenz einstellen lassen. Hier sollen die Auswirkungen der unterschiedlichen Latenzzeiten auf die Immersion untersucht werden. Diese Messwerte dienen dazu, eine klare Vorgabe für die Bildwiederholfrequenz zu finden.

Wir versprechen uns davon verwendbare Ergebnisse für die Weiterentwicklung von virtueller Realität. Wir wollen darstellen, wie unverzichtbar haptisches Feedback in VR-Umgebungen für die Immersion ist und wollen aufzeigen, welche Auswirkungen verschiedene Latenzzeiten auf das Erfahren der virtuellen Realitäten des Anwenders hat.

¹ VR = virtuelle Realität. Nachfolgend immer mit VR abgekürzt.

Einleitung

„Virtual Reality“ ist in den letzten Jahren für mehr Menschen zugänglich geworden als es früher der Fall war. Aus einer teuren technischen Spielerei ist ein massenmarktaugliches Unterhaltungssystem geworden. Mit VR-Systemen wie der *Oculus Rift*, *HTC Vive* oder der *Playstation VR* lassen sich mittlerweile in vielen Wohnzimmern virtuelle Welten erleben, sei es mit High-End-VR-Systemen oder in der mobilen Variante mit dem eigenen Smartphone, so wird prognostiziert, dass die Anzahl der Virtual-Reality-Nutzer von 6,5 Millionen im Jahr 2016 auf 24,4 Millionen im Jahr 2020 steigen wird.²

Die Entwicklung zu immer ausgereifteren Anwendungen, die virtuelle Welten realistischer und immersiver erlebbar machen, sind am Markt zu erkennen. Daher befasst sich die vorliegende Thesis mit den Möglichkeiten, die es aufbauend auf den aktuellen Stand der Technik, möglich machen, durch haptisches Feedback ein immersiveres Erlebnis in virtuellen Welten zu generieren.

Zu Beginn werden einige grundlegende Begriffe zum Thema „Virtual Reality“ erläutert und ein Einblick in den technischen Stand und die bereits heute bestehenden verschiedenen Anwendungsbereiche gegeben. Es wird gezeigt, wie durch aktuelle Technik bereits auf haptisches Feedback zurückgegriffen wird und mit Hilfe der Untersuchungen wird ein Ausblick auf die Verbesserungsmöglichkeiten der Immersion durch haptisches Feedback gegeben.

Des Weiteren wird auf die Latenz eines VR-Systems eingegangen und erforscht, welche Auswirkungen diese auf das Erlebnis in der virtuellen Umgebung hat.

² eMarketer. n.d. *Prognose zur Anzahl der Virtual-Reality-Nutzer weltweit von 2016 bis 2020 (in Millionen)*. Statista. Zugriff am 11. September 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/426237/umfrage/prognose-zur-anzahl-der-aktiven-virtual-reality-nutzer-weltweit/>

Was ist VR?

Der Begriff der „Virtuellen Realität“, in Kurzform „VR“ beschreibt eine computergenerierte Umgebung in der sich, im Idealfall, frei bewegt und mit der interagiert werden kann. (vgl. Vince, 2004, S. 1-2)

Objekte in der virtuellen Welt können berührt und manipuliert werden. Die Sinneseindrücke entsprechen dabei denen der Realität. So ist die *Idealvorstellung* einer virtuellen Realität, die auch schon im Jahr 1964 von E. Sutherland definiert wurde, folgende:

„The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal. With Appropriate programming such a display could literally be the Wonderland into which Alice walked.“ (Sutherland, 1965)

Ein virtuelles System bietet durch eine besonders natürliche und intuitive Form der Interaktion mit der simulierten dreidimensionalen Umgebung eine besondere Art der Mensch-Maschinen-Interaktion.

Das VR-System besteht dabei aus Hardware und Software, die in der Lage sind, die virtuelle Realität zu erzeugen. Die dargestellte Realität stellt eine virtuelle Welt mit erzeugten Objekten dar, die bestimmte Verhaltensbeschreibungen unterliegen und eine gezielte Anordnung im Raum besitzen. Die vom VR-System dargestellte Welt ist die virtuelle Umgebung für den Nutzer, in der er sich bewegt und mit der er interagiert. (vgl. Dörner *et al.*, 2013, S. 7)

„Um eine perfekte virtuelle Realität zumindest in Ansätzen zu verwirklichen, müssen Reize erzeugt werden, die einen Menschen die virtuelle Realität wahrnehmen lassen.“ (vgl. Dörner *et al.*, 2013, S. 5)

So würden in einer optimalen virtuellen Umgebung alle Reize, die der Mensch wahrnimmt, durch den Computer generiert werden, und zwar in einer Qualität, wie der Mensch sie in der realen Welt gewohnt ist. So hätten also auch Handlungen und Interaktionen den gleichen Effekt wie in der realen Welt.

Da die visuelle Wahrnehmung eines jeden Menschen die Realität auf individuelle Weise interpretiert und es keinen eindeutigen Zusammenhang zwischen der Realität und der visuellen

Wahrnehmung des einzelnen gibt, kann die Wahrnehmung auf verschiedene Weise manipuliert werden. (vgl. Dörner *et al.*, 2013)

So ist der Ansatz von virtueller Realität, eine Realität zu schaffen, die die Sinnesreize soweit täuscht, dass sich der Betrachter mit der Unterscheidung zwischen der virtuellen, computergenerierten Welt und der realen Welt, die ihn umgibt, schwertut und so auch körperlich auf virtuell erschaffene Reize reagiert.

Die Reize, die über die virtuelle Realität angesteuert werden, betreffen zum größten Teil die visuelle Wahrnehmung, da diese die größte Informationsquelle zur Wahrnehmung der Umgebung darstellt. Weitere wichtige Reize, die eine Immersion in der virtuellen Welt verstärken, bedienen die auditive Wahrnehmung, so wie die taktile und die haptische Wahrnehmung, auf die im Kapitel *Hardware* später genauer eingegangen wird. (vgl. Dörner *et al.*, 2013)

Ein weiterer wichtiger Aspekt für eine glaubwürdige virtuelle Welt, in die der Anwender eintauchen kann, ist neben der grafischen, detaillierten Darstellung der Welt, und der virtuellen Präsenz in dieser Welt auch die physische Interaktionsmöglichkeit. So hat der Benutzer der eine VR-Welt betritt, das Bedürfnis, mit dieser interagieren zu können. Dafür müssen Interaktionen möglich sein, deren Auswirkungen in der Welt simuliert und dargestellt werden. Diese Art der Simulation bestimmt das Verhalten der virtuellen Welt. Zum einen werden Aktionen des Menschen, der mit der Welt interagiert, zum anderen aber auch interaktionsunabhängige Dinge wie Wasserbewegungen, Tag-Nacht-Wechsel und ähnliches, simuliert.

Dabei kann entweder versucht werden, die Realität möglichst genau abzubilden, indem sich auf physikalische Gesetze bezogen wird, oder aber es wird die Möglichkeit genutzt, die Welt frei zu gestalten und eigene Verhaltensweisen von Objekten zu definieren, um Welten zu schaffen, die abseits eines perfekten realistischen Abbilds der Realität liegen. (vgl. Dörner *et al.*, 2013)

Seit wann gibt es VR und wie hat es sich entwickelt?

Die Entwicklung virtueller Realitäten fand schon in den 60er Jahren statt, in denen Sutherland im Rahmen seiner Forschung zu immersiven Technologien das Werk „The Ultimate Display“ verfasste und darin erste theoretische Ansätze zum Erleben virtueller Welten festhielt. Mit der Etablierung der heutigen VR-Systeme am Massenmarkt ist die Technologie für alle zugänglich geworden und hat den Status einer technischen Spielerei hinter sich gelassen und ist zu einem ernstzunehmenden Medium geworden.

Dabei kombiniert VR auf technischer Seite mehrere Technologien miteinander, um ein immersives Erlebnis entstehen zu lassen, dass die Möglichkeiten die ein herkömmliches Desktop-System bietet, übersteigt. Um den Anwender in eine virtuelle Realität zu integrieren, werden Head-Mounted-Displays eingesetzt, kurz HMDs. Diese HMDs sind 3D-Displays, mit einem stereoskopischen Sichtfeld das ein Tiefengefühl vermittelt. Diese Displays werden, je nach System, mit unterschiedlichen Motion-Tracking-Systemen ergänzt. Die Motion-Tracking-Systeme übertragen Körper- und Kopfbewegungen des Anwenders im realen Raum auf die virtuelle Welt in der die Positionsveränderung berechnet und anschließend auf dem Display des Anwenders angezeigt werden.

Zur Bedienung und Steuerung in der virtuellen Welt kommen oftmals neue, eigens für die neuen Anwendungsmöglichkeiten erstellte Eingabegeräte zum Einsatz, auf die zu einem späteren Zeitpunkt noch genauer eingegangen wird. (vgl. Parisi, 2015)

Anwendungsbereiche

Der Markt, der für die Masse erschwinglicher VR-Systeme, hat sich in den letzten Jahren stark erweitert. Nachdem 2013 die *Oculus Rift* erschienen ist, sind diesem Beispiel viele Hersteller gefolgt, so auch Samsung mit „*Gear VR*“ und *Google* mit „*Cardboard VR*“ in der Smartphone-Variante, *HTC Vive* und *Playstation VR* im High-End-Bereich. Durch diese Vielfalt an Anbietern und VR-Systemen steht der Zugang zu virtuellen Welten vielen Leuten offen. Dadurch sind besonders in den letzten Jahren die Einsatzgebiete von VR-Systemen enorm gewachsen. (vgl. Parisi, 2015)

In der Vergangenheit war der Zugang zu VR-Systemen für die Öffentlichkeit eingeschränkt. VR-Systeme wurden zum größten Teil nur in wissenschaftlichen Institutionen, der Industrie oder im Militär eingesetzt. Mit der Weiterentwicklung der Head-Mounted-Displays (HMD) haben Hersteller wie *Oculus*, *HTC* und *Sony* das Potential der VR-Technik in der Unterhaltungsindustrie erkannt und mit den aktuellen VR-Systemen, massenmarktaugliche Geräte auf den Markt gebracht, welche das Feld der Anwendungsmöglichkeiten neben Gaming und Unterhaltung auf viele weitere Gebiete erweitern.

So werden auch heute schon VR-Systeme in vielen Branchen auf unterschiedliche Art eingesetzt. So tauchen sie vermehrt in der Industrie, in der Medizin, der Forschung oder der Touristik auf. Beispielsweise werden sie in der Industrie benutzt um Prototypen neuer Autos zu visualisieren oder bei der Herstellung Fertigungsprozesse zu optimieren, Gefahrensituationen zu vermindern und Kosten einzusparen. Große Firmen beginnen VR-Systeme zu nutzen, um Konferenzen in virtuellen Räumen abzuhalten oder Geschäftsdaten mit Hilfe der VR-Technik zu visualisieren. Auch im Verkauf werden VR-Systeme eingesetzt, so kann dem Kunden das Produkt in der virtuellen Realität in 360° präsentiert werden und Kundenwünsche, sowie Änderungen, können direkt übernommen und visualisiert werden.

In der Medizin können Operationen geplant und geübt werden, oder die VR-Technik für Therapiezwecke eingesetzt werden. Ebenso nutzt das Militär VR-Systeme, um ihr Personal auszubilden und für bestimmte Situationen zu trainieren. (vgl. Leopold, H., Untermeier, A., 2017)

VR-Systeme

VR-Anwendungen nutzen unterschiedliche Systeme zur Realisierung der virtuellen Realität. Neben den VR-Brillen, auf die im Kapitel Hardware näher eingegangen wird, gibt es auch raumfüllende VR-Systeme mit fest im Raum angebrachten Displays und Tracking Systemen. Ein Beispiel für diese Art der VR-Technik sind die sogenannten „CAVES“, bei denen der Nutzer komplett von Bildschirmen umgeben ist und die Bewegungen per Tracking-Systemen integriert werden. Diese Form der Interaktion mischt *Augumented Reality* und *Virtual Reality*. Dabei ist es leicht möglich, reale Objekte in die Systeme zu integrieren was zu einem Hohen Grad der Immersion führen kann. (vgl. Jerald, 2016, S. 32-34)

Im Dezember 2013 kam die *Oculus Rift* auf den Markt und hat mit ihrem Erfolg maßgeblich dazu beigetragen, dass sich die weiteren VR-Systeme am Markt etablieren konnten. Finanziert wurde das Ganze zuerst über eine Kickstarter Kampagne, die ihr Ziel nach nur 4 Stunden bereits erreicht hatte. Nach dem Erfolg des Release wurde das Unternehmen 2014 an Facebook verkauft. *HTC* hat sich anschließend mit seinem eigenen VR-System, der *HTC Vive* ebenso am Markt etabliert wie *Sony* mit der *Playstation VR*. (vgl. Staudacher, 2016)

Die Anzahl der VR-Systeme auf dem Markt wächst auch heute noch ständig und neue Anbieter versuchen, sich dem Trend anzuschließen und sich auf dem Markt zu etablieren. So soll die Anzahl der Nutzer von Virtual-Reality Systemen bis zum Jahr 2020 auf 24,4 Millionen Nutzer weltweit ansteigen und bietet somit ein großes Potential für Hardware- und Softwarehersteller.³ Bei der Bekanntheit der VR-Systeme lässt sich zwischen Smartphone-VR-Brillen und High-End-VR-Brillen unterscheiden. Bei den Smartphone-Systemen zählen die Versionen *Samsung Gear VR* und *Google Cardboard* zu den beliebtesten. Im Bereich der High-End-VR-Systeme haben sich die VR-Brillen der *Sony Playstation VR*, *Oculus Rift* und der *HTC Vive* als Marktführer etabliert.⁴

³ eMarketer. n.d. *Prognose zur Anzahl der Virtual-Reality-Nutzer weltweit von 2016 bis 2020 (in Millionen)*. Statista. Zugriff am 11. Juli 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/426237/umfrage/prognose-zur-anzahl-der-aktiven-virtual-reality-nutzer-weltweit/>.

⁴ Statista. n.d. *Welche der folgenden VR-Brillen sind Ihnen bekannt?*. Statista. Zugriff am 11. Juli 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/685856/umfrage/umfrage-zur-bekanntheit-von-ausgewahlten-vr-brillen-unter-gamern-in-deutschland/>.

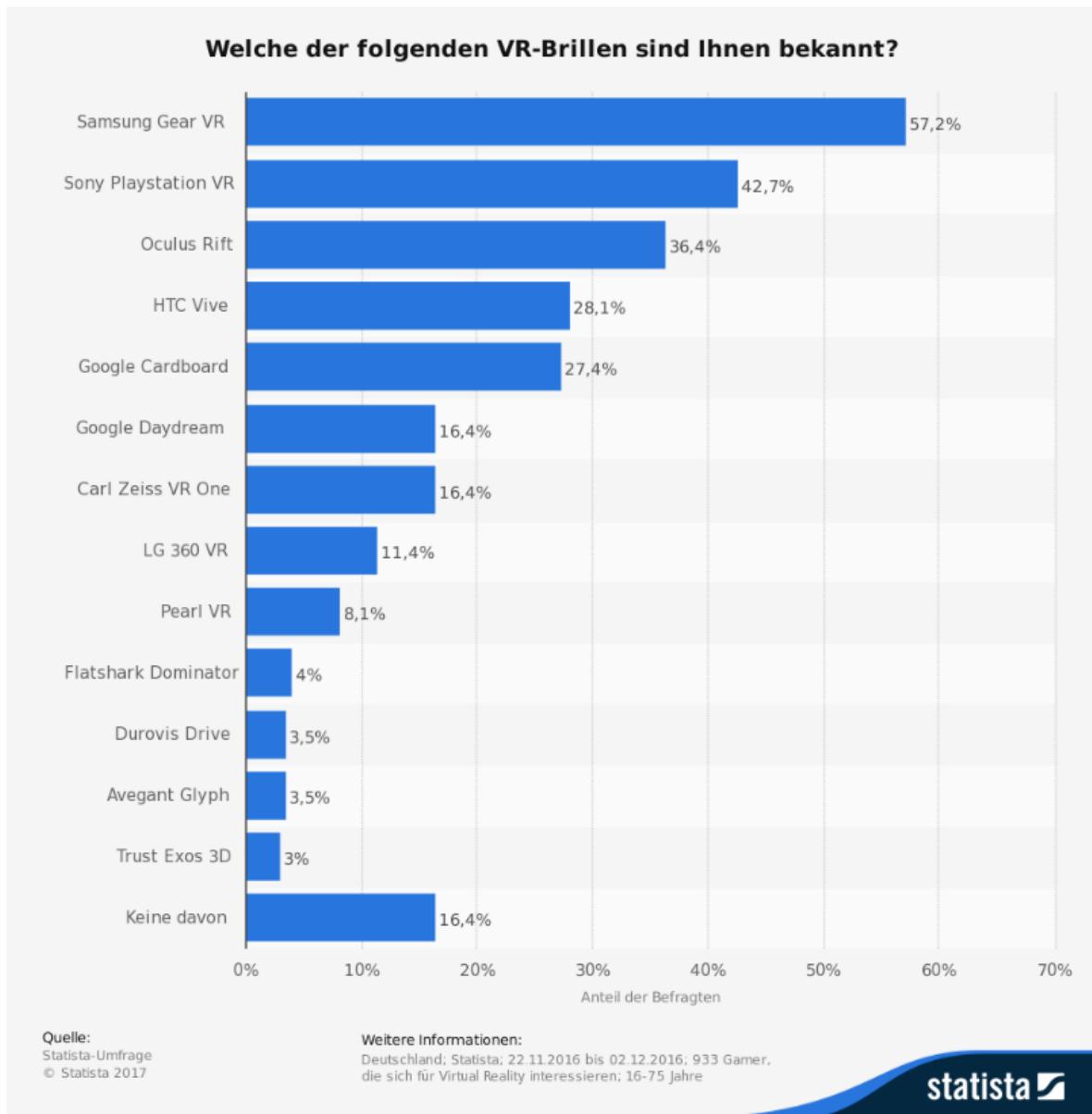


Abbildung 1: <https://de.statista.com/statistik/daten/studie/685856/umfrage/umfrage-zur-bekanntheit-von-ausgewahlten-vr-brillen-unter-gamern-in-deutschland/>

Hardware

„The screen is a window through which one sees a virtual world. The challenge is to make that world look real, act real, sound real, feel real.“ (Sutherland, 1965)

Die Hardware hat das Ziel, dem Anwender das Eintauchen in eine virtuelle Realität zu ermöglichen und ihn diese Realität mit einem hohen Grad der Immersion erfahren zu lassen, so dass er sich präsent in der virtuellen Welt fühlt und sich wie in der realen Welt verhält. (vgl. Dörner *et al.*, 2013, S. 129)

Der Kern eines VR-Systems ist das Head-Mounted-Display (HMD), welches der Anwender auf dem Kopf trägt, um sich die visuelle Ausgabe der Anwendung vor den Augen darstellen zu lassen.

Da sich die Arten der virtuellen Realität in „Virtual Reality“, „Augmented Reality“ und „Mixed Reality“ unterteilen lassen, kommen zur Realisierung verschiedene Varianten des HMDs zum Einsatz. So können die Systeme in „Non-See-Through-“, „Optical-See-Through-“ und „Video-See-Through-“ HMDs unterteilt werden. „Non-See-Through“-HMDs blockieren dabei alle äußeren visuellen Reize und sind damit die beste Variante für immersive VR-Anwendungen. Der visuelle Inhalt, der dabei über das Display wiedergegeben wird, besteht nur aus computergeneriertem Inhalt. „Optical-See-Through“-HMDs ermöglichen es, computergenerierte Inhalte in das visuelle Sichtfeld des Nutzers einzubinden. Damit eignen sie sich am besten für die Umsetzung von AR-Anwendungen. „Video-See-Through“-HMDs sind eine Mischform der beiden zuvor beschriebenen Varianten des HMDs. Dabei wird das reale Bild der Umgebung aufgezeichnet und in Echtzeit mit virtuellen Inhalten ergänzt. (vgl. Jerald, 2016, S. 29-32)

Die Head-Mounted-Displays (HMDs) nutzen die Kombination mehrerer Bildebenen, realistischer optischer Verzerrung und spezieller Linsen, um ein stereoskopisches Bild zu erzeugen, dass die Augen als 3-Dimensional interpretieren. (vgl. Parisi, 2015)

Ein wichtiger Aspekt für die Immersion ist die Größe des Sichtfelds, das die VR-Brille liefert, auch Field-of-View genannt. Das Field-of-View (FOV) gibt den horizontalen und vertikalen Winkel ausgehend vom Auge des Nutzers an. Je größer das Field-of-View ist, desto größer ist die Immersion, da die virtuelle Welt weiträumiger angezeigt wird. Die aktuellen Systeme schaffen ein diagonales Sichtfeld von 110°. (vgl. Dörner *et al.*, 2013)

Ein weiterer wichtiger Aspekt der Hardware ist die Bildqualität der verschiedenen Systeme, diese hat sich seit den ersten Prototypen der neuen VR-Systeme stetig weiter verbessert. Die Auflösung liegt bei *HTC Vive* und *Oculus Rift* mit 2160x1200 Pixeln bei beiden Systemen auf dem gleichen Level. Für die Bildqualität entscheidend sind neben der Auflösung auch die Pixeldichte, Matrixart, Displaygröße, sowie die verbauten Linsen.

Durch Headtracking-Systeme wird die Lage, Neigung, Drehung und Beschleunigung der Kopfrotation erfasst. Wo sich der Anwender im Raum befindet, wird mit dem *Positional Tracker* erfasst. Dieser Tracker berechnet die Position der VR-Brille im Raum und überträgt die Bewegungen in die virtuelle Realität. Solche Bewegungen sind ein wichtiger Faktor der Immersion. (*VR Brillen Vergleich - VR-Nerds*)

Eine Bildwiederholrate von 90 Hz sorgt für eine möglichst geringe Verzögerung zwischen Bewegungen des HMDs und vermindert so das Risiko von auftretender Motion-Sickness beim Anwender. (vgl. Staudacher, 2016)

Oculus Rift

Die *Oculus Rift* kommt im Paket mit der VR-Brille, der Tracking-Kamera, einer Fernbedienung und einem *Xbox-One*-Controller. Das HMD besitzt im Gegensatz zur *HTC Vive* integrierte Kopfhörer.



Abbildung 2: <https://www.oculus.com/press-kit/>

Das Head-Mounted-Displays der Rift hat eine Auflösung von 2160 x 1200 Pixel (1080 x 1200 pro Auge) auf 2 OLED Bildschirmen und einer Bildfrequenz von 90 Hz. (vgl. Prasuethsut, 2016)

Das Erkennen der Position des Spielers im Raum realisiert die *Oculus Rift* über ein Position-Tracking. Dabei wird mit einer Kamera der Standort der im HMD integrierten Infrarot-LEDs ermittelt. Die so ermittelte Bewegung wird in die virtuelle Umgebung übertragen und der Spieler kann sich in einer begrenzten Fläche von ca 1,5 bis 2 Metern frei bewegen. Da dieser Radius relativ begrenzt ist, setzen viele Anwendungen darauf, dass sie im Sitzen oder einer festen stehenden Position gespielt werden. Um das Tracking zu verbessern können weitere Tracking-Kameras eingesetzt werden, die den Trackingbereich der ersten Kamera erweitern um ein rundum Tracking zu ermöglichen. Dadurch werden bessere Positions- und Bewegungsdaten geliefert und somit ist auch das freie Bewegen im Raum ähnlich zur *HTC Vive* möglich. (vgl. Nguyen, 2017)

Die Kopfbewegungen des Benutzers werden direkt in die virtuelle Welt übertragen, dabei simulieren manche Anwendungen Körperbewegungen nach, was zu einem Einbruch der Immersion führen kann wenn die ausgeführte Bewegungen nicht mit dem dargestellten Inhalt übereinstimmen. (vgl. Staudacher, 2016)



Abbildung 3: <https://www.oculus.com/press-kit/>

Die *Oculus Rift* wurde, wie bereits erwähnt, zu Beginn mit einem *Xbox-One*-Controller von Microsoft ausgeliefert aber seit Dezember 2016 ist das Paket auch mit den *Oculus-Touch*-Controllern erhältlich. Diese können auch separat gekauft werden, um ein bereits vorher bestehendes VR-System zu erweitern. Mit den von *Oculus* entwickelten Controllern sollen sich Anwendungen durch intuitive Handbewegungen steuern lassen. (vgl. *Oculus Rift / Oculus*, 2017)

Die Touch-Controller verbessern dadurch deutlich die Bedienbarkeit und die Immersion der *Oculus Rift*, sind komfortabler und fühlen sich natürlicher in der Benutzung an, als die Controller der *HTC Vive*. (vgl. Nguyen, 2017)

HTC Vive

Ebenso wie die *Oculus Rift* setzt das VR-System von *HTC* auf 2 OLED- Displays mit einer Auflösung von 2160 x 1200 Pixeln (1080 x 1200 pro Auge). Das Blickfeld beträgt bei der *HTC Vive* 110°. Durch die von *Valve* entwickelte *Lighthouse*-Technik lässt sich die Position und Bewegung des HMDs sowie der *HTC-Vive*-Controller auf einer Fläche von bis zu 5 x 5 Metern tracken und in den virtuellen Raum übertragen.



Abbildung 4: <https://www.vive.com/us/pr/newsroom-gallery/>

Dabei wird der Raum durch zwei gegenüberliegenden Infrarot-Lasern bis zu 100 Mal pro Sekunde gescannt und die Position und Lage des HMDs und der Controller exakt wiedergegeben. Auch bei diesem System kann die Genauigkeit durch das Anbringen weitere Sensoren verbessert werden. Dadurch kann der Anwender in einem größeren Umfeld im Raum bewegen, als es bei der *Oculus Rift* der Fall ist, was zu einer höheren Immersion führt, da sich der Nutzer durch die freie Bewegungsmöglichkeit und das natürliche Umschauen im Raum schneller in eine Szenerie hineinversetzt. (vgl. McCormick, 2015)

Ein weiterer Vorteil ist die Front-Kamera der *HTC Vive*, da diese es ermöglicht, seine reale Umgebung zu sehen, ohne die Brille abnehmen zu müssen.

Peripherie

Die Eingabegeräte der aktuellen VR-Systeme setzen alle auf eine intuitive Steuerung und einfache Bedienbarkeit. Zwar funktionieren alle Systeme noch über klassisch in der Hand gehaltene Controller, diese wurden aber für die speziellen Anforderungen, die ein VR-System mit sich bringt, angepasst.

Die Eingabemöglichkeiten in einem VR-System sind umfangreicher als in einem normalen Desktop-System. So können Interaktionen per Knopfdruck getätigten werden, ebenso ist es aber auch möglich, komplexe VR-Systeme zu realisieren, die auf Gestenerkennung, Bewegungstracking oder Spracherkennung reagieren.

Um diese Art der Interaktion in einer virtuellen Welt zu integrieren, braucht es Systeme, die die benötigten Daten erfassen. So werden im Falle eines Hand-Trackings die Rotation und die Translation der Hand im Raum erfasst und anschließend werden die Bewegungen und Interaktionen in der virtuellen Welt umgesetzt und die grafische Positionsänderung auf dem visuellen Ausgabegerät dargestellt. Der Umfang und die Genauigkeit eines solchen Trackings ist vom Ziel abhängig, das erreicht werden soll. Zum Beispiel ein einfaches Tracking der Kopfbewegungen, Tracking von Bewegungen im Raum oder genaues Tracking der einzelnen Fingergeelenke.

Leap Motion

Für immersive Erlebnisse sind neue Eingabemöglichkeiten wie sie die *Leap Motion* bietet am besten geeignet. Die *Leap Motion* hat sich mit ihrem Controller auf die Gestensteuerung spezialisiert und ermöglicht ein Hand Tracking für VR Anwendungen.



Abbildung 5: <http://gallery.Leap Motion.com/download/press/LM%20Mount+Vive-Front.png>:

Dadurch ist es möglich, mit den eigenen Händen in der virtuellen Welt auf natürliche Weise zu interagieren. Die Bewegung der Hand und der einzelnen Finger werden durch die in der *Leap Motion* verbauten monochrome Infrarot-Kameras getrackt und auf ein virtuelles Modell der Hände übertragen. Das Kamerasytem kann auf eine Distanz von ungefähr einem Meter die genaue Fingerposition und angezeigte Gesten erkennen und auf das digitale Skelettmodell übertragen werden. Das von drei Infrarot-LEDs ausgestrahlte Licht wird reflektiert und von den Infrarotkameras die mit bis zu 200 Bildern pro Sekunde arbeiten eingefangen und von der *Leap Motion* Software analysiert und interpretiert. So liegt die Genauigkeit der Messung bei 0,7 Millimetern. (vgl. Weichert *et al.*, 2013)

Das Tracking-System wurde für die Gestenerkennung am Desktop-System entwickelt und soll Handbewegungen und komplexe Gesten, die über dem Tisch liegenden Device ausgeführt werden, erkennen und diese auf Anwendungen auf dem PC übertragen. Mit dem Fortschritt in der VR-Technik hat sich auch die Technik der *Leap Motion* weiterentwickelt. So konzentrierte sich

der Hersteller vermehrt auf die Handerkennung in VR-Anwendungen und entwickelte eine verbesserte Hand-Tracking-Engine. (vgl. Bloch, 2016)

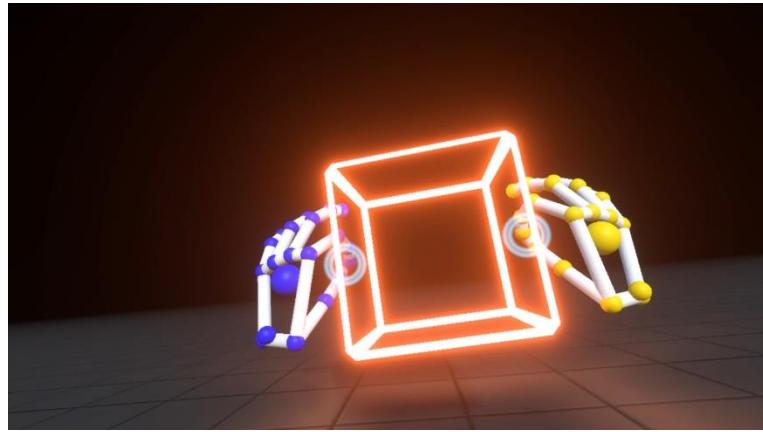


Abbildung 6: <https://www.Leap Motion.com/press#117>

So ist die Anbringung an ein VR-Headset und auch die Integration der Steuerungsmöglichkeiten in VR-Anwendungen mittlerweile fester Bestandteil der Soft- und Hardware der *Leap Motion*.

Da das Trackingsystem der *Leap Motion* ohne zusätzliche Marker funktioniert, ist es störanfälliger bei wechselnden Lichtverhältnissen. Ebenso kann die Zuordnung der virtuellen Modelle durcheinandergeraten, wenn sich mehrere Hände im Bereich der Kamera befinden. Dadurch kann es auch zu einer Unterbrechung des Trackings kommen.

Auf den ersten Blick ist das Benutzen der eigenen Hände in der virtuellen Welt der perfekte Ersatz für in der Hand gehaltene Controller, aber die Interaktion mit den Händen gestaltet sich in der Anwendung schwieriger als vorerst angenommen. So fehlt dem Anwender die Möglichkeit, beim Berühren rein virtueller Objekte ein haptisches Feedback zu bekommen und die Hände müssen für eine genaue Erkennung vor den Sensor gehalten werden, was keine natürliche Körperhaltung garantiert und auf Dauer störend sein kann. Ebenso ist eine konstante Qualität des Trackings und der Gestenerkennung nicht garantiert, was ebenso ein störender Faktor bei der Handsteuerung darstellt.

Wenn das Tracking funktioniert und die Anwendung die Möglichkeiten, die das *Leap Motion*-System bietet, gut einsetzt, kann durch die Integrierung der Hände schnell ein hohes Maß an Präsenz erzeugt werden. Das Interagieren in der virtuellen Welt mit den eigenen Händen ist

intuitiver aber langsamer als die Interaktion mit den herkömmlichen Eingabegeräten. Außerdem muss die neue Form der Interaktion erst vom Anwender erlernt werden, was wiederrum Zeit in Anspruch nimmt und eine Einstiegshürde darstellt.

Mit der Auslegung auf VR wurde die Geschwindigkeit des Trackingvorgangs verbessert und die Latenzzeit reduziert. Verbessert wurde auch die Qualität des Trackings bei Verdeckung in Winkeln, die vorher zu Problemen geführt haben. Auch die Stabilität des Trackings bei schlechten Lichtverhältnissen und beim Berühren oder Halten von Gegenständen wurde verbessert, so, dass sich die Möglichkeiten für VR/AR-Anwendungen erweitert haben. Geplant wird nun auch das der Leap-Controller direkt in VR-Headsets verbaut wird, um dadurch eine direkte Integration des Hand-Trackings in VR-Systemen zu ermöglichen. (vgl. Bloch, 2016)

Immersion und Präsenz

„Ein wesentliches Potential von VR als Mensch-Maschine-Schnittstelle liegt in der Möglichkeit, dem Nutzer die Illusion der Anwesenheit in der dargestellten virtuellen Welt zu suggerieren.“ (Dörner *et al.*, 2013, S. 33)

Für diese Illusion der Anwesenheit sind die Präsenz und die Immersion zwei wichtige Faktoren. Die Präsenz, die der Spieler wahrnimmt, ist das Gefühl, sich selbst in der virtuellen Umgebung zu befinden, so dass diese Umgebung real für ihn wird und Reize aus der realen Welt für ihn ausgeblendet werden. Der Spieler verhält sich, als wäre er in der realen Welt.

Der Grad der Immersion erhöht sich dabei durch mehrere Faktoren. So kommt es beispielsweise darauf an, in welchem Umfang und mit welcher Genauigkeit Sinneseindrücke simuliert werden, zum Beispiel in der visuellen Darstellung der Körperbewegungen, die aus den echten Bewegungen durch technische Systeme integriert werden. Ebenso spielt es eine Rolle, wie weit der Anwender in der Lage ist, mit der virtuellen Realität zu interagieren und wie die Welt auf diese Interaktionen reagiert. Wie weit der Nutzer durch das technische System in eine Welt integriert wird, beispielsweise durch 360°-Tracking oder ein großes Sichtfeld des HMDs, kommt auch auf technische Faktoren wie die Bildwiederholfrequenz, Bildschirmauflösung oder Tracking-Genauigkeit an. (vgl. Jerald, 2016, S. 45)

Was ist Immersion?

„Immersion beschreibt den Eindruck, dass sich die eigene Wahrnehmung in der echten Welt vermindert und die Identifikation in der virtuellen Welt steigert.“ (VR Brillen Vergleich - VR-Nerds, 2017)

Die Immersion beschreibt im Falle von VR-Anwendungen, wie durch die technischen Systeme das Eintauchen in die virtuelle Welt realitätsnah umgesetzt wird. Wenn eine hohe Immersion vorhanden ist, dann ist auch das Gefühl der Präsenz des Anwenders gegeben. So beschreibt Immersion objektiv den Prozess und die Art des Eintauchens und die Präsenz die subjektive Wahrnehmung in der virtuellen Welt. Umso höher der Grad der Immersion ist, umso größer kann auch die Präsenz in der virtuellen Realität sein. Das Gefühl der Präsenz bricht aber sehr

leicht durch äußere Einflüsse ein wie zum Beispiel Störgeräusche aus der realen Umgebung oder der Abbruch des Trackings in der virtuellen Anwendung.

2D-Darstellungen oder Videos bieten nur eine sehr geringe Immersion, wenn aber ein 3D-Video mit einem VR-System betrachtet wird, wächst das Gefühl, in die virtuelle Welt einzutauchen. Es fehlen aber weiterhin Interaktionsmöglichkeiten, die das Gefühl der Immersion steigern. Wenn sich der Anwender in der virtuellen Umgebung frei bewegen und mit ihr interagieren kann dann wird es für den Nutzer zu einem immersiven VR-Erlebnis. (vgl. Leopold, H., Untermeier, A., 2017)

Wie lässt sich Immersion aufbauen?

Bei VR-Anwendungen gibt es technisch viele Möglichkeiten, eine hohe Immersion zu erzeugen. So tragen Faktoren, wie zum Beispiel die übertragenen Bewegungsdaten des Anwenders, das große Sichtfeld des Head-Mounted-Displays und Eingabegeräte, die ein haptisches Feedback liefern, zu einer hohen Immersion bei und lassen damit den Anwender in virtuelle Welten eintauchen.

Immersion entsteht auch immer durch die glaubhafte Gestaltung der virtuellen Welt. Wenn die Welt, in der sich bewegen werden kann in sich stimmig ist und logisch erscheint kann sie schnell vom Anwender verstanden werden und er findet sich in ihr zurecht.

Wenn die Glaubhaftigkeit der virtuellen Welt gegeben ist, lassen sich bei Anwendern auch körperliche Reaktionen feststellen. So lassen sich in simulierten Gefahren- oder Stresssituationen, typische körperliche Reaktionen wie eine schnellere Atmung oder eine erhöhte Herzfrequenz feststellen, wie im späteren Kapitel zum haptischen Feedback noch genauer erläutert wird. Auch wenn sich die Person jeder Zeit im Klaren ist, dass die virtuelle Welt nicht echt ist und die reale Umgebung keine Gefahr darstellt, ist der Mensch dazu bereit, sich soweit in die fiktive Situation hineinzuversetzen, dass die reale Umgebung komplett ausgeblendet wird. Durch diese gewillte Akzeptanz des eigentlich nicht realen, lassen sich einfach glaubhafte virtuelle Umgebungen realisieren, in der sich der Anwender präsent fühlt und die für ihn realistisch erscheinen. (vgl. Dörner *et al.*, 2013, S.33)

Was stört die Immersion?

Das Gefühl in der virtuellen Realität präsent und eingetaucht zu sein, ist von vielen Faktoren abhängig. So kann der Einbruch der Framerate oder eine erhöhte Latenz zum Einbruch der Immersion führen und die virtuelle Welt wird vom Benutzer nicht mehr als realistisch angesehen.

Haptisches Feedback

Für die Wahrnehmung eines Menschen im virtuellen Raum sind die wichtigsten Sinne der visuelle, der akustische und der haptische Sinn. Der visuelle und akustische Sinn wird im Vergleich zu einem Desktop-System bei VR-Systemen nicht in einem größeren Umfang angesprochen. Doch bezogen auf das haptische Erlebnis bietet ein VR-System größere Möglichkeiten als es das haptische Feedback der klassischen Eingabemethoden mit Maus und Tastatur bieten. So reagiert die VR-Umgebung auf 3D-Eingabegeräte oder durch Bewegungen des Anwenders, die anschließend in der virtuellen Welt repräsentiert werden.

Die haptische Wahrnehmung, also das Erfühlen von Gegenständen und ihren Eigenschaften wie die Größe, die Oberflächenstruktur und das Gewicht, ist besonders in VR-Anwendungen ein wichtiger Aspekt für die Präsenz und Immersion des Anwenders. So soll es dem Anwender durch das haptische Feedback möglich gemacht werden, die virtuelle Welt, wie die reale Welt wahrzunehmen.

Ein gutes Beispiel für die Auswirkungen, welche die haptische Wahrnehmung auf die Immersion in einer virtuellen Realität hat, zeigt ein Experiment, das zur physiologischen Messung der Präsenz in virtuellen Umgebungen in stressvollen Situationen von M. Meehan durchgeführt wurde. Dabei wurde die Hypothese aufgestellt, dass wenn die virtuelle Umgebung den Status erreicht, dass sie sich real anfühlt, eine physische Reaktion bei der Versuchsperson erkennbar wird, die mit Reaktionen, die in der echten Welt entstehen, gleichzusetzen sind. Je größer die Präsenz im virtuellen Raum, desto größer auch die körperliche Reaktion. Die Versuchspersonen sollten in einem ersten virtuellen Testraum mehrere Objekte berühren, die im virtuellen Raum dargestellt waren aber auch als reales Objekt im Raum existierten, so dass sie auch wirklich berührt werden konnten. Durch dieses haptische Erlebnis im ersten Raum steigerte sich die Präsenz der Probanden. Anschließend betraten sie einen virtuellen Raum, in dessen Mitte ein tiefes Loch dargestellt wurde. Diesem Loch näherten sie sich über ein Holzbrett, welches so platziert war, dass sie eine echte Kante am virtuellen Abgrund fühlen konnten. Durch die im ersten Raum durch Haptik gesteigerte Präsenz und Immersion empfanden die Testpersonen auch diese Umgebung als glaubwürdig und bedrohlich, was sich in einer messbaren Steigerung der Herzfrequenz widerspiegelte. (vgl. Meehan *et al.*, 2002)

Wie wird haptisches Feedback realisiert?

Passives Feedback entsteht durch reale Objekte, die in die virtuelle Welt integriert sind und berührt werden können. Demgegenüber gestellt, wird aktives haptisches Feedback durch den Computer generiert, um dadurch die Interaktion mit verschiedenen virtuellen Objekten zu simulieren. Haptische Reize können beispielsweise durch Vibrationen oder elektrische Stimulation erzeugt werden. (vgl. Jerald, 2016, S. 36-43)

Da mit den herkömmlichen Controllern, die meistens bei virtuellen Systemen verwendet werden, das haptische Feedback begrenzt ist, gibt es mehrere prototypische Projekte, die sich mit verschiedenen technischen Möglichkeiten auseinandersetzen, um Haptik zu simulieren, darunter Handschuhe, Ganzkörperanzüge oder verschiedene Eingabegeräte mit haptischem Feedback.

Um Bewegungen in der virtuellen Welt als natürlich wahrzunehmen, braucht der Anwender in der Regel 60 Bilder pro Sekunde. Um haptisches Feedback in Echtzeit zu erzeugen, das natürlich und echt wahrgenommen wird, müssen haptische Reize in der Regel 1000-mal pro Sekunde an den Menschen abgegeben werden. (vgl. Dörner *et al.*, 2013, S. 7)

Ein überzeugendes haptisches Feedback besteht dann, wenn die Gegenstände, die in der virtuellen Welt gesehen und berührt werden, auch in Wirklichkeit gefühlt werden können. Größe und Form muss bei den dargestellten Objekten genauso sein, wie bei den realen Objekten, die berührt werden. Die Oberflächenstruktur der gesehenen und der realen Objekte müssen dabei übereinstimmen.

Latenzzeit

Für das Erleben einer glaubhaften virtuellen Welt ist die Latenz in einem VR-System ein entscheidender Faktor.

VR-Systeme sollten im Idealfall in Echtzeit arbeiten. Dies bedeutet, dass die Verarbeitung einer Bewegung oder Interaktion zu dessen visueller Darstellung in möglichst kurzen und konstanten Zeitabständen geliefert werden sollte, damit keine wahrnehmbare Verzögerung entsteht. Wenn zwischen der Nutzereingabe und der Antwort des Systems eine Verzögerung stattfindet, ist diese Zeitdauer der Verzögerung die Latenz des Systems.

Für den Anwender ist es wichtig, dass von ihm ausgeführte Interaktionen oder Bewegungen ohne große Latenz ausgeführt werden. Ansonsten kommt es zu einem Einbruch der Immersion, da die ausgeführte Handlung und die virtuelle Repräsentation nicht mehr übereinstimmen. Die Latenz des VR-Systems sollte deshalb unterhalb der Wahrnehmungsschwelle liegen. (vgl. Dörner *et al.*, 2013, S. 196)

Wenn in der virtuellen Welt eine Situation vorliegt, in der virtuelle Gegenstände durch den Raum verschoben werden, dann bezieht sich die Latenz auf die Zeit, die das System braucht, um die Nutzereingabe und das Tracking zu verarbeiten, den Zeitverlust, den die Übertragung auf das Rechensystem mit sich bringt und die Zeit bis die berechnete Bewegung des Gegenstands auf dem visuellen System ausgegeben wird.

Ist die Latenz so hoch, dass die grafische Darstellung nicht mehr in einer Geschwindigkeit aktualisiert werden kann, die der Benutzer als kontinuierliche Bildfolge wahrnimmt, die optisch mit der ausgeführten Bewegung übereinstimmt, nimmt der Grad der Immersion ab und die Präsenz in der virtuellen Welt ist nicht mehr gegeben.

Bei einem VR-System mit HMD und Head-Tracking spielt die Latenz eine große Rolle. Da sich der Benutzer durch die natürliche Interaktion in der virtuellen Welt schnell an diese gewöhnt, kann eine hohe Latenz negative Folgen haben, da durch die Diskrepanz der Wahrnehmung der virtuellen Welt und der realen Bewegung, Motion-Sickness auftreten kann.

Da in der virtuellen Realität neben der grafischen Darstellung auch die auditiven und haptischen Simulationen erfolgen, kann es geschehen, dass für die verschiedenen Kanäle unterschiedliche Latenzzeiten entstehen und damit eine Asynchronität der Simulation nicht ausgeschlossen ist,

was ebenso zu einem gestörten Erlebnis der virtuellen Welt und auch zu Motion Sickness führen kann. (vgl. Dörner *et al.*, 2013, S.56)

Motion Sickness beschreibt hierbei ein auftretendes Gefühl von Übelkeit, Benommenheit oder Schwindelgefühle. Diese auftretende Motion Sickness lässt sich durch eine geringe Latenz der VR-Systeme reduzieren, in dem der Unterschied zwischen den simulierten und den empfundenen Bewegungen so gering wie möglich gehalten wird.

Durch welche Faktoren wird die Latenzzeit beeinflusst?

Latenzen treten an mehreren Stellen im VR-System auf. Die Tracking-Latenz tritt zwischen der tatsächlichen Bewegung des Anwenders und der Bereitstellung der Daten im System zur Simulation der virtuellen Welt auf. Manche Tracking-Systeme filtern ihre Daten, um zum Beispiel aufgezeichnete Bewegungsdaten zu bereinigen, was zu höheren Latenzzeiten führen kann. Es kann auch sein, dass der zum Tracking verwendete Computer ein anderer ist als der, der die visuelle Darstellung erzeugt. Dann sind gegebenenfalls die durch die Netzwerkübertragung entstehenden Latenzen mit einzuberechnen. (vgl. Jerald, 2016, S. 187)

Der Transport der Daten innerhalb des VR-Systems kann ebenso zu einer sogenannten Transportlatenz führen. Beim Zusammenschließen von mehreren Trackingsystemen, also zum Beispiel bei einem HMD mit Head-Tracking und zusätzlichem Tracking der Hände, müssen die Unterschiede der verschiedenen Latenzzeiten berücksichtigt werden. Bei mehreren Latenzzeiten für unterschiedliches Tracking gibt der Teil des Trackingsystems mit der höchsten Latenz die Gesamlatenz des Trackings vor. Die Simulationslatenz entsteht bei der Berechnung der simulierten Welt und gegebenenfalls auch durch das Warten auf Tracking-Daten, wenn mehrere Systeme zusammengeschlossen sind. (vgl. Dörner *et al.*, 2013, S. 195-206)

Wenn die Welt im VR-System berechnet wurde, muss sie an das HMD übermittelt werden, wo dann die grafische Repräsentation wiedergegeben wird. Das Rendering des Inhaltes, der wiedergegeben werden soll, führt zu einer so genannten Generierungslatenz. Die Verzögerung des Renderings entsteht dabei zwischen dem Zeitpunkt, an dem die Daten zur Visualisierung eines Frames im grafischen System bereitstehen bis zu dem Punkt, an dem der Frame auf der visuellen Ausgabe dargestellt wurde. Diese Latenzzeit ist abhängig von dem visuellen Inhalt und der Qualität, in der dieser im Frame dargestellt werden soll und der Software und Hardware, die zur Berechnung zu Verfügung steht. (vgl. Jerald, 2016, S. 187-194)

Nach der Generierung werden die Daten an das Ausgabegerät weitergegeben und dargestellt. Bei der dabei auftretenden Verzögerung handelt es sich um die Darstellungslatenz. Eine Form der Auswirkung dieser Latenz ist der Einbruch der Bildwiederholfrequenz, im englischen *frames per second*, kurz *fps*, was zu einer gestörten visuellen Darstellung führt, die die Immersion verringert.

Die gesamte Latenz eines Systems setzt sich also aus der Summe aller einzelnen auftretenden Latenzen im VR-System zusammen. Da die Latenzen nicht immer konstant sind und es durch die Kommunikation der verschiedenen Systeme zu hohen Schwankungen kommen kann, wird versucht, neben einer möglichst geringen Gesamtlatenz auch möglichst geringe Schwankungen sicherzustellen, da diese vom Anwender störender empfunden werden können als eine höhere, aber gleichbleibende Gesamtlatenz. (vgl. Dörner *et al.*, 2013, S. 195-206)

Vuforia

Für den Versuch, reale Objekte in die virtuelle Welt zu übertragen um damit das Berühren und Bewegen des virtuellen Objektes mit dem haptischen Gefühl des realen Objektes zu verbinden, wurden mehrere Möglichkeiten in Betracht gezogen, wie reale Objekte in die virtuelle Welt übertragen werden können.

Die Software von *Vuforia* bietet ein Markertracking, dass für Augmented-Reality-Anwendungen auf dem Smartphone ausgelegt ist. Unter Augmented-Reality (AR) wird eine Form, der durch virtuelle Inhalte erweiterten Realität, verstanden. Dabei wird die reale Welt mit virtuellen Objekten überlagert. Diese Überlagerung folgt der dynamischen Bewegung des Blickfeldes des Betrachters. Für die Darstellung einer Augmented-Reality, wird üblicherweise die reale Umgebung durch eine Kamera aufgenommen, die dynamischen Veränderungen der Blickrichtung, somit in der Regel auch die Position und Lage der verwendeten Kamera getrackt und anschließend werden die virtuellen Objekte bezogen auf die getrackten Koordinatensysteme in die reale Umgebung integriert und anschließend grafisch als Überlagerung, des von der Kamera aufgenommenen Bildes, dargestellt.

Vuforia war zu Beginn auf Video See-Through AR-Anwendungen ausgelegt, welche das Scannen eines Markers mit Hilfe des Smartphones ermöglicht, mit anschließender Darstellung eines virtuellen Objektes, welches das von der Kamera aufgenommen Videosignal an der getrackten Position überlagert und auf dem Display des Smartphones angezeigt wird. Mittlerweile arbeitet *Vuforia* auch mit der optischen See-Through-Technik, wie sie zum Beispiel *Microsoft Hololens* entwickelt, ebenso werden aktuelle VR-Systeme von *Vuforia* unterstützt.

Bei einem Marker-Tracking werden in der Regel klar festgelegte Marker verwendet, die durch Schwellwertfilter schnell im Video gefunden werden können. Für das beste Ergebnis werden schwarzweiß Marker benutzt, da diese auch bei einfachen Kameras eingesetzt werden können und bei schwierigen Lichtverhältnissen am besten erkannt werden. Mit einem Marker lässt sich lediglich die Position wiedergeben. Für die Berechnung der Position und der Orientierung sind mehrere Marker notwendig, deren Größe und Abstände zueinander dem Tracking-System bekannt sein müssen. (vgl. Dörner *et al.*, 2013, S. 104-107)

Allgemein ist darauf zu achten, dass die Marker in der richtigen Größe gewählt werden, da bei zu großen Markern ein Annähern an das Objekt zur teilweisen Verdeckung des Markers führen kann, wodurch das Tracking unterbrochen werden kann. Wenn der Marker zu klein gewählt wird, kann es dazu kommen, dass von der Kamera zu wenig Marker-Punkte erkannt werden und so die Mustererkennung nicht mehr gewährleistet ist. Ebenso entscheidend für ein gutes Tracking-Ergebnis ist die Auflösung der verwendeten Kamera, sowie die Lichtverhältnisse und die Winkel, mit der die Kamera die Marker erfasst.

Wenn die Marker erstellt sind und die Anwendung gestartet wird, nimmt die Kamera das Videobild auf und sucht im Bild nach zusammenhängenden Liniensegmenten. Danach wird die erkannte Fläche mit den erstellten Markern verglichen und bei einer Übereinstimmung des erkannten Markers mit einem vorher definierten, wird aus der Position des Markers die Position und Lage der Kamera zum Marker berechnet. (vgl. Dörner *et al.*, 2013, S. 256-259)

Der Vorteil des von *Vuforia* verwendeten Marker - Systems ist es, dass die Marker selbst online erstellt und anschließend ausgedruckt werden können, um diese in eigenen AR/VR-Anwendungen zu verwenden.

Vuforia hat mit *VuMark* ihr eigenes System für die Erstellung von Markern entwickelt, welches jedem Entwickler ermöglicht, individuelle Marker zu erstellen, die sich unter anderem auch in Firmenlogos oder ähnliches einbinden lassen. Um mit den erstellten Markern ein gutes Trackingergebnis zu erzielen, müssen bei der Erstellung einige Dinge beachtet werden. *Vuforia* hat für ihre *VuMarks* einen eigenen Design Guide festgelegt.

So bestehen die nach dem Design Guide erstellten Marker aus folgenden fünf Elementen: Contour, Border, Clear Space, Code Elements und der Design Area.

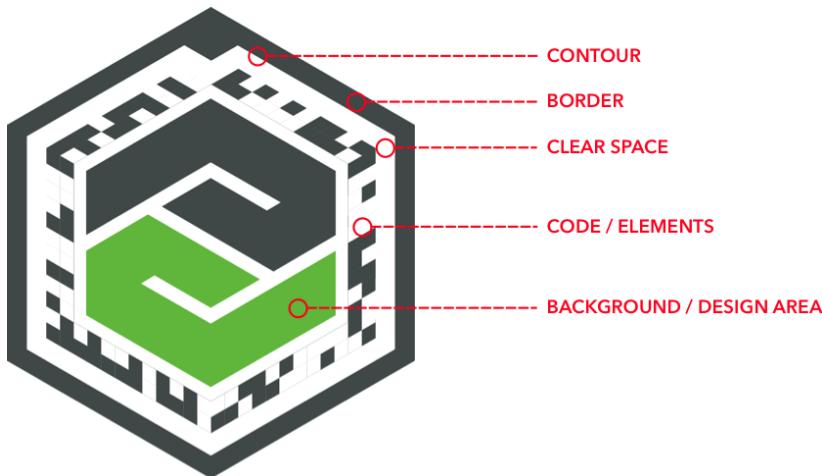


Abbildung 7:<https://library.Vuforia.com/articles/Training/VuMark>

Die „Contour“ bezeichnet dabei den Bereich, der als erstes von *Vuforias* Algorithmus erfasst wird und sollte für eine gute Erkennung aus geraden Linien bestehen, aus denen sich zwischen vier und 20 Seiten ergeben und deren Winkel zueinander nicht mehr als 150° betragen. Nach der Erkennung der „Contour“ sucht der Algorithmus nach dem „Code“ und liest daraus die „ID“ der *VuMark*. Die „Contour“ ist dabei nicht unbedingt im „Design“ sichtbar, sondern bezieht sich vielmehr auf den Kontrast, der zwischen „Border“ und „Clear Space“ entsteht.

Die „Border“ gibt den äußeren Rand und damit die Form des *VuMark* an. An diesem äußeren Rand schließt sich der „Clear Space“ an, der entweder innerhalb oder außerhalb der „Border“ liegt und sicherstellt, dass genug Kontrast vorhanden ist, damit der Algorithmus die äußeren Umrisslinien erfassen kann.

Jeder *VuMark* enthält einen eigenen „Code“, der die visuelle Repräsentation der „ID“ der jeweiligen Marke darstellt. Dieser „Code“ besteht aus visuellen Elementen, einem Datentyp und der Länge der „ID“, die durch die visuellen Elemente dargestellt wird. Dabei besitzt jedes Element die Zustände hell und dunkel, die bei der Erstellung auf exakt der gleichen Position übereinanderliegen müssen. Der einzigartige „Code“ setzt sich dann aus einem Anteil von hellen und dunklen Elementen zusammen.

In der „Design Area“ kann alles eingefügt werden, was für das Marker-Tracking nicht benutzt wird und kann daher frei gestaltet und zum Beispiel mit Logos versehen werden.

Am besten lassen sich die *VuMarks* nach den Richtlinien des „Design Guides“ mit *Adobe Illustrator* erstellen, da dafür mehrere Skripte vorhanden sind, die in das Programm eingebunden werden können und das Erstellen erleichtern. Das Setup-Skript erzeugt einem dabei automatisch die für das Erstellen des Markers notwendigen Layer in *Adobe Illustrator*. Des Weiteren gibt es ein Skript, um den selbst erzeugten Marker auf die vorgegebenen Kriterien zu testen und eines, um den fertigen Marker als SVG-Datei zu exportieren. Die exportierte SVG-Datei kann dann anschließend im *Vuforia Developer Portal* im *Target Manager* hochgeladen werden, um die notwendigen Daten für die *Unity Engine* und andere Entwicklungsumgebungen zu exportieren. (vgl. *VuMark*, 2017)

Neben den *VuMarks* gibt es bei *Vuforia* noch weitere Möglichkeiten, Marker zu erstellen. Die einfachste Variante sind dabei die „Image Targets“, welche aus einfachen Bildern bestehen. Dabei müssen keine vorher spezifizierten Schwarz-Weiß-Wechsel oder besondere Strukturen vorhanden sein, sondern *Vuforia* nimmt die im Bild vorhandenen Merkmale und vergleicht sie mit den vorher festgelegten Marker-Daten. Diese Marker-Daten erhält *Vuforia* durch das vorher im *Target Manager* hochgeladene Referenzbild des Markes. Im Target-Manager wird auch eine automatische Bewertung hinsichtlich der Erkennungs- und Trackingwahrscheinlichkeit vorgenommen.



Abbildung 8: <https://library.Vuforia.com/articles/Training/Image-Target-Guide>

Bei der Bilderkennung analysiert *Vuforia* Kontrastunterschiede im Bild, daher ist es wichtig, bei der Auswahl des Bildes, das zum Tracking genutzt werden soll, darauf zu achten, dass Kontraste gut dargestellt sind und dass das Bild viele unterschiedliche Details enthält. Um von der Kamera gut erfasst werden zu können, kommt es auch auf die Größe, die Auflösung und die Druckqualität des fertigen Markers an.

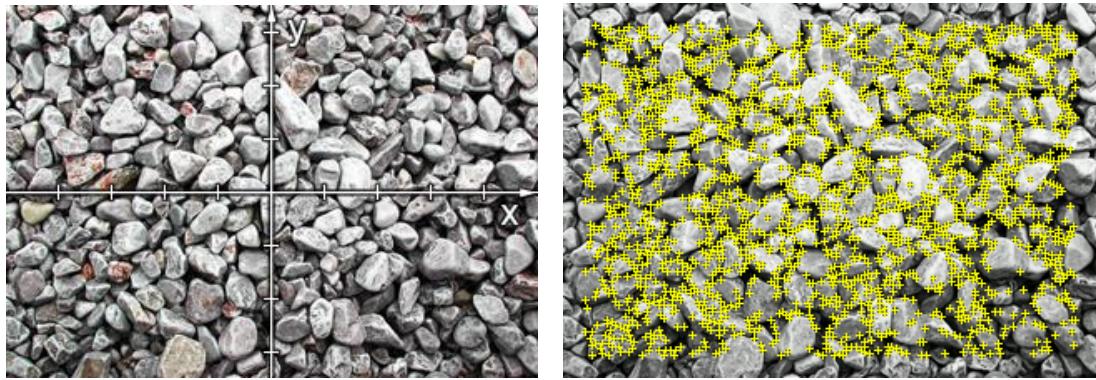


Abbildung 9: <https://library.Vuforia.com/articles/Solution/Optimizing-Target-Detection-and-Tracking-Stability>

Bei der Ausführung des Trackings sollte auf die Lichtverhältnisse geachtet werden, da der festgelegte Marker gut für die Kamera sichtbar sein muss, daher sollte bei der Lichtsituation auch darauf geachtet werden, dass auf den Marker einfallendes Licht nicht zu Spiegelungen führt die das Erkennen durch die Kamera erschweren. Ebenso sollte die Kamera auf den Marker fokussiert sein und der Winkel zwischen Kamera und Marker nicht zu spitz. Wurde der Marker im Videobild erkannt wird die Position des Bildes so lange getrackt bis das Bild außerhalb des Aufnahmebereichs der Kamera ist. (vgl. *Image Targets*, 2017)

Zusätzlich zu den einfachen Image-Targets können sogenannten Multi-Targets erstellt werden. Diese bestehen aus mehreren geometrisch angeordneten Image-Targets. Dabei ist die Position der einzelnen Bilder innerhalb des Multi-Targets ausgerichtet auf den Mittelpunkt des geometrischen Objektes. Durch die relativen Positionen der Bilder zueinander kann das ganze Target durch das Tracken eines einzelnen Bildes rekonstruiert werden. Die Erstellung von Multi-Targets geschieht ähnlich wie bei den Image-Targets über Vuforias Target-Manager, dabei werden zuerst die einzelnen Bilder festgelegt und danach, wie sie zueinander angeordnet werden sollen. (vgl. *Multi-Targets*, 2017)

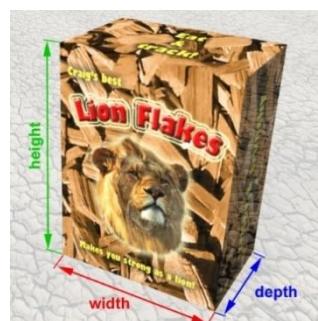


Abbildung 10: <https://library.Vuforia.com/articles/Training/Multi-Target-Guide>

Mit *Vuforia*s Objekterkennung lassen sich neben Bildern und Grafiken auch Objekte als Tracking-Targets verwenden. Die Objekterkennung kann für AR-Anwendungen genutzt werden, mit denen sich zu den getrackten Objekten weitere Informationen anzeigen lassen oder zusätzliche 3D-Grafik eingeblendet werden können.

Der dafür von *Vuforia* bereitgestellte Object-Scanner kann auf *Vuforia*s Webseite heruntergeladen und auf dem Smartphone installiert werden, was anschließend das Scannen von 3-dimensionalen Gegenständen ermöglicht. Die Gegenstände, welche zur Objekterkennung verwendet werden, sollten für eine gute Erkennung des Scanners undurchsichtig, starr, und aus wenig beweglichen Teilen bestehen. Die Oberfläche des Objekts sollte hohe Kontraste aufweisen und nicht verformbar sein, damit ein gutes Tracking-Ergebnis erzielt werden kann.

Zum Scannen des Objektes mit dem Smartphone sollte im Idealfall eine gut ausgeleuchtete Lichtsituation geschaffen werden, die keine Schatten auf dem Objekt erzeugt. Zum positionsgetreuen Einstimmen des Objekts wird vorher eine Tracking-Vorlage ausgedruckt, die dafür sorgt, dass die Position und Lage des Objektes später richtig ausgerichtet wird.



Abbildung 11: <https://library.Vuforia.com/articles/Training/Vuforia-Object-Scanner-Users-Guide.html>

Auf dieser Vorlage wird das zu scannende Objekt platziert und anschließend die Objekterkennung über die App auf dem Smartphone gestartet. Zur genauen Erkennung darf das Objekt während des Vorgangs nicht bewegen werden. Die Kamera wird nun um das Objekt herumbewegt, um es von allen Seiten zu erfassen. Dabei werden erfolgreich eingescannte Bereiche mit einer grünen Fläche markiert.

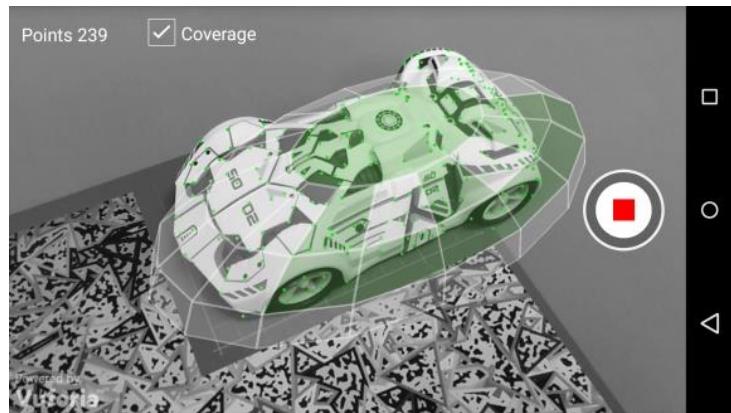


Abbildung 12: <https://library.Vuforia.com/articles/Training/Vuforia-Object-Scanner-Users-Guide.html>

Nach dem Beenden des Vorgangs erscheint eine Zusammenfassung der Objekterkennung mit einer Punktebewertung zur Qualität des Ergebnisses. Neben der Bewertung kann die Objekterkennung auch in der App getestet werden. In diesem Test wird die AR-Version angezeigt und bei einem erfolgreich eingescannten Objekt erscheint ein Würfel, der beim Bewegen des Objektes die Rotation und Translation übernimmt. Wenn das Ergebnis nicht zufriedenstellend ist, kann der Scavorgang fortgesetzt werden, um das Tracking zu verbessern. Ist der gesamte Vorgang abgeschlossen, werden die Objekt-Daten gespeichert und können anschließend im Target-Manager hochgeladen werden, wo sie dann zur Verwendung in eigenen AR-Anwendungen zur Verfügung stehen. (vgl. *Vuforia Object Scanner*, 2017)



Abbildung 13: <https://library.Vuforia.com/articles/Training/Vuforia-Object-Scanner-Users-Guide.html>

Voruntersuchung

Die Überlegung zur Umsetzung der Integration von realen Objekten in eine VR-Anwendung war es, ein Objekt so mit Markern zu versehen, dass diese mit einer Kamera und passender Software erfasst werden können, um anschließend das reale Objekt als virtuelles Objekt in der VR-Anwendung darzustellen. In der einfachsten Form wäre das ein realer Würfel, der in den Händen gehalten wird und der ebenso in der virtuellen Anwendung gesehen und manipuliert werden kann.

Die zur Verfügung stehenden VR-Systeme waren die *Oculus Rift* und die *HTC Vive*. Beide Systeme sollten mit der *Leap Motion* kombiniert werden, um das Tracken der Hände zu ermöglichen und diese als virtuelles Abbild in die Anwendung zu integrieren. Das Integrieren des realen Objektes sollte durch das Tracking des Gegenstandes durch ein geeignetes Marker-System realisiert werden.

Resultat der Voruntersuchung

Nach der Abwägung der uns zu Verfügung stehenden technischen Geräte auf die Eignung zur Umsetzung unserer Überlegung haben wir uns für eine Kombination aus der *HTC Vive* und der *Leap Motion* entschieden. Für das Marker-Tracking viel die Entscheidung auf die Software von *Vuforia*.

Der Hauptgrund für das VR-System der *HTC Vive* war, dass dieses eine in das HMD integrierte Front-Kamera besitzt, welche sich nach unseren Überlegungen für ein markerbasiertes Tracking eignen würde. Die *Leap Motion* wurde als Device zum Handtracking ausgewählt, da gegenüber anderen Handerkennungssystemen, die meist noch einen prototypischen Charakter aufweisen, bereits eine Integration in Unity 3D und den Systemen der *HTC Vive* und *Oculus Rift* besteht. Ebenso entschieden wir uns bei der Software für den Einsatz der *Vuforia* Software zur Realisierung des Markertrackings, da dieses sich mit einem einfachen Kamerabild, wie es die Front-Kamera der *HTC Vive* bietet, umsetzen lässt und ebenfalls bereits eine Integration in Unity3D und VR-Systeme besteht.

Kombinieren der Devices

Die *HTC Vive* musste nun mit der *Leap Motion* und dem *Vuforia*-Marker-Tracking kombiniert werden. Zunächst wurde hierzu die *Leap Motion* mit der *HTC* verbunden. Im ersten Schritt wird die *Leap Motion* an der *HTC Vive* befestigt. Hierfür kann ein von *Leap Motion* angebotener Befestigungsbausatz verwendet werden. Nun folgt die Installation der *Leap Motion* Software „Orion“. Diese beinhaltet das Verwaltungsprogramm „*Leap Motion Control Panel*“ der *Leap Motion*, welches unter anderem auch zur Kalibrierung der *Leap Motion* benötigt wird. Dieser Vorgang ist im Programm unter dem Tab: „Fehlersuche“, links unten, unter Gerät neu kalibrieren, genauer beschrieben.⁵

Um nun die *Leap Motion* in der *Unity Engine* anzuwenden, benötigt man das „*Unity Core Assets*“ Package, das auf der *Leap Motion* Website angeboten wird.⁶

Wenn das „*Leap Motion Control Panel*“ keine Fehler meldet, können nun in der *Unity Engine* im Project-Fenster unter „Assets/Leap Motion/Core/Examples“, verschiedene Beispielszenen geöffnet werden, die einen beispielhaften Aufbau der *Leap Motion* in Unity zeigt. Darunter befindet sich auch die „*Leap Hands Demo (VR)*“, deren Hierarchy, in ähnlicher Kombination in der finalen Szene, Anwendung findet.

Die Einrichtung der *HTC Vive* benötigt ebenfalls einige im folgenden beschriebenen Schritte. Hier war es notwendig, das *SteamVR*-Plugin aus dem Unity Asset Store in die Szene zu importieren. Bevor es weitergehen kann, muss nun zunächst das kostenlose *SteamVR Room - Setup* auf Steam installiert werden, da hier die Raumvermessung durchgeführt werden kann. Im Programm *SteamVR* finden sich unter den Einstellungen auch die Kameraeinstellungen, die für die Verwendung des *Vuforia*-Marker-Trackings notwendig sind. Dazu muss die Kamera im Einstellungsmenü aktiviert werden (siehe Abb. 14). Das Einstellungsmenü bietet auch einen Streaming Test, der bei Fehlern in der *Unity Engine* zur Fehlersuche genutzt werden kann. Meldet z.B. Kamera aktiv kein wie in der Abbildung 14 gezeigten grünen Haken, sondern ein rotes Kreuz, ist klar, dass ein Fehler vorliegt. Dieser Fehler konnte durch einen Programmneustart von Steam VR oder des Computers behoben werden.⁷

⁵ *Leap Motion & HTC Vive Setup*: <https://developer.leapmotion.com/vr-setup/vive>

⁶ *Leap Motion Unity Core Assets*: <https://developer.leapmotion.com/unity#116>

⁷ SteamVR setup https://support.steampowered.com/steamvr/HTC_Vive/

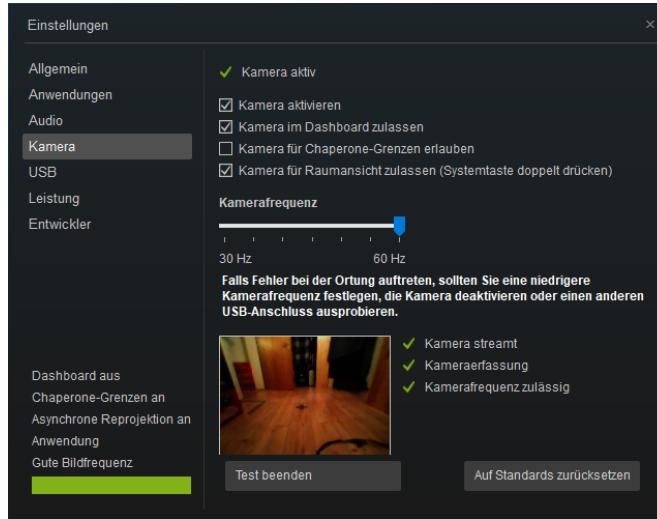


Abbildung 14: Screenshot Vive-Einstellungen

Weiterhin nützlich für eine Fehlersuche bei Programmfehlern, ist das oben erwähnte Programm *SteamVR*, das anzeigt, ob die Infrarotsensoren aktiv sind. (Siehe Abb. 15)

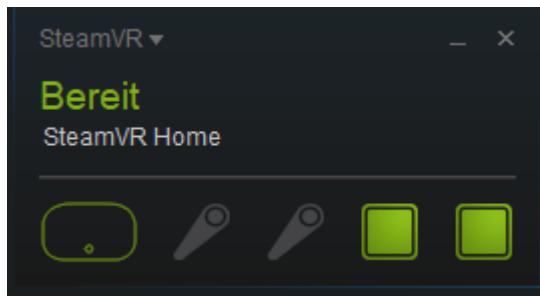


Abbildung 15: Screenshot SteamVR Menü

Um nun die *HTC Vive* und die *Leap Motion* in eine Szene zu integrieren, müssen folgende Prefabs aus dem Ordner: „Assets/SteamVR/Prefabs/“ in die Hierarchy integriert werden:

- [SteamVR]
- [CameraRig]

Sollte die *HTC Vive* nicht in der *Unity Engine* funktionieren, muss im *Unity Engine*-Menü unter „Edit/Project Settings/Player Settings“, auf dem Reiter „Other Settings“, der Haken für „Virtual Reality Supported“ auf „true“ gesetzt werden.

Steam VR sorgt in der Szene unter anderem dafür, dass ein hellblauer Rahmen in der *Unity Engine* angezeigt wird, welcher das Bewegungsfeld in der VR anzeigt. Das „CameraRig“ ist für das Positionstracking der *HTC Vive* im Raum und das Kamerarendering, wie auch die Controllerverwaltung zuständig.

Um die *Leap Motion* zu verwenden, muss das „Prefab“ aus dem Ordner: „Assets/*Leap Motion*/Core/Prefabs/LMHeadMountedRig“ auf das in der „Hierarchy“ befindliche „[CameraRig]/Camera (head)“ platziert werden, damit sich die Hände mit dem HMD mitbewegen. Die Controller unter „[CameraRig]“ wurden nicht benötigt und können daher entfernt werden.

Damit nun auch das *Vuforia*-Marker-Tracking funktionieren kann, muss die hierfür vorgesehene *Unity Engine* Extension(legacy) von der *Vuforia*-Website in die *Unity Engine* integriert werden.⁸ Ist dieser Schritt abgeschlossen, kann wie auch bei den anderen Devices ein „Prefab“ benutzt werden. Dieses findet sich unter: „Assets/*Vuforia*/Prefabs/ARCamera“. Dieses muss nun wie auch das „LMHeadMountedRig“ unter „[CameraRig]/Camera(head)“ in der „Hierarchy“ platziert werden. Auf der „ARCamera“ befindet sich auch das „*Vuforia* Behaviour Skript“ das später grundlegend für die Verwendung wird.

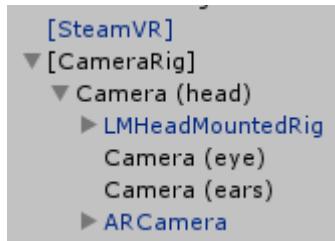


Abbildung 16: Screenshot: Unity Hierarchy

Da allerdings die Verwendung der *HTC Vive*- Frontkamera, wie im *Vuforia*-Forum beschrieben⁹, nicht zum gewünschten Ergebnis führte, wurde versucht, die „*Vuforia*-Configuration“ zu umgehen. Die *HTC Vive* wurde bei der „*Vuforia*-Configuration“ unter „Webcam/Camera Devices/*HTC Vive*“ zwar aufgeführt, konnte aber keine positiven Ergebnisse mittels Multitarget-Tracking¹⁰ erzielen. Da zuvor eine handelsübliche Webcam, für einen Test der *Vuforia*-App mittels Multitarget erfolgreich abgeschlossen war, stand diese Webcam in der „*Vuforia*-Configuration“ unter Webcam als „DroidCam Source3“ zu Verfügung. Dies wird im späteren Verlauf noch einmal relevant. Zunächst wurde nach der Ursache auf Seiten der *HTC Vive*-Kamera gesucht. Dazu wurde versucht, die Kamera per Skript als Webcam zu verwenden. Da dieser Versuch ebenfalls keine Wirkung zeigte, wurde nach Durchsuchen der *SteamVR* „Packages“ im Ordner Extras, die „*SteamVR*_TestTrackedCamera“ – Textur gefunden. Diese Textur konnte in

⁸ *Vuforia* SDK *Unity*: <https://developer.Vuforia.com/downloads/sdk?d=windows-30-16-4815&retU>

⁹ *Vuforia* *HTC Vive* Support: <https://developer.Vuforia.com/forum/digital-eyewear/htc-vive-support>

¹⁰ Siehe Kapitel: Erstellung eines *Vuforia* Multitargets

der *Unity Engine* als Webcam genutzt werden. Um diese allerdings als *Vuforia* Kamera zu nutzen, waren noch einige Schritte notwendig. Beim Ausführen der *Vuforia*-App wurde festgestellt, dass *Vuforia* in der „Hierachy“ eine neue Szene erzeugt. (siehe Abb. 17)

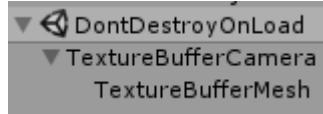


Abbildung 17: Runtime Scene ScreenShot

Diese hatte als unterstes Objekt das „TextureBufferMesh“, auf dem unter Materials das Material der Webcam lag. Wurde diese mit der „SteamVR_TestTrackedCamera“-Textur ausgetauscht, konnte erfolgreich das Multitarget erkannt werden. Um diese selbstständig, zur Laufzeit auszutauschen, wurde das Skript „Texture_Changer“¹¹ angelegt und der „ARCamera“ hinzugefügt. Des Weiteren braucht die „ARCamera“ noch das „Steam_VR_Test_TrackedCamera“-Skript.

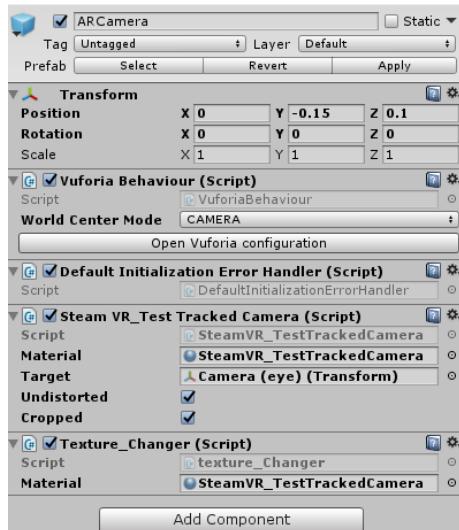


Abbildung 18: Unity Inspector Screenshot ARKamera

Die in Abbildung 18 ersichtlichen Änderungen der Transformwerte der AR-Kamera waren notwendig, da zwischen der *Leap Motion*-Kamera und der Frontkamera der *HTC Vive* ein Versatz vorhanden war.

Frühere Versuche zeigten, dass es auf Grund der unterschiedlichen Kamerasysteme zu einem dynamischen räumlichen Versatz zwischen Target Objekt und den getrackten *Leap Motion* Händen kam. Beispiel: wurde die Hand in der VR um 10 cm nach vorne bewegt, so hatte sich

¹¹ Siehe Anhang: Skripts: [Texture_Changer](#)

das Target Objekt um 15 cm bewegt. Dieser Versatz in der räumlichen Tiefe war nicht linear.

Dies wurde zwar versucht, per Skript auszugleichen, war aber nicht erfolgreich.

Ein weiterer Ansatz, zur Erstellung eines Umrechnungsfaktors um diese Verschiebung auszugleichen, hätte aber den Zeitplan überschritten. Dieser hätte vorgesehen, soweit der theoretische Ansatz, eine Positionsmessung des Multitargets im realen Raum, zu verschiedenen Abständen zur Kamera durchzuführen. Diese Messdaten hätten dann mit den VR-Daten des Multitargets abgeglichen werden können, um so eine passende, relative korrekte Position zu erhalten.

Erstellung eines *Vuforia* Multitargets

Wie im Kapitel Voruntersuchung beschrieben, fiel die Wahl der Tracking-Methode auf das *Vuforia* Multitarget-Tracking. Dazu muss zuerst eine Auswahl der zu verwendenden Texturen getroffen werden¹². Diese werden dann am echten Würfel befestigt. Danach wird auf der Webseite von *Vuforia* eine Datenbank erstellt, ein neues Target hinzugefügt und ein „Cuboid“ ausgewählt. Hier müssen die „Dimension“-Werte angegeben werden. Bei dem im Versuch verwendeten Würfel betrugen diese 6x6x6 cm. Nun können die entsprechenden Texturen auf die *Vuforia* Datenbank hochgeladen werden. Nach Abschluss dieses Vorgangs ist es möglich, die Daten als *Unity Engine* „Package“ herunterzuladen und in die *Unity Engine*-Szene zu integrieren. Dazu wird in der „*Vuforia* – Configuration“ des Weiteren die Datenbank und die Texturen geladen.

In der *Unity Engine*-Szene ist es nun notwendig, aus dem *Vuforia*-Ordner, „Prefabs“ ein Multitarget der „Hierarchy“ hinzuzufügen. Auf dem Multitarget muss nun die zu ladende Datenbank eingestellt werden. Das nun darzustellende Objekt muss in der „Hierarchy“ dem Multitarget untergeordnet werden. Nun ist die Versuchsszene erstellt und es kann mit der weiteren Versuchsvorbereitung begonnen werden.

¹² Siehe Kapitel: VR-Systeme-*Vuforia*

Demografischer Überblick der Testprobanden

In diesem Abschnitt wird ein demografischer Überblick bezüglich der Probanden gegeben, die am Versuch teilgenommen haben. Die einzelnen Versuchsaufbauten werden in den entsprechenden Unterkapiteln beschrieben.

Nachfolgende Zahlen und Grafiken des Nutzer Tests finden sich vollständig im Anhang wider. Alle Probanden haben explizit ihr Einverständnis gegeben, dass ihre Testdaten verwendet werden dürfen. Bei den minderjährigen Probanden gaben die Eltern ihr Einverständnis.

An dem Probandentest nahmen 39 Personen teil. Davon sind 14 Probanden weiblich, 24 männlich. Ein Proband hat kein Geschlecht angegeben.

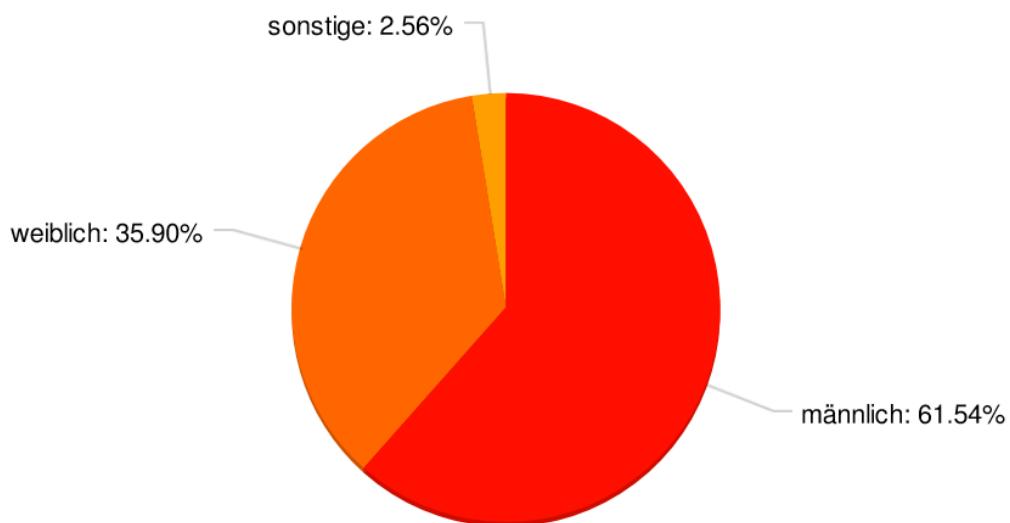


Abbildung 19. Geschlechterverteilung

Die Altersverteilung liegt zwischen 16 Jahren und 60 Jahren. Von 39 Probanden tragen 14 eine Brille. Der Großteil der Probanden besteht aus Arbeitnehmern. Die zweitgrößte Gruppe sind Studenten.

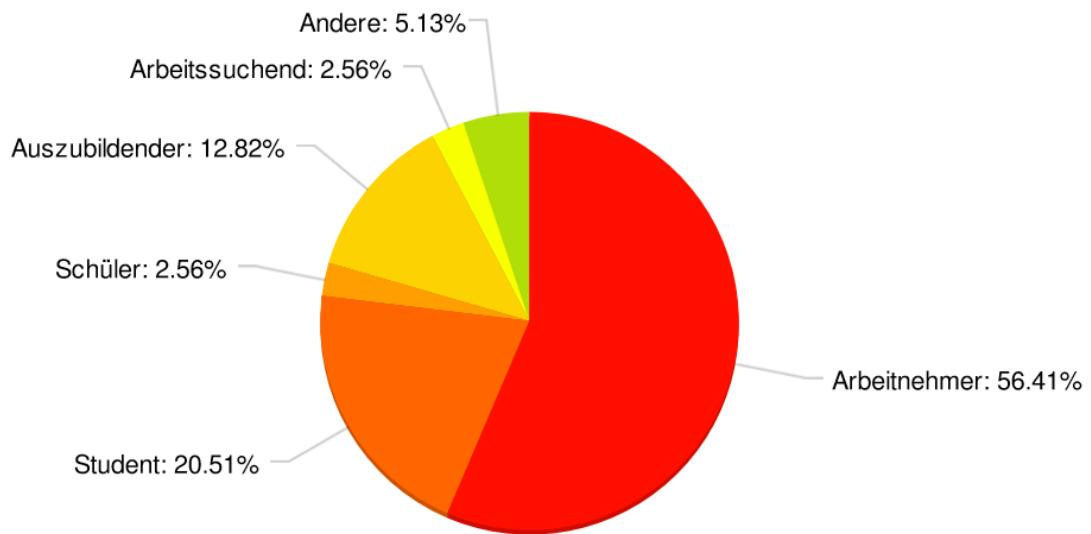


Abbildung 20: Berufsverteilung der Probanden

Untersuchung der Immersion durch ein VR-Video

Dieses Kapitel beschreibt die Problemstellung, den Aufbau, Ablauf und die Auswertung des ersten Versuchs der mit Hilfe eines VR-Videos den Probanden eine Einführung in die VR geben soll. Untersucht wurde hier, wie immersiv das Video auf die Probanden wirkt. Des Weiteren wird das Erleben von klassischem Fernsehen und VR-Videos verglichen.

Problemstellung

Aufgabe war es, eine Versuchsumgebung zu erstellen, die es einem Probanden ermöglicht, sich in die virtuelle Welt hinein zu fühlen und zu vergleichen, ob haptische Interaktion oder virtuelle Interaktion bevorzugt wird.

Um einem VR-unerfahrenen Probanden dieses Hineinfühlen so einfach wie möglich zu machen und einen späteren Vergleichswert für die verschiedenen Interaktionsmöglichkeiten zu liefern, wurde am Anfang des Probandentests ein VR-Video gezeigt, dass sich grafisch, wie auch inhaltlich an den späteren Interaktionsuntersuchungen orientiert. Dieses anfängliche VR-Video wurde aber auch dazu benutzt, einen Vergleichswert der Immersion zwischen einem VR-Video und interaktiven VR-Anwendungen zu erhalten.

Hier war unter anderem die Fragestellung, ob die Probanden bei einem möglicherweise immersiven Video, es vermissen würden, zu interagieren, weil sie z.B. in einem so hohen Grad in die Welt hineinversetzt waren, dass sie es sich gewünscht hätten, oder ob sie das VR-Video lediglich als solches wahrnahmen.

Versuchsaufbau

Das folgende Bild zeigt den Aufbau des VR-Video Versuchs.

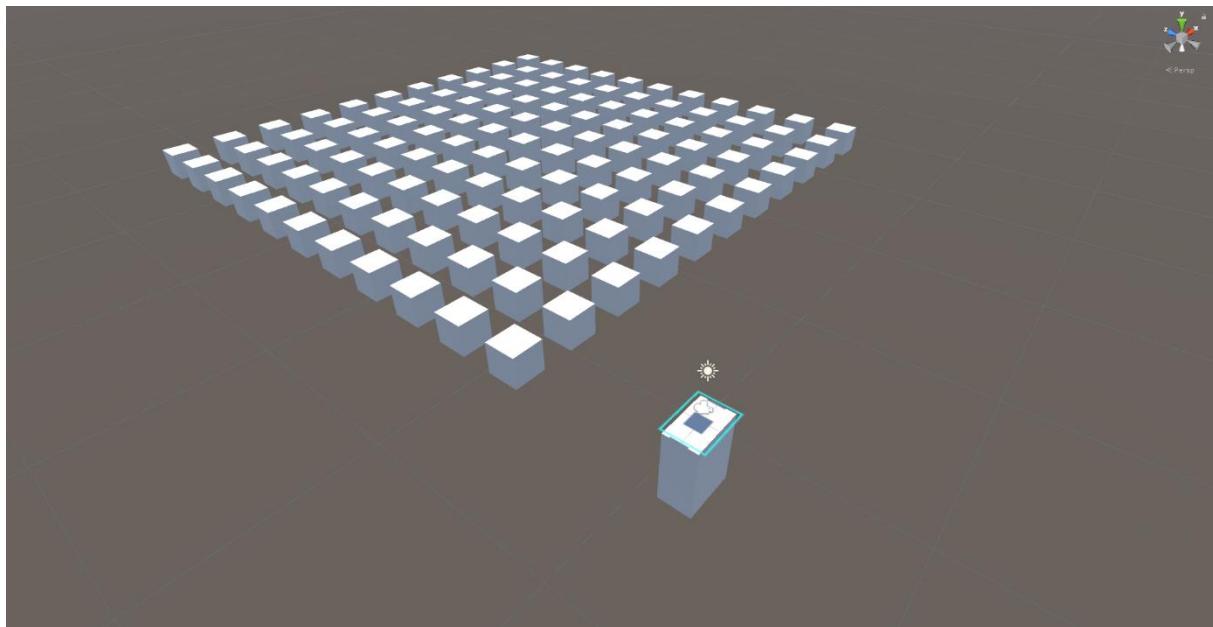


Abbildung 21: Screenshot aus der Unity Engine

Wie im oben gezeigten Bild ersichtlich ist, ist im mittleren Bereich eine Würfelmatrix zu finden. Rechts unterhalb befindet sich der Punkt, an dem der Spieler startet bzw. der Raum, in dem sich der Proband bewegen kann. Dieser ist markiert durch den blauen Rand. Bei diesem Versuch wurde allerdings in der Mitte des Raums ein Stuhl aufgestellt, auf den sich der Proband setzt, um die Immersion auf ein reines VR-Video Erlebnis, also umschauen, zu beschränken.

Versuchsablauf

Zuerst wurde der Proband auf die möglichen Risiken von VR-Nutzung hingewiesen, um sicherzugehen, dass keine Gefährdung für ihn oder das technische Equipment besteht. Dazu wurde er auf folgende Risiken hingewiesen: ¹³

- Es muss die blau dargestellte Raumbegrenzung beachtet werden
- Es dürfen keine Rauschmittel eingenommen worden sein
- Es besteht die Möglichkeit, eines auftretenden Übelkeitsgefühls
- Wenn dieses Übelkeitsgefühl eintritt, muss dies dem Studienleiter sofort mitgeteilt werden
- Die Anweisungen des Studienleiters sind zu beachten
- Bitte nicht das HMD alleine abnehmen
- Bitte nicht das HMD alleine aufziehen
- Bitte nichts mutwillig zerstören
- Achten sie auf die Länge des Kabels
- Das HMD kann sich bei ruckartigen und unvorteilhaften Bewegungen lösen, wie z.B. beim nach vorne lehnen. Daher sollten diese nur mit erhöhter Achtsamkeit vollführt werden

Vor dem Start des VR-Videos, wurde der Proband darauf hingewiesen, dass er nun ein reines Video sehen wird, sich entspannen kann und es auf sich wirken lassen soll.

¹³ Für die Zustimmung über diese Hinweise wurde eine Unterschrift der Probanden eingefordert. Aus Gründen des Datenschutzes sind diese Nicht im Anhang aufgeführt, sind ist auf Anfrage bei den Studienleitern einzusehen



Abbildung 22: Foto eines Probanden beim VR-Video

Während des Versuchsablaufs bewegten sich die in Abbildung 10 gezeigte Würfelflattix um den Probanden herum, über ihn herüber und durch ihn hindurch, damit der Proband angeregt wird, sich in der VR-Umgebung umzuschauen.

Versuchsauswertung

Die Standardabweichung wird in folgenden Auswertungen aller Tabellen mit dem Symbol  gekennzeichnet. Das arithmetische Mittel mit dem Symbol .

Bei den Fragen, wie Nutzer das getestete empfunden haben, war nur eine einfache Nennung möglich, bei den darauffolgenden Fragen, ob sie die getesteten Anwendungen anderen vorziehen würden, ist eine Mehrfachnennung möglich. In Zusammenhang mit jeder Grafik der Mehrfachnennungen ist die Anzahl der Teilnehmer aufgeführt.

Die Frage, ob die Probanden schon einmal eine VR-Brille benutzt haben, wurde wie folgt beantwortet:

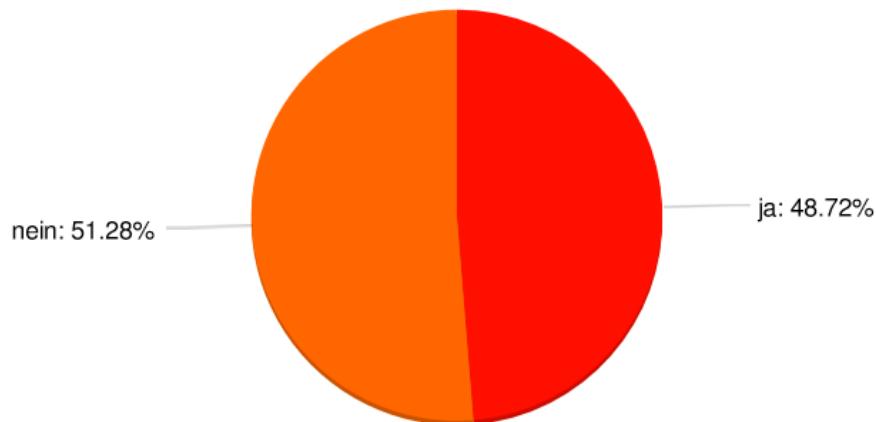


Abbildung 23: VR-Brillennutzung

19 der Probanden haben vor diesem Test schon einmal eine genutzt, davon besitzen aber nur vier Teilnehmer eine eigene VR-Brille. Die meisten Teilnehmer haben eine VR-Brille privat bei Freunden oder auf Messen, wie zum Beispiel der *Gamescom*, benutzt.

Wie im Vorherigen beschrieben, haben sich die Probanden ein VR-Video angesehen, dass sie anschließend bewerten sollten.

	Sehr (1)	Gut (2)	Ein bisschen (3)	Eher nicht (4)	Fast Nicht (5)	Gar Nicht (6)	\emptyset	\pm
	Σ	%	Σ	%	Σ	%	Σ	%
In wie weit konnten Sie s...	8x	20,51	23x	58,97	6x	15,38	2x	5,13
Konnten Sie alles um sic...	9x	23,08	19x	48,72	6x	15,38	2x	5,13
	-	-	-	-	3x	7,69	2,33	1,32

Tabelle 1: Wie haben Sie das gezeigte VR Video empfunden?

Aus den Ergebnissen lässt sich ableiten, dass sich die meisten Nutzer sehr gut bis gut in die virtuelle Welt einfühlen konnten. Nur fünf Probanden konnten sich eher nicht oder gar nicht einfühlen.

Des Weiteren wurde die Frage gestellt, ob Nutzer Filme lieber in VR oder lieber an einem Bildschirm schauen möchten. Die Probanden, die Filme lieber in VR sehen möchten, haben wie folgt geantwortet:

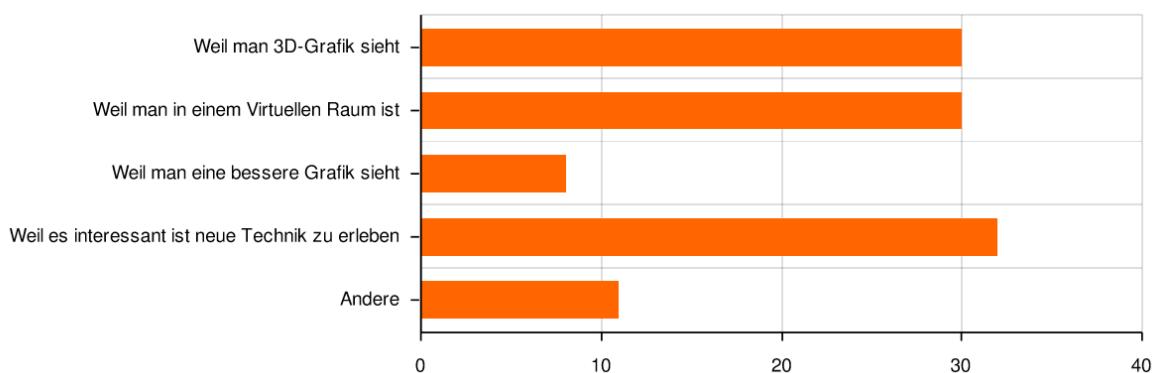


Abbildung 24: Würden Sie Filme lieber in VR oder lieber am Bildschirm schauen? (35 Teilnehmer)

Um die 30 Probanden haben ihre Motivationen damit begründet, dass man in VR 3D Grafiken sehen kann und somit in einem virtuellen Raum ist, sowie dass sie es interessant finden, neue Technik zu erleben. In einem freien Antwortfeld wurden weitere Gründe angegeben, die beispielsweise angeben, dass sich die Nutzer in VR umsehen können und dass das ihnen ein beseres Gefühl der Immersion vermittelt. Die User befinden sich so mitten im Geschehen und werden nicht von äußeren Einflüssen abgelenkt.

Die Probanden, die Filme lieber an Bildschirmen sehen möchten, haben dafür folgende Gründe angegeben:

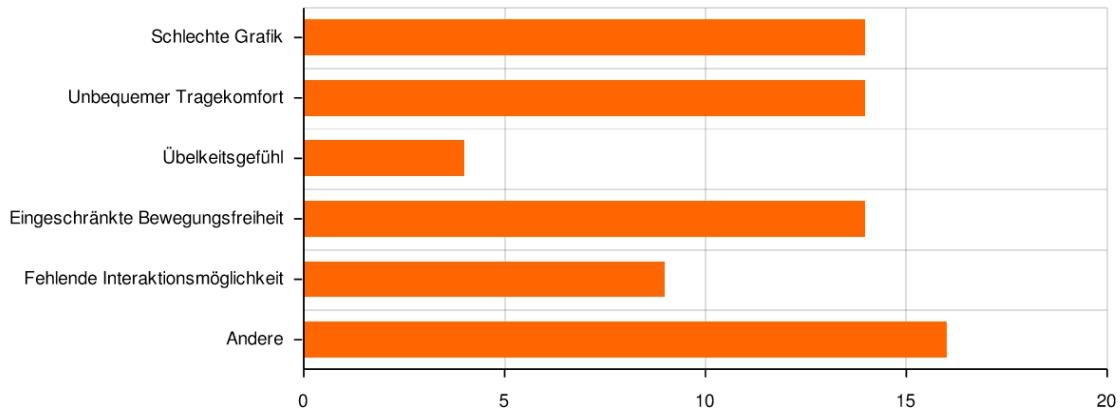


Abbildung 25: Lieber am Bildschirm, wie in VR (31 Teilnehmer)

Der Großteil findet die Technik noch unausgereift, was sich in schlechter Grafik niederschlägt. Auch der unbequeme Tragekomfort einer VR Brille wird als Grund angegeben, sowie die eingeschränkte Bewegungsfreiheit und die dadurch fehlende Interaktionsmöglichkeit. Nur vier Probanden haben Angst vor Übelkeit bei längerem Tragen einer VR Brille. Im freien Antwortfeld taucht drei Mal die Befürchtung auf, dass beim Schauen eines Films in VR etwas verpasst werden könnte, da nicht immer alles auf einmal gesehen werden kann.

Untersuchung der Immersion in einer VR-Anwendung mit visuellem Feedback

Um einen direkten Vergleichswert für die Probanden, wie auch für uns zu bieten, wurde in diesem Versuch zwar virtuelle Interaktion ermöglicht, aber auf haptisches Feedback verzichtet. Eine Ausnahme bildeten hier der Boden und die eigenen Hände, diese wurden aber vernachlässigt. Der Boden war in der *Unity Engine* durch einen im Raum stehenden Würfel Symbolisiert (siehe Abb. 21), und die Hände wurden von der *Leap Motion* getrackt.

Problemstellung

Aufgabe war es, den Probanden ein visuelles Feedback zu geben und gleichzeitig eine Versuchsumgebung zu schaffen, die sie mit den anderen Versuchen vergleichen konnten. Dazu wurde in den folgenden drei Versuchen derselbe Versuchsaufbau mit unterschiedlichen Interaktionsmöglichkeiten genutzt. Bei Versuch zwei und drei wurde die Reihenfolge randomisiert. Versuch eins und vier wurden in ihrer Reihenfolge belassen, um Versuch eins auch als Einführungsvideo zu verwenden. Versuch vier wurde am Schluss belassen, um den Probanden die Erfahrung mit dem haptischen und visuellen Feedback nicht zu beeinträchtigen.

Versuchsaufbau

Die Grundlage dieses Versuchs war wie auch in Versuch eins, der in der Mitte am Boden befindliche Würfel, der als Grundfläche diente. Der Startpunkt war auch hier der Nullpunkt in der *Unity Engine* Szene (siehe Abb. 26).

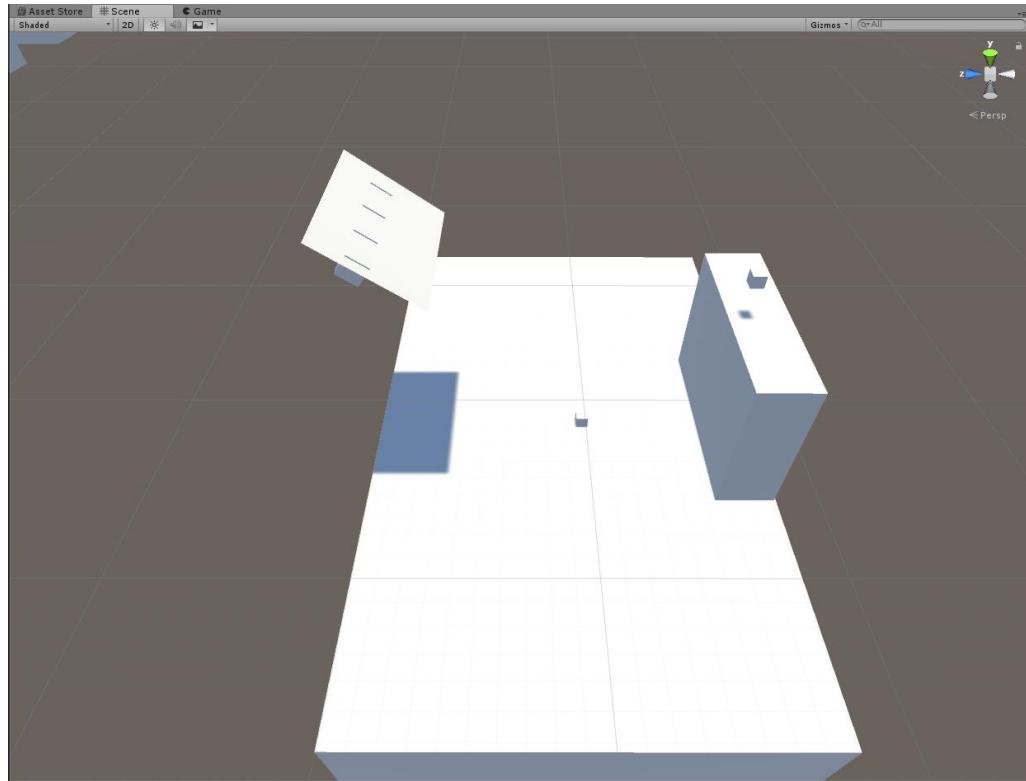


Abbildung 26: Szene Versuch 2 – 4

Wie in Abbildung 26 zu erkennen ist, befinden sich mehrere virtuelle Gegenstände in der Szene. Die wie folgt benannt wurden:

- Auf der linken Seite das sogenannte Panel oder auch Bedienfeld, das zur Steuerung der in Abbildung 27 gezeigten Würfelmatrix dient.
- Auf der rechten Seite ein virtueller Tisch, über dem ein drehbarer, aber räumlich fixierter Würfel schwebt. Die Bewegung dieses Würfels ermöglichte es, den in Abbildung 28 gezeigten rechten Würfel mit zu bewegen.
- In der Mitte auf dem Boden des Raumes befindet sich ein zweiter Würfel, der ebenfalls der Interaktion dient. Dieser kann aber im Gegensatz zum schwebenden Würfel gegriffen und durch den Raum bewegt werden.

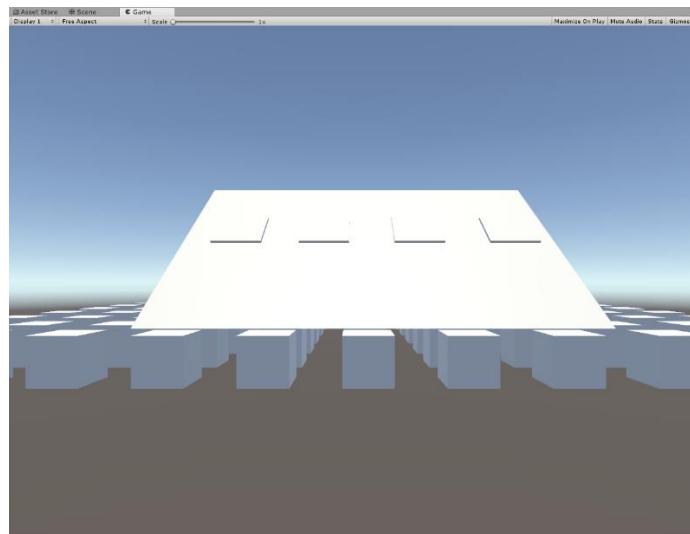


Abbildung 27: Startszene Bedienpult mit Würfelmatrix

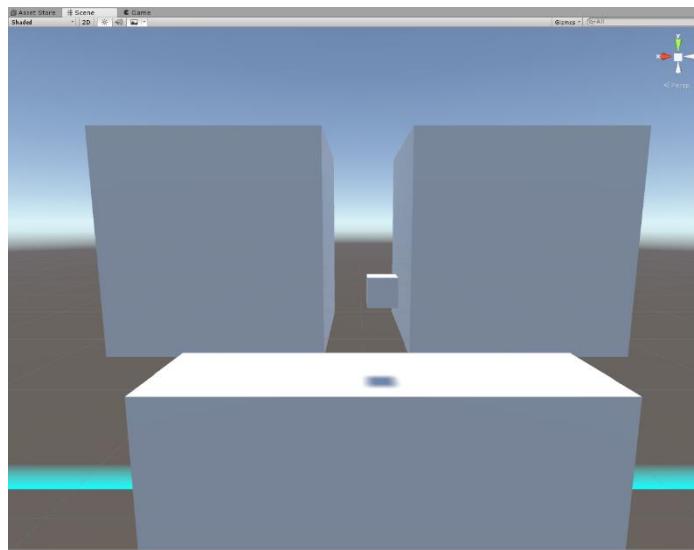


Abbildung 28: Im Hintergrund, die indirekt steuerbaren Würfel

An dieser Stelle sollte nochmals erwähnt werden, dass hier im Gegensatz zu Versuch 3 alle oben gezeigten Objekte ausschließlich in der virtuellen Welt vorhanden waren. Mit Ausnahme der Bodenfläche.

Versuchsablauf

Im Versuchsablauf wurde der Proband nun gebeten aufzustehen, damit der aus Versuch 1 vorhandene Stuhl entfernt werden konnte. Nun war es zunächst seine Aufgabe, das virtuelle Bedienfeld (Abb.27) zu nutzen, um die Würfelmatrix zu steuern, um ein Gefühl für die visuelle Interaktion zu bekommen. Dieses Bedienfeld war dazu in der Lage, die Würfelmatrix nach rechts und links, wie auch nach vorne und hinten zu bewegen.

Um diesen Versuch zeitlich zu begrenzen, wurde ein Zeitgeber mit maximal 30 Sekunden eingestellt. Anschließend wurde der Proband darauf hingewiesen, dass sich hinter ihm auf dem Boden der aufhebbare Würfel befand. Einige der Probanden hatten hier beim aufheben des Würfels Probleme. Gelang es dem Probanden, ihn aufzuheben, wurden er angewiesen den Würfel in die andere Hand zu nehmen, um hier einen besseren Einblick über die Möglichkeiten zu erhalten. Nach weiteren 30 Sekunden wurde der Proband darum gebeten, den Würfel auf den virtuellen Tisch zu legen, wie in Abbildung 28 ersichtlich.

Zum Abschluss dieses Versuchens wurde dem Probanden erklärt, dass der virtuelle, über dem rechts in der Szene befindlichen Tisch, schwebende Würfel, zur Manipulation des in Abbildung 28 rechten, großen Würfels, dient.

Versuchsauswertung

In diesem Versuch konnten die Probanden mit ihren Händen, die sie in VR sehen konnten, einen Würfel bewegen. Dazu bekamen sie nur visuelles Feedback in der VR.

Die Frage, ob die Probanden schon einmal eine *Leap Motion* benutzt haben, wurde vor den Tests wie folgt beantwortet:

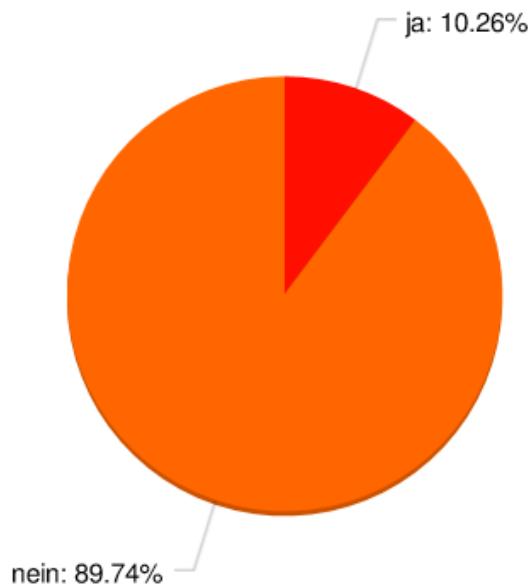


Abbildung 29: Haben Sie schon einmal eine Leap Motion genutzt

Nur vier Probanden haben vor diesem Test eine *Leap Motion* benutzt.

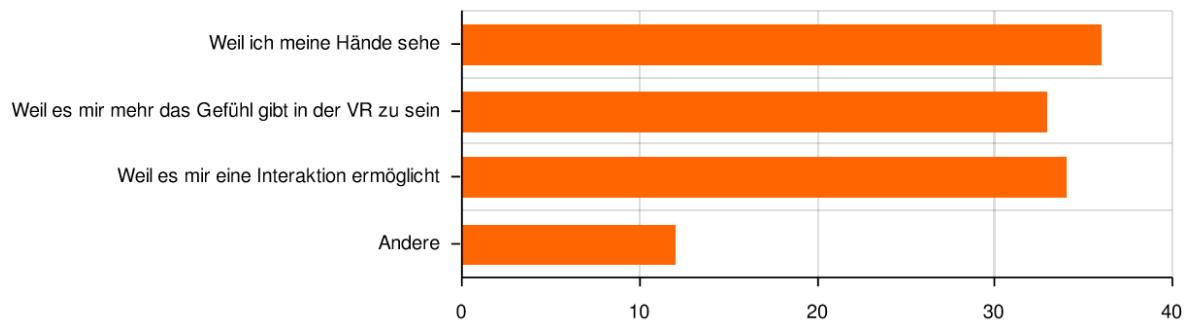
	Sehr (1)		Gut (2)		Ein bisschen (3)		Eher nicht (4)		Fast nicht (5)		Gar nicht (6)		\emptyset	\pm
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%		
In wie weit konnten Sie s...	15x	38,46	12x	30,77	9x	23,08	3x	7,69	-	-	-	-	2,00	0,97
Konnten Sie alles um sic...	19x	48,72	12x	30,77	5x	12,82	2x	5,13	-	-	1x	2,56	1,85	1,11

Tabelle 2: Wie empfanden Sie die Anwendung, bei der Sie einen virtuellen Würfel bewegen

und ein virtuelles Bedienfeld nutzen konnten?

Aus den Ergebnissen lässt sich ableiten, dass sich die meisten Nutzer sehr gut bis gut in die virtuelle Welt einfühlen konnten. Nur drei Probanden konnten sich eher nicht oder gar nicht einfühlen und alles um sich herum vergessen.

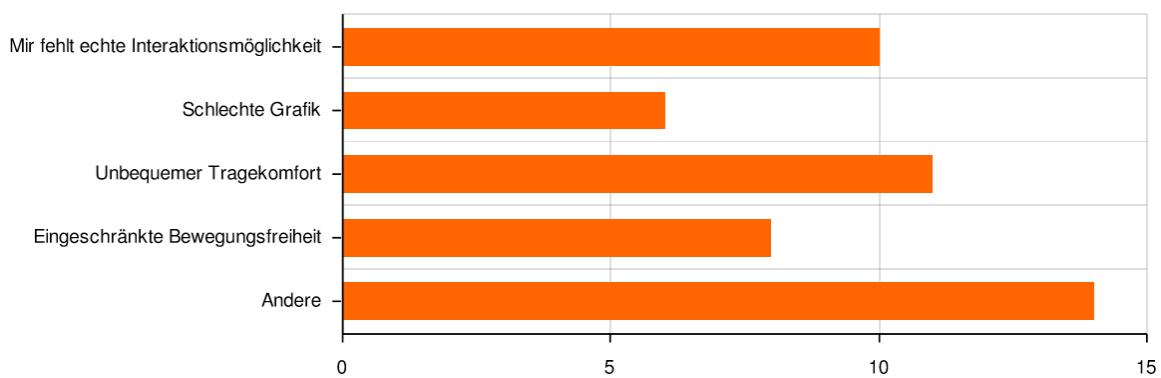
Im Folgenden wurden die Teilnehmer gefragt, ob sie die VR-Brille in Zusammenhang mit der Handerkennung, anderen VR Anwendungen vorziehen würden.



*Abbildung 30: Würden Sie den Gebrauch der VR-Brille + Leap Motion anderen VR Anwendungen vorziehen?
Ja, weil: (37 Teilnehmer)*

Fast alle Testteilnehmer wählten bei dieser Frage die drei folgenden Gründe aus: Sie würden die getestete Anwendung vorziehen, da sie ihre Hände sehen können und besser interagieren können sowie das Gefühl haben, mehr in der virtuellen Realität zu sein.

Im freien Antwortfeld gaben die Nutzer an, dass die Handsteuerung intuitiver ist und man so mit den Händen mehr Möglichkeiten hat. Das gibt den Nutzern ein besseres Gefühl der Kontrolle.



*Abbildung 31: Würden Sie den Gebrauch der VR-Brille + Leap Motion anderen VR Anwendungen vorziehen? Nein, weil.
(24 Teilnehmer)*

In Bezug auf die Auswahlmöglichkeiten sind hier die fehlende Interaktionsmöglichkeit und der unbequeme Tragekomfort am häufigsten genannt worden. Vierzehn Nutzer haben sonstige Gründe angegeben, die sich fast ausschließlich auf das störende Kabel beim Tragen einer VR-

Brille beziehen und auf die unausgereifte Technik, die als verbesserungswürdig angesehen wird. Eine Person hat bemängelt, dass nur die Hände gesehen werden und nicht die Beine, sowie der restliche Körper.

Zusammenfassend lässt sich sagen, dass der Großteil der Probanden die Handerkennung in Verbindung mit der VR bevorzugt.

Untersuchung der Immersion in einer VR-Anwendung mit haptischem Feedback

In dieser Versuchsszene war es notwendig, den Probanden das haptische Feedback zu ermöglichen. Dazu wurde, wie erwähnt, die Szene aus Versuch Zwei, mit leichten Modifikationen, genutzt.

Problemstellung

Die vorliegende Problemstellung befasste sich mit der Erstellung eines Versuchs für das haptische Feedback.

Versuchsaufbau

Um in der Problemstellung erwähnte Aufgabe zu bewerkstelligen, wurde die selbe *Unityszene* wie in Versuch Zwei verwendet. Allerdings wurde hier an der Position des virtuellen Bedienfeld im echten Raum ein den Maßen entsprechender Aufbau eingefügt (siehe Abbildung 23).



Abbildung 32: reale Bedinpnultnachbildung

Des Weiteren befand sich in dieser Szene auch das Multitarget der *Vuforia*-App¹⁴. (Siehe Abbildung 23). Dieser Würfel (Abbildung 24), war wie der schwebende Würfel aus dem Versuch Zwei mit visuellem Feedback nun in der Lage, den Linken großen Würfel¹⁵ durch Drehung mit zu manipulieren. Die Größe des Würfels betrug in diesem Fall 6 cm x 6 cm x 6 cm.



Abbildung 33: Würfel für die Vuforia - Erkennung

¹⁴ Die Erstellung des Multitargets ist in Kapitel: *Vuforia* und Kapitel: Erstellung eines *Vuforia* Multitargets genauer beschrieben

¹⁵ Siehe Abbildung 28

Versuchsablauf

Der Versuchsablauf startete in diesem Fall auf der rechten Seite bei den großen Würfeln¹⁶.

Um den Würfel für die Probanden sichtbar zu machen, war es notwendig, ihn direkt vor die AR-Kamera der *HTC-Vive* zu halten. Damit die Kamera, bzw. die *Vuforia*-Software den Würfel erfassen konnte. Anschließen konnte den Probanden der Würfel übergeben werden und sie konnten damit interagieren.

Diese Interaktion war zeitlich begrenzt auf 30 Sekunden. Im Anschluss wurde der Würfel wieder entfernt und der Proband wieder an das Bedienfeld gebeten. Hier wurde aber im Gegensatz zum virtuellen Bedienfeld der Proband genaustens instruiert und angewiesen, sehr langsame Bewegungen mit den Händen zu vollführen bis er sich an das reale Bedienfeld gewöhnt hatten, um Verletzungen zu vermeiden.

¹⁶ Siehe Abbildung 28

Versuchsauswertung

In diesem Versuch konnten die Probanden mit ihren Händen, die sie in VR sehen konnten, einen Würfel bewegen. Dazu bekamen sie auch haptisches Feedback indem sie einen echten Würfel in der Hand hatten und eine reale Oberfläche berühren konnten. Anschließend wurden den Probanden die Fragen: *In wie weit konnten Sie Sich in die virtuelle Welt hineinversetzen?* und *Konnten Sie alles um sich herum vergessen?* Gestellt.

	Sehr (1)		Gut (2)		Ein bisschen (3)		Eher nicht (4)		Fast nicht (5)		Gar nicht (6)		\emptyset	\pm
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%		
In wie weit konnten Sie s...	17x	43,59	11x	28,21	8x	20,51	3x	7,69	-	-	-	-	1,92	0,98
Konnten Sie alles um sic...	14x	35,90	18x	46,15	3x	7,69	2x	5,13	2x	5,13	-	-	1,97	1,06

Tabelle 3: Wie empfanden Sie die Anwendung, bei der Sie einen echten Würfel bewegen und ein echtes Bedienfeld benutzen konnten?

Aus den Ergebnissen lässt sich ableiten, dass sich die meisten Nutzer sehr gut bis gut in die virtuelle Welt einfühlen konnten. Nur vier Probanden konnten sich eher nicht oder gar nicht einfühlen und alles um sich herum vergessen.

Im Folgenden wurden die Teilnehmer gefragt, ob sie die VR Brille in Zusammenhang mit der Handerkennung und im Zusammenhang mit realen Objekten, anderen VR Anwendungen vorziehen würden.

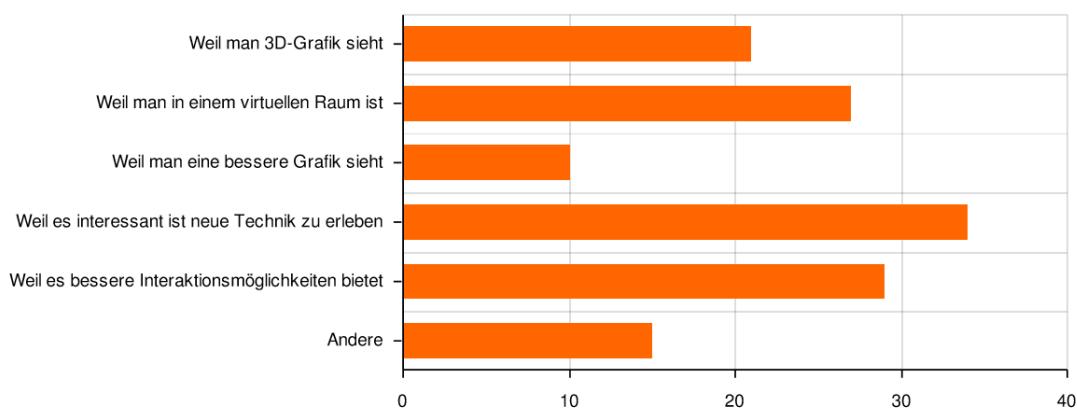


Abbildung 34: Würden Sie den Gebrauch der VR-Brille + Leap Motion + Object Tracking anderen VR-Anwendungen vorziehen? Ja, weil (36 Teilnehmer)

Der Großteil gab hier an, dass sie die getestete Anwendung bevorzugen, da sie es interessant finden, neue Technik zu erleben. Auch oft genannt wurde, dass die Interaktionsmöglichkeiten

größer sind und dass man sich in einem virtuellen Raum befindet, in der 3D Grafiken vorhanden sind.

Im freien Antwortfeld beziehen sich die Gründe hauptsächlich auf das haptische Feedback. Durch dieses haptische Feedback fühlen sich die Probanden in ihrer Immersion bestärkt.

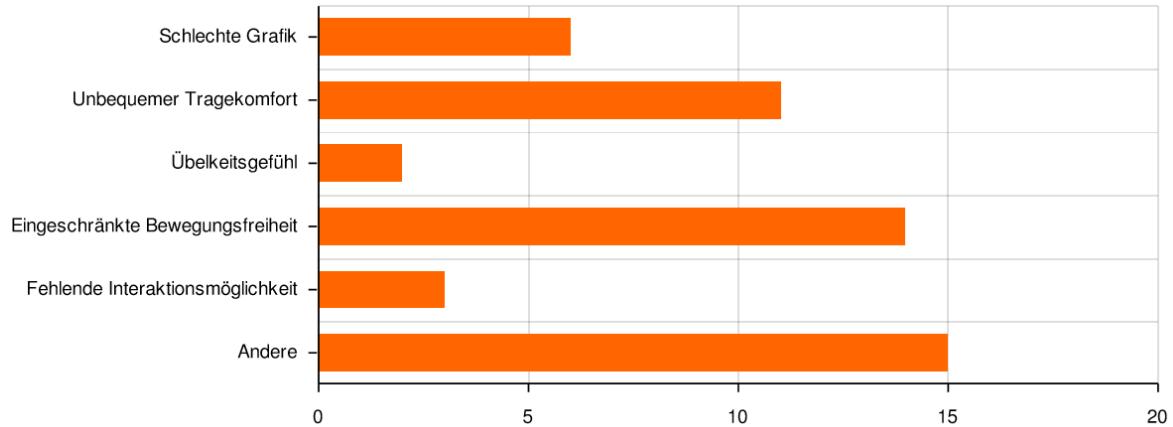


Abbildung 35: Würden Sie den Gebrauch der VR-Brille + Leap Motion + Object Tracking anderen VR-Anwendungen vorziehen? Nein, weil. (23 Teilnehmer)

Die Probanden, die mit nein geantwortet haben, geben als Gründe hauptsächlich den unbequemen Tragekomfort und die eingeschränkte Bewegungsfreiheit an. Im freien Antwortfeld waren die Gründe insbesondere das als störend empfundene Kabel und die Angst, sich an den realen Gegenständen zu verletzen.

Untersuchung der Immersion in einer VR-Anwendung mit erhöhter Latenzzeit

Diesem Versuch war das Ziel gesetzt, die empfundene Latenzzeit zu messen.

Problemstellung

Aufgabe war es, belastbare Messwerte bezüglich der Empfindung der Latenzzeit zu erzeugen.

Dazu war es notwendig, eine konstante fps-Zahl zu erzeugen.

Versuchsaufbau

Um nun in der vorhandenen Szene eine konstante fps-Zahl zu erzeugen, war zunächst geplant die *Unity Engine* Klasse: „*Application.targetFrameRate*“ zu verwenden. Allerdings findet sich in der Dokumentation dieser Klasse folgender Satz:

„When using VR Unity will use the target frame rate specified by the SDK and ignore values specified by the game.“ (Unity3D, 2017)

Daraus schloss sich, dass die fps nur mit dem in der *UnityEngine* befindlichen *SteamVR* Skript ändern ließe. In Abbildung (18) unten ist die entsprechende Stelle aufgeführt.¹⁷

```

393      // Ensure various settings to minimize latency.
394      Application.targetFrameRate = -1;
395      Application.runInBackground = true; // don't require companion window ↵
396      focus
397      QualitySettings.maxQueuedFrames = -1;
398      QualitySettings.vSyncCount = 0; // this applies to the companion ↵
399      window

```

Abbildung 36: Application_TargetFrameRate_SteamVR

Da die Änderung des Codes in Zeile 394, keine Wirkung zeigte, wurde ein Alternativplan entworfen. Dieser sah vor, die gerenderten Bilder der Kamera direkt zu beeinflussen. Dies wurde mit dem Skript: *fps_manipulator*¹⁸ bewerkstelligt. Dieses Skript war in einer leeren Unity Szene dazu in der Lage, die fps-Zahl relativ konstant bei einem eingestellten Wert von 92 als Target-fps im *fps_manipulator* bei einem Mittelwert von 44,86fps und einer Standartabweichung von 1,15, zu halten. Die Target-fps entsprach nicht den geplanten fps, war aber wie erwähnt dazu in der Lage, diese relativ konstant zu halten. Die Varianz betrug in diesem Fall 1,32. Bei diesen Werten wurden die anfänglichen Abweichungen bei 0 – 160fps ausgelassen, da diese den Mittelwert stark verfälschten, wie aus der Tabelle 4 und den Abbildungen 37 und 38 ersichtlich ist.

An dieser Stelle muss erwähnt werden, dass die von der *Unity Engine* mit dem Skript gemessenen und im angezeigten Rendering Statistics Window¹⁹ nicht den Bildern pro Sekunde entspricht, sondern:

¹⁷ Der gesamte Code befindet sich im Anhang unter Skripts/SteamVR_Render

¹⁸ Der gesamte Code befindet sich im Anhang unter Skripts/fps_manipulator

¹⁹ Unity 3D: Rendering Statistics Window: <https://docs.unity3d.com/Manual/RenderingStatistics.html>

„The amount of time taken to process and render one game frame (and its reciprocal, frames per second). Note that this number only includes the time taken to do the frame update and render the game view; it does not include the time taken in the editor to draw the scene view, inspector and other editor-only processing.“ (Unity3D, 2017)

Daraus folgt, dass die gemessenen Werte nicht den Bildern pro Sekunde entsprechen, sondern der Zeit, die ein Frame braucht, um sich zu aktualisieren und im Gameview der *Unity Engine* darzustellen. Die Messdauer war bei jeder Messung auf 1000 Frames beschränkt.

	Mittelwert	Standartabweichung	Varianz
fps: 0- 1000	51,9109109	17,44705909	304,399871
fps: 160 - 1000	44,863258	1,152262099	1,32770794

Tabelle 4: Messung der fps - Zahl einer leeren UnitySzene

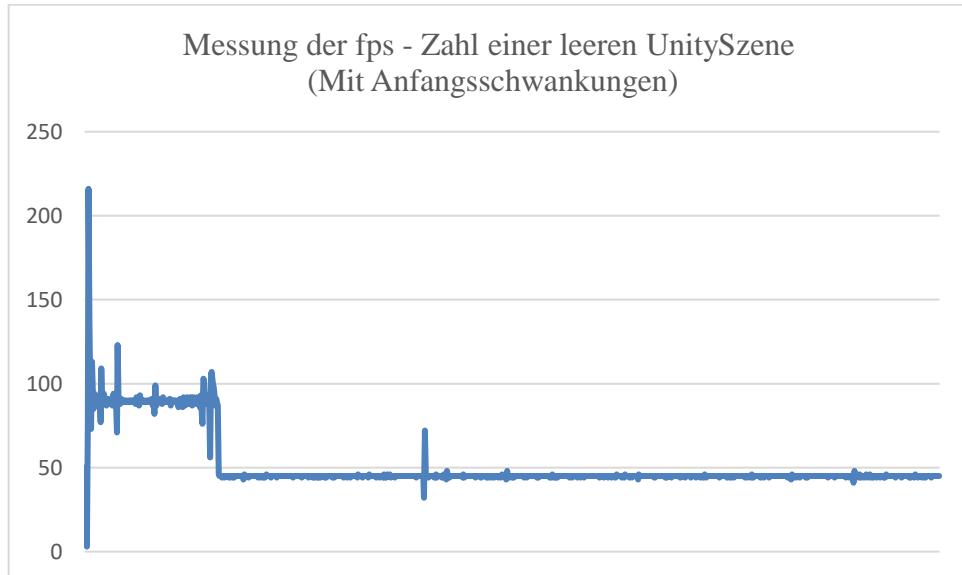


Abbildung 37: Messung der fps-Zahl einer leeren UnitySzene (Mit Anfangsschwankungen)

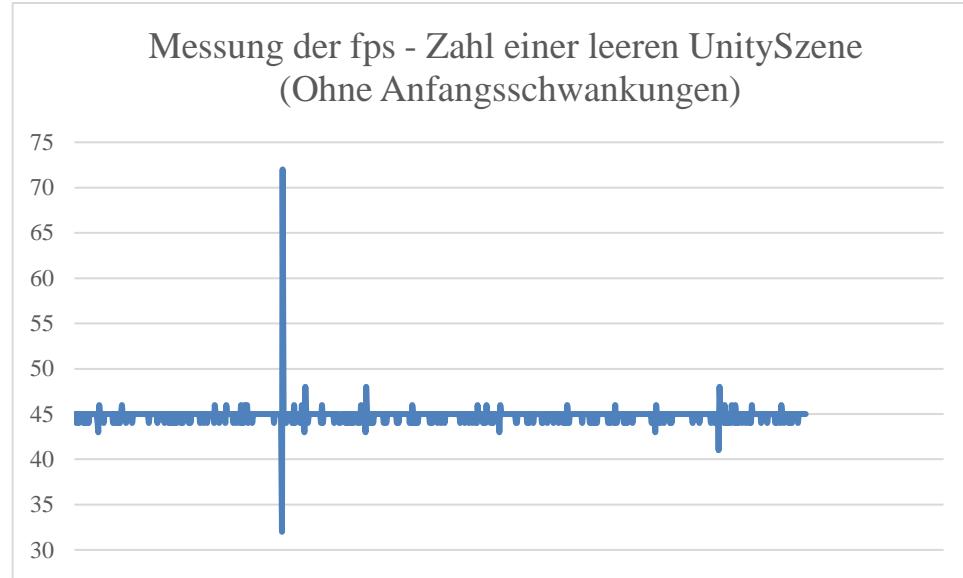


Abbildung 38: Messung der fps-Zahl einer leeren UnitySzene (Ohne Anfangsschwankungen)

Diese in der leeren Unity Szene erzeugten Messwerte lieferten die Grundlage für das weitere Vorgehen. Was außer Acht gelassen wurde, waren die Schwankungen, die in der Versuchsszene auftraten. Diese wurden in der Versuchsumgebung ausschließlich visuell über das „Rendering Statistics Window“ überwacht. Diese Schwankungen sind in der Versuchsauswertung genauer aufgeführt.

Aus Gründen der Vereinfachung erschien es sinnvoll, den Probanden bei der fps-Schätzung eine Hilfestellung zu geben. Deshalb wurden wie auch dem Anhang (Probandentest) zu entnehmen ist, Zehnerschritten zwischen 10 und 50 vorgegeben.

Versuchsablauf

Wie schon erwähnt, wurde für diesen Versuch auf die Szene des rein visuellen Feedbacks zurückgegriffen²⁰, um den Probanden einen möglichst guten Vergleichswert zwischen einer Szene ohne voreingestellte fps-Zahl und mit voreingestellter fps-Zahl zu bieten.

Bei der Abfrage der einzelnen fps-Werte, wurde zuerst eine fps-Zahl erzeugt. War diese konstant, wurde die Schätzung des Probanden abgefragt. In mehreren Testläufen ergab sich, dass gewisse fps-Zahlen nicht konstant gehalten werden konnten, wenn bestimmte Interaktionen vollzogen wurden. Z.B. fiel die voreingestellte fps-Zahl von 50, auf ca. 30-40, sobald die *Leap Motion* die Hände erkannt hatte. Dies führte dazu, dass die Probanden um entsprechende fps zu sehen, bestimmten Anweisungen folgen mussten.

- Für 50fps durfte ausschließlich geradeaus auf die Würfelmatrix geschaut werden und der Kopf nach rechts und links bewegt werden.
- Für 40fps durfte ausschließlich mit dem virtuellen Würfel Tisch interagiert werden und danach die Hände aus dem Bild genommen werden.
- Für 30fps durften ausschließlich die Hände vor dem Gesicht bewegt werden.
- Für 20fps durfte jegliche Interaktion ausgeführt werden, da diese augenscheinlich konstant waren.
- Für 10fps galt das Selbe, wie für 20 fps.

Um die visuell überwachten Schwankungen dennoch einer Latenzzeit zuzuordnen, wurde entsprechend des Wertes auf- oder abgerundet. Lag ein Wert beispielsweise bei ca. 46fps, so wurde in der Messwerttabelle 50 eingetragen.

Nachdem die Probanden ihre Schätzung für einen fps-Wert abgegeben hatten, wurden sie jeweils dazu befragt, in wie weit sie sich von dieser Framerate in ihrem Spielerlebnis beeinflusst fühlten²¹.

²⁰ Siehe Dateianhang: Assets/MySzenen/Untersuchung/Untersuchung *Leap&Vuforia*

²¹ Siehe Anhang: Fragebogen: **Wie sehr hat diese Framerate Ihr Spielerlebnis beeinflusst?**

Diese Skala war aufgeteilt in: Wie sehr hat diese Framerate Ihr Spielerlebnis beeinflusst?

- Gar nicht
- Ein bisschen
- Stark
- Sehr stark

Wobei im Datensatz²² zur besseren Verarbeitung den Aussagen folgende Werte zugeordnet wurden:

- Gar nicht = 0
- Ein bisschen = 1
- Stark = 2
- Sehr stark = 3

Um diese dann anschließend sinnvoll in die Auswertung zu integrieren, wurden diese mit (*-10) multipliziert.

²² Siehe Dateianhang

Versuchsauswertung Latenzversuch

Die nun folgenden Kapitel beschreiben die einzelnen Versuchsreihen von 50 fps-10 fps. Um diese jeweils auszuwerten, wurden sie nach dem Versuch bereinigt und überarbeitet. Auch hier wurden die Messungen randomisiert.

Messung mit 50 Bildern pro Sekunde

Die Auswirkung der Latenz auf die Immersion und ihre entsprechende Schätzung ist im folgenden Diagramm aufgeführt. Hier ist zu erwähnen, dass bei allen Diagrammen zu diesem Versuch die ungültigen Werte entfernt wurden. Dies hat die Ursache, dass zur Zeit des Probandentestes bei dieser Einstellung kein konstanter Wert generiert werden konnte und diese Messung für ungültig erklärt wurde²³.

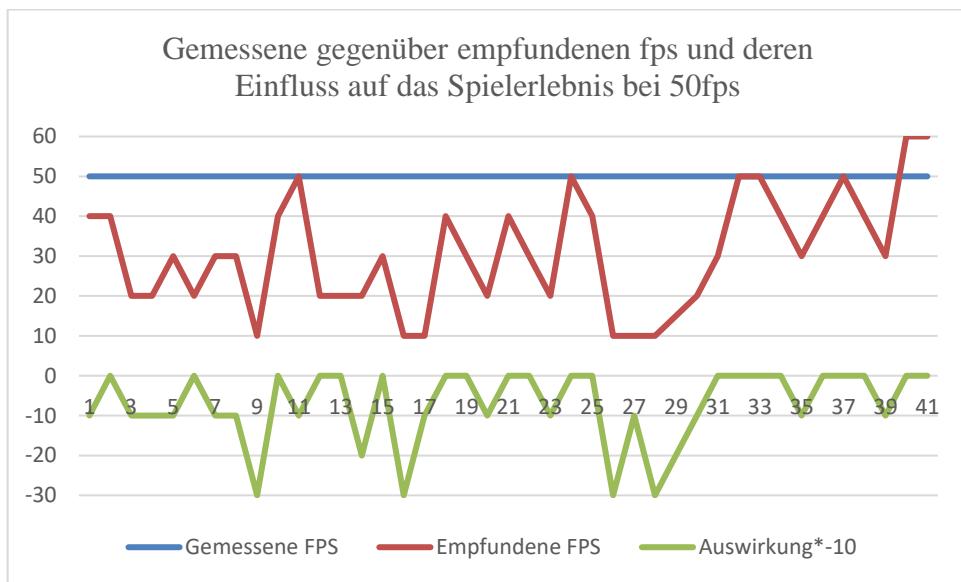


Abbildung 39: Gemessene gegenüber empfundenen fps und deren Einfluss auf das Spielerlebnis bei 50Fps

Die gemessenen fps wurden hier wie im Kapitel Versuchsaufbau erwähnt, visuell überwacht und entsprechend gerundet. Daher bildet sie trotz der gemessenen Abweichungen²⁴ eine konstante Linie. Die empfundene Latenzzeit ist hier der negativen Auswirkung auf das Spielerlebnis gegenübergestellt. Da die Auswirkungswerte aus den Fragebögen, wie im Versuchsablauf beschrieben, von Null bis Vier reichen, wurden diese zur verbesserten Darstellung mit -10 multipliziert. In dem oben dargestellten Diagramm heißt das also, umso tiefer der Auswirkungswert liegt, umso störender empfanden die Probanden die empfundene fps-Zahl.

Eine tiefergehende Auswertung hat bei dieser Latenzzeituntersuchung nicht stattgefunden. Dies begründet sich darin, dass sich bei einer Überprüfung der unten aufgeführten Messung, Zweifel an der Aussagekraft des Versuchs ergaben.

²³ Siehe Dateianhang: ungültige Werte sind bei empfundener Latenz mit 0 gekennzeichnet.

²⁴ Siehe Abbildung 40: Schwankungen der fps bei einer Einstellung auf 50fps

Um die Aussagekraft des Versuches zu belegen, wurde eine Messung durchgeführt, ob die fps konstant genug sind, um ein belastbares Ergebnis zu liefern. Die hier erhobenen Daten wurden wie in den folgenden Diagrammen aufgeführt und bereinigt, um die Analysewerte nicht zu verfälschen und eine verbesserte Darstellung zu ermöglichen.²⁵ In diesem Fall wurden alle Werte über 100fps und unter 20fps nicht berücksichtigt, da diese Überschreitungen auf Grund ihrer Einzigartigkeit zu vernachlässigen sind²⁶.

Dies waren folgende Werte:

- Wert 3: mit 3fps
- Wert 7: 105 fps
- Wert 161:145 fps
- Wert 213: 166 fps

Das unten gezeigte Diagramm veranschaulicht die bei einer Einstellung auf 50 fps auftretenden Schwankungen vom ersten gemessenen Frame, bis hin zu Frame 100. Dieser Bereich wurde gewählt, um die Daten übersichtlicher darstellen zu können.

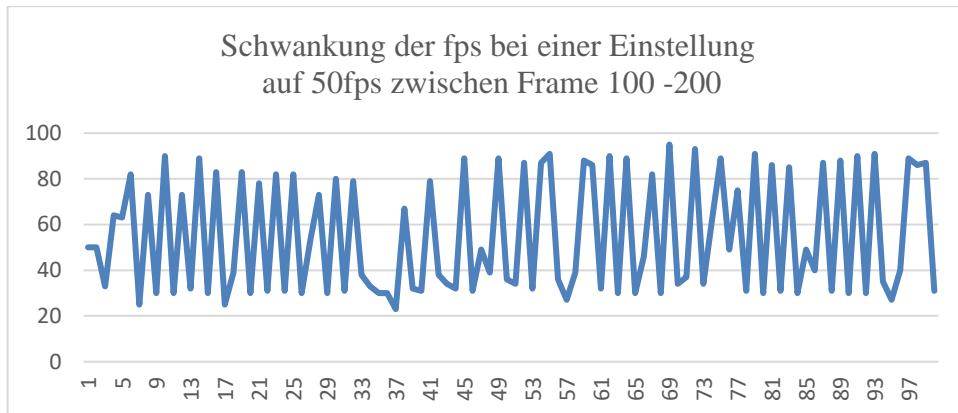


Abbildung 40: Schwankung der fps bei einer Einstellung

Dieses Diagramm macht deutlich, welchen Schwankungen die fps während des Versuchs mit 50 fps unterlagen.

²⁵ Das Diagramm der Orginaldaten findet sich im Anhang unter Latenzversuch.

²⁶ Zum Beleg wurden in Tabelle xx die Unbereinigten zum Vergleich mit den bereinigten Daten aufgeführt

Durch die Auswertung der Messdaten, ergaben sich dann folgende statistische Werte für die Messung.

	Mittelwert (gerundet)	Standartabweichung (gerundet)	Varianz (gerundet)
Unbereinigt 0-1000fps	0- 51,08	23,25	540,41
	Mittelwert	Standartabweichung	Varianz
Bereinigt 0-1000fps	50,87	22,72	516,17
	Mittelwert	Standartabweichung	Varianz
Diagramm fps 0 – 100	54,73	25,40	645,51

Tabelle 5: Schwankung der fps bei einer Einstellung auf 50fps

Aus der Tabelle 5 ergibt sich, dass der Mittelwert bei 51,08 fps lag, eine Standardabweichung von 23,25 und eine Varianz von 540,41 auftrat. Dies kann hier bezüglich der Latenzschätzung durch die Probanden nur eingeschränkt als aussagekräftig beschrieben werden²⁷, da die Messwerte außerhalb der Toleranz lagen. Die im Folgenden aufgeführten Versuche (40fps-10fps) unterlagen ähnlichen Problemen. Diese wurden aber mit geringerer fps-Einstellung deutlich konstanter.

²⁷ Hier fehlt noch der entsprechende Abschnitt über Die latenzschwankungen in entsprechenden Games

Messung mit 40 Bildern pro Sekunde

Die Auswirkung der Latenz von 40fps auf die Immersion und ihre entsprechende Schätzung ist im folgenden Diagramm aufgeführt.

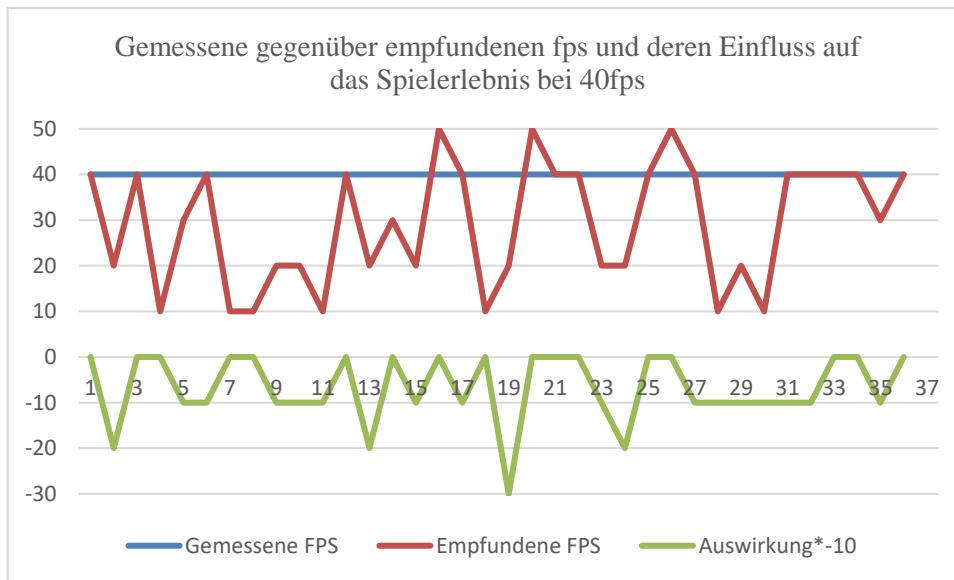


Abbildung 41: Gemessene gegenüber empfundenen fps und deren Einfluss auf das Spielerlebnis bei 40fps

Von einer tiefergehenden Auswertung wurde auch hier abgesehen, da auch hier in der unten aufgeführten Tabelle 6, Zweifel an der Aussagekraft des Versuchs ergaben.

Das unten gezeigte Diagramm veranschaulicht die bei einer Einstellung auf 40fps auftretenden Schwankungen vom 100. gemessenen Frame bis hin zum 200. Dieser Bereich wurde gewählt, um die Daten übersichtlicher darzustellen zu können.

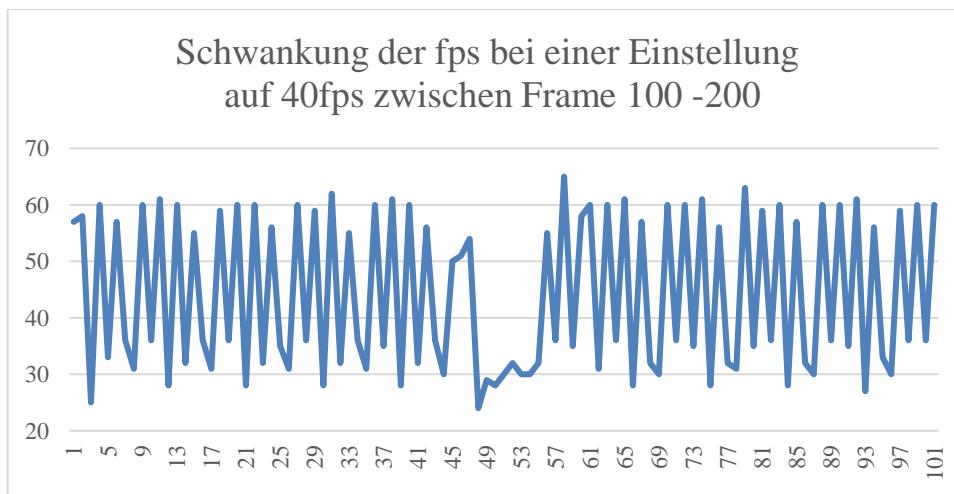


Abbildung 42: Schwankung der fps bei einer Einstellung

Dieses Diagramm macht deutlich, welchen Schwankungen die fps während des Versuchs mit 40fps unterlagen.

Die folgende Tabelle dient zur Darstellung der Genauigkeit der fps. Die verwendeten Messdaten wurden wie auch bei der Messung zu 50fps bereinigt, um die Analysewerte nicht zu verfälschen.²⁸

- Wert 3 mit 3 fps.

Durch die Auswertung der Messdaten, ergaben sich dann folgende statistische Werte für die Messung.

	Mittelwert (gerundet)	Standartabweichung (gerundet)	Varianz (gerundet)
Unbereinigt 0-1000fps	41,08	13,18	173,74
	Mittelwert	Standartabweichung	Varianz
Bereinigt 0-1000fps	41,12	13,13	172,45
	Mittelwert	Standartabweichung	Varianz
Diagramm 100 – 200fps	43,88	13,65	186,425

Tabelle 6: Schwankung der fps bei einer Einstellung auf 40fps

Auch bei dieser Tabelle ergibt sich wie bei Tabelle 5 aus dem Versuch mit 50fps, dass die statistischen Abweichungen zu hoch waren, um wirklich belastbare Ergebnisse zu liefern.

²⁸ Das Diagramm der Orginaldaten findet sich im Anhang. Die Messwerte befinden sich auf dem Datenanhang

Messung mit 30 Bildern pro Sekunde

Da bei dieser Messung die fps-Werte konstant waren, wie in Tabelle 7 ersichtlich, konnte hier nun eine tiefergehende Bewertung vorgenommen werden.

	Mittelwert (gerundet)	Standartabweichung (gerundet)	Varianz (gerundet)
Unbereinigt 0-1000fps	29,24	3,82	14,59
Bereinigt 0-1000fps	29,27	3,72	13,91

Tabelle 7: Schwankung der fps bei einer Einstellung auf 30fps

Hier lag der Mittelwert mit einer Standardabweichung von 3,82 in der notwendigen Eingrenzung von +/- 5. Mit einigen Ausnahmen lagen bei dieser fps-Zahl alle Probanden nahe der gemessenen Zahl. Entsprechend der Schätzung der Probanden veränderte sich auch die Auswirkung. Die Auswirkung der Latenz von 30 fps auf die Immersion und ihre entsprechende Schätzung ist im folgenden Diagramm aufgeführt.

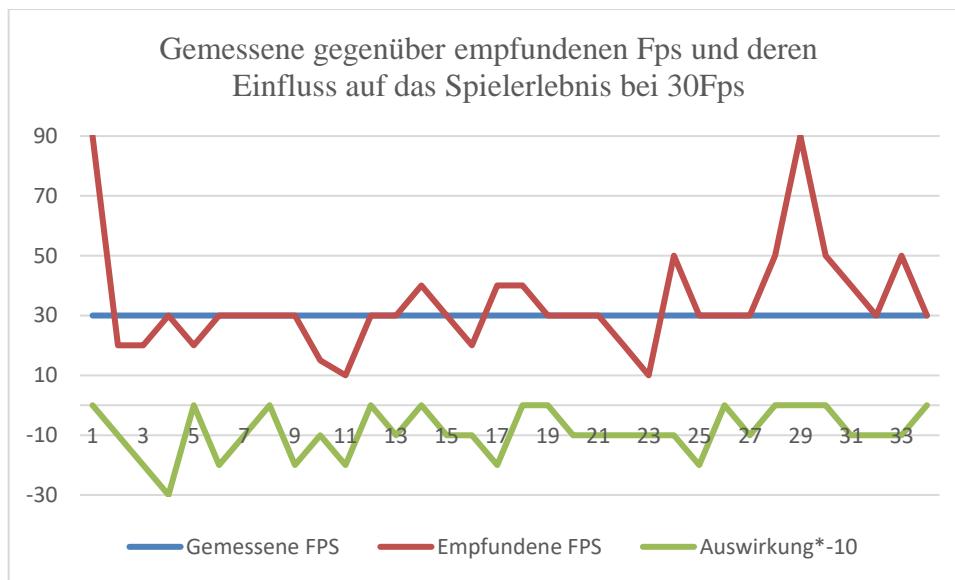


Abbildung 43: Gemessene gegenüber empfundenen Fps und deren Einfluss auf das Spielerlebnis bei 30Fps

Wie oben im Diagramm dargestellt, sinkt das positiv empfundene Spielerlebnis entsprechend der Schätzung der Probanden. Hier ist zu beachten, dass die Originalmesswerte der Auswirkung *-10 genommen wurden, um die Darstellbarkeit zu verbessern.

Messung mit 20 Bildern pro Sekunde

Aus dem unten dargestellten Diagramm geht hervor, wie auch beim Versuch mit 30fps, dass sich hier die empfundene Latenzzeit der Auswirkung angleicht. Also bei mehr empfundenen fps wurde auch ein besseres Spielerlebnis festgestellt.

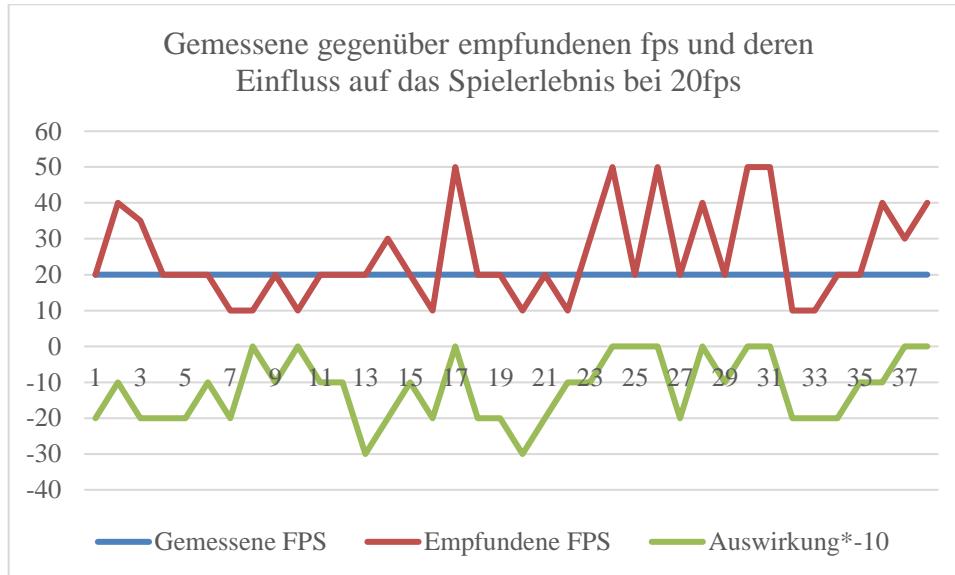


Abbildung 44: Gemessene gegenüber empfundenen Fps und deren Einfluss auf das Spielerlebnis bei 20fps

Die unten dargestellten Werte zeigen die Werte der Messung bei 20 fps. Auch hier lagen die Werte der Standardabweichung mit 3,15 klar im Toleranzbereich.

	Mittelwert (gerundet)	Standartabweichung (gerundet)	Varianz (gerundet)
Unbereinigt 0-1000fps	21,30	3,15	9,94
	Mittelwert	Standartabweichung	Varianz
Bereinigt 0-1000fps	21,94	3,05	9,32

Tabelle 8: Schwankung der fps bei einer Einstellung auf 20fps

Messung mit 10 Bildern pro Sekunde

Das unten dargestellte Diagramm zeigt, dass die Probanden hier eindeutig in der Lage waren, 10 fps zu schätzen. Hier ist festzustellen, dass die Auswirkung nicht immer den zu erwartenden schlechtesten Wert ergab. Hier war es bei manchen Probanden der Fall, dass sich diese von 10 fps nicht beeinflusst fühlten. Dies liegt wahrscheinlich daran, dass den Probanden in den frühen Versuchen von Eins bis Fünf kein Vergleichswert vorlag.²⁹

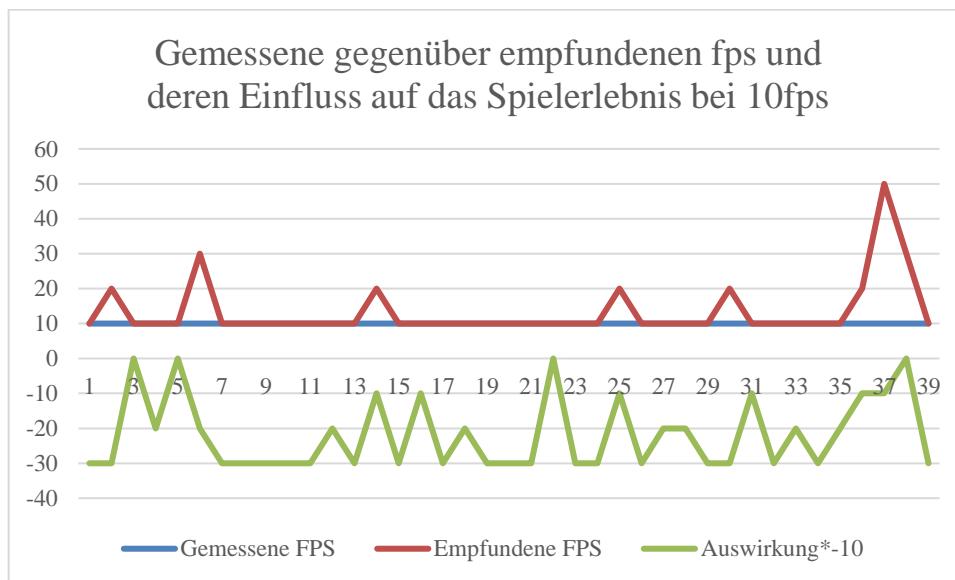


Abbildung 45: Gemessene gegenüber empfundenen fps und deren Einfluss auf das Spielerlebnis bei 10fps

Die folgenden Daten dienen zur Darstellung der Genauigkeit der fps. Die verwendeten Messdaten wurden entsprechend bereinigt, um die Analysewerte nicht zu verfälschen.³⁰ Wie auch bei den beiden vorangegangen Versuchen, war die Standardabweichung im Rahmen der Toleranz.

	Mittelwert (gerundet)	Standartabweichung (gerundet)	Varianz (gerundet)
Unbereinigt 0-1000fps	10,00	4,10	16,83
	Mittelwert	Standartabweichung	Varianz
Bereinigt 0-1000fps	9,59	4,10	16,85

Tabelle 9: Schwankung der fps bei einer Einstellung auf 10fps

²⁹ Genauere Ausführung in der Gesamtauswertung und dem Ausblick

³⁰ Das Diagramm der Orginaldaten findet sich im Anhang. Die Messwerte befinden sich auf dem Datenanhang

Gesamtauswertung des Fragebogens – Gegenüberstellung und Auflistung

In der nachfolgenden Gesamtauswertung bezieht sich:

- Versuch 1 auf Untersuchung der Immersion durch ein VR-Video,
- Versuch 2 auf Untersuchung der Immersion in einer VR-Anwendung mit visuellem Feedback,
- Versuch 3 auf Untersuchung der Immersion in einer VR-Anwendung mit haptischen Feedback,
- Versuch 4 auf Untersuchung der Immersion in einer VR-Anwendung mit erhöhter Latenzzeit.

Die untenstehende Tabelle zeigt unter Angabe des Geschlechts das durchschnittliche Alter der Probanden, die statistische Abweichung, das kleinste und größte angegebene Alter, sowie die Anzahl der Probanden des jeweiligen Geschlechtes.

		2._Alter?				
		Mean	Std Dev	N	Min	Max
3._Geschlecht	männlich	26.17	7.94	N=24	19.0	58.0
	sonstige	25.00	nan	N=1	25.0	25.0
	weiblich	28.36	15.44	N=14	16.0	60.0

Tabelle 10: Alter nach Geschlecht

In der Gesamtauswertung der Fragebögen hat sich bei insgesamt 24 männlichen Probanden ein Durchschnittsalter von 26,17 Jahren ergeben, bei den 14 weiblichen Testpersonen liegt der Durchschnitt bei 28,36 Jahren, eine sonstige Person hat ein Alter von 25 Jahren angegeben. Auffällig ist hier, dass die Verteilung von Frauen und Männern ungleich von 24 zu 14 und einem Sonstigen ist. Dies ist bei der weiteren Auswertung der erhobenen Daten zu berücksichtigen.

Der Unterschied bei der Immersion in Bezug auf das Geschlecht zeigt bei allen 3 untenstehenden Versuchen eine höhere Immersion bei den männlichen Probanden. Die Fragen konnten mit

Werten zwischen 1 und 6 bewertet werden, aufsteigend von 1 (*sehr*) bis 6 (*gar nicht*).³¹ Alle drei unten aufgeführten Tabellen stellen das durchschnittliche Ergebnis der Antworten zu den beiden Fragen: *In wie weit konnten sie sich in die virtuelle Welt hineinversetzen?* und *Konnten Sie alles um sich herum vergessen?* bezogen auf das Geschlecht der Probanden dar. Angegeben sind auch die statistischen Abweichungen, die minimale und maximale Bewertung und die Gesamtzahl der Befragten aufgeteilt auf das jeweilige Geschlecht.

		Versuch 1 Vergessen					Versuch 1 Hineinversetzen				
		Mean	Std Dev	N	Min	Max	Mean	Std Dev	N	Min	Max
Ge- schlecht	männ- lich	2.17	1.31	N=24	1.0	6.0	1.88	0.61	N=24	1.0	3.0
	sonstige	4.00	nan	N=1	4.0	4.0	2.00	nan	N=1	2.0	2.0
	weiblich	2.50	1.34	N=14	1.0	6.0	2.36	0.93	N=14	1.0	4.0

Tabelle 11: Versuch 1 nach Geschlecht

So liegt bei den Antworten des ersten Versuchs der Durchschnittswert bei der Frage 1: *In wie weit konnten sie sich in die virtuelle Welt hineinversetzen?* bei den männlichen Testpersonen bei 1,88 und bei den weiblichen Personen bei 2,36. Bei der Frage 2: *Konnten Sie alles um sich herum vergessen?* liegen die Werte bei den männlichen Teilnehmern bei 2,17 zu 2,50 bei den weiblichen.

		Versuch 2 Vergessen					Versuch2 Hineinversetzen				
		Mean	Std Dev	N	Min	Max	Mean	Std Dev	N	Min	Max
Ge- schlecht	männlich	1.83	1.20	N=24	1.0	6.0	1.92	0.78	N=24	1.0	3.0
	sonstige	1.00	nan	N=1	1.0	1.0	1.00	nan	N=1	1.0	1.0
	weiblich	1.93	1.00	N=14	1.0	4.0	2.21	1.25	N=14	1.0	4.0

Tabelle 12: Versuch 2 nach Geschlecht

Der zweite Versuch bestätigt die aus dem ersten Versuch hervorgehenden Tendenzen und zeigt, dass bei Frage 1 mit 1,92 bei männlichen zu 2,21 bei weiblichen Testpersonen und bei Frage 2 mit 1,83 (m) zu 1,93 (w) sich die männlichen Testpersonen wieder mehr in die VR-Anwendung hineinversetzen können.

³¹ Gesamte Auflistung des Fragebogens im Anhang

		Versuch 3 Hineinversetzen					Versuch 3 Vergessen				
		Mean	Std Dev	N	Min	Max	Mean	Std Dev	N	Min	Max
Ge- schlecht	männ- lich	1.88	1.03	N=24	1.0	4.0	1.96	1.00	N=24	1.0	5.0
	sonstige	2.00	nan	N=1	2.0	2.0	2.00	nan	N=1	2.0	2.0
	weiblich	2.00	0.96	N=14	1.0	4.0	2.00	1.24	N=14	1.0	5.0

Tabelle 13: Versuch 33 nach Geschlecht

Bei Versuch 3 ergab sich für Frage 1 ein durchschnittlicher Wert bei den Antworten der männlichen Probanden von 1,96 und bei den weiblichen von 2,00. Bei Frage Zwei lag der Wert bei den Männern bei 1,88 und bei den Frauen bei 2,00. Wie auch Versuch Zwei, belegt Versuch Drei eine zwar geringer werdende, aber immer noch vorhandene Abweichung bei den Frauen und Männern. Von Versuch Eins zu Zwei ist auch festzustellen, dass Versuch Eins bei den Frauen weniger immersiv war als Versuch Zwei. Zwischen Versuch Zwei und Drei ergibt sich dies bezüglich keine signifikante Änderung.

Im Verlauf von Versuch Eins bis Drei zeigten sich, im Verhältnis von Frauen zu Männern, folgende Tendenzen.

Das unten gezeigte Diagramm zeigt den Verlauf der Durchschnittswerte bzgl. des Geschlechts über die Drei Versuche.

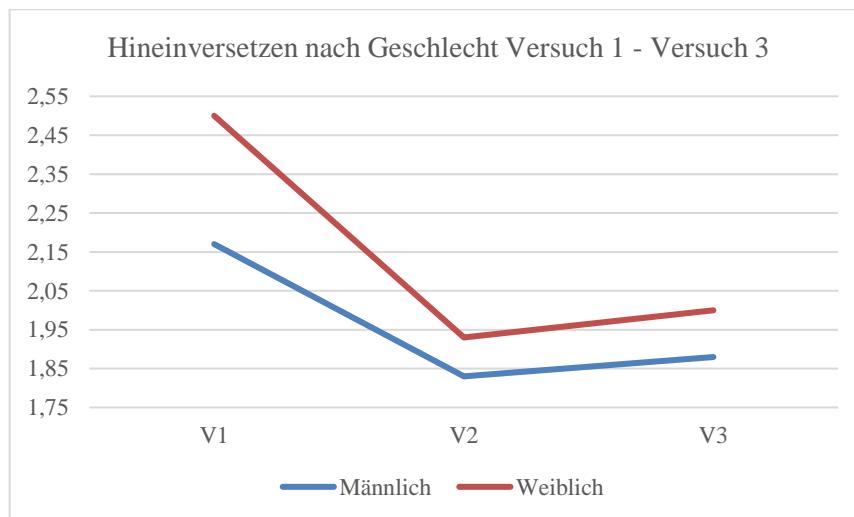


Abbildung 46: Hineinversetzen nach Geschlecht Versuch1 - Versuch3

Aus der gezeigten Grafik lässt sich schlussfolgern, dass bei allen Versuchen, die Männer sich besser in die virtuelle Welt hineinversetzen konnten, als die Frauen. Dabei zeigt sich auch, dass von Versuch Eins zu Versuch Zwei das Hineinversetzen stark ansteigt. Bei Versuch 3 fällt dies aber wieder leicht ab. Dies könnte ein Hinweis darauf sein, dass sich Männer wie Frauen bei rein visuellem Feedback besser hineinversetzen können. Diese Abweichung ist aber zu klein, um hier klare Aussagen zu treffen.

Der unten gezeigte visuell dargestellte Datensatz gibt Auskunft darüber, in wie weit die Männer und Frauen die reale Umgebung vergessen konnten, von Versuch Eins bis Drei.

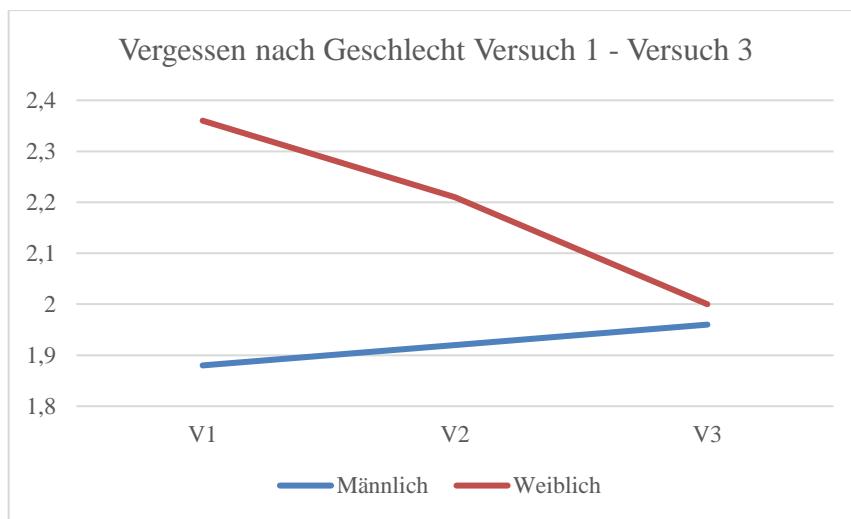


Abbildung 47: Vergessen nach Geschlecht Versuch 1 - Versuch 3

Bei der Aufschlüsselung der Versuche nach dem Geschlecht und der Frage, in wie weit die Probanden alles um sich herum vergessen konnten, zeigt sich eindeutig, dass sich die Männer und Frauen von Versuch Eins bis Versuch Drei, in ihrem Empfinden angleichen.

Die nächste Tabelle zeigt, unter Angabe des Geschlechts, das durchschnittliche Ergebnis der Antworten auf die Frage: *Wie oft nutzen Sie Interaktive Anwendungen bzw. Computerspiele auf Desktop-PCs, Handys, Tablets, etc. im Alltag*, angegeben in Wochenstunden. Ebenso lässt sich aus der Tabelle die statistische Abweichung, das kleinste und größte Ergebnis, sowie die Anzahl der Probanden des jeweiligen Geschlechtes entnehmen.

		Wochenstunden Zocken				
Geschlecht		Mean	Std Dev	N	Min	Max
	männlich	15.08	13.49	N=24	0.0	42.0
	sonstige	70.00	nan	N=1	70.0	70.0
	weiblich	6.79	7.74	N=14	0.0	20.0

Tabelle 14: Wochenstunden nach Geschlecht

Die durchschnittliche Anzahl der Wochenstunden liegt bei den Männern bei 15,08 und bei den Frauen bei 6,79 Stunden. Die Anzahl der Männer ist gegenüber den Frauen auch deutlich höher und die maximale Anzahl der Wochenstunden liegt mit 42 Stunden bei den männlichen Probanden deutlich über dem Maximalwert von 20 Stunden bei den Frauen.

Die drei nachfolgenden Tabellen vergleichen die Antworten der beiden Fragen bezüglich der Immersion mit den Angaben der Wochenstunden.

Da 15 der 39 Testpersonen Brillenträger sind, haben wir untersucht, ob bei Brillenträgern eine andere Wahrnehmung der Immersion besteht. Die Tabelle gibt den Mittelwert der drei Versuche an, unterteilt in Brillenträger oder nicht.

		Mittelwert Versuche 1-3		
Sehhilfe		Mean	Std Dev	N
	ja	2.08	0.91	N=15
	nein	1.97	0.73	N=24

Tabelle 15: Mittelwert Versuch 1 – 3 nach Sehhilfe

Anhand der Werte von 2,08 bei Brillenträgern zu 1,97 bei Testpersonen, die keine Brille tragen, lässt sich kein deutlicher Unterschied in der Immersion feststellen.

Die durchschnittlichen Frageergebnisse der Versuche Eins bis Drei in Abhängigkeit der bisherigen VR-Brillen Nutzung sind unten aufgelistet.

		Vergessen? V1		Hineinver- setzen? V1		Vergessen? V2		Hineinver- setzen? V2		Vergessen? V3		Hineinver- setzen? V3	
		Mean	Std Dev	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
Vr- Brille- Ge- nutzt	ja	2.32	1.20	2.16	0.69	1.63	0.90	1.79	0.92	1.58	0.61	1.74	0.87
	nein	2.35	1.46	1.95	0.83	2.05	1.28	2.20	1.01	2.35	1.27	2.10	1.07

Tabelle 16: Nutzung VR-Brille nach Versuch 1 bis 3

Diese zeigen, mit einer Ausnahme bei Versuch Eins, eine klar positivere Bewertung bzgl. der Fragen bei den Probanden, die schonmal eine VR-Brille genutzt haben. Also konnten sich die VR-Nutzer besser hineinversetzen als die, die noch keine benutzt hatten.

In einer weiteren Gegenüberstellung haben wir die Werte der empfundenen Framerate mit dem Einfluss auf die Immersion anhand der Frage: *Wie sehr hat diese Framerate Ihr Spielerlebnis beeinflusst?* verglichen.

Die Tabelle zeigt die Werte der empfundenen Framerate, von 10 bis 50 in Zehnerschritten und den jeweils dazu ermittelten durchschnittlichen Wert, wie weit die Immersion durch die Framerate gestört wurde. Dabei wurden die vier Antwortmöglichkeiten (Sehr stark, stark, ein bisschen, gar nicht) auf die Zahlen 0 bis 3 ausgelegt, um die Daten statistisch vergleichen zu können.

	Gesamteinfluss		
	Mean	Std Dev	N
Empfundene Framerate	50.0	0.22	0.43 N=18
	40.0	0.35	0.55 N=31
	30.0	0.88	0.81 N=34
	20.0	1.36	0.80 N=45
	10.0	1.94	1.14 N=54

Tabelle 17: Gesamteinfluss nach empfundener Framerate

Die Gegenüberstellung zeigt, dass mit einer steigenden Framerate der negative Einfluss auf die Immersion abnimmt. Bei empfundenen 10 fps liegt der Wert bei 1,94 was einem starken Einfluss entspricht. Bei empfundenen 50 fps liegt der Wert bei 0,22, das Spielerlebnis wurde also so gut wie gar nicht durch die Framerate beeinflusst.

	Empfundene_Latenz_V1		Einfluss_1	
	Mean	N	Mean	N
Gemessene_Latenz_V1	10.0	13.33	N=39	2.18 N=39
	20.0	25.00	N=38	1.18 N=38
	30.0	33.97	N=34	0.88 N=34
	40.0	28.38	N=37	0.65 N=37
	50.0	30.61	N=41	0.73 N=41

Tabelle 18: Empfundene Latenz und Einfluss nach der gemessenen Latenz

Die unten gezeigte Tabelle listet das Verhältnis der folgenden Fragen auf:

- Hat Sie die *Leap Motion* technisch soweit überzeugt, dass die feste Integration einen Aufpreis rechtfertigen würde?
- Welchen Aufpreis wären Sie bereit zu zahlen?
- Hat Sie die Objekterkennung soweit überzeugt, dass die feste Integration einen Aufpreis rechtfertigen würde?
- Welchen Aufpreis wären Sie bereit zu zahlen?

<i>Leap Motion</i>			
Gesamt	Ja	Mittelwert	Standardabweichung
39	28	99,64	64,94
Objekterkennung			
Gesamt	Ja	Mittelwert	Standartabweichung
39	16	85	62,25

Tabelle 19: Zahlungsbereitschaft für *Leap Motion* und Objekterkennung

Aus der Tabelle geht hervor, dass bei der *Leap Motion* 28 von 39 Probanden, dazu bereit wären einen Aufpreis von ca. 100€ im Durchschnitt zu zahlen. Hier zeichnet sich ein klarer Wunsch für die Handerkennung ab. Für die Objekterkennung gilt dies nicht, hier haben von 39 Probanden, 16 mit ja geantwortet. Diese wären im Durchschnitt dazu bereit, einen Aufpreis von ca. 85€ zu zahlen. Das heißt nicht, dass die Probanden sich keine Handerkennung und Objekterkennung wünschen, sondern dass nicht alle dazu bereit sind, für den aktuellen Stand der Technik einen Aufpreis zu zahlen.

Die unten dargestellten Fragen stellen die Probanden vor die Wahl, welche Gründe es für sie gibt, eine VR-Brille zu kaufen oder was dagegenspricht. Hier waren Mehrfachnennungen möglich und Antworten konnten in Ja und Nein gegeben werden.

Können Sie sich vorstellen eine VR-Brille zu kaufen? Wenn ja, weil: (Vor den Versuchen)

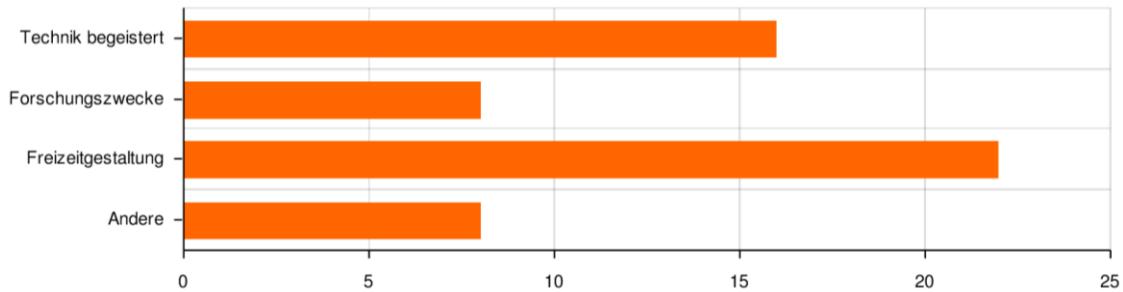


Abbildung 48: Können Sie sich vorstellen eine VR-Brille zu kaufen? Wenn ja, weil: (Vor den Versuchen)

Können Sie sich vorstellen eine VR-Brille zu kaufen? Wenn ja, weil: (Nach den Versuchen)

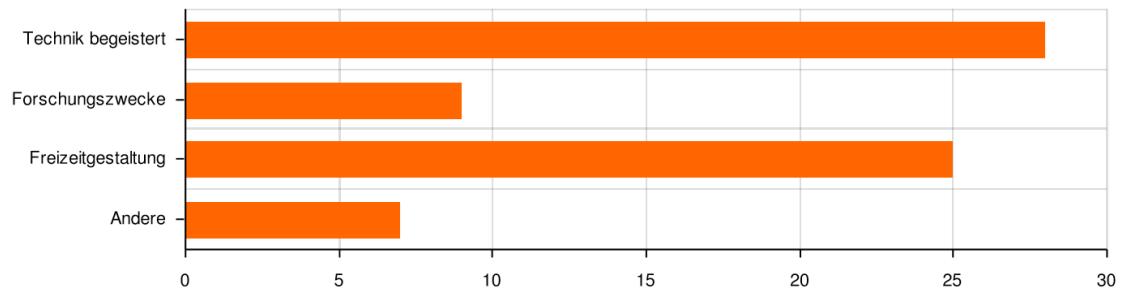


Abbildung 49: Können Sie sich vorstellen eine VR-Brille zu kaufen? Wenn ja, weil: (Nach den Versuchen)

Die Gründe für eine positive Kaufbereitschaft zeigen im Vergleich, dass nach den Versuchen die Begeisterung ein größeres Kriterium für den Kauf einer VR-Brille war.

Auf dieser Seite sind die Gründe aufgeführt, die gegen einen Kauf einer VR-Brille sprechen würden.

Können Sie sich vorstellen eine VR-Brille zu kaufen? Wenn nein, weil: (Vor den Versuchen)

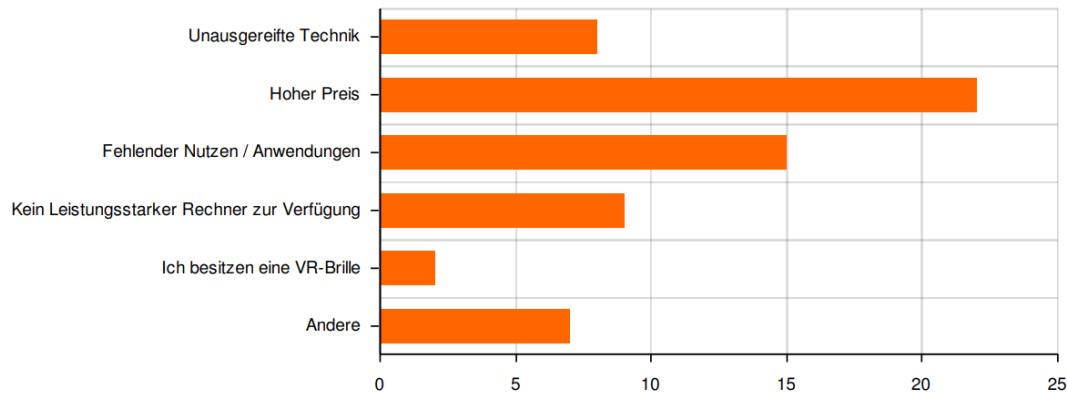


Abbildung 50: Können Sie sich vorstellen eine VR-Brille zu kaufen? Wenn nein, weil: (Vor den Versuchen)

Können Sie sich vorstellen eine VR-Brille zu kaufen? Wenn nein, weil: (Nach den Versuchen)

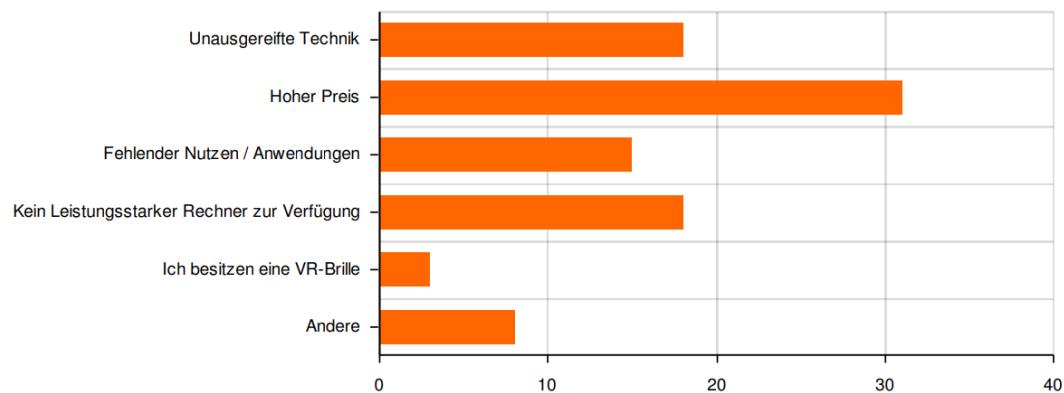


Abbildung 51: Können Sie sich vorstellen eine VR-Brille zu kaufen? Wenn nein, weil: (Nach den Versuchen)

Bei den Gründen, die dagegensprechen, eine VR-Brillen zu kaufen, zeigt sich im Vorher-Nachher-Vergleich, dass der hohe Preis als Grund zwar leicht sinkt, die unausgereifte Technik aber ansteigt.

Im Vergleich zu den Gründen dafür schließt sich hieraus, dass zwar die Technik die Probanden begeistert, aber dabei noch Verbesserungsbedarf besteht.

Resümee und Ausblick

In diesem Kapitel wird die praktische Arbeit reflektiert und ein Ausblick auf vertiefende Untersuchungsmöglichkeiten gegeben.

In der technischen Vorbereitung gestaltete sich die Kombination des Multitarget-Tracking schwierig, da zwischen der Kamera der *HTC Vive* und den Kameras der *Leap Motion* ein Versatz vorhanden war. Dieser wurde zwar versucht, mit dem im Anhang zu findenden Skript „Transformation“ auszugleichen, dies führte aber nicht zum Erfolg. Dieses Skript ermöglichte zwar eine Koordinatentransformation, konnte aber den räumlichen Versatz nicht ausgleichen. Daher wurde dieser Ansatz nicht weiterverfolgt, da es den Zeitplan weit überschritten hätte. Die direkte, technische Integration von Systemen wie der *Leap Motion* in HMDs wird in Zukunft das Zusammenführen von Hand- und Gestensteuerung, wie auch die Objekterkennung vereinfachen.

Für die aussagekräftigere Bewertung der Ergebnisse wäre es vorteilhafter gewesen, eine ausgeglichene Geschlechterverteilung zu haben. Zudem wäre auch wünschenswert gewesen, wenn die Altersverteilung sich nicht auf die Zwanziger konzentriert hätte, bzw. wenn mehr Probanden an der Studie teilgenommen hätten. Dasselbe Problem trat auch bei der Frage: Wie oft nutzen Sie Interaktive Anwendungen bzw. Computerspiele auf Desktop-PCs, Handys, Tablets, etc. im Alltag, auf. Hier war der Großteil der Nutzungsstunden zwischen Null und 20 verteilt, was die statistischen Ausreißer weniger nutzbar machte.

Bei Versuch Eins wäre es sicher interessant gewesen, aktuelle High-End Grafik zu verwenden um damit die Immersion der Probanden zu testen. Das war aber im Umfang der Thesis nicht möglich. Es hätte des Weiteren dazu geführt, dass die Vergleichbarkeit der Versuche untereinander nur eingeschränkt bewertbar wäre.

Zu Versuch Zwei und Drei hätte ein zusätzlicher Versuch mit den aktuellen Eingabegeräten der aktuellen VR-System zur Vergleichbarkeit mit diesen beigetragen, allerdings lag unser Fokus darauf, die Integration von realen Objekten umzusetzen. Beim randomisierten Wechsel zwischen Versuch Zwei und Drei fiel auf, dass bei der Reihenfolge Versuch Zwei, dann Drei, bei den Probanden ein Überraschungseffekt eintrat. Dieser bezog sich auf das plötzliche feststellen, dass das Bedienpult nun real vorhanden war. Bei der Reihenfolge Drei zu Zwei war das weniger zu bemerken, da dort mit dem realen Bedienpult begonnen wurde. Hier wäre es interessant

gewesen, diesen Effekt zu messen und festzuhalten. Da dies aber erst später im Test festgestellt werden konnte, war hier keine Änderung mehr möglich.

Versuch Drei hätte von einem exakteren Würfeltracking profitiert. Dies zeigt sich auch dadurch, dass die technische Ausgereiftheit von den Probanden bemängelt wurde. Hier wäre es spannend, weitergehende Untersuchungen anzustellen, in wie weit sich die realen Objekte mit der Nutzerinteraktion kombinieren lassen und welche Auswirkungen diese auf die Wahrnehmung hat.

Die in den Latenzversuchen aufgetretenen Unklarheiten werfen Fragen auf. Wie z.B. im Abschnitt für die Messung mit 10 Bildern pro Sekunde. Hier war nicht belegbar, ob die Schätzungsicherheit bzw. Unsicherheit der Probanden auf destehende oder mangelnde Kenntnis des fps- Eindrucks, oder aber auf die ansteigende Zahl der Versuche, zurückzuführen ist. Es ist eine abgeänderte Messung notwendig, um festzustellen, ob die Probanden bei jeder Erstschatzung gut oder schlecht liegen, und ob sich diese Werte mit zunehmender Erfahrung verbessern. Auch wäre es hier notwendig, mit den höheren fps Zahlen konstantere Werte zu erhalten. Da diese starken Schwankungen unterlagen. Diese sind wahrscheinlich auf das Skript und die Messweise der Systeminternen *Unity Engine* fps zurück zu führen. Das im Anhang angefügtes Skript³² würde hier eventuell Abhilfe schaffen. Auch über die Technik der fps-Beeinflussung, die wie erwähnt durch das Ansprechen des Kamera-Renderers bewerkstelligt wurde, muss nochmals verbessert werden. Eine weitere Fragestellung ist, ob die Vorgaben der fps-Werte von 10 – 50 die Messung beeinflusst haben, da es nachvollziehbar erscheint, wenn der Proband bei der ihm empfunden langsamsten fps-Zahl, Zehn antwortet. Sind keine Vorgaben gemacht, werden sich bei diesen Ergebnissen möglicherweise Differenzen zu den im Kapitel *Latzversuch Auswertung* erwähnten Messwerten ergeben.

³² Siehe unter Anhang FPS_Messung_2

Fazit

In dieser Thesis hatten wir uns das Ziel gesetzt, eine Integration realer Objekte in die virtuelle Welt zu bewerkstelligen. Dabei wollten wir die Auswirkungen auf die Immersion testen, um ihre Relevanz zu belegen. Durch die Tatsache, dass wir eine Doppelthesis geschrieben haben, konnten wir unser Forschungsumfeld um den Bereich der Latenz erweitern.

Die Objekterkennung schien für uns besonders interessant, weil wir herausfinden wollten, ob den User eine visuelle oder eine haptische Interaktion mehr überzeugt. Das fest integrierte Bedienpult hätte dafür ausgereicht, wir wollten aber auch die Integration von beweglichen Objekten realisieren.

Die grundsätzliche Funktion der Objekterkennung war in unseren Versuchen gegeben, allerdings hätten wir uns mehr Interaktionsmöglichkeiten in der Testanwendung gewünscht. Es wäre unter anderem interessant gewesen, zu untersuchen, welche Auswirkungen eine Interaktion mit mehreren beweglichen Objekten, oder die gemeinsame Interaktion in einer Multiplayer-Umgebung, auf das Spielerlebnis haben.

Die Umsetzung der Handerkennung mit der *Leap Motion* war für uns problemlos umzusetzen. Nur die technische Integration in die VR-Systeme in Kombination mit der Objekterkennung weist noch Hürden auf. Hinsichtlich des Plug & Play Ansatzes sind hier noch einige Schritte zu beachten, um dem Consumer-Markt den Zugang zu erleichtern.

Unsere Forschung zur Auswirkung der Latenzzeit auf das Spielerlebnis haben wir anhand der Bildwiederholfrequenz gemessen, es wäre aber auch interessant gewesen, die Latenzzeit in einer Netzwerkumgebung zu messen und die Auswirkung auf die Mensch-zu-Mensch-Kommunikation zu erforschen.

Durch unsere Recherche und Forschung in Richtung der VR-Technik, konnten wir die Erkenntnis gewinnen, dass sich der Markt um VR in Richtung einer intuitiveren Interaktion entwickelt. Diese Erkenntnis zeigte sich auch in der von uns durchgeführten Befragung, da viele der Probanden sich eine realitätsnähere Präsenz im virtuellen Raum, mit einer natürlicheren und intuitiveren Bedienbarkeit gewünscht hätten.

Mit unseren Untersuchungen haben wir die von uns angedachten Problemstellungen behandelt, hätten sie aber gerne umfangreicher und in einem größeren Ausmaß untersucht.

Durch die Arbeit in diesem Themengebiet konnten wir interessante Erkenntnisse über die Auswirkungen von Interaktionsmöglichkeiten, Latenzzeiten und der Integration realer Objekte auf die Immersion sammeln und es hat sich gezeigt, dass der Umfang des Themengebiets mehrere Möglichkeiten für unterschiedliche Folgeuntersuchungen bietet.

Literaturverzeichnis

- Dörner, R., Broll, W., Grimm, P. and Jung, B. (Eds.) (2013), *Virtual und Augmented Reality (VR/AR): Grundlagen und Methoden der Virtuellen und Augmentierten Realität*, eXamen.press, Springer Vieweg, Berlin, Heidelberg.
- Jerald, J. (2016), *The VR book, human-centered design for Virtual Reality*, ACM; M&C, [San Rafael].
- Meehan, M., Insko, B., Whitton, M. and Brooks, F.P. (2002), “Physiological measures of presence in stressful virtual environments”, *ACM Transactions on Graphics*, Vol. 21 No. 3.
- Parisi, T. (2015), *Learning virtual reality: Developing immersive experiences and applications for desktop, Web, and mobile*, O'Reilly, Sebastopol CA.
- Sutherland, I.E. (1965), “The ultimate display”, In The Congress of the International Federation of Information Processing (IFIP), pp. 506–508.
- Vince, J. (2004), *Introduction to virtual reality*, Springer Berlin Heidelberg, London, Berlin, Heidelberg [u.a.].
- Weichert, F., Bachmann, D., Rudak, B. and Fisseler, D. (2013), “Analysis of the accuracy and robustness of the leap motion controller”, *Sensors (Basel, Switzerland)*, Vol. 13 No. 5, pp. 6380–6393.

Onlinequellen

- Bloch, Y. (2016), “Orion – Neue VR Hand Tracking Engine von Leap Motion”, available at: <https://www.vrnerds.de/orion-neue-vr-hand-tracking-engine-von-leap-motion/> (accessed 18 September 2017).
- “Image Targets” (2017), available at: <https://library.vuforia.com/articles/Training/Image-Target-Guide> (accessed 24 September 2017).
- Leap Motion & HTC Vive Setup, , available at: <https://developer.leapmotion.com/vr-setup/vive> (accessed 20 September 2017).
- Leap Motion Unity Core Assets, , available at: <https://developer.leapmotion.com/unity#116> (accessed 20 September 2017).
- Leopold, H., Untermeier, A. (2017), “Ungeahnte Dimensionen. Virtual Reality - Immersion, Interaktion und Insights”, available at: <https://www.research-results.de/fachartikel/2017/ausgabe-4/ungeahnte-dimensionen.html> (accessed 18 September 2017).
- McCormick, J. (2015), “Examining the Valve/HTC Vive Ecosystem: Basic Lighthouse Operation”, available at: <https://metaversing.com/2015/03/23/examining-the-valvehtc-vive-ecosystem-basic-lighthouse-operation/> (accessed 18 September 2017).
- “Multi-Targets” (2017), available at: <https://library.vuforia.com/articles/Training/Multi-Target-Guide> (accessed 24 September 2017).
- Nguyen, T. (2017), “The best VR headset”, available at: <http://www.pcgamer.com/the-best-vr-headset/> (accessed 18 September 2017).
- “Oculus Rift | Oculus”, available at: <https://www.oculus.com/rift/> (accessed 18 September 2017).
- Prasuethsut, L. (2016), “Oculus Rift review”, available at: <https://www.wearable.com/oculus-rift/oculus-rift-review> (accessed 18 September 2017).
- Staudacher, R. (2016), “Oculus Rift und HTC Vive im Vergleich: Welches VR-System ist besser?”, available at: <http://www.pcgameshardware.de/Virtual-Reality-Hardware-258542/Tests/Oculus-Rift-und-HTC-Vive-Vergleich-Test-1194231/> (accessed 18 September 2017).
- SteamVR setup, available at: https://support.steampowered.com/steamvr/HTC_Vive/ (accessed 20 September 2017).
- Unity3D: DOCUMENTATION C#, available at: <https://docs.unity3d.com/SkriptReference/Application-target-FrameRate.html> (accessed 20 September 2017).
- Unity3D: DOCUMENTATION C#, available at: <https://docs.unity3d.com/Manual/RenderingStatistics.html> (accessed 20 September 2017).
- Vuforia Developer Portal: Support, available at: <https://developer.vuforia.com/forum/digital-eyewear/htc-vive-support> (accessed 24 September 2017).
- “VR Brillen Vergleich - VR-Nerds”, available at: <https://www.vrnerds.de/vr-brillen-vergleich/> (accessed 18 September 2017).
- “Vuforia Object Scanner” (2017), available at: <https://library.vuforia.com/articles/Training/Vuforia-Object-Scanner-Users-Guide.html> (accessed 24 September 2017).
- “VuMark” (2017), available at: <https://library.vuforia.com/articles/Training/VuMark> (accessed 24 September 2017).

Abbildungsverzeichnis

Abbildung 1: https://de.statista.com/statistik/daten/studie/685856/umfrage/umfrage-zur-bekanntheit-von-ausgewahlten-vr-brillen-unter-gamern-in-deutschland/	Fehler! Textmarke nicht definiert.
Abbildung 2: https://www.oculus.com/press-kit/	14
Abbildung 3: https://www.oculus.com/press-kit/	15
Abbildung 4: https://www.vive.com/us/pr/newsroom-gallery/	16
Abbildung 5: http://gallery.Leap Motion.com/download/press/LM%20Mount+Vive-Front.png:	18
Abbildung 6: https://www.Leap Motion.com/press#117	19
Abbildung 7: https://library.Vuforia.com/articles/Training/VuMark	31
Abbildung 8: https://library.Vuforia.com/articles/Training/Image-Target-Guide	32
Abbildung 9: https://library.Vuforia.com/articles/Solution/Optimizing-Target-Detection-and-Tracking-Stability	33
Abbildung 10: https://library.Vuforia.com/articles/Training/Multi-Target-Guide	33
Abbildung 11: https://library.Vuforia.com/articles/Training/Vuforia-Object-Scanner-Users-Guide.html	34
Abbildung 12: https://library.Vuforia.com/articles/Training/Vuforia-Object-Scanner-Users-Guide.html	35
Abbildung 13: https://library.Vuforia.com/articles/Training/Vuforia-Object-Scanner-Users-Guide.html	35
Abbildung 14: Screenshot Vive-Einstellungen	38
Abbildung 15: Screenshot SteamVR Menü	38
Abbildung 16: Screenshot: Unity Hierarchy	39
Abbildung 17: RunTime Scene ScreenShot	40
Abbildung 18: Unity Inspector Screenshot ARKamera	40
Abbildung 19: Geschlechterverteilung	43
Abbildung 20: Berufsverteilung der Probanden	44
Abbildung 21: Screenshot aus der Unity Engine	47
Abbildung 22: Foto eines Probanden beim VR-Video	49
Abbildung 23: VR-Brillennutzung	50
Abbildung 24: Würden Sie Filme lieber in VR oder lieber am Bildschirm schauen? (35 Teilnehmer)	51
Abbildung 25: Lieber am Bildschirm, wie in VR (31 Teilnehmer)	52
Abbildung 26: Szene Versuch 2 – 4	55
Abbildung 27: Startszene Bedienpult mit Würfelmatrix	56
Abbildung 28: Im Hintergrund, die indirekt steuerbaren Würfel	56
Abbildung 29: Haben Sie schon einmal eine Leap Motion genutzt	58
Abbildung 30: Würden Sie den Gebrauch der VR-Brille + Leap Motion anderen VR Anwendungen vorziehen? Ja, weil: (37 Teilnehmer)	59
Abbildung 31: Würden Sie den Gebrauch der VR-Brille + Leap Motion anderen VR Anwendungen vorziehen? Nein, weil. (24 Teilnehmer)	59
Abbildung 32: reale Bedinpultnachbildung	63
Abbildung 33: Würfel für die Vuforia - Erkennung	63
Abbildung 34: Würden Sie den Gebrauch der VR-Brille + Leap Motion + Object Tracking anderen VR-Anwendungen vorziehen? Ja, weil (36 Teilnehmner)	65

Abbildung 35: Würden Sie den Gebrauch der VR-Brille + Leap Motion + Object Tracking anderen VR-Anwendungen vorziehen? Nein, weil. (23 Teilnehmer)	66
Abbildung 36: Application_TargetFrameRate_SteamVR	69
Abbildung 37: Messung der fps-Zahl einer leeren UnitySzene (Mit Anfangsschwankungen)	70
Abbildung 38: Messung der fps-Zahl einer leeren UnitySzene (Ohne Anfangsschwankungen)	71
Abbildung 39: Gemessene gegenüber empfundenen fps und deren Einfluss auf das Spielerlebnis bei 50Fps	75
Abbildung 40: Schwankung der fps bei einer Einstellung	76
Abbildung 41: Gemessene gegenüber empfundenen fps und deren Einfluss auf das Spielerlebnis bei 40fps	78
Abbildung 42: Schwankung der fps bei einer Einstellung	78
Abbildung 43: Gemessene gegenüber empfundenen Fps und deren Einfluss auf das Spielerlebnis bei 30Fps	80
Abbildung 44: Gemessene gegenüber empfundenen Fps und deren Einfluss auf das Spielerlebnis bei 20fps	81
Abbildung 45: Gemessene gegenüber empfundenen fps und deren Einfluss auf das Spielerlebnis bei 10fps	82
Abbildung 46: Hineinversetzen nach Geschlecht Versuch1 - Versuch3	85
Abbildung 47: Vergessen nach Geschlecht Versuch 1 - Versuch 3	86
Abbildung 48: Können Sie sich vorstellen eine VR-Brille zu kaufen? Wenn ja, weil: (Vor den Versuchen)	90
Abbildung 49: Können Sie sich vorstellen eine VR-Brille zu kaufen? Wenn ja, weil: (Nach den Versuchen)	90
Abbildung 50: Können Sie sich vorstellen eine VR-Brille zu kaufen? Wenn nein, weil: (Vor den Versuchen)	91
Abbildung 51: Können Sie sich vorstellen eine VR-Brille zu kaufen? Wenn nein, weil: (Nach den Versuchen)	91

Tabellenverzeichnis

Tabelle 1: Wie haben Sie das gezeigte VR-Video empfunden? S.52

Tabelle 2: Wie empfanden Sie die Anwendung, bei der Sie einen Würfel bewegen und ein virtuelles Bedienfeld nutzen konnten? S. 59

Tabelle 3: Wie empfanden Sie die Anwendung, bei der Sie einen echten Würfel bewegen und ein echtes Bedienfeld benutzen konnten? S. 66

Tabelle 4: Messung der fps Zahl in eine leeren Unity Szene. S.70

Tabelle 5: Schwankung der fps bei einer Einstellung auf 50 fps. S.77

Tabelle 6: Schwankung der fps bei einer Einstellung auf 40 fps. S.79

Tabelle 7: Schwankung der fps bei einer Einstellung auf 30 fps. S. 80

Tabelle 8: Schwankung der fps bei einer Einstellung auf 20 fps. S. 81

Tabelle 9: Schwankung der fps bei einer Einstellung auf 10 fps. S. 82

Tabelle 10: Alter nach Geschlecht. S. 83

Tabelle 11: Versuch 1 nach Geschlecht. S. 84

Tabelle 12: Versuch 2 nach Geschlecht. S.84

Tabelle 13: Versuch 3 nach Geschlecht. S.85

Tabelle 14: Wochenstunden nach Geschlecht. S. 87

Tabelle 15: Mittelwert Versuch 1-3 nach Sehhilfe. S. 87

Tabelle 16: Nutzung VR-Brille nach Versuch 1-3. S. 88

Tabelle 17: Gesamteinfluss nach empfundener Framerate. S. 88

Tabelle 18: Empfundene Latenz und Einfluss nach gemessener Latenz. S. 89

Tabelle 19: Zahlungsbereitschaft für *Leap Motion* und Objekterkennung. S. 89

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelor-Thesis selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die aus fremden Quellen direkt oder indirekt übernommenen Gedanken als solche kenntlich gemacht habe. Die Arbeit habe ich bisher keinem anderen Prüfungsamt in gleicher oder vergleichbarer Form vorgelegt. Sie wurde bisher auch nicht veröffentlicht.

_____ (Ort, Datum) _____ (Unterschrift)

_____ (Ort, Datum) _____ (Unterschrift)

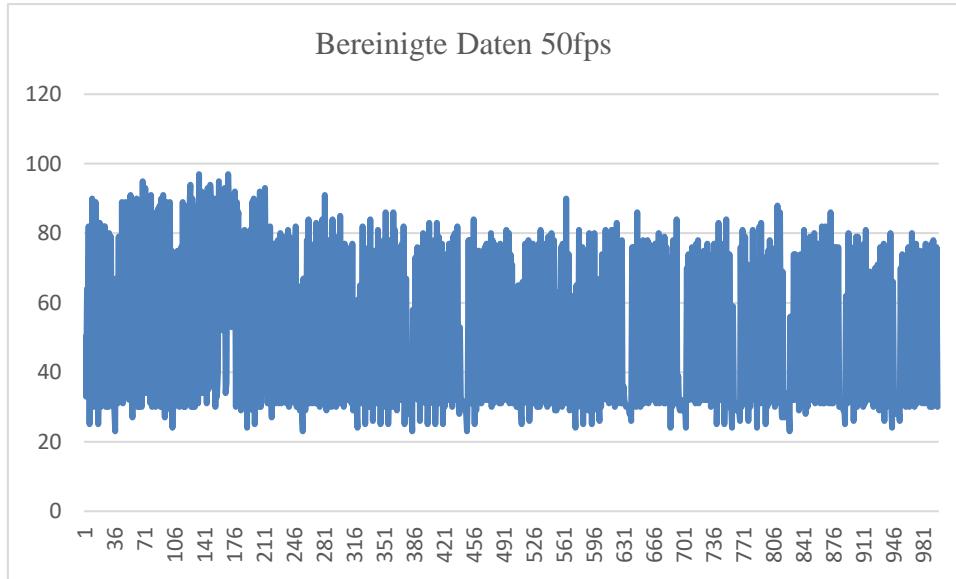
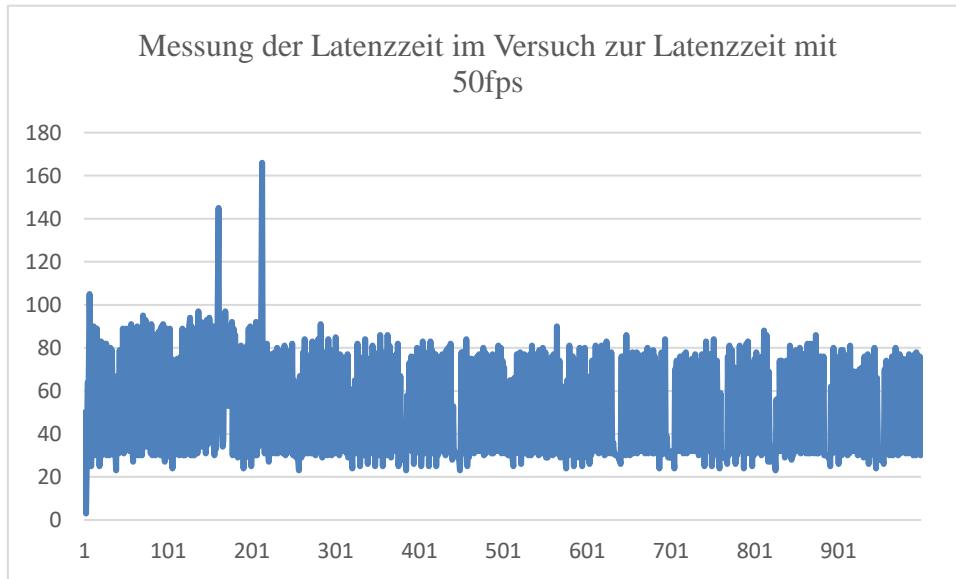
Arbeitsaufteilung

Bei allen Arbeitsschritten waren wir beide beteiligt. Daher ist eine Aufteilung nur soweit möglich, dass bei der schriftlichen Ausarbeitung Markus Weiß seinen Schwerpunkt auf die Kapitel: Kombinieren der Devices, sowie der vier verschiedenen Untersuchungen gelegt hat. Maximilian Kanisch sich auf den Teil der Einleitung, der VR-Systeme und der Voruntersuchung konzentrier. Dabei wurden aber alle Teile in gemeinsamer Absprache verfasst.

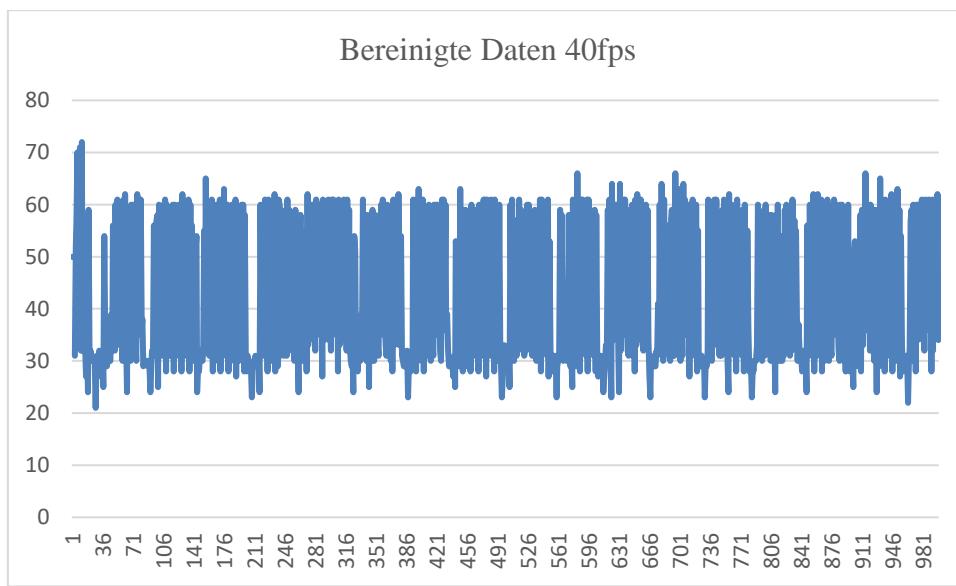
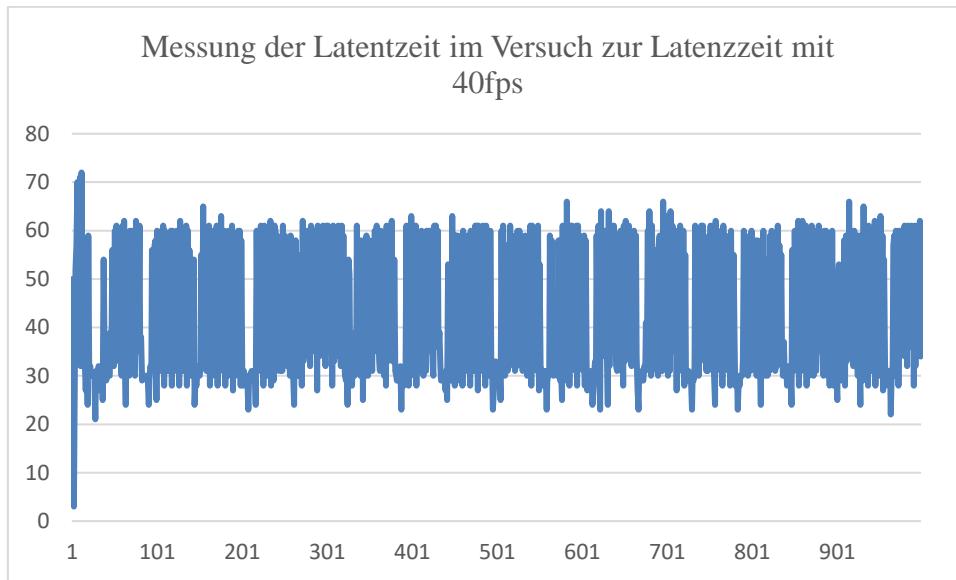
Anhang

Latenzversuch Auswertung

Messung mit 50 Bildern pro Sekunde



Messung mit 40 Bildern pro Sekunde



Integration realer Objekte in den Virtuellen Raum Version 5.0

1. Identifikationsdaten *

Anzahl Teilnehmer: 39

1. Spalte

- | ID | |
|----|------|
| | - 5 |
| | - 6 |
| | - 7 |
| | - 8 |
| | - 9 |
| | - 10 |
| | - 11 |
| | - 12 |
| | - 13 |
| | - 14 |
| | - 15 |
| | - 16 |
| | - 17 |
| | - 18 |
| | - 19 |
| | - 20 |
| | - 21 |
| | - 22 |
| | - 23 |
| | - 24 |
| | - 25 |
| | - 26 |
| | - 27 |
| | - 28 |
| | - 29 |
| | - 30 |
| | - 31 |
| | - 32 |
| | - 33 |
| | - 34 |
| | - 35 |
| | - 36 |
| | - 37 |
| | - 38 |
| | - 39 |
| | - 40 |
| | - 41 |
| | - 42 |
| | - 43 |

2. Alter? *

Anzahl Teilnehmer: 39

 Alle 4 vorangegangenen Antworten anzeigen

- 26
- 39
- 26
- 18
- 27
- 27
- 24
- 24
- 22
- 20
- 19
- 22
- 22
- 24
- 25
- 17
- 27
- 25
- 56
- 22
- 24
- 22
- 53
- 21
- 25
- 22
- 60
- 58
- 22
- 16
- 22
- 19
- 20
- 24
- 22

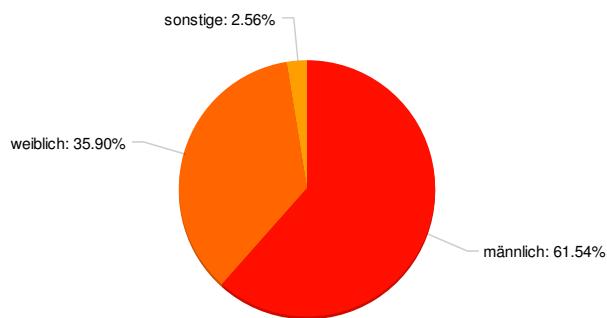
3. Geschlecht *

Anzahl Teilnehmer: 39

24 (61.5%): männlich

14 (35.9%): weiblich

1 (2.6%): sonstige

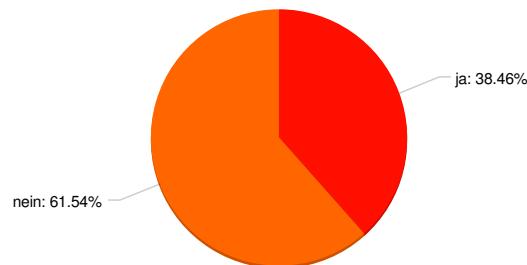


4. Sind Sie Brillenträger *

Anzahl Teilnehmer: 39

15 (38.5%): ja

24 (61.5%): nein



5. Beruf *

Anzahl Teilnehmer: 39

22 (56.4%): Arbeitnehmer

8 (20.5%): Student

1 (2.6%): Schüler

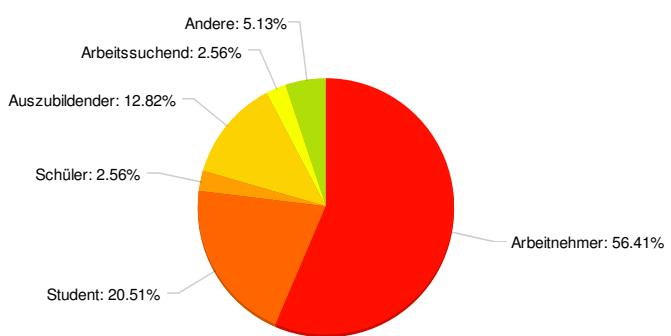
5 (12.8%): Auszubildender

1 (2.6%): Arbeitssuchend

2 (5.1%): Andere

Antwort(en) aus dem
Zusatzfeld:

- Selbstständig
- Selbstständig



6. Wie oft nutzen Sie Interaktive Anwendungen bzw. Computerspiele auf Desktop-PCs, Handys, Tablets, etc. im Alltag *

Anzahl Teilnehmer: 39

 Alle 4 vorangegangenen Antworten anzeigen

- 1
- 6
- 0
- 1
- 0
- 1
- 0
- 3
- 4
- 2
- 2
- 5
- 1
- 1
- 7
- 0
- 0
- 0
- 0
- 2
- 0
- 1
- 0
- 3
- 1
- 3
- 0
- 0
- 1
- 0
- 3
- 2
- 4
- 1
- 1

7. Wie oft nutzen Sie Interaktive Anwendungen bzw. Computerspiele auf Desktop-PCs, Handys, Tablets, etc. im Alltag *

Anzahl Teilnehmer: 39

 Alle 4 vorangegangenen Antworten anzeigen

- 4
- 42
- 1
- 7
- 0
- 7
- 0
- 21
- 32
- 20
- 14
- 40
- 8
- 7
- 70
- 1
- 0
- 0
- 0
- 20
- 0
- 20
- 5
- 20
- 10
- 23
- 0
- 0
- 7
- 1
- 35
- 20
- 30
- 7
- 7

8. Besitzen Sie einen VR-Fähigen Rechner? *

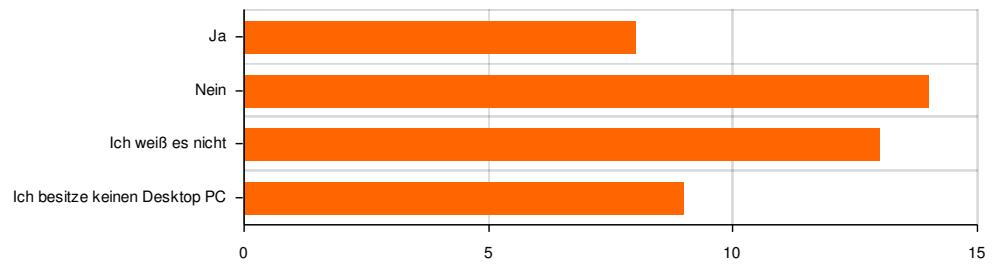
Anzahl Teilnehmer: 39

8 (20.5%): Ja

14 (35.9%): Nein

13 (33.3%): Ich weiß es nicht

9 (23.1%): Ich besitze keinen Desktop PC

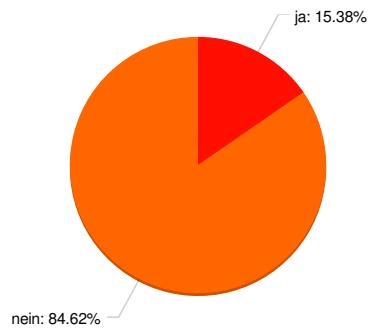


9. Kennen Sie das Inputdevice LeapMotion? *

Anzahl Teilnehmer: 39

6 (15.4%): ja

33 (84.6%): nein

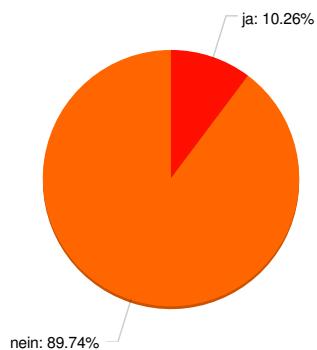


10. Haben Sie schon einmal eine Leap Motion benutzt? *

Anzahl Teilnehmer: 39

4 (10.3%): ja

35 (89.7%): nein

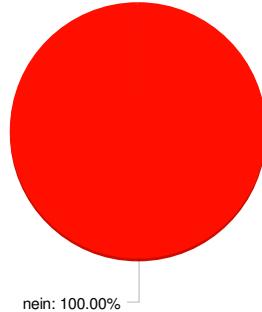


11. Besitzen Sie eine LeapMotion? *

Anzahl Teilnehmer: 39

- (0.0%): ja

39 (100.0%): nein

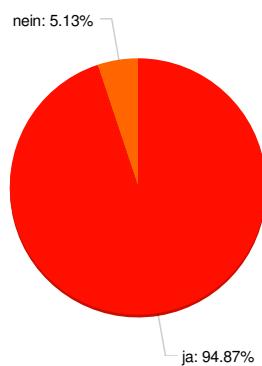


12. Wissen Sie was eine VR-Brille ist? *

Anzahl Teilnehmer: 39

37 (94.9%): ja

2 (5.1%): nein

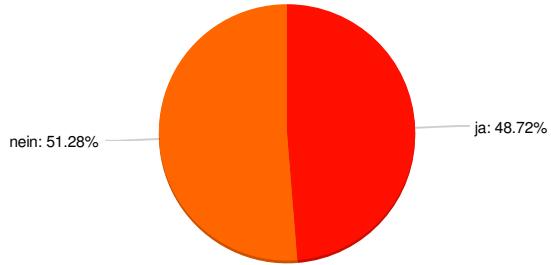


13. Haben Sie schon einmal eine VR Brille (Oculus, HTC, PSVR etc.) genutzt? *

Anzahl Teilnehmer: 39

19 (48.7%): ja

20 (51.3%): nein

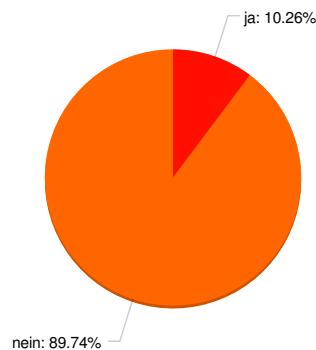


14. Besitzen Sie eine VR-Brille? *

Anzahl Teilnehmer: 39

4 (10.3%): ja

35 (89.7%): nein



15. Können Sie sich vorstellen eine VR-Brille zu kaufen?

Anzahl Teilnehmer: 31

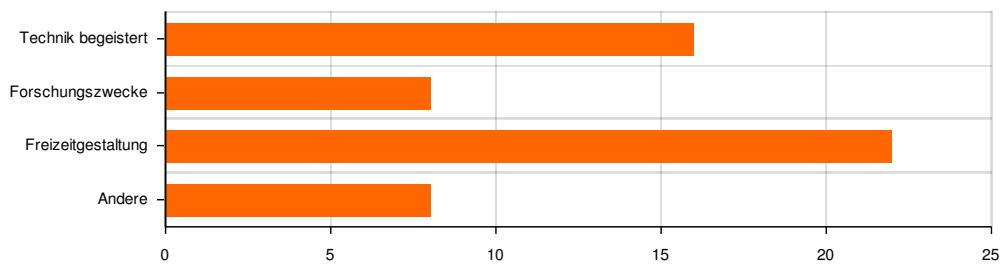
16 (51.6%): Technik
begeistert

8 (25.8%):
Forschungszwecke

22 (71.0%):
Freizeitgestaltung

- (0.0%): Prestigegründe

8 (25.8%): Andere



Antwort(en) aus dem
Zusatzfeld:

- hab eine
- Verbesserung in der Interaktion zwischen Computer und Mensch
- Weil ich keinen Fernseher mehr brauche
- Zukunftsorientierung
- Weil ich es cool finde
- Vielleicht keine Ahnung
- Ich habe noch keine genutzt
- Wir haben eine bei der Arbeit daher muss ich mir keine kaufen

16. Können Sie sich vorstellen eine VR-Brille zu kaufen?

Anzahl Teilnehmer: 34

8 (23.5%): Unausgereifte
Technik

22 (64.7%): Hoher Preis

15 (44.1%): Fehlender
Nutzen / Anwendungen

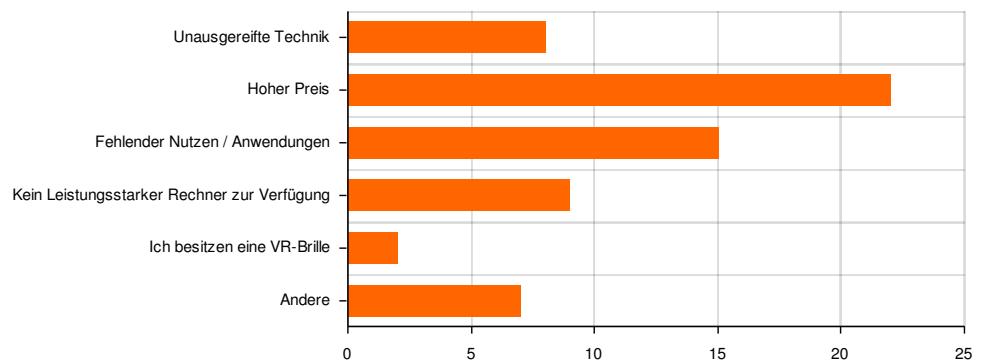
9 (26.5%): Kein
Leistungsstarker Rechner
zur Verfügung

2 (5.9%): Ich besitzen eine
VR-Brille

7 (20.6%): Andere

Antwort(en) aus dem
Zusatzfeld:

- Der nutzen ist noch 'nicht groß genug
- Aus Angst wenn etwas passiert (das ich in Horrorspielen erschrecke)
- Weil ich das Momentane wie Maus und Tatstatur gewohnt bin
- Kein Nutzen weil ich daran kein Interesse hab
- Kein Nutzen weil ich daran kein Interesse hab
- Übelkeitsgefühl
- Kein Verlangen danach in VR zu verschwinden



17. Haben Sie grundsätzlich Interesse an VR-Anwendungen wie VR-Computerspielen, Filmen etc.?

Anzahl Teilnehmer: 38

24 (63.2%): Weil man 3D-Grafik sieht

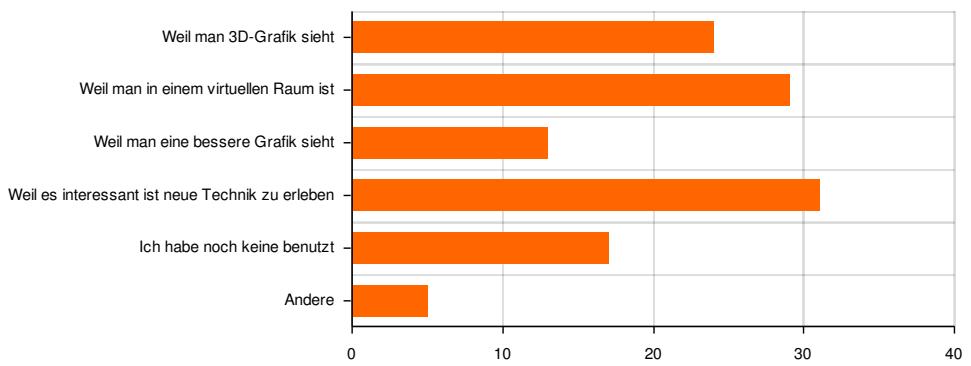
29 (76.3%): Weil man in einem virtuellen Raum ist

13 (34.2%): Weil man eine bessere Grafik sieht

31 (81.6%): Weil es interessant ist neue Technik zu erleben

17 (44.7%): Ich habe noch keine benutzt

5 (13.2%): Andere



Antwort(en) aus dem Zusatzfeld:

- Immersion
- Um die Spielerfahrung zu verbessern / das Gesamtkonzept muss stimmen um meine Gewohnheiten zu ändern
- Neugier
- Forschungszwecke
- VR-Pornos

18. Haben Sie grundsätzlich Interesse an VR-Anwendungen wie VR-Computerspielen, Filmen etc.?

Anzahl Teilnehmer: 28

4 (14.3%): Schlechte Grafik

4 (14.3%): Unbequemer Tragekomfort

10 (35.7%): Übelkeitsgefühl

5 (17.9%): Eingeschränkte Bewegungsfreiheit

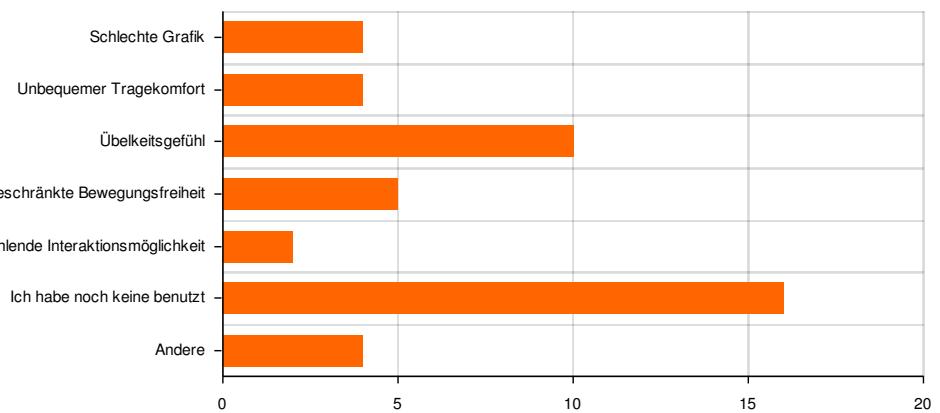
2 (7.1%): Fehlende Interaktionsmöglichkeit

16 (57.1%): Ich habe noch keine benutzt

4 (14.3%): Andere

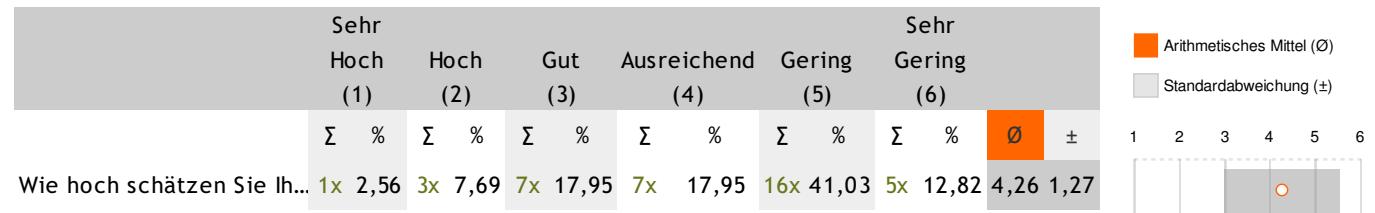
Antwort(en) aus dem Zusatzfeld:

- habe interesse daran
- Kabel / keine Hände
- Nicht genügend Platz zuhause
- Das man um sich herum nichts mehr Mitbekommt



19. Wie hoch schätzen Sie Ihr Wissen um die Virtuelle Realität(VR) ein? *

Anzahl Teilnehmer: 39



20. Wenn Sie schon einmal eine VR-Brille benutzt haben, in welchem Zusammenhang haben Sie diese benutzt? *

Anzahl Teilnehmer: 39

8 (20.5%): Messen
(Ausstellungen, Gamescom,
etc.)

12 (30.8%): Privat bei
Freunden

4 (10.3%): Ich besitze eine
eigene

4 (10.3%): Beruflich

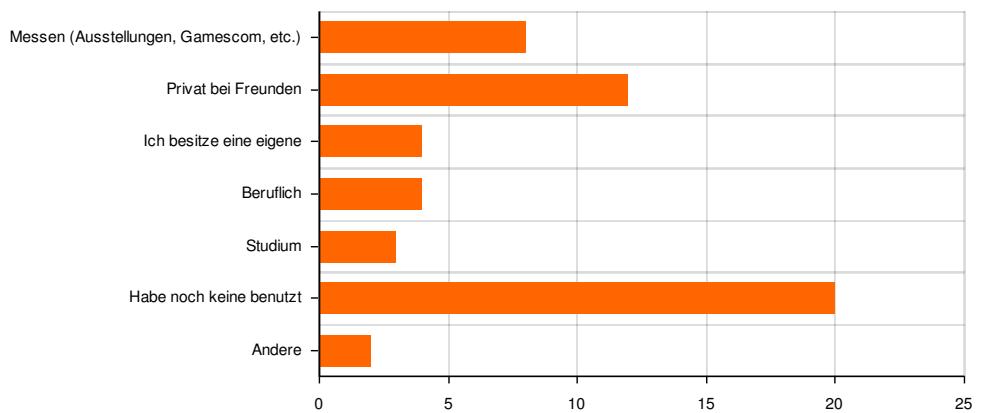
3 (7.7%): Studium

20 (51.3%): Habe noch keine
benutzt

2 (5.1%): Andere

Antwort(en) aus dem
Zusatzfeld:

- Europapark
- Europapark



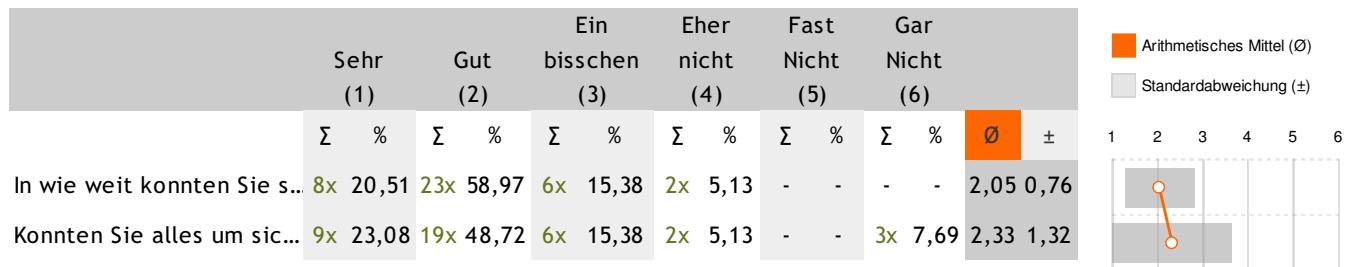
21. Wie viele Stunden haben Sie insgesamt eine VR-Brille genutzt? *

Anzahl Teilnehmer: 39

Alle 4 vorangegangenen Antworten anzeigen

22. Wie haben Sie das gezeigte VR-Video empfunden? *

Anzahl Teilnehmer: 39



23. Würden Sie Filme lieber in VR oder lieber am Bildschirm anschauen?

Anzahl Teilnehmer: 35

30 (85.7%): Weil man 3D-Grafik sieht

30 (85.7%): Weil man in einem Virtuellen Raum ist

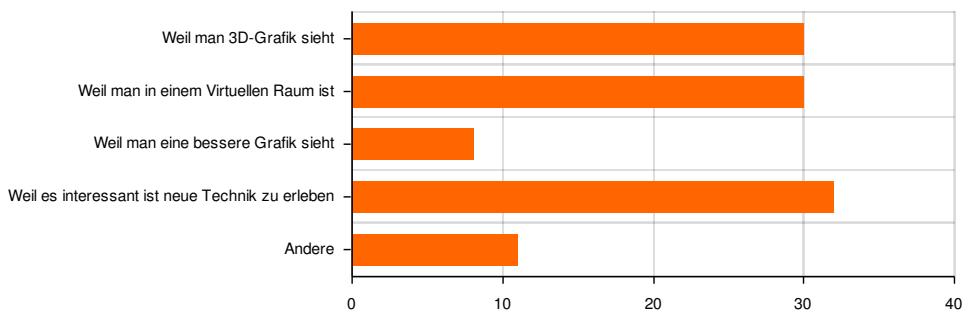
8 (22.9%): Weil man eine bessere Grafik sieht

32 (91.4%): Weil es interessant ist neue Technik zu erleben

11 (31.4%): Andere

Antwort(en) aus dem Zusatzfeld:

- Wenn es entsprechende Filme mit der passenden Technologie gibt
- Weil ich das das gesamte Geschehen beobachten kann, also auch neben dran/ komplett anderes gefühl wie am bildschirm
- Man kann sich voll auf den Film konzentrieren / wenn es spezielle VR-Filme gibt/ Für solid works anwendungen wäre es super
- Man kann den Blickwinkel selbst bestimmen
- Weil man nicht abgelenkt wird von anderen sachen
- Interaktivität weil man hinschauen kann wo man will
- Rundumsicht
- Weil es mir das Gefühl gibt mittendrin zu sein
- Hohe Immersion / man fühlt sich wie ein Cyborg / es ist technik die die Sinne erweitert und die Sinnwahrnehmung verändert / Keine Störfaktoren
- Beim Film ist besseres eintauchen möglich, intensivberes erlebnis (Immersion)
- Ein VR-Film ist packender



24. Würden Sie Filme lieber in VR oder lieber am Bildschirm anschauen?

Anzahl Teilnehmer: 31

14 (45.2%): Schlechte Grafik

14 (45.2%): Unbequemer
Tragekomfort

4 (12.9%): Übelkeitsgefühl

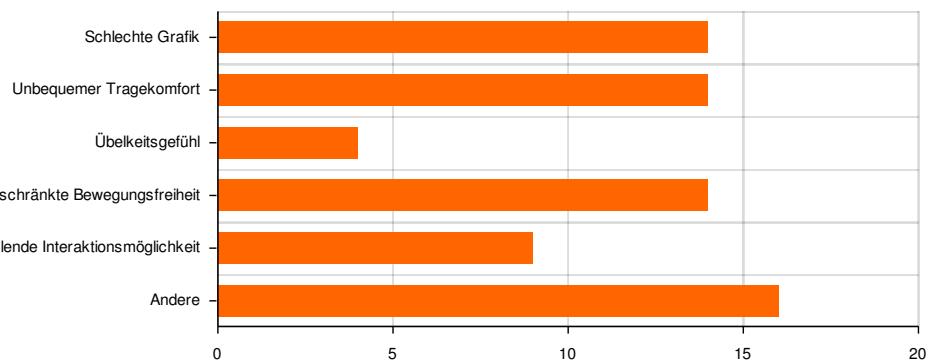
14 (45.2%): Eingeschränkte
Bewegungsfreiheit

9 (29.0%): Fehlende
Interaktionsmöglichkeit

16 (51.6%): Andere

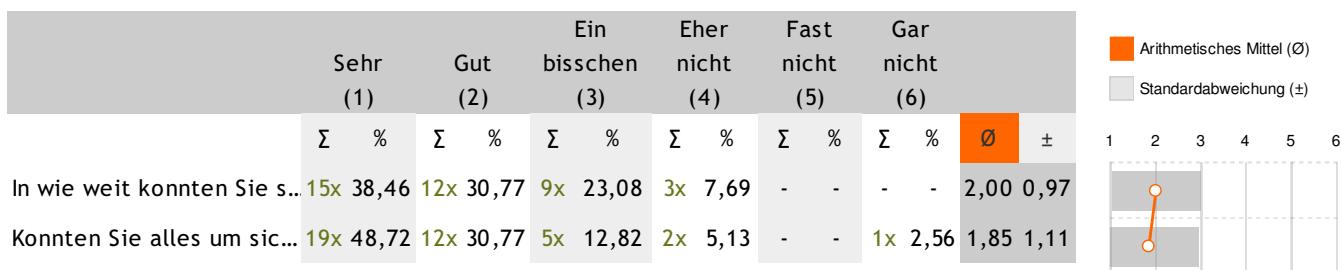
Antwort(en) aus dem
Zusatzfeld:

- Wenn ich die Handlung hinter mir verpasse
- Unausgereifte Technik
- Lieber zocken als Filme schauen
- Kabel beschränkung
- Gewohnheit am Fernseher
- Eingeschränktes Sichtfeld
- Ein Virtueller Körper hätte die Immersion erhöht
- Wenn ich die Handlung hinter mir verpasse
- Kabel beschränkung
- Streßig weil man dem Geschehen hinterher sein muss / man könnte was verpassen
- Kabel beschränkung
- In der Gruppe Film schauten mit der VR mindert die Kommunikations untereinander
- Kabel / das man sich darin verlieren könnte
- Ich könnte keine Horrofilme schauen
- Im VR-Film was trinken nicht möglich
- Kabel beschränkung



25. Wie empfanden Sie die Anwendung, bei der Sie einen virtuellen Würfel bewegen und ein virtuelles Bedienfeld nutzen konnten? *

Anzahl Teilnehmer: 39



26. Würden Sie den Gebrauch der VR-Brille + LeapMotion(die Handerkennung), anderen VR Anwendungen vorziehen?

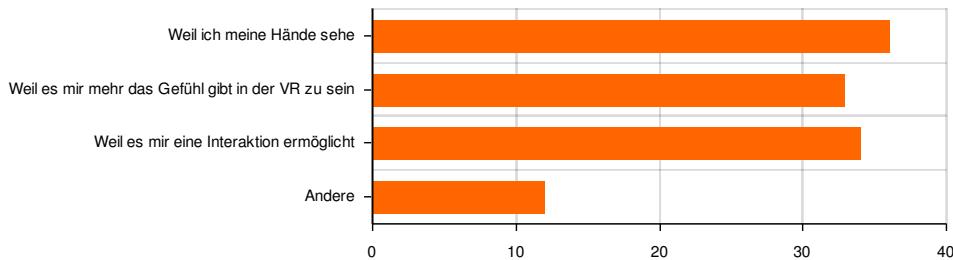
Anzahl Teilnehmer: 37

36 (97.3%): Weil ich meine Hände sehe

33 (89.2%): Weil es mir mehr das Gefühl gibt in der VR zu sein

34 (91.9%): Weil es mir eine Interaktion ermöglicht

12 (32.4%): Andere



Antwort(en) aus dem Zusatzfeld:

- höhere Immersion weil du deine echten Hände siehst
- Hatte die Interaktion in anderen Spielen weniger
- Eine Hand echt, Eine Hand Kontroller, physisches HUD
- Ich hab andere noch nicht probiert
- Ausschließlich Hände würden reichen
- Virtuelle Gegenstände sind zu Sachen fähig, die Reale nicht sind
- Handsteuerung ist Intuitiver
- Weil man mit den Händen mehr Möglichkeiten hat
- Es gab mir eine bessere Kontrolle
- Ein sehr gutes Gefühl
- Es ist eine komplett andere Welt
- Ist witzig

27. Würden Sie den Gebrauch der VR-Brille + LeapMotion(die Handerkennung), anderen VR Anwendungen vorziehen?

Anzahl Teilnehmer: 24

10 (41.7%): Mir fehlt echte

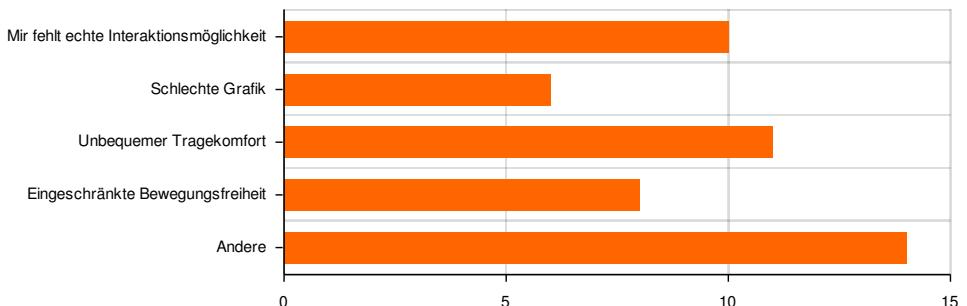
Interaktionsmöglichkeit

6 (25.0%): Schlechte Grafik

11 (45.8%): Unbequemer

Tragekomfort

- (0.0%): Übelkeitsgefühl



8 (33.3%): Eingeschränkte Bewegungsfreiheit

14 (58.3%): Andere

Antwort(en) aus dem

Zusatzfeld:

- Unausgereifte Technik

- Weil ich alle Interessant finde

- Unausgereifte Technik

- Unausgereifte Technik ; weil man kein haptisches Feedback hat

- Kabel

- Unausgereifte Technik

- Unausgereifte Technik / Kabel

- Unausgereifte Technik / Das man die Brille halten musste

- Kabel

- Kabel

- Bessere Erkennung der Hände, besseres Ansprechen. Interaktion mit Vvrtuellem würfel war nicht sehr realistisch

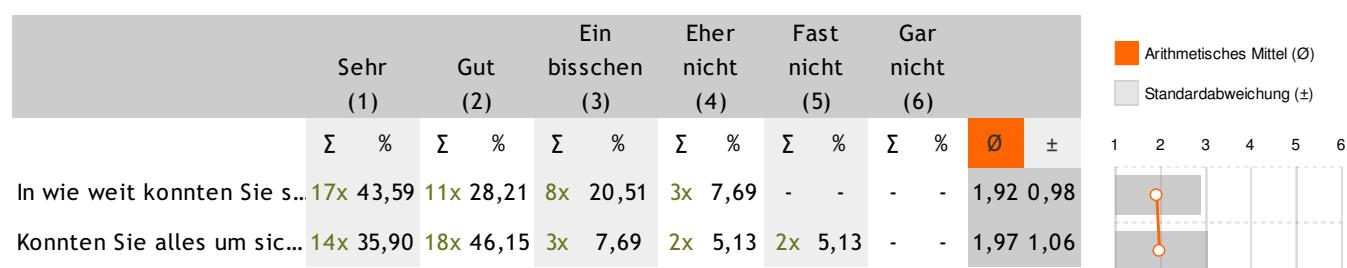
- Man sieht die Beine Nicht, man sollte den ganzen Körper sehen

- Kabel

- Kabel

28. Wie empfanden Sie die Anwendung, bei der Sie einen echten Würfel bewegen und ein echtes Bedienfeld bewegen konnten? *

Anzahl Teilnehmer: 39



29. Würden Sie den Gebrauch der VR-Brille + Leap Motion(Handerkennung) + ObjectTracking(echter Würfel, echtes Bedienfeld), anderen VR Anwendungen vorziehen?

Anzahl Teilnehmer: 36

21 (58.3%): Weil man 3D-Grafik sieht

27 (75.0%): Weil man in einem virtuellen Raum ist

10 (27.8%): Weil man eine bessere Grafik sieht

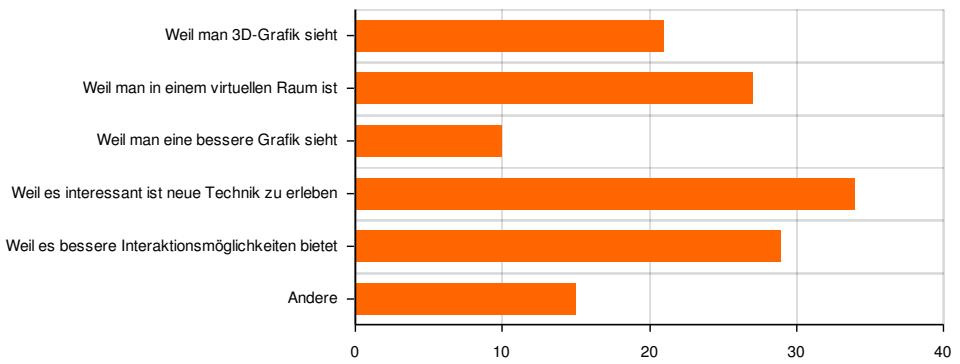
34 (94.4%): Weil es interessant ist neue Technik zu erleben

29 (80.6%): Weil es bessere Interaktionsmöglichkeiten bietet

15 (41.7%): Andere

Antwort(en) aus dem Zusatzfeld:

- Haptisches Feedback
- weil man die Haptik hat, einen Teil der die Immersion verstärkt
- Haptisches Feedback, Immersion
- Wenn die Technik mit den realen Gegenständen ausgereift ist, würde ich es dem Virtuellen vorziehen
- Höhere Immersion durch haptisches Feedback
- Man hat keine störenden Controller in der Hand
- Anwendungsgabhängig
- weil man die Haptik hat, einen Teil der die Immersion verstärkt
- Ich würde eine Mix aus real und virtuelle Vorziehen / was echtes das universell genutzt werden kann / es fühlt sich echter an
- Wenn man bestimmte Geräte und Bedienfelder nutzen könnte die es in echt auch gibt
- Haptisches Feedback
- Es macht das Erlebnis realer
- Haptisches Feedback sorgt Immersion
- Haptisches Feedback
- Man kann interagieren



30. Würden Sie den Gebrauch der VR-Brille + Leap Motion + ObjectTracking, anderen VR Anwendungen vorziehen?

Anzahl Teilnehmer: 23

6 (26.1%): Schlechte Grafik

11 (47.8%): Unbequemer Tragekomfort

2 (8.7%): Übelkeitsgefühl

14 (60.9%): Eingeschränkte Bewegungsfreiheit

3 (13.0%): Fehlende Interaktionsmöglichkeit

15 (65.2%): Andere

Antwort(en) aus dem Zusatzfeld:

- ja

- Mögliche Risiken, und Verletzungsgefahren, weil man nicht einschätzen kann wo sie wirklich stehen

- Je nach dem was man macht kann man es nicht bewegen weil das Pult zum Beispiel bewegen muss

- In dem vorgegebenen Feld keine Bewegungseinschränkung / Ein Menü für anderen Geräte zum Auswählen zum Beispiel Stereoanlage anschalten

- Weil mit realen Würfeln nicht das Virtuelle möglich ist / Technisch unausgereift

- Controller sind Präziser

- Diese Dinge würde nur ich benutzen wollen, es würde mich stören wenn sie andere benutzen

- Kabel, HMD ist schwer, es drückt nach unten
- Nicht ausgereifte Technik
- Wenn man sich an echten Gegenständen möglicherweise verletzen könnte

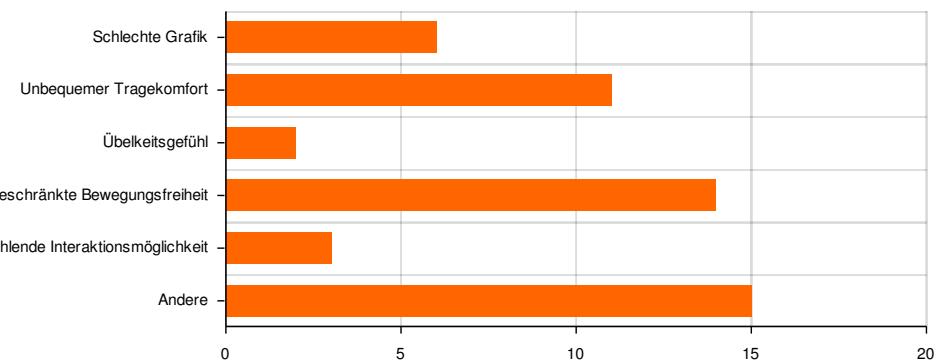
- Wegen des Würfels eingeschränkte Bewegungsfreiheit

- Kabel

- Kabel, HMD ist schwer, es drückt nach unten

- Man kann den echten Raum nicht wahrnehmen bei Gefahr

- Kabel



31. Was glauben Sie wie viel FPS. das waren? Eine Zahl zwischen 0 - 100 Fps *

Anzahl Teilnehmer: 39

	Gemessene Latenz	Empfundene Latenz
Versuch 5	- 0	- 0
	- 0	- 0
	- 50	- 60
	- 40	- 40
	- 50	- 60
	- 40	- 40
	- 10	- 10
	- 10	- 10
	- 10	- 10
	- 40	- 40
	- 10	- 10
	- 10	- 20
	- 40	- 30
	- 10	- 10
	- 40	- 40
	- 20	- 10
	- 20	- 20
	- 10	- 10
	- 20	- 20
	- 10	- 10
	- 10	- 20
	- 10	- 10
	- 10	- 10
	- 10	- 10
	- 20	- 40
	- 20	- 30
	- 10	- 20
	- 10	- 50
	- 30	- 40
	- 10	- 30
	- 10	- 10
	- 30	- 30
	- 30	- 50
	- 30	- 30
	- 20	- 40
Versuch 4	- 40	- 40
	- 30	- 90
	- 10	- 10
	- 10	- 10
	- 10	- 10
	- 10	- 10
	- 20	- 20
	- 20	- 30
	- 20	- 20
	- 20	- 10
	- 10	- 10
	- 15	- 15
	- 20	- 50
	- 20	- 20
	- 20	- 20
	- 20	- 10
	- 20	- 20
	- 10	- 10
	- 50	- 30
	- 20	- 10
	- 10	- 10
	- 20	- 30
	- 20	- 50
	- 20	- 20
	- 15	- 10
	- 20	- 50
	- 20	- 20
	- 20	- 40
	- 30	- 50
	- 10	- 20
	- 20	- 20
	- 20	- 50
	- 10	- 10

	- 20	- 50
	- 20	- 10
	- 10	- 10
	- 10	- 10
	- 10	- 10
	- 10	- 10
Versuch 3	- 10	- 10
	- 10	- 20
	- 40	- 0
	- 20	- 20
	- 20	- 20
	- 20	- 20
	- 30	- 50
	- 40	- 40
	- 40	- 10
	- 40	- 20
	- 20	- 10
	- 30	- 30
	- 50	- 30
	- 40	- 50
	- 10	- 10
	- 40	- 40
	- 10	- 10
	- 50	- 40
	- 10	- 10
	- 40	- 40
	- 40	- 20
	- 40	- 20
	- 40	- 40
	- 40	- 50
	- 50	- 40
	- 40	- 10
	- 50	- 30
	- 30	- 30
	- 30	- 30
	- 30	- 50
	- 20	- 10
	- 40	- 20
	- 20	- 10
	- 20	- 20
	- 20	- 20
Versuch 2	- 20	- 20
	- 20	- 40
	- 20	- 35
	- 30	- 30
	- 30	- 30
	- 50	- 15
	- 40	- 30
	- 30	- 30
	- 50	- 20
	- 30	- 15
	- 30	- 10
	- 40	- 40
	- 30	- 30
	- 30	- 30
	- 30	- 40
	- 30	- 30
	- 30	- 20
	- 50	- 30
	- 30	- 40
	- 30	- 40
	- 30	- 30
	- 50	- 50
	- 40	- 10
	- 0	- 0
	- 40	- 10
	- 30	- 30
	- 40	- 20

- 40	- 20
- 40	- 10
- 50	- 50
- 50	- 40
- 40	- 40
- 40	- 20
- 30	- 30
- 30	- 20
- 40	- 30
- 40	- 20
- 30	- 10
- 40	- 50
Versuch 1	
- 30	- 90
- 50	- 40
- 30	- 20
- 50	- ist vorgegeben
- 40	- 40
- 30	- 20
- 50	- 40
- 50	- 20
- 30	- 30
- 50	- 20
- 40	- 20
- 50	- 30
- 50	- 20
- 50	- 20
- 50	- 30
- 50	- 20
- 50	- 30
- 50	- 30
- 50	- 10
- 50	- 40
- 30	- 20
- 50	- 50
- 50	- 20
- 50	- 20
- 30	- 30
- 50	- 20
- 50	- 30
- 50	- 10
- 50	- 10
- 50	- 10
- 50	- 40
- 50	- 30
- 50	- 20
- 40	- 40
- 40	- 10
- 50	- 40
- 50	- 30
- 50	- 20
- 50	- 50
- 50	- 40
- 50	- 10
- 50	- 10
- 50	- 10

32. Wie sehr hat diese Framerate Ihr Spielerlebnis beeinflusst? *

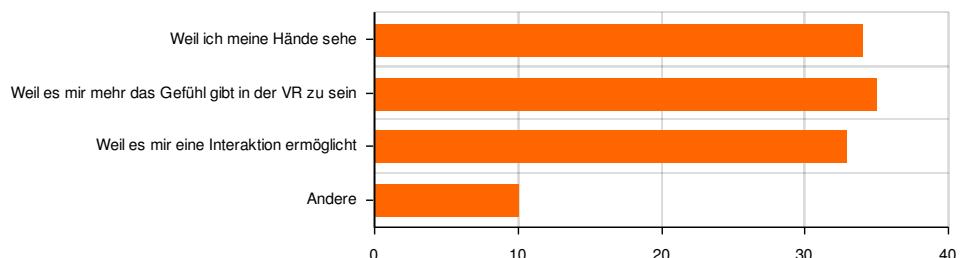
Anzahl Teilnehmer: 39

	Sehr Stark		Stark		Ein Bisschen		Gar Nicht	
	Σ	%	Σ	%	Σ	%	Σ	%
Versuch 1	5x	12,82	4x	10,26	12x	30,77	18x	46,15
Versuch 2	2x	5,13	7x	17,95	17x	43,59	13x	33,33
Versuch 3	7x	17,95	8x	20,51	14x	35,90	12x	30,77
Versuch 4	12x	30,77	10x	25,64	8x	20,51	9x	23,08
Versuch 5	10x	25,64	6x	15,38	11x	28,21	12x	30,77

33. Würden Sie sich eine feste Integration der LeapMotion(Handerkennung) in VR Anwendungen wie Computerspielen wünschen?

Anzahl Teilnehmer: 37

34 (91.9%): Weil ich meine Hände sehe



35 (94.6%): Weil es mir mehr das Gefühl gibt in der VR zu sein

33 (89.2%): Weil es mir eine Interaktion ermöglicht

10 (27.0%): Andere

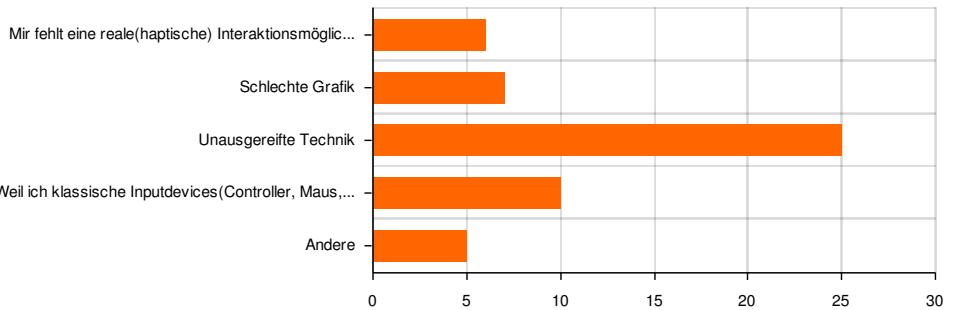
Antwort(en) aus dem Zusatzfeld:

- Die Leapmotion soll abnehmbar sein
- Ist etwas anderes wenn man die Hände sieht, Hände sind vertrauter / Mit der Leap hatte ich das Gefühl richtig drin zu sein
- Wenn ich einen echten Würfel anfassen muss sind echte Hände besser
- Ist doof wenn ich meine Hände nicht sehe, ist nicht so Immersiv
- Super
- Es macht es einfacher es einzuschätzen.
(Größenverhältnisse)
- Immersive
- mehr Realismus
- Es ist realer wenn ich meine Hände sehe
- Ist cool

34. Würden Sie sich eine feste Integration der LeapMotion(Handerkennung) in VR Anwendungen wie Computerspielen wünschen?

Anzahl Teilnehmer: 32

6 (18.8%): Mir fehlt eine reale(haptische) Interaktionsmöglichkeit



7 (21.9%): Schlechte Grafik

25 (78.1%): Uausgereifte Technik

10 (31.3%): Weil ich klassische Inputdevices(Controller, Maus, Tastatur) gewohnt bin

5 (15.6%): Andere

Antwort(en) aus dem Zusatzfeld:

- ja
- Vivecontroller sind besser
- Weil das Leap Tracking unausgereift ist
- Reale Hände
- Verbesserungsfähig

35. Würden Sie sich eine feste Integration von realen Objekten(Würfel, Bedienfeld) in die VR wünschen?

Anzahl Teilnehmer: 38

27 (71.1%): Weil man ein reales(haptisches) Feedback bekommt

31 (81.6%): Weil es interessant ist, neue Technik zu erleben

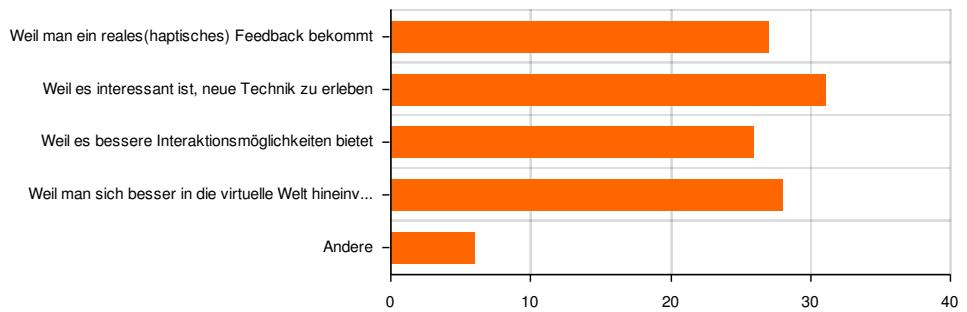
26 (68.4%): Weil es bessere Interaktionsmöglichkeiten bietet

28 (73.7%): Weil man sich besser in die virtuelle Welt hineinversetzen kann

6 (15.8%): Andere

Antwort(en) aus dem Zusatzfeld:

- Wenn die Technik besser funktioniert
- Wenn man reale Sachen in die virtuelle Welt besser integriert bsp. Risiko echt auf dem Brett und Animationen um einen herum in VR
- Wenn die Technik besser funktioniert
- Mehr Realismus durch haptisches Feedback
- 360° Sicht / Das Ruckeln hat gestört /
- Würfel hat nicht mit VR-Würfel übereingestimmt



36. Würden Sie sich eine feste Integration von Realen Objekten(Würfel, Bedienfeld) in die VR wünschen?

Anzahl Teilnehmer: 30

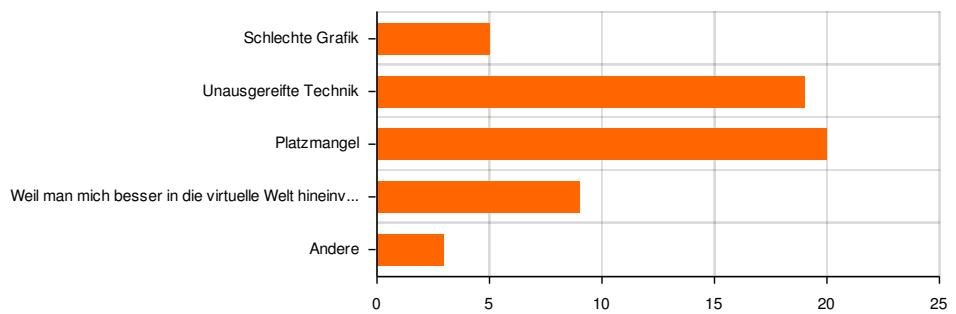
5 (16.7%): Schlechte Grafik

19 (63.3%): Unausgereifte
Technik

20 (66.7%): Platzmangel

9 (30.0%): Weil man mich
besser in die virtuelle Welt
hineinversetzen kann

3 (10.0%): Andere



Antwort(en) aus dem

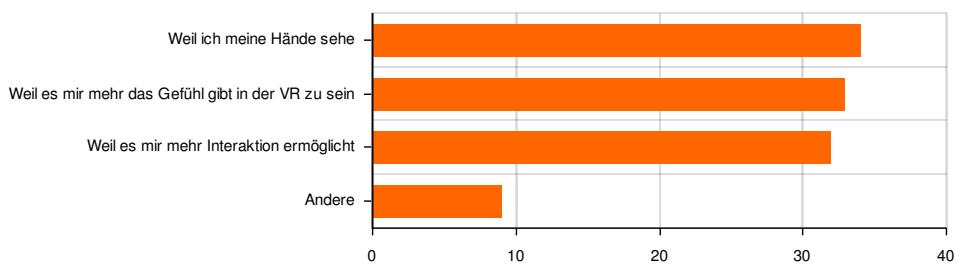
Zusatzfeld:

- ja
- Weil es den Spielfluss stört wenn man an echte Sachen rankommt / weil das Tracking noch zu ungenau ist
- Reales Feedback schränkt in den Möglichkeiten ein

37. Würden Sie mehr interaktive Anwendungen wie Computerspiele mit der VR-Brille Nutzen wenn eine LeapMotion(Handerkennung) und das Objecttracking(Würfelerkennung), wie auch reale Gegenstände im Raum integriert wären?

Anzahl Teilnehmer: 38

34 (89.5%): Weil ich meine Hände sehe



33 (86.8%): Weil es mir mehr das Gefühl gibt in der VR zu sein

32 (84.2%): Weil es mir mehr Interaktion ermöglicht

9 (23.7%): Andere

Antwort(en) aus dem Zusatzfeld:

- Weil ich dann keine Angst haben muss etwas im Raum kaputt zu machen (am besten alle Objekte integriert)
- Man kann mehr empfinden, Stifness, macht es immersiver
- Höherer Funfaktor
- Als Kombination von Händen und Controllern
- Es fühlt sich echter an
- Sicherheitsgefühl
- Es erzeugt eine höhere Immersion
- Mit den Händen ist eine bessere Verschmelzung mit der VR möglich, wenn ich meine Hände sehe und damit reale Sachen anfassen kann kann ich besser eintauchen
- Beides zu kennen ist sehr interessant Real u. Virtuell um einen Vergleich zu haben

38. Würden Sie mehr interaktive Anwendungen wie Computerspiele mit der VR-Brille nutzen, wenn eine LeapMotion(Handerkennung) und das ObjectTracking(Würfelerkennung), wie auch reale Gegenstände im Raum integriert wären

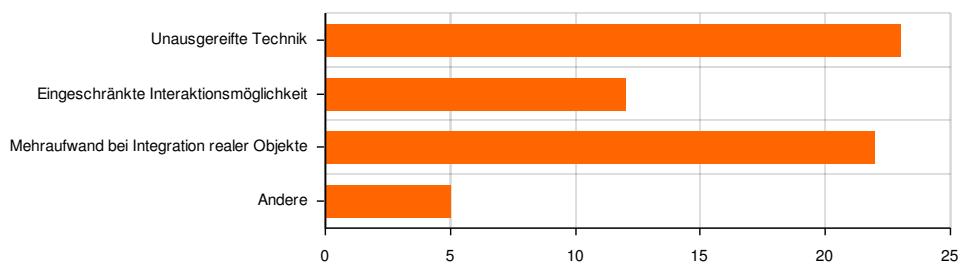
Anzahl Teilnehmer: 27

23 (85.2%): Unausgereifte Technik

12 (44.4%): Eingeschränkte Interaktionsmöglichkeit

22 (81.5%): Mehraufwand bei Integration realer Objekte

5 (18.5%): Andere



Antwort(en) aus dem Zusatzfeld:

- ..
- Mich würde es stören, wenn diese Inputdevices nur zur Monetarisierung verwendet werden
- Ohne Kabel, Gewicht der Brille
- Man weis das es echt ist deswegen nimmt es die Immersion
- Beides zu kennen ist sehr interessant Real u. Virtuell um einen vergleich zu haben

39. Können Sie nach dem Verwenden dieser Anwendungen sich vorstellen, eine VR-Brille zu kaufen?

Anzahl Teilnehmer: 31

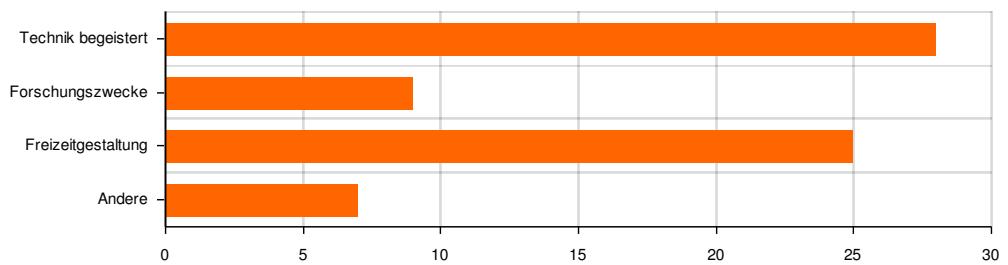
28 (90.3%): Technik
begeistert

9 (29.0%):
Forschungszwecke

25 (80.6%):
Freizeitgestaltung

- (0.0%): Prestigegründe

7 (22.6%): Andere



Antwort(en) aus dem
Zusatzfeld:

- Ich besitze eine
- VR-Porn
- Virtueller Chat/ Virtuelle Telephonie
- Intresse an dem kommenden Anwendungen
- Weil ich Teilhaben möchte an der Entwicklung dieser Technik
- Es macht mega Spaß
- Zukunftsorientierung

40. Können Sie nach dem Verwenden dieser Anwendungen sich vorstellen, eine VR-Brille zu kaufen?

Anzahl Teilnehmer: 35

18 (51.4%): Unausgereifte
Technik

31 (88.6%): Hoher Preis

15 (42.9%): Fehlender
Nutzen / Anwendungen

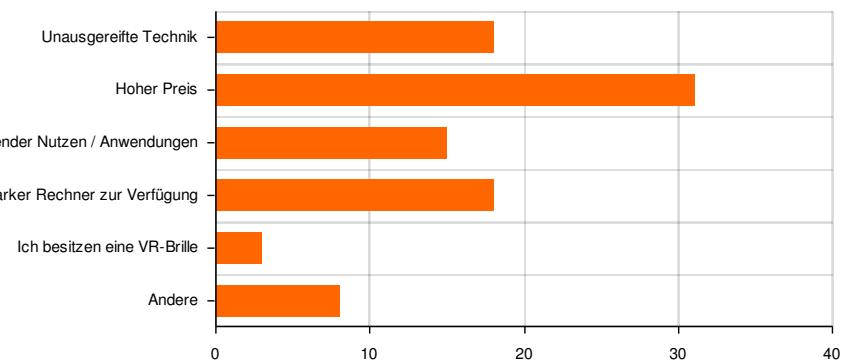
18 (51.4%): Kein
Leistungsstarker Rechner
zur Verfügung

3 (8.6%): Ich besitzen eine
VR-Brille

8 (22.9%): Andere

Antwort(en) aus dem
Zusatzfeld:

- Ich besitze eine
- PappVR
- Tragekomfort
- Unnötig da ich keine Spiele nutze
- Es ist anstrengend: Tragekomfort. Anstrengend für die Augen wegen der Pixel
- Nicht Gesellschaftsfähig
- Vier Beamer in einem Raum ist besser
- Kopfschmerzen / Verlust der Wahrnehmung des reelen Raumes

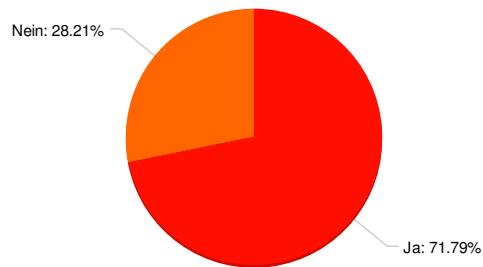


41. Hat Sie die Leap Motion technisch soweit überzeugt, dass die feste Integration einen Aufpreis rechtfertigen würde? *

Anzahl Teilnehmer: 39

28 (71.8%): Ja

11 (28.2%): Nein



42. Welchen Aufpreis wären Sie bereit zu zahlen?

Anzahl Teilnehmer: 39

 Alle 4 vorangegangenen Antworten anzeigen

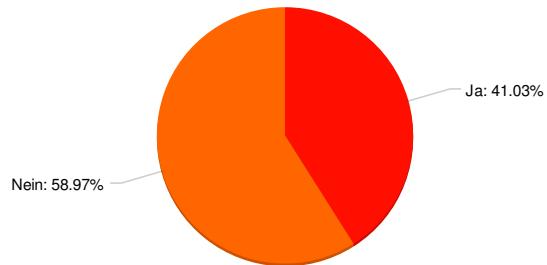
- 0
- 30
- 30
- 50
- 150
- 100
- 150
- 40
- 250
- 0
- 100
- 30
- 0
- 50
- 0
- 200
- 30
- 50
- 50
- 30
- 50
- 100
- 200
- 0
- 0
- 200
- 100
- 0
- 200
- 0
- 50
- 50
- 150
- 150
- 0

43. Hat Sie die Objekterkennung soweit überzeugt, dass die feste Integration einen Aufpreis rechtfertigen würde? *

Anzahl Teilnehmer: 39

16 (41.0%): Ja

23 (59.0%): Nein



44. Welchen Aufpreis wären Sie bereit zu zahlen? *

Anzahl Teilnehmer: 39

 Alle 4 vorangegangenen Antworten anzeigen

- 0
- 0
- 0
- 50
- 0
- 0
- 150
- 0
- 200
- 0
- 20
- 40
- 20
- 0
- 0
- 0
- 30
- 0
- 0
- 0
- 0
- 150
- 0
- 0
- 0
- 100
- 0
- 80
- 0
- 0
- 50
- 0
- 50
- 150
- 50

45. Welche Komponenten würden Sie sich in der VR wünschen? *

Anzahl Teilnehmer: 39

24 (61.5%): Reale Objekte

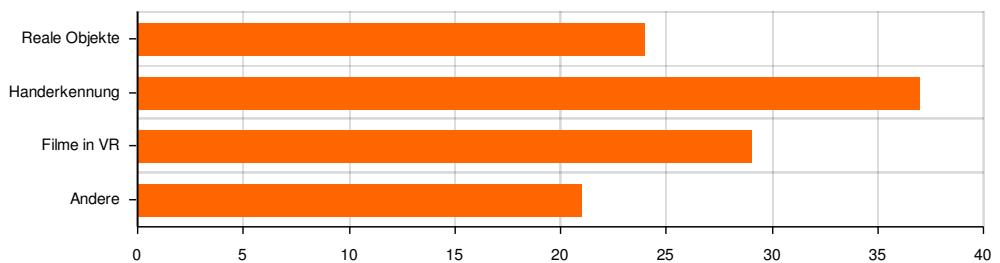
37 (94.9%): Handerkennung

29 (74.4%): Filme in VR

21 (53.8%): Andere

Antwort(en) aus dem
Zusatzfeld:

- Alle Sinne sollen angesprochen werden
- Office Anwendungen, Musikanwendungen, Künstlerische Gestaltungen
- 3D sculpting, Virtuelle chats
- Interaktionsgesten / ein komplettes HUD mit der Leap wäre perfekt (MinorityReport) /
- CAD-Software, Für Körperlich eingeschränkte,
- Das man auf der stelle laufen kann, (Omni besser ausgereift)
- Bessere erkennung der Gegenstände im Weg / Umwelterkennung
- Games in VR
- Ich würde mir wünschen das es nicht nur Unterhaltungszwecken dient sondern auch zur erleichterung zur Intregration mit anderen Geräten (z.B. Musikanlage, Fernseher)
- Dich stört die beschränktheit des Raumes ich möchte herumlaufen können
- VR-Modelling
- Filme aus neugier
- Kabellos
- Wenn die realen Objekte richtig ausgereift wären
- Anwendung im Berufsalltag / Schulungen / Berufsausbildungen
- Bei Zombiespielen Warnhinweis wenn Gegner kommen
- Zu Lernzwecken / Für ältere Menschen damit Sie dinge erleben könnten die Sie sonst nicht erleben könnten /
- Spiele
- Gewehre; Schwerter,
- Beim Film Interagieren / Star Wars Xwing Commander /
- ganzer Körper in VR
- Shooter in VR



46. Kritik und Anregungen

Anzahl Teilnehmer: 32

- War geil, war richtig fett!
- Die Integration von realen Objekten bietet mir Persönlich eine Starken Mehrwert.
- Ich hoffe ihr bekommt eine 1.0 nach dem ganzen Feuerscheiß
- Technik, die Erkennung vom Würfel es gibt nur einen kleinen Handlungsspielraum
- Ist gut gemacht, die Antworten auf die Tests sind vielsagend lassen wenig Raum für weitere Einwürfe, ihr habt euch Gedanken gemacht
Ich würde mir Interaktionsdarstellungen auf Virtueller Ebene wünschen (der virtuelle Elemente wie der Tisch war richtig gut, sollte aber besseres visuall Feedback bekommen)
- Ich will mein Geld nicht investieren weil es unreif wirkt.
- Würfel auf Panel verschieben!
Würfel auf dem Boden war schlecht als Würfel zu erkennen.
- Echte Knöpfe auf dem Bedienfeld.
Der Würfel sollte besser erkannt werden.
- Bedienfelder durch Blickrichtung bzgl. Film.
- Die Virtuelle Bedienoberfläche reagiert zu früh.
Ich wünsche mir das die Einsatzbereiche ausgebaut werden. (Wie Integration von Musikanlage etc.)
Interaktion mit einem Virtuellen Tutorial für echte Anwendungen Bsp. CarMechanikVR für lernende Kfzler
Virtuelles Buch lesen.
- Man sollte die Grafik und die Erkennung noch verbessern.
Warnhinweise zu Realitätsverlust !
vielleicht den kompletten Körper sehen
Sound
Omni! laufen
- Mit den Interaktionsmöglichkeiten war es sehr interessant und gut umgesetzt, vor allem durch die Haptik.
- Ich wünsche mir einen Virtuellen Körper. (Nicht nur die Hände)
Kabellos
 - Das Prinzip ist cool aber die Technik ist nicht ausgereift. Mehr zum Vergnügen nicht wirklich nützlich.
Man kann in einer VR-Brille die Atmosphäre viel besser einfangen, im Vergleich zu Casualgames.
Sehr angenehm! Hier gabs Bier!
 - Ich fand die VR sehr faszinierend und immersiv.
Es hat mir Spaß gemacht an dem Projekt teilzunehmen.
Die Technik sollte ausgereicht sein.
Ich freue mich auf mein Bier!
Die Anleitung durch den Studienleiter war super.
Ich würde gerne Hogwarts in VR sehen, durch die Gänge laufen etc. wäre der Hammer.
 - Fand es interessant eine VRbrille zu testen.
 - Mehr Bier!
 - Ich fand es gut.
Ich würde mir eine ausgereiftere Technik wünschen, bessere Reaktionszeit.
 - Richtig geil, ich würde es mir kaufen wenn ich das Geld dazu hätte.
Mir stört die Einschränkung durch das Zimmer beim Laufen.
 - Ich finde es interessant es zu untersuchen.
 - Die Objekterkennung ist unausgereift.
 - Bessere Einführung in die Technik.
Clipping bei den Händen und den Gegenständen.
 - VR-Porno wäre klasse. Kim Jong Un ist scheiße. Der SC Freiburg wird Meister. Die Technik ist zwar noch nicht ausgereift, und es lohnt sich damit umzugehen, da wir in einer Zeit leben die sich vorwärts bewegt. Es wäre für jeden Mensch von Vorteil sich mit dieser fortschreitenden Technik zu beschäftigen.
 - Es geht an die Augen, Gewichtsreduktion der Brille, vom Kabel weg, Sensortechnik der Handerkennung,
 - Gut erklärt! Guter Versuchsaufbau!
 - Die geringe Pixelzahl strengt die Augen an. Da die Handerkennung nicht ausgereift ist. Das Interaktionen durch Hand und Knopf drücken gegeben ist.
 - Mit dem Controller in der VR fand ich besser, Handerkennung war trotzdem gut.
Ich M/22 habe einen dicken brutal brachialen und unfassbar schönen Penis!

- Die Technik müsste flüssiger laufen.

Die Hände waren cool.=)

Der Schnubbi hat das ganz toll gelöst um mit dem Bierkastenturm reale Objekte einzubauen!

Der Schnubbi hat das auch toll gemacht die Probanden vorab im dunkeln zu lassen. Deswegen war es umso interessanter!

Die Idee an sich ist gut, weil VR auch die meisten Interessiert.

Eine ausgearbeitetere Umgebung wäre wünschenswert gewesen, (Aber Zeitrahmen)!

Es war eine Schöne ruhige Atmosphäre, die Anweisungen waren klar und verständlich.

Gerne wieder. Gruß an das Auswertungsteam.

- Es kamen viele Fragen sehr oft die ich gleich beantworten wollte.

Ich hatte schon die OculusRift auf, die HTC Vive war zur schwer, aber die HTC Vive war angenehmer vom Schaumstoff

Ich finde es gut das VRspiele kurz und knackig gehalten, (geradliniges Spiel) (zum jetzigen Stand der Technik)

Preisleistungsverhältnis ist ok / Spiele sind günstig und packend (emotionalen Aspekten)

Sehr Interessant das man die Umwelt um sich herrum vergisst / Sound ist sehr wichtig

Oculus TouchController sehr spannend da sie im Gegensatz zur Leap immer funktionieren

- Hat alles geklappt

- Unangenehmes Tragegefühl

Kabel

Zu teuer

Betreuer hat mich nicht ausgelacht, das war nett.

Interessante Technik, würde es aber trotzdem nicht kaufen.

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using Leap;
5  using LeapInternal;
6
7  public class collidorTrigger : MonoBehaviour {
8
9      // string für Triggername
10     private string triggerButtonName;
11
12     void Start ()
13     {
14         // Speichert localen Triggernamen
15         triggerButtonName = transform.name;
16     }
17
18     private void OnTriggerEnter(Collider other)
19     {
20         // erkennt den Trigger mit den Leap Händen
21         sendMessageToPanel();
22     }
23
24     // Sendet message an die Inputmessenger
25     public void sendMessageToPanel()
26     {
27         SendMessageUpwards("PanelInputManager", triggerButtonName);
28     }
29
30 }
31
```

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class fps_manipulator : MonoBehaviour {
7
8      // Die in der Szene verwendeten Kameras
9      public Camera camHead;
10     public Camera camEye;
11
12     // Die einstellbare Zielfps
13     public float targetfps = 80;
14
15     // Die zeit die zu warten sind in Sekunden
16     private float secondsToWait = 0.000001f;
17     // Variable für die InGameFPS
18     private float gameFps;
19
20     void Update()
21     {
22         // speichert die FPS in die variable
23         gameFps = 1.0f / Time.deltaTime;
24
25         // Wenn die Ingamefps größer als die geplante FPS
26         if (gameFps > targetfps)
27         {
28             // Starte die Coroutine
29             StartCoroutine(DelayedRendering());
30         }
31         // wenn die GameFps kleiner als die Zielfps
32         else if(gameFps < targetfps)
33         {
34             // Render die Kameras
35             camEye.Render();
36             camHead.Render();
37         }
38         else
39         {
40             // ansonsten Render die Kamera, das auf jeden fall ein Bild entsteht
41             camEye.Render();
42             camHead.Render();
43         }
44     }
45
46     public IEnumerator DelayedRendering()
47     {
48         // solange bis die zeit vergangen ist
49         while (true)
50         {
51             // Render die Kamera
52             camEye.Render();
53             camHead.Render();
54             // Gehe aus der Coroutine nach der abgelaufenen Zeit
55             yield return new WaitForSeconds(secondsToWait); ↗
```

```
56         }
57     }
58 }
59
```

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using System.IO;
5
6  public class writeFile : MonoBehaviour
7  {
8      // Zeit der Messung in Sekunden
9      public int measureTimeInFps = 100;
10     // String um in txt zu schreiben
11     private string s = "";
12     // float um fps zu runden
13     private float r;
14     // Wie oft er Pro Sekunde messen soll
15     private float updateRate = 1;
16     // Zählt die Frames bis zur nächsten Messung
17     private float frameCount;
18     // die Zeit die ein Frame braucht
19     private float deltaTime = 0;
20     // Fps als float
21     private float fps = 0;
22     // zählt fps
23     private float fpsCount;
24
25     private void Update()
26     {
27
28         // zählt framecount hoch
29         frameCount++;
30         // solange fpscount < Messzeit ist
31         if (fpsCount < measureTimeInFps)
32         {
33             // erhöht variable um die Zeit die der letzte Frame gebraucht hat
34             deltaTime += Time.deltaTime;
35
36             // sobald die Zeit der Frames > als 1 / updaterrate
37             if (deltaTime > 1.0 / updateRate)
38             {
39                 // frames = zahl der frames / durch die die sie brauchten
40                 fps = frameCount / deltaTime;
41                 // Counter zurücksetzen
42                 frameCount = 0;
43                 // verringert deltaTime um 1 / updaterrate
44                 deltaTime -= 1.0f / updateRate;
45                 // setzt r als die gerundeten gameFps
46                 r = Mathf.Round(fps);
47                 // Convertiert die GameFps zu einem String
48                 s = r.ToString();
49                 // Übergibt den String an die Funktion
50                 WriteString(s);
51                 // Setzt Dauer Count nach oben
52                 fpsCount++;
53             }
54         }
55     else
56     {
```

```
57         print("Messungsende");
58     }
59
60
61 }
62
63 static void WriteString(string s)
64 {
65     //Ruft das File auf
66     string path = "Assets/Resources/test.txt";
67     //Erzeugt ein Streamwriter Objekt
68     StreamWriter writer = new StreamWriter(path, true);
69     // Schreibt den String in das File
70     writer.WriteLine(" " + s + " ");
71     // Beendet den Schreibvorgang
72     writer.Close();
73 }
74 }
```

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PanelInputEventListener : MonoBehaviour {
6
7      // Name des gesetzten Trigger
8      private string triggerName;
9      // Die zu manipulierende Würfelmatrix
10     public Transform cubeMatrix;
11     // Einstellbare bewegungsgeschwindigkeit
12     public float InputParameter = 1f;
13
14     // Eventlistener für das Inputevent
15     public void PanelInputManager(string name)
16     {
17         // Speichere Tiggername
18         triggerName = name;
19         // Ruft switchcase auf
20         buttonManager();
21     }
22
23     private void buttonManager()
24     {
25         // Switch case für die entsprechenden Interaktionen
26         switch (triggerName)
27         {
28             case "InputButton_Trigger_1":
29                 cubeMatrix.transform.Rotate(Vector3.forward * -InputParameter * ↵
30                     Time.deltaTime, Space.World);
31                 break;
32             case "InputButton_Trigger_2":
33                 cubeMatrix.transform.Rotate(Vector3.forward * InputParameter * ↵
34                     Time.deltaTime, Space.World);
35                 break;
36             case "InputButton_Trigger_3":
37                 cubeMatrix.transform.Rotate(Vector3.right * -InputParameter * ↵
38                     Time.deltaTime, Space.World);
39                 break;
40             case "InputButton_Trigger_4":
41                 cubeMatrix.transform.Rotate(Vector3.right * InputParameter * ↵
42                     Time.deltaTime, Space.World);
43                 break;
44         }
45     }
46 }
```

```
...\\MixRealityApp\\Assets\\SteamVR\\Scripts\\SteamVR_Render.cs 1
 1 //===== Copyright (c) Valve Corporation, All rights reserved.
 2 =====
 3 // Purpose: Handles rendering of all SteamVR_Cameras
 4 //
 5 //
 6 =
 7 using UnityEngine;
 8 using System.Collections;
 9 using Valve.VR;
10
11 public class SteamVR_Render : MonoBehaviour
12 {
13     public bool pauseGameWhenDashboardIsVisible = true;
14     public bool lockPhysicsUpdateRateToRenderFrequency = true;
15
16     public SteamVR_ExternalCamera externalCamera;
17     public string externalCameraConfigPath = "externalcamera.cfg";
18
19     public ETrackingUniverseOrigin trackingSpace =
20         ETrackingUniverseOrigin.TrackingUniverseStanding;
21
22     static public EVREye eye { get; private set; }
23
24     static private SteamVR_Render _instance;
25     static public SteamVR_Render instance
26     {
27         get
28         {
29             if (_instance == null)
30             {
31                 _instance = GameObject.FindObjectOfType<SteamVR_Render>();
32
33                 if (_instance == null)
34                     _instance = new GameObject
35                         ("[SteamVR]").AddComponent<SteamVR_Render>();
36             }
37             return _instance;
38         }
39     void OnDestroy()
40     {
41         _instance = null;
42     }
43
44     static private bool isQuitting;
45     void OnApplicationQuit()
46     {
47         isQuitting = true;
48         SteamVR.SafeDispose();
49     }
50
51     static public void Add(SteamVR_Camera vrcam)
```

```
52     {
53         if (!isQuitting)
54             instance.AddInternal(vrcam);
55     }
56
57     static public void Remove(SteamVR_Camera vrcam)
58     {
59         if (!isQuitting && _instance != null)
60             instance.RemoveInternal(vrcam);
61     }
62
63     static public SteamVR_Camera Top()
64     {
65         if (!isQuitting)
66             return instance.TopInternal();
67
68         return null;
69     }
70
71     private SteamVR_Camera[] cameras = new SteamVR_Camera[0];
72
73     void AddInternal(SteamVR_Camera vrcam)
74     {
75         var camera = vrcam.GetComponent<Camera>();
76         var length = cameras.Length;
77         var sorted = new SteamVR_Camera[length + 1];
78         int insert = 0;
79         for (int i = 0; i < length; i++)
80         {
81             var c = cameras[i].GetComponent<Camera>();
82             if (i == insert && c.depth > camera.depth)
83                 sorted[insert++] = vrcam;
84
85             sorted[insert++] = cameras[i];
86         }
87         if (insert == length)
88             sorted[insert] = vrcam;
89
90         cameras = sorted;
91     }
92
93     void RemoveInternal(SteamVR_Camera vrcam)
94     {
95         var length = cameras.Length;
96         int count = 0;
97         for (int i = 0; i < length; i++)
98         {
99             var c = cameras[i];
100            if (c == vrcam)
101                ++count;
102        }
103        if (count == 0)
104            return;
105
106        var sorted = new SteamVR_Camera[length - count];
107        int insert = 0;
```

```
108         for (int i = 0; i < length; i++)
109         {
110             var c = cameras[i];
111             if (c != vrcam)
112                 sorted[insert++] = c;
113         }
114
115         cameras = sorted;
116     }
117
118     SteamVR_Camera TopInternal()
119     {
120         if (cameras.Length > 0)
121             return cameras[cameras.Length - 1];
122
123         return null;
124     }
125
126     public TrackedDevicePose_t[] poses = new TrackedDevicePose_t
127         [OpenVR.k_unMaxTrackedDeviceCount];
128     public TrackedDevicePose_t[] gamePoses = new TrackedDevicePose_t[0];
129
130     static private bool _pauseRendering;
131     static public bool pauseRendering
132     {
133         get { return _pauseRendering; }
134         set
135         {
136             _pauseRendering = value;
137
138             var compositor = OpenVR.Compositor;
139             if (compositor != null)
140                 compositor.SuspendRendering(value);
141         }
142     }
143
144     private WaitForEndOfFrame waitForEndOfFrame = new WaitForEndOfFrame();
145
146     private IEnumerator RenderLoop()
147     {
148         while (Application.isPlaying)
149         {
150             yield return waitForEndOfFrame;
151
152             if (pauseRendering)
153                 continue;
154
155             var compositor = OpenVR.Compositor;
156             if (compositor != null)
157             {
158                 if (!compositor.CanRenderScene())
159                     continue;
160
161                 compositor.SetTrackingSpace(trackingSpace);
162             }
163         }
164     }
165 }
```

```
163         var overlay = SteamVR_Overlay.instance;
164         if (overlay != null)
165             overlay.UpdateOverlay();
166
167         RenderExternalCamera();
168     }
169 }
170
171 void RenderExternalCamera()
172 {
173     if (externalCamera == null)
174         return;
175
176     if (!externalCamera.gameObject.activeInHierarchy)
177         return;
178
179     var frameSkip = (int)Mathf.Max(externalCamera.config.frameSkip, 0.0f);
180     if (Time.frameCount % (frameSkip + 1) != 0)
181         return;
182
183     // Keep external camera relative to the most relevant vr camera.
184     externalCamera.AttachToCamera(TopInternal());
185
186     externalCamera.RenderNear();
187     externalCamera.RenderFar();
188 }
189
190 float sceneResolutionScale = 1.0f, timeScale = 1.0f;
191
192 private void OnInputFocus(bool hasFocus)
193 {
194     if (hasFocus)
195     {
196         if (pauseGameWhenDashboardIsVisible)
197         {
198             Time.timeScale = timeScale;
199         }
200
201         SteamVR_Camera.sceneResolutionScale = sceneResolutionScale;
202     }
203     else
204     {
205         if (pauseGameWhenDashboardIsVisible)
206         {
207             timeScale = Time.timeScale;
208             Time.timeScale = 0.0f;
209         }
210
211         sceneResolutionScale = SteamVR_Camera.sceneResolutionScale;
212         SteamVR_Camera.sceneResolutionScale = 0.5f;
213     }
214 }
215
216 void OnQuit(VREvent_t vrEvent)
217 {
218 #if UNITY_EDITOR
```

```
219     foreach (System.Reflection.Assembly a in
220         System.AppDomain.CurrentDomain.GetAssemblies())
221     {
222         var t = a.GetType("UnityEditor.EditorApplication");
223         if (t != null)
224         {
225             t.GetProperty("isPlaying").SetValue(null, false, null);
226             break;
227         }
228     #else
229         Application.Quit();
230     #endif
231     }
232
233     private string GetScreenshotFilename(uint screenshotHandle,
234         EVRScreenshotPropertyFilenames screenshotPropertyName)
235     {
236         var error = EVRScreenshotError.None;
237         var capacity = OpenVR.Screenshots.GetScreenshotPropertyName
238             (screenshotHandle, screenshotPropertyName, null, 0, ref error);
239         if (error != EVRScreenshotError.None && error !=
240             EVRScreenshotError.BufferTooSmall)
241             return null;
242         if (capacity > 1)
243         {
244             var result = new System.Text.StringBuilder((int)capacity);
245             OpenVR.Screenshots.GetScreenshotPropertyName(screenshotHandle,
246                 screenshotPropertyName, result, capacity, ref error);
247             if (error != EVRScreenshotError.None)
248                 return null;
249             return result.ToString();
250         }
251     }
252
253     private void OnRequestScreenshot(VREvent_t vrEvent)
254     {
255         var screenshotHandle = vrEvent.data.screenshot.handle;
256         var screenshotType = (EVRScreenshotType)vrEvent.data.screenshot.type;
257
258         if (screenshotType == EVRScreenshotType.StereoPanorama)
259         {
260             string previewFilename = GetScreenshotFilename(screenshotHandle,
261                 EVRScreenshotPropertyFilenames.Preview);
262             string VRfilename = GetScreenshotFilename(screenshotHandle,
263                 EVRScreenshotPropertyFilenames.VR);
264
265             if (previewFilename == null || VRfilename == null)
266                 return;
267
268             // Do the stereo panorama screenshot
269             // Figure out where the view is
270             GameObject screenshotPosition = new GameObject
271                 ("ScreenshotPosition");
272             screenshotPosition.transform.position = SteamVR_Render.Top
```

```
    (.transform.position;
267     screenshotPosition.transform.rotation = SteamVR_Render.Top      ↵
    (.transform.rotation;
268     screenshotPosition.transform.localScale = SteamVR_Render.Top      ↵
    (.transform.lossyScale;
269     SteamVR_Utils.TakeStereoScreenshot(screenshotHandle,           ↵
        screenshotPosition, 32, 0.064f, ref previewFilename, ref      ↵
        VRfilename);
270
271         // and submit it
272         OpenVR.Screenshots.SubmitScreenshot(screenshotHandle,       ↵
            screenshotType, previewFilename, VRfilename);
273     }
274 }
275
276 void OnEnable()
277 {
278     StartCoroutine(RenderLoop());
279     SteamVR_Events.InputFocus.Listen(OnInputFocus);
280     SteamVR_Events.System(EVREventType.VREvent_Quit).Listen(OnQuit);
281     SteamVR_Events.System(EVREventType.VREvent_RequestScreenshot).Listen   ↵
        (OnRequestScreenshot);
282 #if UNITY_2017_1_OR_NEWER
283     Application.onBeforeRender += OnBeforeRender;
284 #else
285     Camera.onPreCull += OnCameraPreCull;
286 #endif
287     var vr = SteamVR.instance;
288     if (vr == null)
289     {
290         enabled = false;
291         return;
292     }
293     var types = new EVRScreenshotType[]
294     { EVRScreenshotType.StereoPanorama };
295     OpenVR.Screenshots.HookScreenshot(types);
296 }
297
298 void OnDisable()
299 {
300     StopAllCoroutines();
301     SteamVR_Events.InputFocus.Remove(OnInputFocus);
302     SteamVR_Events.System(EVREventType.VREvent_Quit).Remove(OnQuit);
303     SteamVR_Events.System(EVREventType.VREvent_RequestScreenshot).Remove   ↵
        (OnRequestScreenshot);
304 #if UNITY_2017_1_OR_NEWER
305     Application.onBeforeRender -= OnBeforeRender;
306 #else
307     Camera.onPreCull -= OnCameraPreCull;
308 #endif
309 }
310
311 void Awake()
312 {
313     if (externalCamera == null && System.IO.File.Exists
        (externalCameraConfigPath))
```

```
313     {
314         var prefab = Resources.Load<GameObject>("SteamVR_ExternalCamera");
315         var instance = Instantiate(prefab);
316         instance.gameObject.name = "External Camera";
317
318         externalCamera = instance.transform.GetChild(0).GetComponent<SteamVR_ExternalCamera>();
319         externalCamera.configPath = externalCameraConfigPath;
320         externalCamera.ReadConfig();
321     }
322 }
323
324 public void UpdatePoses()
325 {
326     var compositor = OpenVR.Compositor;
327     if (compositor != null)
328     {
329         compositor.GetLastPoses(poses, gamePoses);
330         SteamVR_Events.NewPoses.Send(poses);
331         SteamVR_Events.NewPosesApplied.Send();
332     }
333 }
334
335 #if UNITY_2017_1_OR_NEWER
336     void OnBeforeRender() { UpdatePoses(); }
337 #else
338     void OnCameraPreCull(Camera cam)
339     {
340         // Only update poses on the first camera per frame.
341         if (Time.frameCount != lastFrameCount)
342         {
343             lastFrameCount = Time.frameCount;
344             UpdatePoses();
345         }
346     }
347     static int lastFrameCount = -1;
348 #endif
349
350     void Update()
351     {
352         // Force controller update in case no one else called this frame to ensure prevState gets updated.
353         SteamVR_Controller.Update();
354
355         // Dispatch any OpenVR events.
356         var system = OpenVR.System;
357         if (system != null)
358         {
359             var vrEvent = new VREvent_t();
360             var size = (uint)System.Runtime.InteropServices.Marshal.SizeOf(typeof(VREvent_t));
361             for (int i = 0; i < 64; i++)
362             {
363                 if (!system.PollNextEvent(ref vrEvent, size))
364                     break;
365             }
366         }
367     }
368 }
```

```
366             switch ((EVREventType)vrEvent.eventType)
367             {
368                 case EVREventType.VREvent_InputFocusCaptured: // another ↵
369                     app has taken focus (likely dashboard)
370                     if (vrEvent.data.process.oldPid == 0)
371                     {
372                         SteamVR_Events.InputFocus.Send(false);
373                     }
374                     break;
375                 case EVREventType.VREvent_InputFocusReleased: // that app ↵
376                     has released input focus
377                     if (vrEvent.data.process.pid == 0)
378                     {
379                         SteamVR_Events.InputFocus.Send(true);
380                     }
381                     break;
382                 case EVREventType.VREvent_ShowRenderModels:
383                     SteamVR_Events.HideRenderModels.Send(false);
384                     break;
385                 case EVREventType.VREvent_HideRenderModels:
386                     SteamVR_Events.HideRenderModels.Send(true);
387                     break;
388                 default:
389                     SteamVR_Events.System((EVREventType)
390 vrEvent.eventType).Send(vrEvent);
391                     break;
392             }
393         }
394     }
395
396     // Ensure various settings to minimize latency.
397     Application.targetFrameRate = -1;
398     Application.runInBackground = true; // don't require companion window ↵
399     focus
400     QualitySettings.maxQueuedFrames = -1;
401     QualitySettings.vSyncCount = 0; // this applies to the companion ↵
402     window
403
404     if (lockPhysicsUpdateRateToRenderFrequency && Time.timeScale > 0.0f)
405     {
406         var vr = SteamVR.instance;
407         if (vr != null)
408         {
409             var timing = new Compositor_FrameTiming();
410             timing.m_nSize = (uint)
411                 System.Runtime.InteropServices.Marshal.SizeOf(typeof
412                 (Compositor_FrameTiming));
413             vr.compositor.GetFrameTiming(ref timing, 0);
414
415             Time.fixedDeltaTime = Time.timeScale /
416                 vr.hmd_DisplayFrequency;
417         }
418     }
419 }
```

414

415

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class texture_Changer : MonoBehaviour {
6
7      // Vuforia Gameobj
8      GameObject textureBufferMesh;
9      // SteamVR Material
10     public Material material;
11
12    void Update () {
13        // wenn der TextureBufferMesh nicht vorhanden dann
14        if(textureBufferMesh == null)
15        {
16            // suche das TexturbufferMesh
17            textureBufferMesh = GameObject.Find("TextureBufferMesh");
18
19            // Wenn das TexturebufferMesh vorhanden
20            if (textureBufferMesh != null)
21            {
22                // Speichere den Meshrenderer des TextureBufferMesh
23                var Renderer = textureBufferMesh.GetComponent<MeshRenderer>();
24                // Weise dem Materialrenderer Das SteamVR Material zu
25                Renderer.material = material;
26            }
27        }
28    }
29 }
30 }
```

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using Leap;
5  using System;
6
7  public class Transformation : MonoBehaviour
8  {
9      // Speichert das zu verschiebende Objekt
10     public GameObject target_Cube;
11
12     // Vector4 um eine 4x4 Matrix zu erstellen
13
14     public Vector4 a = new Vector4(Mathf.Sqrt(3) / 2, -0.5f, 0, 1.825f);
15     public Vector4 b = new Vector4(0.5f, 0, 0, 3.85f);
16     public Vector4 c = new Vector4(0, 0, 1, 0);
17     public Vector4 d = new Vector4(0, 0, 0, 1);
18
19     // Die Matrix um die Transformiert wird
20     private Matrix4x4 TransformMatrix;
21
22
23     void Start()
24     {
25         // Vectoren in die 4x4 Matrix speichern
26         TransformMatrix.SetRow(0, a);
27         TransformMatrix.SetRow(1, b);
28         TransformMatrix.SetRow(2, c);
29         TransformMatrix.SetRow(3, d);
30
31         // Variable mit den Positions値en der Matrix
32         Vector3 targetPosVar = target_Cube.transform.position;
33
34         // 4x4 Matrix mit den Transformations値en des zu verschiebenden
35         // Objekts
36         Matrix4x4 TargetCube = Matrix4x4.TRS(target_Cube.transform.position,
37             target_Cube.transform.rotation, target_Cube.transform.localScale);
38         // 4x4 Matrix mit den neuen Positions値en
39         Matrix4x4 TargetPositionMatrix = TargetCube * TransformMatrix;
40
41         // ändert die Position des Würfels nach der Matrix
42         //target_Cube.transform.position = new Vector3(TargetPositionMatrix[0, 3] * targetPosVar.x, TargetPositionMatrix[1, 3] * targetPosVar.y, TargetPositionMatrix[2, 3] * targetPosVar.z);
43         // ändert die Rotation des Würfels nach der Matrix
44         target_Cube.transform.localEulerAngles = new Vector3(Mathf.Cos(TargetPositionMatrix[0, 0]), -Mathf.Sin(TargetPositionMatrix[0, 1]), Mathf.Sin(TargetPositionMatrix[1, 0]));
45     }
46
47 }
48
```

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using System.IO;
5
6  public class writeFile : MonoBehaviour {
7
8      public int measureTimeInFps = 1000;
9      private float gameFps = 0;
10     private string s = "";
11     private float r;
12     public bool onOff = false;
13
14     private void Update()
15     {
16         print(gameFps = 1.0f / Time.deltaTime);
17         //print(Time.frameCount);
18
19         // Schaltet Writescrypt an
20         if (onOff)
21         {
22             // Wenn die EngineFps < die zu messende Zeit
23             if (Time.frameCount < measureTimeInFps)
24             {
25                 // setzt die gameFps = den EngineFps
26                 gameFps = 1.0f / Time.deltaTime;
27                 // setzt r als die gerundeten gameFps
28                 r = Mathf.Round(gameFps);
29                 // Convertiert die GameFps zu einem String
30                 s = r.ToString();
31                 // Übergibt den String an die Funktion
32                 WriteString(s);
33             }
34             else
35             {
36                 Debug.Log("Measure has ended!");
37             }
38         }
39     }
40
41     static void WriteString(string s)
42     {
43         //Ruft das File auf
44         string path = "Assets/Resources/test.txt";
45         //Erzeugt ein Streamwriter Objekt
46         StreamWriter writer = new StreamWriter(path, true);
47         // Schreibt den String in das File
48         writer.WriteLine(" " + s + " ");
49         // Beendet den Schreibvorgang
50         writer.Close();
51     }
52 }
```