|  |  |  |
| --- | --- | --- |
| C:\_Hochschule Heilbronn\6_AG MCI\4_Organisation\0_Corporate Design\1_Logo\UniTyLab_Logo.jpg |  | D:\Desktop\HHN Original (Vektor).emf |

Bachelor

Thesis

**Konzeption und Evaluation von Benutzerkonditionierung**

**in Virtual Reality**

|  |  |
| --- | --- |
| **Autor** | Robert Zlomke |
| **Studiengang** | Angewandte Informatik  Hochschule Heilbronn |
| **Matrikel-Nr.** | 193045 |
| **Abgabe** | 15.04.2021 |
| **Referent** | Prof. Dr.-Ing. Gerrit Meixner |
| **Korreferent:** | Philip Schäfer |

* 1. Danksagung
  2. Abstract

Virtuelle Umgebungen zu Fuß zu erkunden ist die realistischste und natürlichste Schnittstelle. Zeitgleich ist dies auch technisch und logistisch am anspruchsvollsten.

Allgemein wurden in bisherigen Arbeiten sich auf die Fortbewegung in übereinstimmenden oder leeren Umgebungen konzentriert. Weniger wurde darauf geachtet, wie sich eine Änderung der Nichtübereinstimmung zwischen der physischen Umgebung und ihrer virtuellen Darstellung auf die Benutzer auswirkt.

**Schlagwörter**: Wissenschaftliche Arbeit, Dokumentvorlage, Hochschule

## Inhaltsverzeichnis

[I. Danksagung 2](#_Toc66192517)

[II. Abstract 3](#_Toc66192518)

[1 Inhaltsverzeichnis 4](#_Toc66192519)

[2 Einleitung 6](#_Toc66192520)

[2.1 Unterpunkt 1 6](#_Toc66192521)

[3 Stand der Technik 7](#_Toc66192522)

[3.1 Konditionierung 7](#_Toc66192523)

[3.2 Virtual Reality 9](#_Toc66192524)

[3.2.1 Natural Walking 9](#_Toc66192525)

[3.2.2 Redirected Walking 9](#_Toc66192526)

[3.2.3 Walkin-In-Place 10](#_Toc66192527)

[3.2.4 Abstahierte Schnittstelle des Gehens 11](#_Toc66192528)

[3.2.5 Laufbänder 11](#_Toc66192529)

[3.2.6 (Bewegungsverhalten) 11](#_Toc66192530)

[4 Problemstellung, Zielsetzung und Vorgehensweise 13](#_Toc66192531)

[4.1 Problemstellung 13](#_Toc66192532)

[4.2 Zielsetzung 13](#_Toc66192533)

[4.3 Vorgehensweise 13](#_Toc66192534)

[4.4 Technologien 13](#_Toc66192535)

[4.4.1 Unity 3D 13](#_Toc66192536)

[4.4.2 Oculus Quest 13](#_Toc66192537)

[4.4.3 Visual Studio 14](#_Toc66192538)

[4.4.4 Programmiersprache C# 14](#_Toc66192539)

[4.4.5 Git Versionskontrolle 15](#_Toc66192540)

[5 Umsetzung 16](#_Toc66192541)

[5.1 Einbinden der Oculus Quest in Unity 3D 16](#_Toc66192542)

[5.2 Entwicklung des Malus 17](#_Toc66192543)

[5.3 Entwicklung der Szenarien 17](#_Toc66192544)

[5.4 Datenerfassung 19](#_Toc66192545)

[5.5 Evaluation 22](#_Toc66192546)

[6 Zusammenfassung, Bewertung und Ausblick 23](#_Toc66192547)

[7 Anhang 24](#_Toc66192548)

[7.1 Abkürzungsverzeichnis 24](#_Toc66192549)

[7.2 Abbildungsverzeichnis 25](#_Toc66192550)

[7.3 Literaturverzeichnis 26](#_Toc66192551)

[7.4 Eidesstattliche Erklärung 28](#_Toc66192552)

## Einleitung

### Unterpunkt 1

Die Erkundung virtueller Umgebungen (VE) durch Gehen ist die realistischste und natürlichste

Schnittstelle für VR-Systeme (Virtual Reality), in denen Benutzer anthropomorphe Charaktere

verkörpern [1]. Es ist auch die technisch anspruchsvollste Schnittstelle aufgrund von

Hardwareproblemen (Tracking-Systeme, zu tragende Instrumente) und Umweltproblemen

(Ungleichheit zwischen den Ausmaßen des VE und dem verfügbaren physischen Raum,

Vorhandensein von physikalischen Elementen, die in der Simulation nicht vorhanden sind).

Obwohl verschiedene Lösungen für diese Probleme vorgestellt wurden, sind diese

nicht immer an häusliche Umgebungen wie Wohnungen, Büros und andere Orte anpassbar, die

nicht explizit als VR-Labor konzipiert sind. [2]

Wir sind der Meinung, dass die inländische VR ein grundlegendes Sprungbrett für ihre breite Akzeptanz darstellt. Diese Umgebungen bieten jedoch unterschiedliche andere Herausforderungen im Vergleich zu VR-Labors: in Häusern, Möbeln schränkt den navigierbaren Raum ein. Stattdessen sind VR-Labore normalerweise groß und leere Räume. Das Einbeziehen von Objekten aus der unmittelbaren physischen Umgebung des Benutzers in die Simulation durch Ersetzen durch nicht übereinstimmende virtuelle Gegenstücke bietet das Potenzial, ein überzeugendes VR-Erlebnis zu bieten. Wenn sich frühere Arbeiten jedoch auf die Bewegung in übereinstimmenden Umgebungen [3] oder leeren Umgebungen konzentrierten [4], wurde weniger darauf geachtet, ob sich eine Änderung der Nichtübereinstimmung zwischen der physischen Umgebung und ihrer virtuellen Darstellung auf das Bewegungsverhalten der Benutzer auswirkt. Wenn ja, gibt es bestimmte Änderungen, die wirksamer sind als andere?

## Stand der Technik

### Konditionierung

Die Pawlowsche Konditionierung ist ein wertvolles Labor Modell, um den Erwerb, die Ausprägung, die Generalisierung und die Hemmung von bedrohungsbezogenem Verhalten bei verschiedenen Spezies zu untersuchen4-6. Zwei pawlowsche Konditionierungsparadigmen haben als Arbeitsgrundlage gedient, um die Mechanismen zu verstehen, die der erlernten Bedrohung und der Rolle der kontextuellen Prozesse zugrunde liegen: die operante Konditionierung und die Kontextkonditionierung. Bei der aversiven Pawlowschen Konditionierung lernt eine Versuchsperson, dass ein neutraler Stimulus in der Umgebung (konditionierter Stimulus, CS, z.B. ein Ton oder ein Bild) das Auftreten eines aversiven Ergebnisses (unkonditionierter Stimulus, US, z.B. ein elektrischer Schock) vorhersagt. Sobald die Versuchsperson die CS-US-Assoziation erlernt hat, löst die Präsentation des CS allein bedrohungsbezogene Abwehrreaktionen aus, wie z. B. Erstarren und erhöhte sympathische Erregung. Die Methode der operanten Konditionierung ist leicht von Versuchstieren auf den Menschen übertragbar, wo einfache sensorische Reize wie Töne und Lichter oder komplexere Reize wie Gesichter und Objektkategorien als CSs dienen können. Die pawlowsche Bedrohungskonditionierung dient nach wie vor als wertvolles und nachvollziehbares Modell für Psychopathologien, die durch akute oder phasische Bedrohungsreaktionen gekennzeichnet sind, die durch diskrete Reize ausgelöst werden, wie z.B. Zwangsstörungen, Phobien und Symptomcluster der PTBS, die Abwehrreaktionen auf Trauma-Erinnerungen beinhalten [siehe Ref. 4 für eine Übersicht].

Bei der pawlowschen Kontextkonditionierung wird der US ausgelöst, während sich die Versuchsperson in einer bestimmten Umgebung (d.h. in der Konditionierungskammer) befindet, aber nicht durch einen diskreten CS signalisiert wird. Bemerkenswert ist, dass der Begriff "Kontext" in der Forschung zum assoziativen Lernen sehr weit gefasst werden kann und von den sensorischen Details der Umgebung (Anblicke, Gerüche, Geräusche usw.) über innere Zustände bis hin zu Zeitwahrnehmungen reicht7. Die Kontextkonditionierung bietet ein geeigneteres Modell für Psychopathologien, die durch anhaltende oder "freischwebende" Angst gekennzeichnet sind, wenn es keinen eindeutigen bedrohungsauslösenden Stimulus in der Umgebung gibt8, wie z.B. bei der Generalisierten Angststörung oder Symptomclustern der PTBS, die hyperarousal in Abwesenheit von bedrohungsauslösenden Hinweisen beinhalten.

Der Kontext interagiert auch mit der Cue-Konditionierung, um die Ausprägung des assoziativen Speichers CS-US zu modulieren. Zum Beispiel könnte ein CS, der den US in einem Kontext vorhersagt, dann ohne den US in einem anderen Kontext präsentiert werden, d.h. Extinktion. Das Testen des CS im ersten (d.h. Akquisitions-), zweiten (d.h. Extinktionskontext) oder einem neuen Kontext kann zeigen, welche CS-Assoziation die Testperson abruft. Die Forschung der neurologischen Verhaltensweisen hat gezeigt, wie der Kontext die Ausprägung von assoziativen CS-US-Erinnerungen beeinflusst7, 9; z.B. können Cue-ausgelöste Reaktionen bei einem späteren Test im Extinktionskontext reduziert werden oder im Extinktionskontext (spontane Erholung) oder im Akquisitions- oder neuen Kontext wieder auftauchen. Insbesondere konzentriert sich der vorliegende Artikel weitgehend auf Paradigmen der Kontextkonditionierung, bei denen nur die räumlich-zeitliche Umgebung den US in Abwesenheit eines diskreten bedrohungsauslösenden Hinweises signalisiert. In der Literatur über den Menschen wurde der Begriff Kontextkonditionierung oft fälschlicherweise verwendet, um Paradigmen zu beschreiben, bei denen der Kontext die Konditionierung des Cues moduliert.

Trotz der zunehmenden Anzahl von Studien zu kontextuellen Prozessen bei Versuchstieren gibt es bis heute nur wenige systematische Untersuchungen zur Kontextkonditionierung beim Menschen (Tabelle 1). Dies ist größtenteils auf die Herausforderung zurückzuführen, einen räumlich-zeitlichen Kontext zu schaffen, der mit einem US gekoppelt werden kann. Bei Versuchstieren wird der Kontext manipuliert, indem der physische Ort des Tieres (z. B. eine andere Konditionierungsbox) oder der Raum (z. B. Veränderung der Wandfarben, der Bodenbeschaffenheit und/oder des Geruchs) verändert wird. Das offensichtlichste Analogon zur Konditionierungskammer für menschliche Probanden ist der Laborversuchsraum. Obwohl der Raum der vorherrschende Kontext ist, ist er oft nicht das Ziel der Kontextkonditionierung. Mit anderen Worten, anders als in der Labor Tierforschung beabsichtigt der Experimentator in Studien zur Konditionierung von Menschen im Allgemeinen nicht, dass die Versuchsperson eine Assoziation zwischen der physischen Umgebung des Testraums (bestehend aus bestimmter Beleuchtung, Möbeln, Computern, Forschungsgeräten usw.) und den USA herstellt.

Die Technologie der virtuellen Realität kann genutzt werden, um kontextuelle Stimuli zu optimieren und die Chance zu erhöhen, dass die Kontextkonditionierung kontextuelle Lernsysteme im Gehirn aktiviert. In erster Linie kann VR ein starkes Gefühl der "Präsenz" hervorrufen, bei dem Menschen denken, sich verhalten, fühlen und das Gefühl haben, sich im virtuellen Raum und nicht in der realen Welt zu befinden34. Ein entscheidendes Merkmal von VR ist das Headset (oder Head Mounted Display), das dazu beiträgt, den sensorischen Input von der äußeren Umgebung zu reduzieren und zusätzlich ermöglicht, dass Kopfbewegungen in visuelle Rotationen übersetzt werden, wodurch eine Bewegungsparallaxe für den Teilnehmer entsteht, wodurch das Gefühl verstärkt wird, in eine virtuelle Umgebung vertieft zu sein und vom gegenwärtigen physischen Ort entfernt zu sein - dies wird als "Immersion" bezeichnet. Bis heute haben nur eine Handvoll Studien zur Kontextkonditionierung VR-Headsets eingesetzt. VR war bisher nicht weit verbreitet, da frühe Systeme kostspielig waren und in einigen Fällen viel Platz benötigten, z. B. [ref. 35]. Mit der kommerziellen Freigabe von VR-Headsets ist die Möglichkeit, die Forschung zur Kontextkonditionierung zu erweitern, nun in Reichweite von fast jedem Forschungslabor, das bereits menschliche Konditionierung mit psychophysiologischer Ausrüstung untersucht. Zugängliche und validierte Protokolle für kommerziell erhältliche VR-Systeme werden sich zunehmend als nützlich für die Untersuchung von Kontextkonditionierung beim Menschen erweisen. [5]

### Virtual Reality

Fortbewegung ist ein wesentliches Merkmal in VR. In vielen immersiven Anwendungsszenarien müssen Benutzer in der Lage sein, ihre Position und ihren Blickwinkel zu ändern. Dies wird mit Hilfe von Bewegungsschnittstellen erreicht, d. h. einer Interaktionstechnik, die durch ein zugrunde liegendes System in der Lage ist, Benutzereingaben in Bewegungen in der VE abzubilden. Aufgrund des breiten Spektrums an 3D-Anwendungen lassen sich diese Interfaces nach der Genauigkeit unterscheiden, mit der sie den Akt des Gehens simulieren. [2]

In den nachfolgenden Unterkapiteln werden die unterschiedlichen Schnittstellen genauer beschrieben.

#### Natural Walking

Als High-Fidelity-Techniken werden solche bezeichnet, die den Akt des Gehens am genauesten reproduzieren. Besonders in den Szenarien, in denen der Benutzer durch einen virtuellen Charakter verkörpert wird, ist das echte Gehen in einem physischen Raum die natürlichste Wahl. Natürliches Gehen bezieht sich in der Regel auf Szenarien, in denen sich die VE vollständig innerhalb des physischen Raums befindet, in dem das VR-System eingesetzt wird. Unter diesen Umständen entspricht ein Schritt in der realen Welt einem Schritt von gleicher Länge in der VE. Diese Fortbewegungsschnittstelle hat sich als präsenzfördernder Faktor erwiesen [6], wenn man sie mit weniger realitätsnahen Fortbewegungstechniken wie Walking-in-Place [1] oder virtuellem Fliegen [7] vergleicht. Es hat sich auch gezeigt, dass natürliches Gehen zu weniger Kollisionen mit Hindernissen führt [8], weniger Trainingszeit erfordert [9], mehr mit realen Trajektorien [10] oder Verhalten [11] übereinstimmt. [2]

#### Redirected Walking

In den Szenarien, in denen Benutzer VEs erkunden müssen, die größer als der verfügbare physische Raum sind, sind natürliche Gehschnittstellen weniger geeignet. "Redirected Walking" bezieht sich auf eine Kategorie von Techniken, die es Benutzern ermöglichen, virtuelle Räume, die größer sind als ihre physische Umgebung, physisch zu erkunden, indem sie verändern, wie der physische Pfad in der realen Welt in der VE wahrgenommen wird [22]. Suma et al. stellten eine Taxonomie von Umleitungstechniken vor [40], die eine frühere Arbeit von Steinicke et al. erweitert [39]. Die Autoren klassifizieren diese Techniken danach, ob sie Translation oder Rotation beeinflussen, nach dem Ausmaß der Veränderung (diskret oder kontinuierlich) und danach, ob sie subtil oder offenkundig sind. Beispiele für subtile Techniken, die das Ausmaß der vom Benutzer zurückgelegten Strecke beeinflussen, sind die "Seven Leagues Boots" (kontinuierlich) [14] und eine Technik von Bruder et al., die die Distanzwahrnehmung durch Selbstbewegungsillusionen kompensiert (diskret) [3]. Offene kontinuierliche Techniken koppeln die translatorische Veränderung mit Navigationshilfen, wie Fahrzeugen oder Aufzügen [11]. Diskrete Techniken nutzen die bekannte Science-Fiction-Metapher von Portalen, um zwischen verschiedenen Teilen einer Umgebung zu navigieren [8, 36]. Reorientierungs-Techniken beeinflussen den Blickwinkel des Benutzers in der Welt. Zu den subtilen Techniken gehört die Arbeit von Razzaque et al. [22] über Redi-rected Walking, die kontinuierlich nicht wahrnehmbaren Verstärkungen der Kopfdrehungen einführen, so dass der Benutzer größere als physische Räume erkunden kann. Diese Veränderungen haben sich als tolerierbar erwiesen, wenn sie innerhalb von Schwellenwerten liegen [37, 38]. Ein Nachteil ist jedoch, dass ein großer räumlicher Radius erforderlich ist, um nicht auffällig zu werden. Dieser Radius wurde mit mindestens 22 m [12] bis zu mehr als 40 m [37] angegeben. Steinicke et al. wendeten Redirected-Walking-Techniken an, um Benutzer zu einem virtuellen Objekt zu führen, das mit einem physischen Proxy übereinstimmt, und zwar entlang eines anderen virtuellen Pfades als dem, den sie in der physischen Welt gegangen sind [39]. Dieser Ansatz wurde mit einem einzigen Proxy-Objekt getestet, da bei einer höheren Anzahl von Proxy-Requisiten die Umlenkungseffekte spürbar werden. Dies macht diese Techniken für kleinere häusliche Umgebungen ungeeignet. Diskrete Techniken ordnen stattdessen die Position einiger architektonischer Merkmale wie z. B. Türöffnungen neu an. Diese Strategie ist jedoch für nicht-abstrakte Umgebungen ungeeignet, in denen die Aufrechterhaltung einer kohärenten räumlichen Beziehung entscheidend ist. Offene kontinuierliche Techniken verlangen explizit, dass der Benutzer sich neu orientiert, wenn er die Grenzen der Umgebung erreicht [48]. Ablenkende Elemente, Objekte oder Figuren, die versuchen, die Aufmerksamkeit des Benutzers zu erregen, indem sie ihn zwingen, seinen Kopf zu drehen, bieten eine Gelegenheit, diese Gewinne einzuführen, während der Benutzer sich dem Stimulus zuwendet [21]. Kohli et al. nutzten eine Drehung der VE, um ein virtuelles Objekt mit einem passenden physischen Proxy auszurichten, während der Benutzer durch die Bewegungen eines Droiden abgelenkt ist [15]. Diskrete Techniken verlangen stattdessen explizit, dass der Benutzer seine Richtung ändert, während die Tracking-Updates eingefroren werden [48]. Neth et al. [19] untersuchten die Auswirkung einer dynamischen Änderung der Verstärkung im Gegensatz zu einer statischen Steuerung auf die durchschnittliche Distanz, die zurückgelegt wird, bevor eine Neuorientierung erforderlich ist. Die Autoren fanden heraus, dass dynamische Regler zu einer signifikant größeren gelaufenen Distanz führten. Sie untersuchten auch den Effekt von kontextuell angemessenen Avatar-Agenten im Gegensatz zu unnatürlichen "Stoppschildern", was die Ergebnisse nicht signifikant beeinflusste. [2]

#### Walkin-In-Place

Walking-in-Place (WIP) ist eine Kategorie von Techniken, bei denen, wie der Name schon sagt, der stationäre Gang des Benutzers auf die Bewegung in der VE abgebildet wird [33, 42]. Eine umgeleitete Version von WIP verhindert, dass der Benutzer in einem dreiseitigen CAVE-Setup auf die fehlende Rückwand schaut [23]. WIP wird typischerweise über Sensoren oder Infrarotmarker implementiert, aber es wurde auch auf einem Wii Balance Board [47] und durch die Erkennung von Fußschatten auf dem Boden demonstriert [51]. Es wurde berichtet, dass WIP weniger realistisch ist als natürliches Gehen und zu Problemen im Zusammenhang mit Latenz und mangelnder Geschmeidigkeit während der Bewegung führt [6]. [2]

#### Abstahierte Schnittstelle des Gehens

"Virtuelles Fliegen" gehört zu den häufigsten und an den weitesten verbreiteten Techniken, bei denen die Manipulation des Blickpunkts über ein Eingabegerät erfolgt, z. B. einen Joystick oder eine Maus- und Tastaturschnittstelle [2]. Eine andere Variante dieses Konzepts, der "Human Joystick", ist eine von McMahan et al. für den Einsatz in CAVE-Systemen entwickelte Technik, bei der die Position des Benutzers in Bezug auf das Zentrum als Bewegungsvektor verwendet wird [18]. [2]

#### Laufbänder

Omnidirektionale Laufbänder unterstützen das Gehen in jede Richtung [35]. Sie erweitern zwar den virtuell begehbaren Raum und simulieren das Gehen besser, aber ihre Größe würde eine radikale Umgestaltung der häuslichen Umgebung erfordern, in der sie zum Einsatz kommen sollen. Laufbänder simulieren auch kein unebenes Terrain oder Hindernisse. [2]

#### (Bewegungsverhalten)

Im Unterschied zu den zuvor zitierten Arbeiten untersuchen wir in dieser Arbeit Faktoren, die das Verhalten der Benutzer beim Gehen in einer VE beeinflussen. Andere Forscher haben das Fortbewegungsverhalten durch die Analyse trajektographischer Maße und kinematischer Daten untersucht. Cirio et al. [4] haben ein Rahmenwerk von trajektographischen Kriterien vorgestellt, das den Realismus von virtuellen mit realen Trajektorien vergleicht. Das Rahmenwerk umfasst neun Metriken, die sich auf die Form, die Leistung und die kinematischen Merkmale beziehen. Bei uneingeschränkten zielgerichteten Aufgaben stimmen die Trajektorien in VE mit denen in einer realen Version der Aufgabe überein [4]. Fink et al. [7] fanden heraus, dass sich Trajektorien zur Hindernisvermeidung nur geringfügig zwischen einer VE und einer angepassten realen Umgebung unterscheiden. Warren und Fajen haben ein Modell der Lenkdynamik vorgeschlagen, das versucht darzustellen, wie Individuen eine Umgebung in Richtung eines Ziels navigieren würden, während sie stationäre oder bewegliche Hindernisse vermeiden [45]. Obwohl dieses Modell unter kontrollierten Bedingungen genau ist, reicht es nicht aus, um das Verhalten von Menschenmengen darzustellen [1]. Ruddle et al. [24] fanden heraus, dass natürliches Gehen die Benutzer dazu veranlasste, um Hindernisse herumzugehen, wohingegen Interfaces mit geringerer Originaltreue eine größere Inzidenz von Hindernisdurchdringungen aufwiesen. Sanz et al. [26] haben Trajektorien zur Hindernisvermeidung in einer 10 m langen immersiven Projektionsumgebung verglichen, in der die Teilnehmer realen physischen Hindernissen und Personen ausweichen mussten, sowie passenden virtuellen (nicht existierenden) Gegenstücken. Ihre Ergebnisse deuten darauf hin, dass die Teilnehmer dazu neigen, einen größeren Abstand (ca. 5 cm mehr) zu verwenden, wenn sie mit virtuellen Hindernissen umgehen als mit physischen. Fajen hat die Rolle der Affordanzwahrnehmung in VEs untersucht und gezeigt, dass Benutzer ihre Körpergröße und Handlungsmöglichkeiten berücksichtigen, wenn sie entscheiden, welche Route sie nehmen [5]. Zum Beispiel bei der Entscheidung, wie man sich bewegt, um einem Hindernis auszuweichen. Passive Haptik kann auch die Trajektorien der Teilnehmer beeinflussen. Im "Pit Room" [13] gab eine erhöhte Kante den Teilnehmern die Illusion, neben einer Grube zu stehen. Folglich folgten ihre Trajektorien der Kante. Das Hauptaugenmerk dieser Arbeit lag auf der Untersuchung, ob die Illusion überzeugend war. In unserer Arbeit sind wir daran interessiert, das Verhalten der Benutzer zu verstehen, wenn sie gezwungen sind, eine Navigationsentscheidung zu treffen, die auf dem Konflikt zwischen den visuellen Reizen der VE und ihrem Wissen über den realen Raum, in dem sie sich befinden, basiert. [2]

## Problemstellung, Zielsetzung und Vorgehensweise

### Problemstellung

Virtuelle Umgebungen zu Fuß zu erkunden ist die realistischste und natürlichste Schnittstelle. Zeitgleich ist dies auch technisch und logistisch am anspruchsvollsten.

Allgemein wurden in bisherigen Arbeiten sich auf die Fortbewegung in übereinstimmenden oder leeren Umgebungen konzentriert. Weniger wurde darauf geachtet, wie sich eine Änderung der Nichtübereinstimmung zwischen der physischen Umgebung und ihrer virtuellen Darstellung auf die Benutzer auswirkt

### Zielsetzung

* Konzipieren eines Szenarios, in dem das Bewegungsverhalten der Nutzer über festgelegte Reize in der VR beeinflussen zu können
* Evaluation, um die benötigten Daten zu sammeln und auszuwerten

### Vorgehensweise

* Szenario wird nach der Definition der Konditionierungsreize entwickelt
* Definition der Reize zur Konditionierung der Benutzer in VR (Malus)
* Schnittstelle zur Oculus Quest implementieren
* Auswahl der Erhebungsmethoden und nötigen Daten
* Evaluierung

### Technologien

Für die bestmögliche Umsetzung des Projekts werden verschiedene Technologien benutzt, die im Nachfolgenden erläutert werden.

#### Unity 3D

Unity 3D ist eine Laufzeit- und für Spiele und andere 3D und anderer interaktive 3D-Grafik-Anwendungen. Ebenso das meistgenutzte Tool im Bereich VR-Entwicklung. In Szenen werden die 3D Umgebungen erstellt. Diese besteht zum größten Teil aus Objekten, Materialien bzw. Texturen, Animationen, Audio und Lichtquellen. So entsteht eine virtuelle Welt. Der Android Support ist sehr umfangreich, was der Entwicklung mit der Oculus Quest zugutekommt. [14]

#### Oculus Quest

Bei der Oculus Quest handelt es sich um eine mobile VR Brille. Sie kann also genutzt werden, ohne zusätzlich einen leistungsstarken Computer oder Konsole zu benötigen. Ebenso werden keine externen Sensoren oder Kameras für das Tracking gebraucht.

Die Auflösung des Displays ist 1.600x1.440 Pixel pro Auge mit einer Wiederholrate von 72Hz. Das Tracking passiert über die Brille selbst. Auf dieser befinden sich an der Vorderseite 4 Kameras, die mit Weitwinkel arbeiten. Mit 6DoF, in Englisch Six Degrees of Freedom, ist die Nutzung aller sechs Freiheitsgrade möglich. Somit stehen viele Steuermöglichkeiten zur Verfügung und eine Bewegung im virtuellen Raum gegeben. Dafür elementar ist die Bedienung mit den zwei dazugehörigen Controllern (siehe Abb. 1: Oculus Quest mit Controller). [15]



Abb. 1: Oculus Quest mit Controller[[1]](#footnote-1)

#### Visual Studio

Visual Studio 2019 ist eine Entwicklungsumgebung mit vielen Tools und Dienste für Entwickler. Die eingesetzte Programmiersprache ist hierbei C#. Die Software eigenen Refactorings vereinfachen die Organisation des geschrieben Codes durch vorgeschlagene Aktionen. IntelliCode von Visual Studio gibt dem Entwickler Unterstützung mit dem Einsatz von künstlicher Intelligenz. Beinhaltend sind unter anderem die Bereitstellung von Codevervollständigung und die Suche nach Codeproblemen. Mit der integrierten Codebereinigung ist es möglich mit einem Klick Warnungen und Vorschläge zu ermitteln und zu beheben.

Mit Debugging kann präzise nach Laufzeitfehlern gesucht werden. [16]

#### Programmiersprache C#

C# ist eine objektorientierte und moderne Programmiersprache. Hier werden sogenannte Objekte zur Kapselung von Funktionalität und Daten genutzt. Dies wird erweitert durch die Möglichkeit der Vererbung und Polymorphie. [17] [18]

#### Git Versionskontrolle

Git ist eine kostenlose Open Source Software. Sie dient bei der Softwareentwicklung zur verteilten Versionskontrolle. Die Nutzung wird heutzutage in Unternehmen und von privaten Entwicklern genutzt. Durch die Plattformunabhängigkeit lässt sich Git in fast jeder Entwicklungsumgebung integrieren und nutzen. Die erste Version des Tools wurde im Jahre 2005 veröffentlicht.

Der Nutzen einer Versionskontroller besteht darin, Änderungen verteilt dem Softwareprojekt hinzufügen zu können. Diese werden zusätzlich protokolliert und sind somit nachvollziehbar. Es kann zu einem späteren Zeitpunkt auch auf frühere Versionen zugegriffen werden.

Die Verwaltung geschieht über dezentrale Git-Repositories, im Gegensatz zu anderer Software zur Versionskontrolle, welche mit einer zentralen Datenbank arbeiten. Jeder Entwickler, der an einem Projekt mit Git arbeitet, besitzt eine lokale Kopie des Repositories. Diese wird durch Programmieren weiterentwickelt und die Änderungen mit den anderen Entwicklern des Projekts geteilt.

Ein weiterer großer Bereich ist das Aufteilen und Zusammenführen von verschiedenen Entwicklungsständen. Hierbei kann man ein Softwareprojekt in Unterverzweigungen, den Branchen aufteilen. Dies wird vor allem genutzt, um parallel zum Hauptzweig an verschiedenen Features zu arbeiten. Zum Schluss werden die Branches wieder mit dem Hauptzweig zusammengeführt.

Es gibt viele Webanwendungen, welche Git nutzen und weitere Funktionen hinzufügen. Die Verwaltung von Softwareprojekten wird damit noch übersichtlicher, zugänglicher, transparenter und für andere Entwickler online verfügbar. Der Entwicklungsfortschritt kann grafisch und in Statistiken dargestellt werden. Auch ein eigenes Wiki über das Projekt kann für die Community erstellt und öffentlich zur Verfügung gestellt werden. Die Webseite GitHub ist ein Beispiel für solch eine Plattform. [19]

## Umsetzung

### Einbinden der Oculus Quest in Unity 3D

Um selbst entwickelte grafische Anwendung bzw. Spiele in Unity 3D ausführen oder testen zu können, muss eine aktive Verbindung von der Oculus Quest zum Computer und der Entwicklungsumgebung hergestellt werden. (Unity Version erzänzen)

Um die Verbindung zum PC erfolgreich herstellen zu können, wird die Software Oculus Link benötigt. Diese wird vom Unternehmen selbst online angeboten. Erforderlich ist ebenso das Einrichten der Oculus Quest Brille mit einem Facebook Account.

In Unity 3D ist es nötig, den Oculus Support einzurichten. Dazu gibt es aktuell noch zwei Möglichkeiten, welche ihre Vor- und Nachteile bieten. Das veraltete XR-Plugin Tool ist schon standardmäßig integriert und bietet dazu die Möglichkeit, OpenVR zu nutzen. Dies wird benötigt, wenn weitere VR-Brillen in der Entwicklungsumgebung genutzt werden sollen. Ein Beispiel ist die HTCVive. Da die Umsetzung einzig mit der Oculus Quest durchgeführt wird, fiel die Entscheidung auf das neue XR-Management. Dies befindet sich in ständiger Weiterentwicklung, somit wird in naher Zukunft ebenso OpenVR unterstützt werden. Dies könnte bedeutsam sein, falls das Projekt weiterentwickelt werden soll.

Ebenso benötigt wird das Asset Oculus Integration, welches sich über den integrierten Asset Store finden und importieren lässt. Darin befinden sich einige Prefabs und Skripte, die benötigt werden, um grundlegende Interaktionen wie das Umschauen und das Gehen in der Virtuellen Welt ermöglichen. Dazu muss das Prefab OVRPlayerController vom Assets Order per Drag and Drop in die Hierarchie der Szene gezogen werden. Zusätzlich wird das Skript CharacterCameraContraint benötigt, welches noch hinzugefügt werden muss.

Ist die Oculus Quest erfolgreich mit Oculus Link verbunden, kann die VR-Brille direkt in der Unity Entwicklungsumgebung über den Play Mode ausgeführt werden. Dies wird vor allem in der Entwicklung benötigt. Für die spätere Evaluation läuft die Anwendung jedoch Standalone, da ohne Kabel und angeschlossenen Computer ein freieres Bewegen der Probanden möglich ist. Dazu werden in den Grafikeinstellungen einige Änderungen vorgenommen und die Plattform wird zu Android gewechselt. Ebenso muss der Account als Developer freigeschalten werden und die Oculus Quest im Entwicklermodus laufen.

### Entwicklung des Malus

Dieser Teil wurde zunächst separat zum Hauptprojekt entwickelt und nach der Fertigstellung integriert. Ziel ist es, einen visuellen und akustischen Malus auszulösen, sobald der Proband vom Weg abkommt. Visuell sichtbar sein soll ein rotes, pulsierendes Blinken im kompletten Sichtfeld. Ebenso soll neben dem visuellen Reiz ein akustisches Piepen ertönen. Die Frequenz beide Stimuli ist zeitgleich und wird bei steigender Dauer schneller.

Für die Umsetzung in Unity 3D wurde das UI Objekt Canvas implementiert. Dies ist dazu da um Inhalte wie z.B. Bilder, Buttons für ein Menü oder Text anzeigen zu lassen. Eine Besonderheit in der Entwicklung einer VR-Anwendung im Vergleich zu einer normalen 3D Umgebung ist, dass das Canvas mit dem Kamera Objekt des OVRPlayerController verknüpft werden muss. Sonst besteht das Problem der fehlenden Anzeige in der VR-Brille.

Unter dem Canvas wurde ein Image Objekt erstellt und die Größe angepasst. Die Farbe wurde Rot wurde als Hintergrundfarbe eingestellt und der Alpha-Wert zunächst auf 0, sodass beim Start der Anwendung die Farbe erstmal nicht sichtbar ist. Für das Erzeugen des Impulses wurde eine Animation erstellt. Diese nimmt den Alpha-Wert des Images und verändert diesen, um so ein Impuls zu erzeugen. Ebenso wurde eine AudioSource Komponente dem Image Objekt hinzugefügt, die dazu benutzt wird, um die Audiodatei einbinden zu können. Diese wurde zuvor per Drag and Drop in die Assets des Projekts gelegt und dient neben dem visuellen zusätzlich noch einen akustischen Reiz. Um das Audio zeitgleich mit dem visuellen Impuls abspielen zu lassen, wurde ein Event PlayAudioSource in der Animation erstellt. Dazu wurde zusätzlich ein Skript dem Image Objekt hinzugefügt, welches das Event implementiert, die AudioSource Komponente aufruft und das Audio abspielt.

Um die Animation ausführen zu können, wird ein Animator benötigt. Dieser bindet die Erstellen Animationen ein und ist später die Schnittstelle, um die Animationen per Skript ausrufen und starten zu können.

Nach erfolgreichem Testen wurde der Canvas als Prefab, der Animator mit den entsprechenden Animationen und der AudioSource in das Hauptprojekt integriert. Der Aufruf des Malus wurde zum späteren Zeitpunkt bei der Entwicklung der Szenarien implementiert.

### Entwicklung der Szenarien

Der zentralste Bestandteil des Projekts ist das Entwickeln der Szenarien. In diesen soll der Proband in der virtuellen Umgebung einen Weg vom Start bis zum Ziel durchlaufen. Die Schwierigkeit dabei ist das sich verschiedene Hindernisse auf dem Weg befinden. Diese Umgebung ist in allen drei Szenarien gleich.

Zu Beginn des Weges bilden ein Karton und eine Kiste eine Gasse, durch die der Proband durchlaufen muss. Nach diesem Hindernis steht ein Tisch halb im Weg hinein, an dem man vorbeikommen muss (siehe Abb. 2: Hindernisse Kisten und Tisch).



Abb. 2: Hindernisse Kisten und Tisch

Daraufhin bilden zwei hohe Zäune eine Gasse und ein liegender Pfosten, quer liegend über dem Weg, die Hindernisse (siehe Abb. 3: Hindernisse Zaun und Pfahl). Bei diesen Gegenständen besteht entweder die Möglichkeit diese zu Umgehen oder durch Hindurchlaufen bzw. drüber steigen zu bewältigen.



Abb. 3: Hindernisse Zaun und Pfahl

Zum Erkennen, ob sich der Proband auf dem Weg befindet, wurden Box Collider implementiert. Zusätzlich dazu wurde ein Player Skript erstellt und als Komponente an den OVRPlayerController Objekt gehängt. Die darin in C# implementierte Methoden erkennt, ob der Proband den Weg betritt oder verlässt. Befindet sich der Proband auf dem Weg und verlässt diesen, wird der akustische und visuelle Malus über den Zugriff auf den Animator aktiviert. Ebenso wird hier auf das DataRecorder Skript zugegriffen. Beim erstmaligen Betreten der Collider wird der Timer des Spiels gestartet. Sobald ein Malus aktiviert wird, wird darüber ebenso der Timer des Malus gestartet und beim wieder betreten des Weges gestoppt.

Ein weiterer Box Collider befindet sich am Ende des Weges. Sobald der Proband diesen betritt, wird die Zeit des Szenarios gestoppt und er bekommt mit einem grünen optischen Impuls und einem Audio signalisiert, dass das Ziel erreicht wurde.

Den Szenaren unterscheiden sich grundsätzlich im zu laufenden Weg. In einem dieser Szenarien stimmt der visuelle mit dem physischen Weg überein. Im visuellen Szenario unterscheidet sich der visuelle, richtige Weg vom physischen Weg. Das erste Szenario ich gleichzusetzen mit dem Physischen, unterscheidet sich jedoch im fehlenden negativen Malus.

Zum Start des Szenarios wurde ein Menü erstellt, über welches entweder ein explizites Szenario oder über den Menüpunkt Start das Spiel mit einer zufälligen Auswahl gestartet werden kann (siehe). Zur Interaktion mit dem Menü kann der linke oder rechte Controller genutzt werden und die Auswahl wird mit der Trigger Taste bestätigt. Für die Implementierung wurde ein Canvas verwendet. Unter diesem wurden die Menüpunkte als Buttons hinzugefügt. Die Funktionalität der Buttons wurde mit dem selbstentwickelten Skript MenuHandler implementiert, welches alle benötigten Methoden enthält um die verschiedenen Szenarien starten zu können. Die Zuweisung der richtigen Methode für jeden Button wurde im OnClick() Bereich des jeweiligen Inspectors zugewiesen. Für die Benutzung der Controller wurde das Prefab UIHelpers aus dem Oculus Integration Asset hinzugefügt. Dies beinhaltet zum einen Objekt LaserPointer, welches dafür zuständig ist den Laser Pointer zum Anvisieren der Buttons zu visualisieren. An selber Stelle befindet sich zum anderen das Objekt EventSystem, was die Funktionalität des Laserpointers und Controller implementiert. Hier lässt sich unter anderem der Controller bzw. Laserpointer auswählen und welche Taste des Controllers zur Eingabe genutzt werden soll.

### Datenerfassung

Für die Evaluation wurden zwei verschiedene Arten von Daten vorgesehen. Quantitative Daten, die zur Laufzeit der Anwendung entstehen, und qualitative Daten durch die spätere Befragung der Probanden mit Hilfe eines Online Fragebogens.

Die Daten, die automatisch von Unity während der Laufzeit der Szenarien getracked werden sind:

* welches Szenario aktiv war
* Zeit, die der Proband braucht, um den kompletten Weg bis zum Ziel zu laufen
* die Anzahl der Malus, die aktiviert wurde
* die Dauer pro Malus, als wie lange der Proband gebraucht hat, um wieder auf den richtigen Weg zurück zu kommen

Dazu wurde das Skript DataRecorder erstellt und an das OVRPlayerController Objekt hinzugefügt. In diesem Skript wurden mit C# Methoden erstellt, um die erforderlichen Daten, wie zuvor erläutert, aufzuzeichnen und zu speichern. Die Methoden werden durch das Player Skript aufgerufen.

Das Skript CSVManager ist für den Export der aufgezeichneten Daten in eine CSV Datei zuständig. Dies ist für das Auslesen und weitere Auswertung der Daten entscheidend. Die Datei Report.csv wird lokal im Assets Ordner Report erstellt bzw. abgelegt. Für den Fall, das Daten erhoben und abgespeichert werden sollen, die CSV Datei aber noch nicht existiert, wird diese und der dazugehörige Ordner automatisch erstellt. Dazu werden standardmäßig folgende Header in gleicher Reihenfolge erstellt:

* Szenario
* Gebrauchte Zeit
* Malus aktiviert

Sobald ein Malus öfter als einmal aktiviert wurde, werden die dazugehörigen Header für die Dauer der einzelnen Malus dynamisch hinzugefügt. Ein Beispiel der CSV Datei mit gefüllten Daten zeigt sich in Abb. 2: CSV Datei mit Beispieldaten.



Abb. 4: CSV Datei mit Beispieldaten

Es wurde ein Online Fragebogen erstellt, um zusätzlich qualitative Daten zu erheben. Für die Umsetzung wurde EvaSys verwendet. Hierbei handelt es sich um eine webbasierte Feedback Management Lösung. Online Umfragen, klassische Umfragen in Papierform oder selbst Hybridumfragen sind möglich. [6] In unserem Fall setzen wir auf eine reine Online-Umfrage, da eine Papier- oder Hybridumfrage keine Vorteile bietet und das Auswerten mit mehr Aufwand zusammenhängt.

Die Fragen der Online Umfrage wurde in verschiedene Fragegruppen zusammengefasst, auf die im folgenden Abschnitt im Einzelnen genauer eingegangen wird. Als erstes geht es um die persönlichen Daten des Probanden (siehe Abb. 3: Fragebogen - Persönliche Daten). Hier wird das Geschlecht und das Alter als Angaben gemacht. Diese wurden als Single Choice Fragen umgesetzt und das Alter in Gruppen unterteilt, was das empirische Auswerten unkomplizierter und übersichtlicher gestaltet.



Abb. 5: Fragebogen - Persönliche Daten

In der zweiten Fragengruppe geht es um allgemeine Angaben der Probanden (siehe Abb. 4: Fragebogen - Allgemeine Angaben). Die Erfahrungen in Bereich Gaming und Virtual Reality werden jeweils in einer Skala von 1 bis 5 abgefragt. Wobei 1 keine Erfahrung und 5 viel Erfahrung bedeutet. Die Anzahl der Ankreuzfelder wurden im kompletten Fragebogen konsistent gehalten. In der dritten Frage geht es um die Art der VR-Technologien, mit welche der Proband schon in Erfahrung getreten ist. Hierbei handelt es sich um eine Multiple Choice Frage, da eine oder mehrere Antworten möglich sind.



Abb. 6: Fragebogen - Allgemeine Angaben

Die nächste Gruppenfrage bezieht sich explizit auf die Durchführung der Szenarien (siehe Abb. 5: Fragebogen - Beurteilung Szenario). Für jedes der drei Szenarien werden dieselbe Art Fragen benutzt, um die Auswirkungen auf den Probanden zu ermitteln. Die Skala im Wert von 1 bis 5 ist hier ebenso, wie zuvor erläutert, die Art der Antwortmöglichkeiten. Der Wert 1 bedeutet Dazu gehört die Natürlichkeit des Laufens, die Projektion des Denkens und Handelns in der virtuellen Umgebung. Das Bewusstsein der physischen Umgebung ist ebenso ein Teil der Fragestellung. Die letzte Frage geht explizit auf den Einfluss des negativen Malus auf den Probanden ein.



Abb. 7: Fragebogen - Beurteilung Szenario

Die letzte Fragegruppe ergibt ein offenes und persönliches Feedback (siehe Abb. 6: Fragebogen - Persönliches Feedback). Der Frage Typ hierbei sind offene Fragen mit unbegrenzter Anzahl an Zeichen.

Hier geht es unter anderem darum, was die Probanden besonders positiv und negativ an der Durchführung der Szenarien fanden. Ein weiterer Punkt ist, was genau die Probanden am negativen Malus beeinflusst hat. Zum Abschluss können konkrete Verbesserungsvorschläge gebracht werden, was besonders Relevant für die Weiterentwicklung des Projektes ist.



Abb. 8: Fragebogen - Persönliches Feedback

Nach der Datenerfassung mit Hilfe des Fragenbogens können die Daten direkt in EvaSys dargestellt und ausgewertet werden.

### Evaluation

Bei der Ankunft in der Studie (siehe Abb. 1 Die Teilnehmer erhielten einen kurzen Überblick über die Studie, der Informationen zum Oculus Rift-Headset und eine allgemeine Beschreibung der Konditionierungsaufgabe enthielt. Nach Einverständniserklärung wurde den Teilnehmern die Verwendung

des Videospiel-Controllers erklärt. (Aufgabenbeschreibung)

## Zusammenfassung, Bewertung und Ausblick

## Anhang

### Abkürzungsverzeichnis

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### Abbildungsverzeichnis

[Abb. 1: Oculus Quest mit Controller 8](#_Toc65407026)

[Abb. 2: CSV Datei mit Beispieldaten 12](#_Toc65407027)

[Abb. 3: Fragebogen - Persönliche Daten 13](#_Toc65407028)

[Abb. 4: Fragebogen - Allgemeine Angaben 13](#_Toc65407029)

[Abb. 5: Fragebogen - Beurteilung Szenario 13](#_Toc65407030)

[Abb. 6: Fragebogen - Persönliches Feedback 14](#_Toc65407031)

### Literaturverzeichnis

|  |  |
| --- | --- |
| [1] | J. N. Templeman, D. P. S. und L. E. Sibert, „Virtual Locomotion: Walking in Place through Virtual Env ironments,“ Dezember 1999. |
| [2] | A. L. Simeone, I. Mavridou und W. Powell, „Altering User Movement Behaviour in Virtual Environments,“ April 2017. |
| [3] | P. Fink, P. Foo und W. H. Warren, „Obstacle avoidance during walking in real and virtual environments,“ Januar 2007. |
| [4] | S. Razzaque, D. Swapp, M. Slater, M. C. Whitton und A. Steed, „Redirected Walking in Place,“ 2002. |
| [5] | M. C. W. Kroes, J. E. Dunsmoor, W. E. Mackey, M. McClay und E. A, „Context conditioning in humans using commercially available immersive Virtual Reality,“ 2017. |
| [6] | [Online]. Available: https://evasys.de/evasys/. [Zugriff am 28 02 2021]. |
| [7] | J. N. Templeman, P. S. Denbrook und L. E. Sibert, „Virtual locomotion: Walking in place through virtual environments,“ Dezember 1999. |
| [8] | D. A. Bowman, D. Koller und L. F. Hodges, „Travel in Immersive Virtual Environments: An Evaluation of Viewpoint,“ 1997. |
| [9] | M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater und F. P. Brooks, „Walking > Walking-in-Place > Flying, in Virtual Environments,“ 1999. |
| [10] | R. A. Ruddle und S. Lessels, „The Benefits of Using a Walking Interface,“ April 2009. |
| [11] | R. A. Ruddle, E. Volkova und H. H. Bülthoff, „Learning to Walk in Virtual Reality,“ Mai 2013. |
| [12] | M. C. Whitton, J. V. Cohn, J. Feasel, P. Zimmons, S. Razzaque, S. .. Poulton, B. McLeod und F. P. Brooks, Jr., „Comparing VE Locomotion Interfaces,“ März 2005. |
| [13] | E. A. Suma, S. L. Finkelstein, M. Reid, S. V. Babu, A. C. Ulinski und L. F. Hodges, „Evaluation of the Cognitive Effects of Travel Technique in Complex Real and Virtual Environments,“ 2010. |
| [14] | [Online]. Available: https://unity.com/de. [Zugriff am 23 02 2021]. |
| [15] | D. Ziesecke, „Alle Infos zur Oculus Quest – technische Daten, Preis und Laufzeit,“ 03 03 2019. [Online]. Available: https://vr-legion.de/news/alle-infos-zur-oculus-quest-technische-daten-preis-und-laufzeit/. [Zugriff am 23 02 2021]. |
| [16] | TerryGLee, „Neues in Visual Studio 2019,“ 10 11 2020. [Online]. Available: https://docs.microsoft.com/de-de/visualstudio/ide/whats-new-visual-studio-2019?view=vs-2019. [Zugriff am 22 02 2021]. |
| [17] | B. Wagner, „Überblick über C#,“ 28 01 2021. [Online]. Available: https://docs.microsoft.com/de-de/dotnet/csharp/tour-of-csharp/. [Zugriff am 23 02 2021]. |
| [18] | O. B. Dr. Veikko Krypczyk, „Einführung in die Programmierung: Objekte, nichts als Objekte,“ 19 04 2016. [Online]. Available: https://entwickler.de/online/development/einfuehrung-programmierung-objektorientierte-programmentwicklung-197372.html. [Zugriff am 24 02 2021]. |
| [19] | S. Augsten, „Definition „Git SCM“ - Was ist Git?,“ 27 08 2019. [Online]. Available: https://www.dev-insider.de/was-ist-git-a-850847/. [Zugriff am 24 03 2021]. |
| [20] | C. Spinger, „Oculus Quest im Test: Virtual Reality für die Masse,“ 30 04 2019. [Online]. Available: https://www.vrnerds.de/oculus-quest-im-test-virtual-reality-fuer-die-masse/. [Zugriff am 27 02 2021]. |
| [21] | A. L. Simeone, E. Velloso und H. Gellersen, „Substitutional Reality: Using the physical environment to design virtual reality experiences,“ Januar 2015. |

### Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Heilbronn, 19. September 2016 Vorname Name

1. https://www.vrnerds.de/wp-content/uploads/2019/04/Oculus-Quest-2-1.jpg [20] [↑](#footnote-ref-1)