



Prototypische Entwicklung einer körperbasierten Mensch-Computer- Interaktion für Virtual Reality unter Berücksichtigung ergonomischer Faktoren

vorgelegt von:
Ravell Heerdegen
761330

Emailadresse:
ravell.heerdegen@student.reutlingen-university.de

Bachelor in Medien- und Kommunikationsinformatik
SS2019

Erstbetreuerin Prof. Dr. Gabriela Tullius
Zweitbetreuer Prof. Dr. Uwe Kloos
Betriebliche Betreuung Manuel Dixken

Eingereicht am: 15.06.2019

Abstract

Dies wird der Abstract und kommt vor dem Inhaltsverzeichnis

Inhaltsverzeichnis

1 Einleitung	4
1.1 Kontext und Motivation	4
1.2 Ziele der Arbeit	4
2 Konzeption und Entwicklung	5
2.1 Grundlagen	5
2.1.1 Virtual reality	5
2.1.2 Head-mounted displays	6
2.1.3 Eingabegeräte	7
2.1.4 Ergonomie	8
2.1.5 3D Benutzeroberflächen	10
2.1.6 Körperbasierte Interfaces	11
2.2 Hinführung zum Konzept	12
2.2.1 Anwendungsanforderungen	12
2.2.2 Richtlinien	12
2.2.3 Interfacepositionierung	14
2.2.4 Verwandte Ideen und Konzepte	15
2.2.5 Interfaceanforderungen	17
2.2.6 Finales Interfacekonzept	18
2.3 Implementation	19
2.3.1 Verwendete Frameworks und Tools	19
2.3.2 Umsetzung	20
2.3.3 Probleme und Hindernisse	24
2.4 Evaluation	25
2.4.1 Aufbau	25
2.4.2 Durchführung	26
2.4.3 Diskussion	27
3 Schluss	28
3.1 Fazit	28
3.2 Zukunftsaussichten und Pläne	28

Abbildungsverzeichnis

1 Virtual-Reality Anwendung SwordPlay [Jr et al., 2010]	6
2 HTC Vive Headset	6
3 HTC Vive Controller	8
4 3D Dateimanager [Kyritsis et al., 2013, p. 243]	10

5	Anwendung von OmniTouch [Harrison et al., 2011, p. 446]	11
6	Wheelpicker-Modell nach [Ehrlich, 2017, p. 33]	15
7	Interaktion mit Wheelpicker-Modell nach [Ehrlich, 2017, p. 33]	16
8	Pie-Modell nach [Ehrlich, 2017, p. 10]	16
9	Interaktion und Erweiterung des Pie-Modells nach [Ehrlich, 2017, p. 10]	16
10	Interaktion und Erweiterung des <i>3d button menu</i> nach [Ehrlich, 2017, p. 36-37]	17
11	Interaktion mit Kachel-Modell	17
12	Finales Konzept <i>decision menu</i>	18
13	Erweiterung des <i>decision menu</i>	19
14	Platzierung anwendungsspezifischer Funktionen	19
15	Interfaceimplementation	22
16	Teleportationsfunktion	22
17	Markieren und Selektieren eines Würfels	23
18	Verwendung der Translationsfunktion	23
19	Rotieren eines Objektes	23
20	Verwendung der Skalierungsfunktion	24
21	Einblendung eines Funktionstitels bei blinder Bedienung	24
22	Angepasster virtueller Raum für Evaluation	26

Tabellenverzeichnis

1	Eingabearten und Systeme	9
2	Anwendungsanforderungen zufällig sortiert	13
3	Vor- und Nachteile verwandter Interface-Designs	17
4	Interfaceanforderungen zufällig sortiert	18
5	System Usability Scale Fragen nach Reihenfolge sortiert	26

1 Einleitung

Dies wird die Einleitung

1.1 Kontext und Motivation

1.2 Ziele der Arbeit

Primärziel dieser Arbeit ist ein körperbasiertes Interface für eine VR-Anwendung zu entwickeln, welches durch kompatible Eingabe- und Ausgabegeräte bedient werden kann. Als Ausgabegerät für Informationen soll ein *head-mounted display* dienen. Die Eingabe soll über Controller geschehen. Ein weiteres Ziel ist, der zu entwickelnden VR-Anwendung einen CAD-Ansatz zu geben. Der Benutzer soll somit die Möglichkeit haben, virtuelle Objekte erstellen, manipulieren und löschen zu können. Das Interface soll unter ergonomischen Gegebenheiten entwickelt und praktisch verwendet werden können. Dafür sollen Nebeneffekte und Designrichtlinien bei und für die Verwendung und Implementation von 3D-User Interfaces berücksichtigt werden. Auch sollen verwandte Arbeiten mit in den Design- und Entwicklungsprozess einfließen, verglichen werden und die Entwicklung des Interfaces beeinflussen. Schließlich soll eine Evaluation die Benutzungsfreundlichkeit und Bedienung des Interfaces und der Anwendung überprüfen und entsprechend nachweisen.

2 Konzeption und Entwicklung

2.1 Grundlagen

Seit dem Beginn der Entwicklung von Lösungen für *Virtual Reality* (VR) in den fünfziger Jahren des zwanzigsten Jahrhunderts, gab es stetig neue und innovative Konzepte, um nicht nur mit, sondern auch in der virtuellen Welt zu interagieren [Novak et al., 2014]. In diesem Kapitel wird zunächst erläutert, um was es sich bei VR genau handelt. Danach wird näher auf *head-mounted displays* (HMDs) eingegangen, welche nach dem ersten Prototypen aus dem Jahre 1968 kontinuierlich weiterentwickelt wurden und heute ein vielversprechendes Ein- und Ausgabegerät für Informationen darstellen [Lamman et al., 2014]. Daran anschließend folgen grundlegende Informationen zu Eingabegeräten wie *tracking devices* oder *vision-based devices*, welche ebenfalls in der Konzeptions-Phase für das zu entwickelnde Interface relevant waren. Des Weiteren werden Details zur Ergonomie aufgeführt, zusammen mit Möglichkeiten und Grenzen durch den menschlichen Körper. Abschließend wird auf *3D user interfaces* (3DUIs) und *body-based interfaces* (BIs) eingegangen, wodurch der Rahmen für das Grundverständnis dieser Arbeit gegeben wird.

2.1.1 Virtual reality

VR erfährt seit der technischen Einführung im Jahre 1968 ein andauerndes Interesse. Sowohl in der Erforschung neuer Meilensteine als auch in der Verwendung der bereits vorhandenen Methoden und Mittel im Alltag. Grundlegend ist VR eine interaktive Simulation, bei welcher die menschlichen Sinne verstärkt oder manipuliert werden, um während der Benutzung mit einem Computer, einem Benutzer das Gefühl zu geben, tatsächlich in einer virtuellen Umgebung zu sein [Novak et al., 2014]. Einerseits kann der gesamte Körper des Benutzers in die virtuelle Umgebung übertragen werden, andererseits reicht es teilweise aus, bestimmte Körperteile wie z.B. die Hände [Mendes et al., 2017] oder die Finger [Riecke et al., 2018a] virtuell abzubilden. Die virtuelle Umgebung besteht dabei aus verschiedenen Objekten, welche untereinander in Beziehung stehen können. Wahrgenommen werden können die platzierten Objekte visuell, auditiv oder haptisch [Novak et al., 2014]. Das bedeutet, dass Objekte entweder der reinen Visualisierung dienen, z.B. ein Foto in einem Raum, die Akustik unterstützen, z.B. Windgeräusche, oder Interaktionen zulassen wie z.B. ein Würfel, welcher geworfen werden kann. Dadurch soll dem Benutzer ein immersives Gefühl und eine sogenannte physische als auch mentale Präsenz vermittelt werden [Zhou and Deng, 2009].

Physische Präsenz bedeutet, dass der Benutzer das Gefühl bekommt, er sei mit seinem Körper in der virtuellen Welt. Mentale Präsenz hingegen bedeutet Ansprüche und Erwartungen an die virtuelle Umgebung zu stellen, und somit ein Teilhabe-Gefühl zu entwickeln. Die mentale Präsenz kann dabei unterschiedliche Stadien annehmen.

Von einer puren Verbundenheit mit einem Computer, bis hin zum Gefühl, dass die virtuelle Welt real ist. Je nach Anwendung müssen physische und mentale Präsenz einen bestimmten Grad erreichen. Dafür notwendig ist das sensorische Feedback. Feedback wird größtenteils visuell vermittelt, z.B. durch das Einfärben eines Objektes bei einer Selektion. Feedback kann jedoch auch haptisch oder auditiv übertragen werden. Ebenfalls wichtig, ist die Perspektive mit der die virtuelle Umgebung wahrgenommen wird. Die Erste-Person Ansicht spiegelt die Sicht eines Menschen in der Realität wider, möglich sind jedoch auch die Zweite-Person Ansicht, sowie die Dritte-Person Ansicht. Letztere auch bekannt als Vogel- oder ISO-Perspektive, dient dem Überblick einer Situation oder Umgebung aus der Ferne [Novak et al., 2014].



Abbildung 1: Virtual-Reality Anwendung Sword-Play [Jr et al., 2010]

2.1.2 Head-mounted displays

HMDs sind Geräte, welche auf oder an dem Kopf eines Benutzers platziert werden [Zhou and Deng, 2009]. HMDs unterscheiden sich u.a. in der Qualität der Auflösung, Farbdarstellung, Sichtfeldweite, Interfacedarstellung sowie Beleuchtung [Lakshminarayanan and Moroi, 2017]. Aktuelle Geräte wie die Oculus Rift oder HTC Vive, letztere zu sehen in Abbildung 2, verwenden

stereoskopische Anzeigen und Aufzeichnungssysteme, um die Kopfbewegungen eines Benutzers zu erfassen und dreidimensionale Bilder darstellen zu können. Um die genaue Position des Kopfes des Benutzers im virtuellen Raum berechnen zu können, kommen Beschleunigungssensoren und Gyroskope zum Einsatz, welche in die HMDs direkt integriert werden [Zhou and Deng, 2009].



Abbildung 2: HTC Vive Headset

Durch HMDs können u.a. kognitive, psychomotorische oder affektive Fähigkeiten erlernt und verbessert werden. Im Falle von kognitiven Fähigkeiten kann z.B. das Erinnerungsvermögen verbessert werden [Jensen and Konradsen, 2018]. Um eine positive Erfahrung während der Benutzung eines HMDs zu gewährleisten, ist die sogenannte *Field of View* (FOV) entscheidend. Exemplare welche vor dem Jahre 2013 erschienen sind, besaßen einen Sichtradius von 25 bis 60 Grad. Der natürliche Sichtradius eines Menschen befindet sich bei 180 Grad. Aktuelle Geräte wie die HTC Vive erreichen eine FOV von über 100 Grad, was sich positiv auf die Erfahrung und dem damit verbundenen Realitätsgefühl auswirkt. Ebenfalls nennenswert ist der Begriff *motion-* bzw. *cybersickness*. Darunter wird ein Übelkeits- oder Schwindelgefühl verstanden, welches bei der Verwendung von HMDs in Verbindung mit VR auftreten kann. *Motionsickness* kann die Immersion beeinträchtigen und dadurch negative Erfahrungen mit HMDs und VR begünstigen. Auch kann *motionsickness* vom Alter, der Erfahrung und des Geschlechts des Benutzers abhängen, was durch diverse Studien nachgewiesen wurde [Jensen and Konradsen, 2018]. Durch die stetige Weiterentwicklung von HMDs, finden die selbigen zunehmend Anwendung in den Bereichen Bildung [Jensen and Kon-

radsen, 2018], Psychologie [Maples-keller et al., 2018], Medizin [Lakshminarayanan and Moroi, 2017] oder in der 3D-Modellierung [Vajak, 2017].

2.1.3 Eingabegeräte

Eingabegeräte sind auch in VR erforderlich, um mit virtuell erstellten Umgebungen interagieren zu können. Je nach Anwendungskontext und Aufgabenstellung können unterschiedliche Eingabegeräte zum Einsatz kommen. Nach [Ortega et al., 2016] kann jedes Eingabegerät einer vordefinierten Kategorie zugeordnet werden. Ortega et al. definieren dazu elf Kategorien. Für diese Arbeit relevant aufgrund der Möglichkeit das Eingabegerät am Körper zu tragen, waren die Kategorien *inertial-sensing*, *data gloves*, *vision-based devices* und *tracking devices*. Bei *inertial-sensing* werden verschiedene Funktionen durch Druck- bzw. Trägheitserfassung ausgelöst. [Harrison et al., 2010] haben dafür ein System entwickelt, welches durch Druck durch einen oder mehrere Finger bedient werden kann. Je nach Druckstärke und Druckpunkt, können damit die im System verbauten Vibrationssensoren angeregt und somit unterschiedliche Interaktionen durchgeführt werden. *Vision-based devices* hingen nutzen Projektoren und Sensoren, um z.B. Gesten oder die Umgebung zu erfassen, um daraus Eingabe- bzw. Ausgabeinformationen zu erstellen. Als Beispiel dient das durch Gesten gesteuerte System SixthSense von [Mistry et al., 2009]. Die am Kopf des Benutzers befestigte Kamera erfasst die Hände und die dargestellten Gesten. Der am Kopf befestigte Projektor wirft Informationen und Interaktionselemente auf Oberflächen aus der Umgebung. Ein weiteres Beispiel für ein auf Gesten und Projektoren basierendes System ist LightSpace [Wilson and Benko, 2010]. Bei LightSpace kommen jedoch im Vergleich zu SixthSense keine Marker an Fingern oder Umgebung vor. Ein durch einen *data glove* gesteuertes System wurde von [Jofre et al., 2016] entwickelt. Der Handschuh besitzt für jeden Finger eigene Sensoren, welche nicht nur die Bewegungen und dessen Geschwindigkeit, sondern auch die Orientierung der Finger und Hand messen und erfassen können. Im Vergleich zu den *vision-based devices*, kommen beim *data glove* keine Projektoren zum Einsatz, da die Informationen direkt vom Handschuh in die an einem Computer laufenden Anwendungen übertragen werden. Unter die *tracking devices* fallen u.a. *finger-tracking*, *hand-tracking*, *head-tracking*, *body-tracking* oder *controller-tracking devices*. Ein System, welches *head-tracking* verwendet, um 3DUIs darzustellen, ist das Projekt *the personal cockpit* von [Ens et al., 2014]. Das von Ens et al. entwickelte System, umfasst ein HMD in Form einer Brille und darin verbaute Sensoren. Die verbauten Sensoren erfassen die Richtung in welche der Träger der Brille schaut, und stellt daraufhin verschiedene Interaktionselemente oder Informationen dar. Ein populäres

Eingabegerät, welches mit *controller-tracking*, VR und HMDs in Verbindung gesetzt wird, ist z.B. der HTC Vive Controller, zu sehen in Abbildung 3.



Abbildung 3: HTC Vive Controller

Durch sogenannte *Positioner*, welche im Raum um den Benutzer herum platziert werden, werden nicht nur die Position und Rotation der Controller, sondern wenn gewünscht der gesamte Körper eines Benutzers erfasst. Knöpfe und Touchpads dienen als Eingabemechanismen und können zusammen mit Bewegung und Rotation der Controller sogar Gesten abbilden und so mit einer 3D-Umgebung interagieren [Lee et al., 2017]. Aufgrund der zahlreichen Einsatzmöglichkeiten, der für VR ausgelegten Hardware, Gestenerkennung und *body-tracking*, sowie der unmittelbaren Verfügbarkeit boten sich die HTC Vive Controller in Kombination mit dem HTC Vive Headset für diese Arbeit am Besten an. Tabelle 1 listet die beschriebenen Eingabearten und dazugehörigen Systeme zusammengefasst auf.

2.1.4 Ergonomie

Um mit virtuellen Objekten oder Umgebungen interagieren zu können, werden u.a. Interfaces benötigt, welche je nach Anwendung unterschiedlich in die 3D-Umgebung eingebunden werden. Dabei sollten nicht nur die Art der Interaktionen, sondern u.a. auch der menschliche Körper mit seinen ergonomischen Möglichkeiten und Limitierungen berücksichtigt werden. Nur somit ist es möglich, Interfaces zu entwickeln, welche komfortabel und effektiv genutzt werden können. Unterschiedliche Interaktionen erfordern unterschiedliche Bewegungen, welche wiederum unterschiedlich komplex und andauernd sein

Eingabeart	Beispielsystem
<i>inertial-sensing</i>	Skinput
<i>data gloves</i>	DG5 Vhand Data Glove 2.0
<i>vision-based devices</i>	SixthSense, LightSpace
<i>tracking devices</i>	The personal cockpit, HTC vive controller

Tabelle 1: Eingabearten und Systeme

können. Die Interaktionen haben dabei verschieden stark ausgeprägte Attribute wie Genauigkeit, Geschwindigkeit, Richtung und Dauer. Aufgrund der menschlichen Physiologie können bestimmte Interaktionen länger und präziser oder nur bedingt und umständlich umgesetzt werden, was bei der Implementierung von 3DUIs in VR zu beachten ist [LaViola et al., 2017]. Die menschliche Hand z.B. bietet durch ihre praktikable Ergonomie eine umfangreiche Palette an Einsatzmöglichkeiten. Darunter das Greifen oder Halten eines Gegenstandes, oder die Möglichkeit Gesten und Posen darzustellen. Ein unangenehmes Maß an Kraft, Dauer und Wiederholung einer Aufgabe, kann sich negativ auf die Umsetzung auswirken, gleich wie gut ein Interface entwickelt wurde. Ein weiterer wichtiger Aspekt in der Gestaltung von 3DUIs unter Berücksichtigung der Ergonomie des Menschen, ist das sogenannte *Gorilla arm syndrome*. Darunter wird das Ermüden der Arme durch lang andauernde und unbequeme Posen verstanden. Interfaces, welche auf Augenhöhe oder sogar darüber platziert sind, begünstigen den Gorilla-Arm Effekt. Darum sollten Interfaces möglichst niedrig gehalten oder Interaktionen näher an den Körper gebracht werden. Das Greifen eines weit entfernten Objektes könnte bspw. durch eine Pointer-Technik ermöglicht werden, um das Ausstrecken des Armes zu vermeiden [LaViola et al., 2017; Hincapie-Ramos et al., 2014]. Ein weiteres Anzeichen für Ermüdung ist ein Haltungswechsel. Deshalb sollten Posen bevorzugt werden, welche komfortabel und energiesparend durchgeführt werden können und somit, die Bequemlichkeit und Leichtigkeit unterstützen. Interfaces, welche sich am Arm oder auf der Hand befinden, haben dabei in Studien großen Zuspruch gefunden, da sich Arm- bzw. Handinterfaces ergonomisch günstig anbieten und viel Raum als auch Möglichkeiten für Interaktionen bieten [Harrison and Faste, 2014]. Komfort ist jedoch nicht gewährleistet unter reiner Berücksichtigung der aufgeführten

Aspekte. Komfort ist ebenfalls bedingt, durch die Darstellung der Interfaces, bezogen auf Form und Gestalt [LaViola et al., 2017].

2.1.5 3D Benutzeroberflächen

3DUIs erfahren seit der Alltagstauglichkeit von VR stetig wachsenden Zuspruch und Anwendung. 3DUIs unterscheiden sich in erster Linie von zweidimensionalen UIs durch die gegebenen *degree of freedom* (DOF) [Weiß et al., 2018; Mendes, 2018; LaViola et al., 2017]. Die DOF geben an, inwieweit in einer virtuellen Umgebung mit virtuellen Objekten interagiert werden kann, bzw. wie stark die Operationsvielfalt im virtuellen Raum ausgeprägt ist. Bei 2DUIs beschränkt sich die Interaktion auf Operationen in x und y Richtung. Das bedeutet, dass 2DUIs wie z.B. ein Dateimanager und dessen Elemente nur in der x und y Richtung positioniert werden können [Kyritsis et al., 2013]. Bei 3DUIs hingegen kommt die Tiefe durch die z-Achse hinzu, wodurch sowohl dreidimensionale Manipulation als auch Rotation ermöglicht werden [Jr et al., 2010]. Die dazugekommene Tiefe kann jedoch auch zu höheren Fehlerraten in der Ausführung von Aufgaben führen, bspw. beim Selektieren eines Objektes, welches sich dicht neben und kaum hinter einem anderen Objekt befindet. Die Bevorzugung von 3DUIs gegenüber 2DUIs hängt demnach von der Anwendung und dessen Kontext ab. [Kyritsis et al., 2013] haben bspw. nachgewiesen, dass sich ein 3D-Dateimanager, aufgezeigt in Abbildung 4, in Verbindung mit VR nur anbietet, wenn die Darstellung der Dateien durch große und auffällig eingefärbte Symbole gegeben ist.

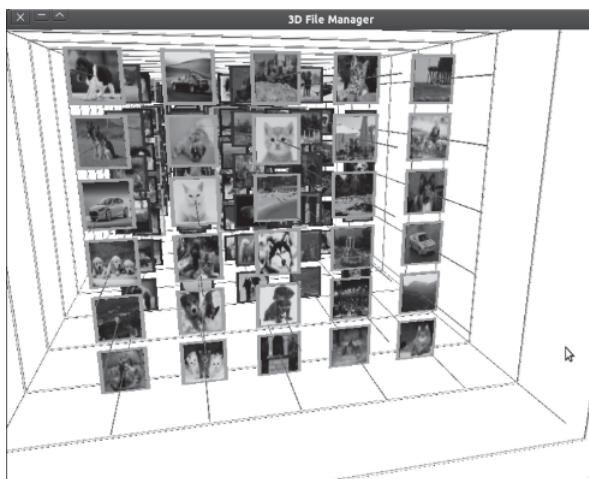


Abbildung 4: 3D Dateimanager [Kyritsis et al., 2013, p. 243]

Eine Eigenschaft die 3DUIs gegenüber 2DUIs mit sich bringen, ist die Vielfalt der Darstellung und Interaktion. So können 3DUIs bspw. frei in der virtuellen Umgebung platziert [Jr et al., 2010], und/oder durch Gesten [LaViola, 2014] gesteuert werden. Die Komplexität von 3DUIs reicht von Interfaces welche sich bei Kopfdrehungen während des Tragens eines HMD mit bewegen, um die Position des 3DUI zu ändern, bis hin zu komplexen Modellierungs- oder Designinteraktionen. Eingesetzt werden 3DUIs u.a. auch in den Bereichen Videospiele, Unterhaltung, Simulation, Bildung oder der Medizin [LaViola et al., 2017]. Da sich 3DUIs so unterschiedlich komplex und vielfältig im Design ausprägen können, wirken sich die selbigen auch unterschiedlich auf die *user experience* (UX) aus, welche der Benutzer beim Interagieren mit den Interfaces erlangt. Um eine einfache und positive Nutzung von 3DUIs sicherstellen zu können, haben [Burmester et al., 2018] bspw. Interaktionsmuster definiert, welche global eingesehen werden können und dadurch einen Standard in der Entwicklung von dreidimensionalen Interfaces bieten sollen.

2.1.6 Körperbasierte Interfaces

BIs versuchen sowohl die Ein- als auch die Ausgabe von Informationen über den menschlichen Körper zu bewerkstelligen [Harrison and Faste, 2014]. Das Ziel ist, den menschlichen Körper so zu instrumentalisieren, sodass keine externen Geräte mehr wie z.B. Smartphones benötigt werden, um Daten verarbeiten, Anwendungen nutzen oder mit anderen Menschen global kommunizieren zu können [Harrison et al., 2010]. Berücksichtigt werden sollte dabei nicht nur die Ergonomie, sondern auch die Nutzergruppe, für die ein BI entwickelt wird. [Harrison and Faste, 2014] haben dazu herausgefunden, dass bestimmte Positionen für körperbasierte Interfaces von Frauen und Männern unterschiedlich streng empfunden wurden. Für die anwesenden Frauen z.B. , war ein Interface im Schulter- oder Oberarmbereich unvorstellbar, da es zu privat oder unangenehm wäre. Allgemein akzeptabel und angenehm, war bei dem Experiment wiederum der Unterarm- und Handbereich. Durch die emotionale Verbundenheit zum eigenen Körper ist es möglich, verschiedenste Arten von körperbasierten Interfaces anzuwenden und zu entwickeln. Die Vielfältigkeit zeigt sich anhand der folgenden Beispiele. Skininput [Harrison et al., 2010] ist ein BI, welches die durch den Körper geleiteten Impulse eines Fingerschlages aufzeichnet und interpretiert. Je nach Stärke und Dauer eines Impulses, können unterschiedliche Aktionen durchgeführt werden. Omni-Touch von [Harrison et al., 2011], zu sehen in Abbildung 5, nutzt eine Microsoft Kinect als Tiefenkamera sowie einen Projektor welche auf der Schulter des Benutzers befestigt werden, um Interfaces ausgehend von der Position des Trägers, beliebig auf Hände und Arme zu projizieren. Die Finger des Trägers

werden dabei aufgezeichnet. Durch Gesten oder simple Berührungen auf dem BI können Benutzer z.B. Objekte skalieren oder zeichnen.

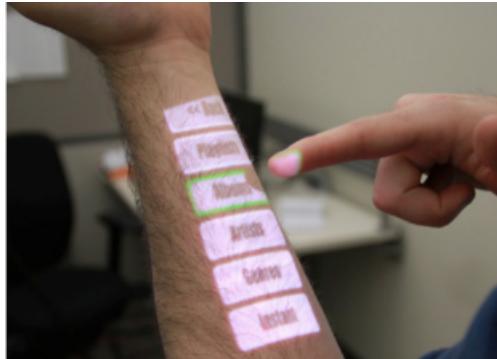


Abbildung 5: Anwendung von OmniTouch [Harrison et al., 2011, p. 446]

Ein weiteres Beispiel stellt PalmRC [Dezfuli et al., 2012] dar. Bei PalmRC wird die Handfläche eines Menschen zum Eingabegerät für Aktionen mit einem TV-Gerät. Die Handfläche besitzt dabei imaginäre Markierungen, an die diverse Funktionen gekoppelt sind, bspw. das Umschalten eines Programmes oder das Hoch- und Runterscrollen durch die Senderliste. BIs haben durch die Verbundenheit zum eigenen Körper den Vorteil, leicht bedienbar und einprägsam zu sein. Ein statisches Interface kann somit einfach erlernt und sogar mit geschlossenen Augen und ausreichender Übung bedient werden [Harrison and Faste, 2014].

2.2 Hinführung zum Konzept

In diesem Kapitel wird näher auf die Aspekte eingegangen, welche sowohl für die Entwicklung der Anwendung als auch für das Interface relevant waren. Zunächst folgen Anforderungen, welche sowohl für das Interface als auch für die Anwendung gestellt wurden. Anschließend werden die Richtlinien aufgelistet, welche bei der Entwicklung des Interfaces berücksichtigt wurden und zusätzlich Grund gab, bestimmte Designideen zu verwerfen. Danach folgt ein Abschnitt über die Wahl der Position des Interfaces. Dabei wird näher darauf eingegangen, welche Vorteile der menschliche Arm mit sich bringt, und warum sich speziell der Unterarm im Kontext dieser Arbeit als beste Wahl für ein Interface angeboten hat. Im Anschluss daran folgen Konzepte von bereits etablierten Designs, sowie Beschreibungen, weshalb ein Design für eine Anwendung mit Modellierungsansatz und körperbasierter Eingabe

in VR geeignet oder nicht geeignet ist. Abschließend wird das finale Konzept für den Prototypen aufgezeigt und beschrieben.

2.2.1 Anwendungsanforderungen

Aus dem Kontext der Arbeit, den aufgeführten ergonomischen Einschränkungen und dem Austausch mit Experten des Fraunhofer IAO in Stuttgart, sind Anforderungen für die zu entwickelnde Anwendung entstanden, welche in Tabelle 2 aufgelistet sind.

1.	Primitive Körper wie Würfel und Kugeln sollen erstellt und gelöscht werden können
2.	Einzelne Objekte sollen selektierbar sein
3.	Selektierte Objekte sollen deselektiert werden können
4.	Objekte sollen in Position, Rotation und Größe manipuliert werden können
5.	Die Rotation und Translation eines selektierten Objektes soll durch eine einhändige Eingabe möglich sein
6.	Der Benutzer bekommt Informationen zur Benutzung von Funktionen
7.	Die Anwendung soll vom Benutzer beendet werden können

Tabelle 2: Anwendungsanforderungen zufällig sortiert

2.2.2 Richtlinien

Bei der Entwicklung von 3DUIs können diverse Richtlinien und Vorgaben beachtet werden, welche für eine positive UX bei der Verwendung des zu entwickelnden Interfaces sorgen sollen. Die Richtlinien und Vorgaben beziehen sich dabei u.a. auf ergonomische Gegebenheiten, als auch auf technische und andere äußere Faktoren. Ein wichtiger Punkt für die Verwendung von Interfaces durch Gesten und Touch, ist die Komplexität und Natürlichkeit der Interaktionen [Mendes, 2018]. Funktionen wie Selektion oder Objektmanipulation sollten einfach und natürlich gestaltet sein. Das Greifen eines Objektes in der Realität kann z.B. demnach exakt in VR nachgestellt werden, um die Immersion zu erhöhen. Ebenfalls sollten die DOF so weit wie möglich, wenn möglich, reduziert werden, um die Fehlerrate zu verringern und einer Überforderung vorzubeugen. Auch *bi-manual interactions*, also zweihändige

Interaktionen, sollten aufgrund der eventuellen Komplexität vermieden werden, da dadurch der Gorilla-Arm Effekt gefördert wird [Mendes, 2018]. Funktionen, welche auf dem Arm der nicht dominierenden Hand platziert werden, sollten nach [Stuerzlinger and Wingrave, 2011] so angebracht werden, sodass andere Funktionen weder geschnitten noch überlagert werden. Auch sollten nach Stuerzlinger et al. schwebende Objekte an andere Objekte befestigt werden, sodass ein natürliches Gefühl für die Elemente entsteht. Für die Selektion und Manipulation von Objekten im virtuellen Raum, sollte außerdem die Überdeckung und Tiefenwahrnehmung berücksichtigt werden. Objekte welche so weit vom Benutzer entfernt sind, sodass ein simples Greifen nicht mehr ausreicht, sollten durch Techniken wie Pointer oder *ray-casting* selektier- und manipulierbar gemacht werden [Jr et al., 2010]. Für die Rotation von Objekten, bedarf es einer Lösung, welche das Handgelenk kaum belastet und die Ergonomie des selbigen berücksichtigt. Ebenso wichtig ist die Entfernung von Funktionen und Interaktionsmöglichkeiten vom Benutzer zum Interface. Weit entfernte Elemente verursachen Zittern und Ungenauigkeit, wodurch die Fehlerrate erhöht und eine negative Erfahrung in der Benutzung mit dem Interface und der Anwendung erlangt wird [Jr et al., 2010]. Desweiteren sollten nach [Riecke et al., 2018b] klammernde oder packende Gesten vermieden und eher auf einhändige, simple Gesten gesetzt werden. Jedoch gilt es nicht nur das Interface und die Interaktionen so einfach wie möglich zu halten, sondern auch die Umgebung. Wichtig hierbei ist das Feedback bei der Verwendung von 3DUIs [LaViola et al., 2017]. Das temporäre und dreidimensionale Feedback sollten dabei übereinstimmend sein, sprich die Bewegungen des Benutzers in der virtuellen Umgebung sind auf die natürlichen Bewegungen des Benutzers in der Realität abgestimmt, um ein positives Nutzererlebnis herzustellen und aufrechtzuerhalten. Zusätzlich in Betracht gezogen werden kann die Textur und der Stil des Interfaces. Je echter und hochwertiger das Interface gestaltet ist, desto mehr wird von der Interaktion und der Anwendung erwartet. Ein gezeichneter oder Karikatur ähnlicher Stil ist bei simplen Anwendungen zu bevorzugen, da sowohl das Erlernen als auch das Verstehen des Interfaces dadurch erleichtert wird [LaViola et al., 2017].

2.2.3 Interfacepositionierung

Die Position eines Interfaces ist mitunter ausschlaggebend, ob eine Anwendung komfortabel und praktisch benutzt werden kann. Aufgrund der bereits beschriebenen ergonomischen Gegebenheiten und dem Hintergrund, dass das Interface in einem Modellierungskontext verwendet werden soll, konnten diverse Positionen und Umsetzungen von vornherein ausgeschlossen werden.

Die zusätzliche Dimension, welche durch 3D hinzu kommt, bringt zwar Probleme wie Überdeckung oder falsche Tiefeneinschätzung mit sich, wirkt sich jedoch positiv auf die Immersion aus und bietet gerade aufgrund der zusätzlichen Tiefe, neue Möglichkeiten für Interaktionen [Stuerzlinger and Wingrave, 2011]. Die Arme haben sich als praktisches Mittel in der Arbeit mit VR erwiesen, da durch die horizontale Ausdehnung genügend Platz für diverse Funktionen vorhanden ist [Harrison et al., 2010]. Außerdem sind Arme gut beweglich und benötigen keine unbequemen Bewegungen um anvisiert werden zu können, was während der Benutzung eines HMD aufgrund der eingeschränkten FOV wichtig ist. Auch aufgrund von zweihändigen Funktionen, bietet sich der untere Arm für ein Interface an. Die nicht dominierende Hand kann dabei die Funktionen visualisieren, während die andere Hand die Funktionen selektiert und verwendet. Die Präzision wird durch die körperbasierte Lösung ebenfalls erhöht, da das Ausstrecken und Heben der Arme, um mit einem Interface interagieren zu können, nicht mehr notwendig ist. Zittern und Verzerrung, was zu ungenauen Eingaben führt, sowie die Ermüdung der Arme fällt damit weg [LaViola et al., 2017; Jr et al., 2010; Hincapie-Ramos et al., 2014]. Auch kann das körperbasierte Interface durch die Nähe zum eigenen Körper, recht große und eindeutige Elemente enthalten, welche die Menünavigation erleichtern können [Stuerzlinger and Wingrave, 2011]. Des Weiteren können durch das Arm-Interface die dazugehörigen Elemente so platziert werden, dass kein Element ein anderes überdeckt. Aktivierte Funktionen blenden nicht benötigte Funktionen aus, oder erweitern ihren Kontext durch Anordnung der Unterlemente in andere Richtungen, sodass andere Funktionen sichtbar bleiben [Harrison and Faste, 2014]. Aufgrund dessen dass in der Realität fliegende oder schwebende Objekte an einem anderen Objekt befestigt sind, bieten sich die Arme als Interfaceposition an, da somit eine Immersion der Verbindung zur Hand und zum eigenen Körper aufgebaut wird [Stuerzlinger and Wingrave, 2011]. Durch die Verbindung zum eigenen Körper ist es durchaus möglich, nach einer gewissen Zeit an Übung, das Interface auch blind bedienen zu können [LaViola et al., 2017].

2.2.4 Verwandte Ideen und Konzepte

Die folgenden Designideen für das Interface unterscheiden sich sowohl in Form als auch in Komplexität, auf welche jeweils eingegangen wird. Für diese Arbeit war es wichtig, ein Interface zu entwickeln, welches durch direkte Eingabemechanismen wie Touch bedient werden kann. Somit konnten Ansätze für Pointer-Techniken in der Designphase ausgeschlossen werden.

3D Wheel picker

Die erste Idee war ein nach [Ehrlich, 2017] sogenannter *wheelpicker*, zu sehen in Abbildung 6. Beim *wheelpicker* werden die verfügbaren Interaktionen auf einem Wheel, sprich einem runden Körper platziert. Der *wheelpicker* kann beliebig breit gestaltet werden. Die Funktionen können durch eine simple Rotation des Armes visualisiert oder auch je nach Winkel und gewünschter Funktionalität deaktiviert werden.



Abbildung 6: Wheelpicker-Modell nach [Ehrlich, 2017, p. 33]

Nachteil des Wheelpicker-Ansatzes ist jedoch die Bedienung. Das blinde Bedienen des *wheelpicker* wird durch die zusätzlich benötigte Rotation des Armes erschwert. Der Ansatz des Deaktivierens von Funktionen welche nicht im korrekten Winkel liegen, kann die blinde Bedienung zusätzlich erschweren. Auch kann die Nichtsichtbarkeit von Bedienelementen aufgrund der Rotation des Armes die Bedienung erschweren [LaViola et al., 2017]. Ebenfalls kann es aufgrund der Anbringung der Funktionen zu Überdeckungen während der Bedienung kommen, zu sehen in Abbildung 7.

Pie menu

Eine weitere Idee war ein an [LaViola et al., 2017] angelehntes *radial* bzw. nach [Ehrlich, 2017] sogenanntes *pie menu*, visualisiert in Abbildung 8. Das *radial* bzw. *pie menu* ist ein kreisförmiges Interface, bei dem die Elemente in gleichmäßigen Abständen nebeneinander platziert werden. Das *pie menu* ist im Vergleich zum *wheelpicker* ein 2D-Modell und bietet dadurch keine Tiefe.

Beim *pie menu* wird auch keine Rotation benötigt, um versteckte Funktionen sichtbar zu machen. Somit wird das Erlernen des Interfaces und die damit verbundene blinde Bedienung gegenüber dem *wheelpicker* erleichtert. Nachteil des *pie menu* ist wiederum die Überdeckung sowie die

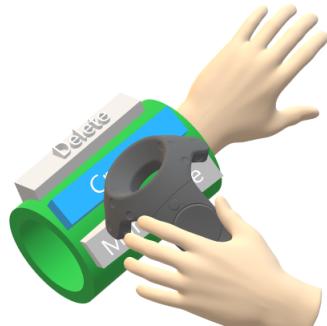


Abbildung 7: Interaktion mit Wheelpicker-Modell
nach [Ehrlich, 2017, p. 33]



Abbildung 8: Pie-Modell nach [Ehrlich, 2017, p. 10]

Anbringung der Elemente auf unterschiedlicher Höhe und Breite, was eine erhöhte Lernkurve zur Folge hat.

3D button menu

Auf Grundlage des *pie menu* entstand der dritte Entwurf, zu sehen in Abbildung 10. Das *3d button menu* unterscheidet sich nur leicht vom *pie menu*, bietet jedoch eine bessere Sichtbarkeit und einheitlich große Elemente. Auch bei diesem Modell entsteht teilweise Überdeckung. Aufgrund der Zweidimensionalität der Elemente fehlt auch hier die Tiefe, was die Bedienung des Interfaces erleichtert [LaViola et al., 2017]. Die Erweiterung gestaltet sich beim Kugel-Modell jedoch als schwierig, da eingeschlossene Elemente existieren, welche bei Erweiterung noch mehr Bedienelemente überdecken und das Interface überladen.

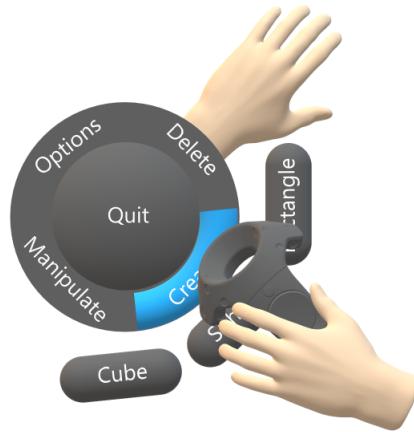


Abbildung 9: Interaktion und Erweiterung des Pie-Modells nach [Ehrlich, 2017, p. 10]

Kachel-Modell

Das bekannte Kachel-Modell von Microsoft wurde ebenfalls in Betracht gezogen, zu sehen in Abbildung 11. Das Kachel-Modell besteht ausschließlich aus zweidimensionalen Rechtecken, und bietet wie das *pie menu* oder das Kugel-Modell keine Tiefe. Vorteilhaft hierbei ist wiederum die simple Anordnung der Elemente. Die Kacheln werden über-, unter oder nebeneinander angeordnet und besitzen alle die gleichen Maße. Die Erweiterung des Kachel-Modells gestaltet sich in der hier beschriebenen Visualisierung schwierig, da ebenfalls wie beim Kugel-Modell eingeschlossene Elemente existieren, welche bei Erweiterung andere Elemente überdecken könnten.

Tabelle 3 listet die Vor- und Nachteile der genannten Designs zusammengefasst auf.

2.2.5 Interfaceanforderungen

Aus den vorangegangenen Richtlinien, Vorzügen der Positionierung und den Vor- und Nachteilen der verwandten Designs, entstanden die in Tabelle 4 aufgelisteten Anforderungen für die Gestaltung des prototypischen Interfaces.

2.2.6 Finales Interfacekonzept

Der folgende Entwurf repräsentiert das finale Konzept für den zu entwickelnden Prototypen. Auf Grundlage der festgelegten Anforderungen und Richtlinien sowie des vorher beschriebenen Kachel-Modells und dessen Vorteile,

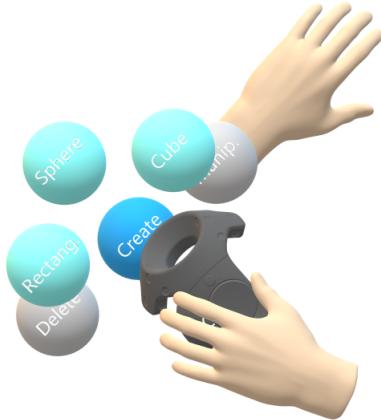


Abbildung 10: Interaktion und Erweiterung des *3d button menu* nach [Ehrlich, 2017, p. 36-37]

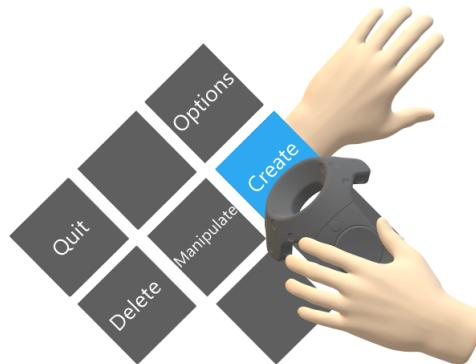


Abbildung 11: Interaktion mit Kachel-Modell

entstand die Idee eines Designs, welches für weniger Überdeckung und Überladung sorgen soll. Dem in Abbildung 12 visualisierten Interface wurde der Name *decision menu* gegeben, da es einer Auswahl bekannt von Fragebögen oder Computer-Abfragen ähnelt, bei welchen die Auswahlmöglichkeiten unter- bzw. nebeneinander platziert werden. Das *decision menu* besteht wie das Kachel-Modell aus zweidimensionalen Bedienelementen, welche jedoch in der ersten Ebene nur nebeneinander angeordnet werden und somit Komplexität reduzieren.

Überdeckung findet aufgrund der horizontalen Ausrichtung der Elemente verringert statt. Die Erweiterung des *decision menu* gestaltet sich leichter als bei den vorherigen Entwürfen, da es keine eingeschlossenen Elemente mehr gibt. Die Erweiterung der Funktionen wird durch Über- bzw. Unterordnung der

Designtyp	Vorteile	Nachteile
<i>3d wheel picker</i>	Deaktivierung nicht relevanter Funktionen durch Rotation	Tiefe und Rotation erschweren blinde Bedienung, Überdeckung durch Tiefe
<i>pie menu</i>	Keine Tiefe, Keine Rotation	Verdeckung, unterschiedliche Höhen und Breiten der Elemente
<i>3d button menu</i>	Einheitliche Elemente, Keine Tiefe	Überdeckung, eingeschlossene Elemente, Überladung
Kachel-Modell	Keine Tiefe, einheitliche Elemente	Eingeschlossene Elemente, Überdeckung

Tabelle 3: Vor- und Nachteile verwandter Interface-Designs

neuen Elemente gelöst. Dadurch werden keine anderen Funktionen überdeckt und bleiben auch bei Erweiterung einer Funktion komplett sichtbar. Die Bedienung wird beim *decision menu* nicht nur durch die Kacheloptik, sondern auch durch die simple Positionierung derselbigen erleichtert. Eine blinde Bedienung ist somit einfacher als bei den vorherigen Modellen, da sowohl Überdeckung als auch Überladung geringer sind.

Funktionen welche keine Auswirkungen auf die virtuell platzierten Objekte haben oder damit nicht in Zusammenhang stehen, werden an der Unterseite des Armes platziert. Dadurch wird die Oberseite entlastet und einer ungewollten Fehlauswahl vorgebeugt.

2.3 Implementation

Dieses Kapitel beschreibt die Umsetzung des Prototypen und der dazugehörigen Anwendung. Zunächst werden die Tools und Frameworks erläutert, welche für die Implementation der Funktionen und Umgebung zu Hilfe genommen wurden. Anschließend wird die praktische Realisierung in Textform zusammengefasst wiedergegeben. Dabei wird auf Probleme und besondere Ereignisse oder Hindernisse näher eingegangen.

1.	Das Interface soll wenn möglich keine dreidimensionalen Körper beinhalten
2.	Das Interface soll ohne Scrollen oder zusätzliche Rotationsmechanismen auskommen
3.	Die Interface-Elemente sollen wenn möglich zweidimensional und einheitlich gestaltet sein
4.	Es sollen keine Elemente eingeschlossen werden welche schwer zu erweitern sind
5.	Die Erweiterung einer Funktion soll keine anderen Funktionen überdecken
6.	Funktionen, welche nicht zur Interaktion mit virtuellen Objekten gedacht sind, sollen sich in beliebiger Weise von den interaktiven Funktionen unterscheiden
7.	Die Bedienung des Interfaces soll den eigenen Körper miteinbeziehen
8.	Es sollen auffällige Farben für selektierte Funktionen und Objekte verwendet werden

Tabelle 4: Interfaceanforderungen zufällig sortiert

2.3.1 Verwendete Frameworks und Tools

Zur Entwicklung der VR-Anwendung und des dazugehörigen körperbasierten Interfaces, wurden diverse Programme verwendet, die sowohl VR-Unterstützung als auch nützliche Extras bieten. Unity vom Unternehmen *Unity Technologies*, diente als Spiel-Engine und Entwicklungsumgebung. Verwendet wurde dabei die Version 2018.3.11f1. Zum Schreiben des benötigten Codes wurde Microsoft's Visual Studio 2017 verwendet. Unity Codeerkennung als optionale Erweiterung wurde in Visual Studio integriert. Zur Benutzung des Vive Headsets sowie der Vive Controller und Vive Kameras, wurde das Tool SteamVR [Valve Corporation, 2014] in Unity importiert. SteamVR ist ausgestattet mit zahlreichen vordefinierten Funktionen und Möglichkeiten, mit VR-Hardware interagieren und VR-Anwendungen entwickeln zu können. Um das Gefühl für den eigenen Körper in VR zu erhöhen und damit die Qualität der Immersion zu verbessern, wurde die Unity-Erweiterung FinalIK [Root-Motion, 2014] verwendet. FinalIK erfasst durch inverse Kinematik und die vorhandenen Vive Kameras die Beschaffenheit des Körpers des Trägers und kann dadurch eine vollständige virtuelle Kopie des Trägers und seiner Bewe-

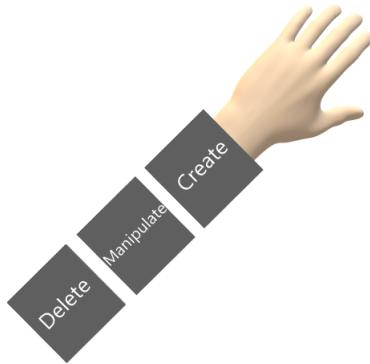


Abbildung 12: Finales Konzept *decision menu*

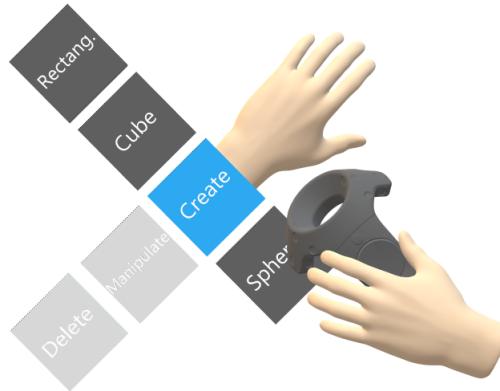


Abbildung 13: Erweiterung des *decision menu*

gungen im virtuellen Raum simulieren. Dadurch ist ein qualitativ hochwertiges Mapping des Interfaces auf den Körper des Benutzers möglich.

2.3.2 Umsetzung

Die Umsetzung setzt sich zusammen aus Elementen, welche die Anwendung und dessen Gestaltung betreffen, als auch aus Elementen speziell für das Interface und die damit verbundenen Interaktionen. Zunächst wird die virtuelle Umgebung beschrieben, in welcher das Interface benutzt werden kann. Anschließend folgen Erläuterungen zur Implementation des Interfaces und der damit verbundenen Interaktion als auch Beschreibungen der Interfacefunktionen und der Benutzung der selbigen.

Anwendung

Die Anwendung ist schlicht und praktikabel gestaltet, da der Fokus dieser

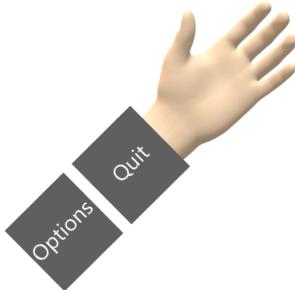


Abbildung 14: Platzierung anwendungsspezifischer Funktionen

Arbeit auf dem Interface und der Verwendung dessen liegt. Die Umgebung besteht aus einem viereckigen Raum ohne Decke, dadurch soll sich der Benutzer nicht eingeengt fühlen und Platz nach oben für Manipulationen haben. Der Boden des Raumes wurde mit einer Teleportations-Komponente versehen, welche durch SteamVR als Vorlage mitgeliefert wurde. Dadurch kann sich ein Benutzer bei Verwendung der entsprechenden Funktion, im Raum an beliebige Positionen und fest im Raum platzierte Teleportationspunkte teleportieren, um mit Objekten zu interagieren oder die Umgebung aus einem anderen Blickwinkel betrachten zu können. Jede Wand wurde mit einer Kollidierungs-Komponente versehen. Somit kann der Benutzer nicht ungewollt aus dem Raum fallen. Der Boden und die Wände besitzen schlichte, unauffällige Texturen, sodass sich der Raum nicht ungewöhnlich oder fremd anfühlt. Auch alle erstellbaren Objekte besitzen Texturen. Die Texturen der primitiven Körper bestehen jedoch ausschließlich aus einfachen Farben wie rot, grün oder blau. Die Texturen der Objekte wurden bewusst in den RGB-Farben gehalten, um die Objekte unterscheiden zu können und die Objekte für jeden subjektiven Kontext passend zu machen. Das Modell des Benutzers ist eine von FinalIK mitgelieferte Vorlage, welche an einen Crashtest-Dummy angelehnt ist. Das Modell nimmt dadurch keine geschlechterspezifische Rolle an, was für eine positive Benutzererfahrung bei allen Geschlechtern sorgen soll. Das Modell wurde mit einer hautähnlichen Farbe versehen, um ein natürlicheres Gefühl bei der Verwendung und Interaktion mit dem Modell herzustellen.

Interface

Das Interface wurde am linken Arm des Dummy-Modells angebracht. Durch das FinalIK-Plugin können dem Dummy-Modell sogenannte *Bones* und Kalibrierungskomponenten zugeordnet werden. Somit kann das Dummy-Modell an

den Körper eines Benutzers angepasst werden, um für eine größere Immersion und genauere Eingaben sowie Bewegungen zu sorgen. Der linke Arm teilt sich somit in mehrere *Bones* auf, welche bei der Positionierung und Fixierung des Interfaces hilfreich waren. Das Interface selbst besteht aus einem *Canvas*, welches im *WorldSpace* platziert wurde. Durch die *WorldSpace* Einstellung, wurde die Anbringung am linken Arm des Benutzers erst möglich, da ein im *ScreenSpace* platziertes Interface direkt vor einer Kamera platziert wird und im dreidimensionalen Raum nicht frei beweglich ist, was für diese Arbeit jedoch wichtig war. Die Rotation und Translation des Interfaces wird hauptsächlich durch das Modell und die zugeordneten *Bones* sowie ein Rotationsskript von FinalIK geregelt. Da die Standardeinstellungen für den Kontext dieser Arbeit nicht ausreichten, wurden Anpassungen und Optimierungen durch zusätzliche selbstgeschriebene Skripte vorgenommen. Auf der ersten Ebene, welche sich auf der oberen Seite des linken Unterarms befindet, wurden die objektspezifischen Funktionen angebracht. Darunter die Funktionen zur Manipulation und Erstellung von Objekten, in der Anwendung *Manipulate* und *Create* genannt. Die englische Sprache wurde für alle Interaktionen, Hinweise und Bezeichnungen verwendet, um die Anwendung international einsetzbar zu machen. Die objektspezifischen Funktionen der ersten Ebene wurden nebeneinander angebracht und erstrecken sich somit auf der horizontalen Achse des Armes des Dummy-Modells. Die zweite Ebene der objektspezifischen Funktionen, erstreckt sich auf der vertikalen Achse des Armes. Das bedeutet, dass erweiterte Funktionen über oder unter den Funktionen der ersten Ebene angebracht sind, ohne die erste Ebene zu verdecken. Zur Erweiterung der Manipulationsfunktion gehören die Standard-Manipulationen *Translate*, *Rotate* und *Scale*, auf welche später in diesem Kapitel eingegangen wird. Die Erweiterung der *Create*-Funktion umfasst die drei primitiven Körper Würfel, Kugel und Zylinder. Anwendungsspezifische Funktionen bzw. Funktionen ohne weitere Kategorisierung, wurden auf der unteren Seite des linken Unterarms des Dummy-Modells angebracht. Dadurch soll der Fokus der Interaktion auf die Objekte verstärkt werden und die Komplexität und Bedienung des Interfaces verringert und erleichtert werden. Zu den nicht kategorisierten bzw. anwendungsspezifischen Funktionen gehört das Löschen von Objekten, in der Anwendung *Delete* genannt, und das Beenden der Anwendung, in der Anwendung mit *Quit* tituliert. Die *Quit*-Funktion befindet sich bewusst weiter hinten am Unterarm, um die Wahrscheinlichkeit eines versehentlichen Schließens der Anwendung zu verringern. Zusätzliche oder neue Funktionen können beliebig auf den Achsen auf der Ober- oder Unterseite des Armes platziert werden, wenn die Ebenen und deren Bedeutung berücksichtigt wird. Die erste Ebene symbolisiert Obermenüs, wohingegen die zweite Ebene Untermenüs bzw. Funktionen

ohne weitere Unterkategorien darstellt. Aufgeklappte, aktive und markierte Funktionen werden farblich hervorgehoben, um ein visuelles Feedback bei der Bedienung des Interfaces zurückzugeben. Nach [Vos, 1978] reagieren Menschen besonders auf gelbe oder grüne sowie auf helle Farben. Daraus resultieren die vergebenen Farben für die verschiedenen visuellen Feedbacks bei der Verwendung des Interfaces, zu sehen in Abbildung 15. Aufgeklappte Funktionen werden demnach türkis dargestellt, aktivierte Funktionen gelb und markierte Funktionen hellblau. Die Standardfarbe für alle Funktionen ist weiß.

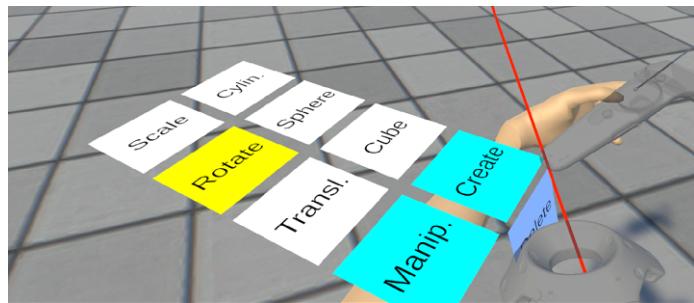


Abbildung 15: Interfaceimplementation

Funktionen

In diesem Abschnitt werden die Interface- und Anwendungsfunktionen und deren Benutzung näher erläutert. Dabei wird der Zweck jeder Funktion beschrieben und die Verwendung und Auswirkung verdeutlicht.

Die Teleportation des Benutzers wird durch eine Touchpad-Funktion ermöglicht. Dazu wird das Touchpad des rechten Controllers gedrückt und gehalten, und anschließend ein beliebiger Punkt im Raum anvisiert. Die Teleportation wird durch eine gestrichelte Kurve und einen Auftreffpunkt repräsentiert, wie in Abbildung 16 zu sehen. Nach Loslassen des Touchpad-Buttons wird der Benutzer zum ausgewählten Ort teleportiert.

Objekte werden durch das Markieren eines Körper-Buttons und anschließendes Drücken des Trigger-Buttons des rechten Controllers erstellt und in der Mitte des Raumes platziert.

Um ein Objekt zu (de-)selektieren, wurde eine Laserpointer-Technik verwendet. Dadurch kann auch mit weiter entfernten Objekten gearbeitet werden, ohne in deren unmittelbarer Nähe sein zu müssen. Der Laserpointer wurde am rechten Controller angebracht und besitzt zwei Farben, welche den aktuellen Status des Laserpointers darstellen. Die Nichtverwendung wird durch einen roten, die Aktivierung durch einen blauen Strahl repräsentiert. Wird der Trigger-Button des rechten Controllers gedrückt, während ein se-

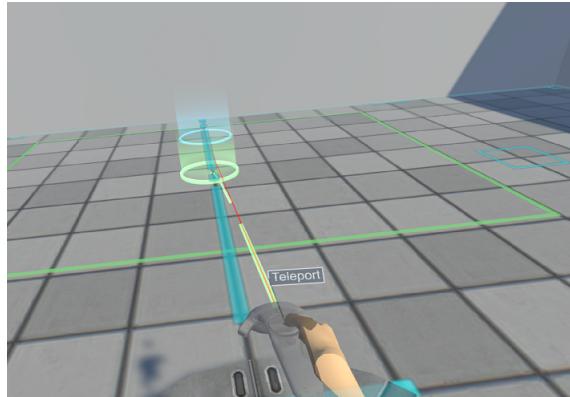


Abbildung 16: Teleportationsfunktion

lektierbares Objekt anvisiert wird, so wird das Objekt ausgewählt und gelb eingefärbt, um die Selektion, visualisiert in Abbildung 17, zu bestätigen. Die Deselektion wird durch erneutes Drücken des Trigger-Buttons durchgeführt.

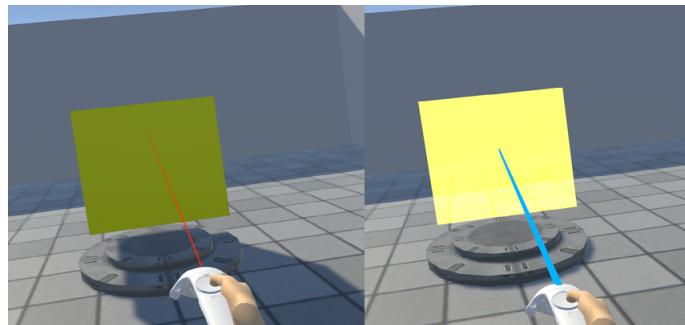


Abbildung 17: Markieren und Selektieren eines Würfels

Die Translation von selektierten Objekten, zu sehen in Abbildung 18, setzt die Aktivierung der Translations-Funktion voraus. Ist die Funktion aktiviert, so kann ein Objekt durch das Drücken und Halten des linken Grip-Buttons frei im Raum bewegt werden. Wird der Grip-Button losgelassen so verweilt das Objekt an der zuletzt erreichten Position.

Eine Rotation setzt ebenfalls eine Aktivierung der Funktion voraus, als auch ein selektiertes Objekt. Sind beide Voraussetzungen gegeben, kann ein Objekt wie in Abbildung 19 durch Drücken und Halten des linken Grip-Buttons am rechten Controller sowie eine beliebige Rotationsrichtung rotiert werden. Die Skalierung ist in dieser Arbeit die einzige Funktion, die eine zweihändige Geste voraussetzt. Nach Aktivierung der Funktion und Auswählen eines Objektes kann ein Objekt skaliert werden, indem der rechte Grip-Button des

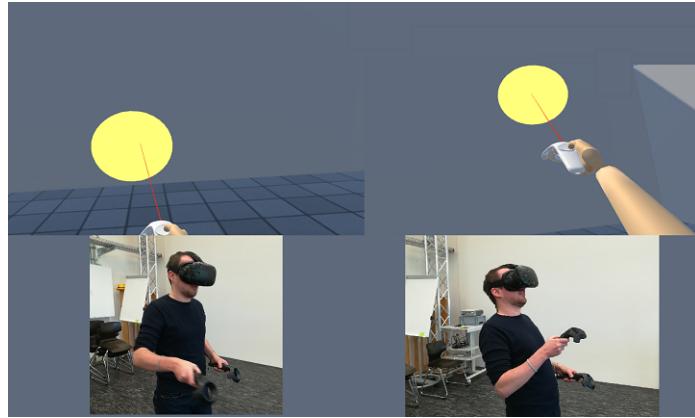


Abbildung 18: Verwendung der Translationsfunktion

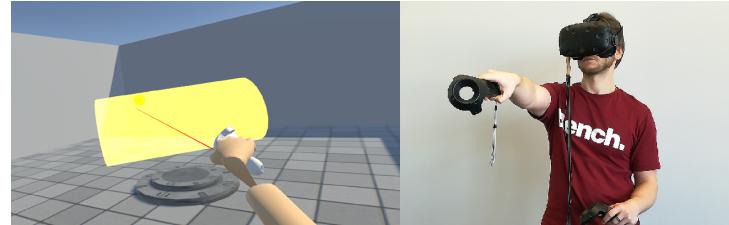


Abbildung 19: Rotieren eines Objektes

linken und der linke Grip-Button des rechten Controllers gedrückt und gehalten werden, und die Controller voneinander weg oder zueinander hin bewegt werden. Die Skalierungs-Geste entspricht der Vergrößern-Verkleinern-Geste, bekannt aus der Verwendung eines Smartphones. Abbildung 20 verdeutlicht die Funktionsweise.

Das Löschen von Objekten, setzt ein ausgewähltes Objekt voraus. Eine Markierung und anschließende Aktivierung durch den Trigger-Button des rechten Controllers löscht das ausgewählte Objekt und hebt die Auswahl auf.

Der Benutzer hat die Möglichkeit die Anwendung jederzeit zu beenden. Dazu wird der Quit-Button markiert und mit Drücken des Trigger-Buttons des rechten Controllers bestätigt. Der Benutzer gelangt daraufhin ins SteamVR-Hauptmenü zurück.

Um das Interface blind bedienen zu können, wurde eine Technik verwendet, welche den aktuellen Sichtbereich des Benutzers speichert und überprüft, ob eine Funktion momentan markiert, aber nicht zu sehen ist. Ist eine Funktion

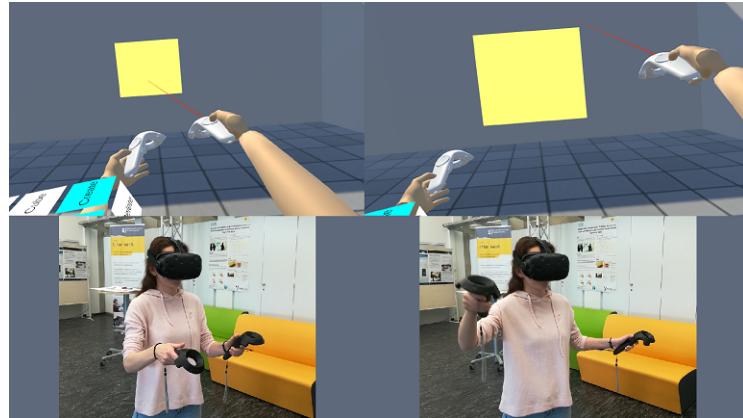


Abbildung 20: Verwendung der Skalierungsfunktion

markiert aber nicht sichtbar, so wird der Titel des markierten Buttons wie in Abbildung 21 vor der virtuellen Kamera angezeigt.

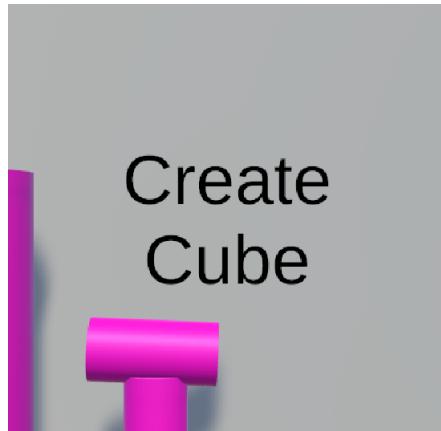


Abbildung 21: Einblendung eines Funktionstitels bei blinder Bedienung

2.3.3 Probleme und Hindernisse

Bei der Entwicklung und Implementation des Interfaces als auch der damit verbundenen Funktionen, kam es regelmäßig zu Verzögerungen. Aufgrund der unterschiedlichen Komplexität dauerten Lösungen unterschiedlich lange und nahmen dadurch einen wesentlichen Teil bei der Entwicklung ein. Problematisch war anfangs der Umgang mit den Erweiterungen. Da SteamVR und FinalIK von unterschiedlichen Herstellern entwickelt werden, kam es teilweise zu Interferenzen bei der Implementation, woraufhin Lösungen gefunden

werden mussten, um beide Erweiterungen ohne Fehler verwenden zu können. Auch die Rotation von Objekten gestaltete sich als schwierig, da eine Lösung implementiert werden sollte, welche abhängig von der aktuellen Position des Benutzers zu jeder Zeit die gleichen Aktionen ausführt. Das bedeutet, dass die unterschiedlichen Achsen von Objekt und Benutzer gespiegelt und passend übersetzt werden mussten, je nachdem in welchem Winkel der Benutzer zum Objekt steht, da nur somit eine stetig übereinstimmende visuelle Repräsentation der Rotation möglich war. Ein weiteres Hindernis war die Platzierung auf und die Rotation des Interfaces mit dem Arm des Dummy-Modells, da sich der Arm standardmäßig nicht mit der Hand mit rotiert hat. Dadurch mussten zusätzliche Skripte entwickelt und weitere mitgelieferte Komponenten eingebunden und konfiguriert werden. Ein kleines, jedoch wichtiges Feature, welches von Anfang bis Ende mehrmals angepasst werden musste, war die Interaktion der Bedienelemente. Eine Lösung zu finden, die UI-Elemente direkt mit dem Controller bedienen zu können, darunter das Aufklappen und Aktivieren von Funktionen, war zu Anfang der Implementationsphase schwierig, wurde jedoch mit ansteigendem Wissen und Kreativität zunehmend einfacher und schließlich gelöst. Kleinere Probleme wie Fehlkonfigurationen, Versuch und Irrtum Ansätze oder Ausfälle der Hardware erschwerten und verzögerten die Implementation zusätzlich, jedoch nicht gravierend.

2.4 Evaluation

In diesem Abschnitt wird näher auf die Evaluation des entwickelten Systems, bestehend aus Anwendung und Interface, eingegangen. Dabei wird u.a. erläutert mit welcher Methode evaluiert wurde, wie die dabei verwendeten Räumlichkeiten und Variablen aussahen, wie das System bewertet wurde und wie die Ergebnisse und deren Interpretation ausgefallen sind.

2.4.1 Aufbau

Für die Evaluation des Systems wird nach dem DECIDE-Framework von [Rogers et al., 2011] vorgegangen. Das Ziel der Evaluation ist, die Benutzungsfreundlichkeit und Bedienung des Systems zu überprüfen. Dafür wird der *System Usability Scale* (SUS) von J. Brooke [Jordan et al., 2014, p. 189-196] als Grundlage für die Befragung der Testpersonen verwendet. Der SUS beinhaltet die folgenden zehn Fragen, aufgelistet in Tabelle 5, welche das System und dessen Benutzungsfreundlichkeit durch Werte von 1 für "Stimme überhaupt nicht zu" bis 5 für "Stimme voll und ganz zu" bewertbar und überprüfbar macht.

1.	I think that I would like to use this system frequently
2.	I found the system unnecessarily complex
3.	I thought the system was easy to use
4.	I think that I would need the support of a technical person to be able to use this system
5.	I found the various functions in this system were well integrated
6.	I thought there was too much inconsistency in this system
7.	I would imagine that most people would learn to use this system very quickly
8.	I found the system very cumbersome to use
9.	I felt very confident using the system
10.	I needed to learn a lot of things before I could get going with this system

Tabelle 5: System Usability Scale Fragen nach Reihenfolge sortiert

Zusätzlich zum SUS wird die Umfrage durch Fragen nach der bisherigen Erfahrung mit VR und HMDs sowie dem Alter und dem Geschlecht erweitert. Die Räumlichkeit in der die Evaluationsdurchführung stattfinden soll, wurde mithilfe des SteamVR-Plugins auf 4.0 x 3.0 Meter gemessen. Die gemessene Größe und damit die Limitierung der Bewegungsfreiheit, wird dem Träger der HTC Vive bei der Durchführung durch rote Laserstrahlen auf dem Bildschirm angezeigt. Der Raum bietet helle Deckenleuchten und ist somit unabhängig von äußerer Umweltgegebenheiten verwendbar. Ebenfalls wurde der virtuelle Raum, zu sehen in Abbildung 22, für die Evaluation angepasst.

Die Testumgebung wurde mit Objekten gefüllt, welche sowohl als Orientierungshilfe zur Erfüllung der gestellten Aufgaben, als auch als Interaktionsmöglichkeit dienen. Zu den Aufgaben gehört das Teleportieren, Erstellen von Objekten, Selektieren und Deselektieren von Objekten, die Translation, Rotation und Skalierung von Objekten, sowie das Löschen von Objekten, die blinde Bedienung und das Beenden der Anwendung. Die Aufgaben sind unterschiedlich schwer und sollen den Fortschritt beim Erlernen der Bedienung des Interfaces, als auch den Funktionsumfang dessen darstellen.

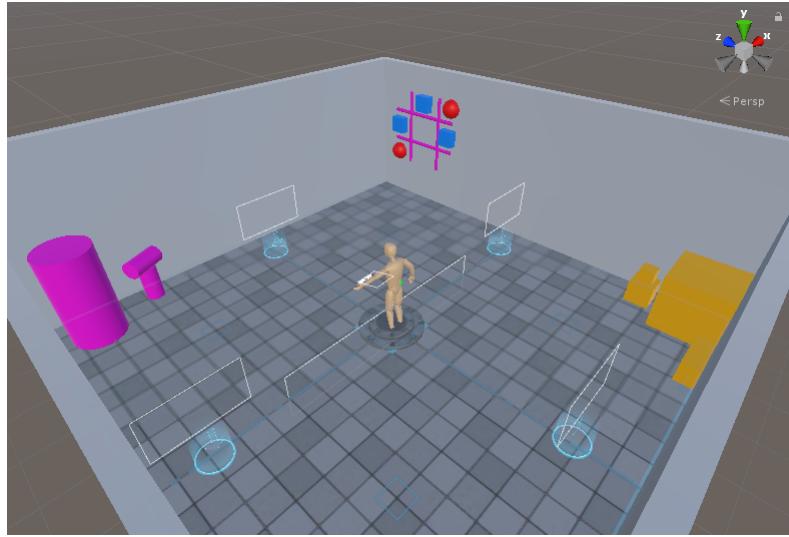


Abbildung 22: Angepasster virtueller Raum für Evaluation

2.4.2 Durchführung

Die Evaluation wurde in einem lichtdurchfluteten Raum mit Raumtemperatur durchgeführt. Männer und Frauen wurden in beliebiger Reihenfolge nacheinander über einen Testzeitraum von acht Stunden hinweg getestet. Dafür musste jeder Proband vorab eine Einverständniserklärung unterzeichnen und bekam daraufhin ein Einleitungsvideo zu sehen, welches auf die anschließenden Aufgaben vorbereitete und die Verwendung des Interfaces darstellte. Jeder Testperson wurde in gleichem Maße bei der Verwendung der Ausrüstung und der Einnahme der Startposition geholfen. Zu Anfang jedes Testdurchlaufes wurde die Testperson kalibriert, um das Dummy-Modell auf die tatsächliche Größe des Testers oder der Testerin anzupassen, und somit die Bedienung des Interfaces zu erleichtern. Für jede Testperson wurde die Zeit für die Erfüllung aller Aufgaben gestoppt. Die Höchstdauer für die Erfüllung aller Aufgaben betrug 20 Minuten. Jeder Testperson wurden die gleichen Aufgabestellungen gegeben, Hilfestellungen wenn nötig geboten, und Hinweise aufgesagt. Die Aufgaben wurden nach Schwierigkeit angeordnet und wurden wenn gewünscht wiederholt. Nach einem Testdurchlauf wurde die jeweilige Testperson gebeten die Umfrage auszufüllen und damit den Test abzuschließen. Probleme wie das Ausfallen der Hardware oder die Falschkalibrierung der Testperson, wurde über den Neustart der Anwendung und dem Erneutem einnehmen der Startposition gelöst. Traten Probleme in-

mitten des Testes auf, so wurde die Anwendung neu gestartet, und die zuletzt aktive Aufgabe wiederholt.

2.4.3 Diskussion

was ist raus gekommen

3 Schluss

Zusammenfassend siehts so aus

3.1 Fazit

Das ist das Fazit

3.2 Zukunftsaussichten und Pläne

Rosige Aussichten

Referenzen

- Michael Burmester, Kristin Haasler, Katharina Schippert, Vivien Engel, Ralph Tille, Daniel Reinhardt, and Joern Hurtienne. Lost in space? 3d-interaction-patterns für einfache und positive nutzung von 3d interfaces. In Steffen Hess and Holger Fischer, editors, *Mensch und Computer 2018 - Usability Professionals*, pages 53–66, Bonn, 2018. Gesellschaft für Informatik e.V. Und German UPA e.V.
- Niloofar Dezfuli, Mohammadreza Khalilbeigi, Jochen Huber, Florian Mueller, and Max Muehlhaeuser. Palmrc: Imaginary palm-based remote control for eyes-free television interaction. In *Proceedings of the 10th European Conference on Interactive TV and Video*, EuroITV ’12, pages 27–34, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1107-6.
- Jan Ehrlich. Exploring Mid-air Interaction for Menu Control in Virtual Reality with regard to Task Performance and User ’ s Preference. *Faculty of Mathematics and Computer Science Department of Computer Science*, 2017.
- Barrett Ens, Rory Finnegan, and Pourang P Irani. The personal cockpit: A spatial interface for effective task switching on head-worn displays. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI ’14*, pages 3171–3180, 2014.
- Chris Harrison and Haakon Faste. Implications of location and touch for on-body projected interfaces. In *Proceedings of the 2014 Conference on Designing Interactive Systems*, DIS ’14, pages 543–552, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2902-6.
- Chris Harrison, Desney Tan, and Dan Morris. Skinput: Appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’10, pages 453–462, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9.
- Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. Omnitouch: Wearable multitouch interaction everywhere. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST ’11, pages 441–450, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0716-1.
- Juan David Hincapie-Ramos, Xiang Guo, Paymahn Moghadasi, and Pourang Irani. Consumed endurance: A metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the 32Nd Annual ACM Conference*

on Human Factors in Computing Systems, CHI '14, pages 1063–1072, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1.

Lasse Jensen and Flemming Konradsen. A review of the use of virtual reality head-mounted displays in education and training. *Education and Information Technologies*, 23(4):1515–1529, 2018. ISSN 1573-7608.

Luis N. Jofre, Graciela B. Rodriguez, Yoselie M. Alvarado, Jacqueline M. Fernandez, and Roberto A. Guerrero. Non-verbal communication interface using a data glove. *CACIDI 2016 - Congreso Argentino de Ciencias de la Informatica y Desarrollos de Investigacion*, 2016.

P.W. Jordan, B. Thomas, I.L. McClelland, and B. Weerdmeester. *Usability Evaluation In Industry*. CRC Press, 2014. ISBN 9781498710411.

Joseph J Laviola Jr, Richard L Marks, and Richard Marksplaystationsony.com. An introduction to 3d spatial interaction with video game motion controllers sony computer entertainment america. 2010.

M. Kyritsis, S. R. Gulliver, S. Morar, and R. Stevens. Issues and benefits of using 3d interfaces: Visual and verbal tasks. In *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems*, Medes '13, pages 241–245, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2004-7.

Vasudevan Lakshminarayanan and Sayoko E Moroi. Head-Mounted Display Technology for Low-Vision Rehabilitation and Vision Enhancement. *American Journal of Ophthalmology*, 176:26–32, 2017. ISSN 0002-9394.

Douglas Lanman, Henry Fuchs, Mark Mine, Ian McDowall, and Michael Abrash. Put on your 3d glasses now: The past, present, and future of virtual and augmented reality. In *ACM Siggraph 2014 Courses*, Siggraph '14, pages 12:1–12:173, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2962-0.

Joseph J. LaViola, Ernst Kruijff, Ryan P. McMahan, Doug Bowman, and Ivan P. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley usability and HCI series. Addison-Wesley, Boston, 2nd edition edition, 2017. ISBN 978-0-13-403432-4.

Joseph J. LaViola, Jr. An introduction to 3d gestural interfaces. In *ACM Siggraph Courses*, Siggraph '14, pages 25:1–25:42, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2962-0.

- Seokwon Lee, Kihong Park, Junyeop Lee, and Kibum Kim. User study of vr basic controller and data glove as hand gesture inputs in vr games. *Proceedings - 2017 International Symposium on Ubiquitous Virtual Reality, ISUVR 2017*, pages 1–3, 2017.
- Jessica L Maples-keller, Brian E Bunnell, Sae-jin Kim, O Barbara, and Behavioral Sciences. The use of virtual reality technology in the treatment of anxiety and other psychiatric disorders. pages 103–113, 2018.
- D Mendes. A survey on 3d virtual object manipulation : From the desktop to immersive virtual environments. *Major Revision from Eurographics Conference*, pages 1–25, 2018.
- D. Mendes, D. Medeiros, M. Sousa, R. Ferreira, A. Raposo, A. Ferreira, and J. Jorge. Mid-air modeling with boolean operations in vr. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 154–157, 2017.
- Pranav Mistry, Pattie Maes, and Liyan Chang. Wuw - wear ur world: A wearable gestural interface. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 4111–4116, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-247-4.
- Domen Novak, Matjaz Mihelj, and Samo Begus. *Virtual Reality Technology and Applications*. Springer Dordrecht, 2014. ISBN 9789400769090.
- Francisco R. Ortega, Fatemeh Abyarjoo, Armando Barreto, Naphtali Rishe, and Malek Adjouadi. *Interaction Design for 3D User Interfaces: The World of Modern Input Devices for Research, Applications, and Game Development*. A. K. Peters, Ltd., Natick, MA, USA, 2016. ISBN 1482216949, 9781482216943.
- Bernhard E. Riecke, Joseph J. LaViola, Jr., and Ernst Kruijff. 3d user interfaces for virtual reality and games: 3d selection, manipulation, and spatial navigation. In *ACM Siggraph 2018 Courses*, Siggraph '18, pages 13:1–13:94, New York, NY, USA, 2018a. ACM. ISBN 978-1-4503-5809-5.
- Bernhard E. Riecke, Joseph J. LaViola, Jr., and Ernst Kruijff. 3d user interfaces for virtual reality and games: 3d selection, manipulation, and spatial navigation. In *ACM Siggraph 2018 Courses*, Siggraph '18, pages 13:1–13:94, New York, NY, USA, 2018b. ACM. ISBN 978-1-4503-5809-5.
- Yvonne Rogers, Helen Sharp, and Jenny Preece. *Interaction design: beyond human-computer interaction ;3rd ed.* J Wiley & Sons, Chichester West Sussex UK, 2011”.

RootMotion. FinalIK Plugin, 2014. URL
<http://www.root-motion.com/final-ik.html>. aufgerufen am 06.04.2019.

Wolfgang Stuerzlinger and Chadwick A. Wingrave. *The Value of Constraints for 3D User Interfaces*, pages 203–223. Springer Vienna, Vienna, 2011. ISBN 978-3-211-99178-7.

Denis Vajak. Combining Photogrammetry , 3D Modeling and Real Time Information Gathering for Highly Immersive VR Experience. *2017 Zooming Innovation in Consumer Electronics International Conference (ZINC)*, pages 82–85, 2017.

Valve Corporation. SteamVR Plugin, 2014. URL
<https://steamcommunity.com/steamvr>. aufgerufen am 01.05.2019.

Judd Vos. Colorimetric and photometric properties of a 2 degree fundamental observer. pages 125—128, 1978.

Y. Weiß, D. Hepperle, A. Sieß, and M. Woelfel. What user interface to use for virtual reality? 2d, 3d or speech—a user study. In *2018 International Conference on Cyberworlds (CW)*, pages 50–57, 2018.

Andrew D. Wilson and Hrvoje Benko. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST ’10, pages 273–282, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0271-5.

Ning-Ning Zhou and Yu-Long Deng. Virtual reality: A state-of-the-art survey. *International Journal of Automation and Computing*, pages 319–325, 2009. ISSN 1751-8520.