

FH JOANNEUM - University of Applied Sciences

**Bewegungserkennung für physiotherapeutische Übungen mit Augmented-,
Virtual- und Mixed Reality Devices**

Masterarbeit

zur Erlangung des akademischen Grades eines/einer

„Diplomingenieurs/Diplomingenieurin für technisch-wissenschaftliche Berufe“

eingereicht am Master-Studiengang Informationsmanagement

Verfasser/in:

Stefan Niedermüller Bsc

Betreuer/in/nen:

FH-Prof. Dipl.-Ing. Dr. Alexander Nischelwitzer

[Graz, 2018]

Ehrenwörtliche Erklärung

Ich erkläre ehrenwörtlich, dass ich die vorliegende Masterarbeit selbstständig angefertigt und die mit ihr verbundenen Tätigkeiten selbst erbracht habe. Ich erkläre weiters, dass ich keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle aus gedruckten, ungedruckten oder dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte sind gemäß den Regeln für gutes wissenschaftliches Arbeiten zitiert und durch Fußnoten bzw. durch andere genaue Quellenangaben gekennzeichnet.

Die vorliegende Originalarbeit ist in dieser Form zur Erreichung eines akademischen Grades noch keiner anderen Hochschule vorgelegt worden. Diese Arbeit wurde in gedruckter und elektronischer Form abgegeben. Ich bestätige, dass der Inhalt der digitalen Version vollständig mit dem der gedruckten Version übereinstimmt.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben kann.

Graz, 21.06.2018

Stefan Niedermüller

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufbau der Arbeit	1
1.2	Begriffsdefinitionen	2
1.2.1	Realitäts-Virtualitäts-Kontinuum	2
1.2.2	Degrees of freedom (DoF)	3
1.2.3	Field of view (FOV)	4
2	Relevante Technologien und Forschungen	7
2.1	Gestenerkennung und Spatial Mapping	7
2.2	Objekterkennung mit der Microsoft HoloLens	9
2.3	Bewegungserkennung in MR Applikationen	11
2.3.1	Inside-Out Bewegungserkennung	13
2.3.2	Outside-In Bewegungserkennung	14
2.4	Schwierigkeiten von VR	15
2.5	Microsoft HoloLens für Betroffene von Alzheimer	16
3	Fokusgruppen und Applikationsziele	19
3.1	Ziele der Applikationen	19
3.2	Zusammenhang zur Problemstellung	20
3.3	Definition und Beschreibung der Zielgruppen	20
3.4	Beschreibung des Inputs der Zielgruppen	21
4	Beschreibung der Applikationen	23
4.1	Funktionalität der Applikationen	23
4.2	Beschreibung der ausgewählten Übungen für die Erkennung	24
4.3	Konfigurationsinterface für Physiotherapeuten und Physiotherapeutinnen	28
4.3.1	Webserver	28
4.3.2	Beschreibung des Konfigurationsinterfaces	29
4.3.3	Schwierigkeiten des Konfigurationsinterfaces	31
5	Beschreibung der Microsoft HoloLens Applikation	33
5.1	Beschreibung verwendeter Hard- und Software	33
5.1.1	Unity 3D	33
5.1.2	Universal Windows Platform	33
5.1.3	Windows Mixed Reality	34
5.1.4	Microsoft HoloLens	37
5.1.5	Spatial Mapping	38
5.1.6	Spatial Audio	40
5.2	Interface der Applikation	41
5.3	Automatische Interface Positionsanpassung	44
6	Bewegungserkennung mit der Microsoft HoloLens	46
6.1	Beschreibung der analysierten Bewegungserkennungsalgorithmen	46
6.1.1	Messen des Abstands zu einer Ursprungsposition	46
6.1.2	Messen der Beschleunigung über Zeit zur Errechnung der Bewegung	47

Inhaltsverzeichnis

6.2	Umsetzung der Bewegungserkennung für die diversen Übungen	48
6.2.1	Erkennung von Kniebeugen	48
6.2.2	Erkennung von Seitschritten	51
6.2.3	Erkennung von Step-Ups	52
6.2.4	Erkennung von Zehen berühren mit den Händen	55
6.2.5	Schwierigkeiten der Bewegungserkennung	57
7	Bewegungserkennung mit der HTC Vive	58
7.1	HTC Vive	58
7.2	Applikation für die HTC Vive	60
7.2.1	Bewegungserkennung in der Applikation	61
7.2.2	Schwierigkeiten der Bewegungserkennung	66
8	Bewegungserkennung mithilfe eines Smartphones	68
8.1	Beschreibung der verwendeten Hard- und Software	68
8.1.1	AR Core	68
8.1.2	Samsung Galaxy S8+	69
8.1.3	Haori Mirror	70
8.2	Erstellung der Applikation	71
8.2.1	Umsetzung der Bewegungserkennung	73
8.2.2	Schwierigkeiten der Bewegungserkennung	75
9	Vergleich der verwendeten Devices	77
9.1	Definition der Messverfahren und Vergleichskriterien der Applikationen	77
9.1.1	Die Genauigkeit der Erkennung der Kopfposition im Raum	77
9.1.2	Die Genauigkeit der Erkennung der Körpergröße	77
9.1.3	Die Genauigkeit der Erkennung der Distanz zwischen zwei HTC Vive Trackern	78
9.1.4	Durchführung der Messungen	79
9.2	Auswertung der Messergebnisse	79
9.3	Vergleich des Loggings	83
9.3.1	Logging der Microsoft HoloLens Applikation	84
9.3.2	Logging der HTC Vive Applikation	85
9.3.3	Logging der Smartphone Applikation	87
10	Usability Tests	89
10.1	Testplanung	89
10.1.1	Pre-Test Fragebogen	89
10.1.2	Post-Test Fragebogen	89
10.2	Testdurchführung	90
10.3	Testauswertung	91
10.3.1	Microsoft HoloLens Applikation	92
10.3.2	HTC Vive Applikation	93
10.3.3	Smartphone Applikation	93
10.3.4	Gesamtauswertung	94
10.4	Diskussion der Ergebnisse	97
11	Conclusio	100

Abbildungsverzeichnis

1.1	Realitäts-Virtualitäts-Kontinuum [61]	3
1.2	Sechs Freiheitsgrade [36]	4
1.3	Menschliches Field of view [25]	5
1.4	Tag-along Visualisierung [25]	6
2.1	Interface Design - Baustein Applikation [10]	8
2.2	Spatial Mapping Ergebnisse - Baustein Applikation [10]	9
2.3	Hardware Setup - Objekterkennung Applikation [13]	9
2.4	Referenzszene - Objekterkennung Applikation [13]	10
2.5	Vergleichsergebnisse - Objekterkennung Applikation [13]	10
2.6	Multiple Objekterkennung [38]	13
2.7	Inside-Out Tracking [23]	14
2.8	Outside-In Tracking [23]	15
2.9	Visualisierung des Blickvektors [1]	17
2.10	Darstellung des Spielfeldes [1]	17
2.11	Spielfeld der dritten Anwendung [1]	18
4.1	Abschluss der Kalibration	23
4.2	Darstellung einer Kniebeuge [27]	25
4.3	Darstellung eines Seitschrittes [24]	26
4.4	Darstellung eines Step-Ups [14]	26
4.5	Darstellung einer Zehenberührung [54]	27
4.6	Darstellung der Webserver-Kommunikation	28
4.7	Erste Version des Konfigurationsinterfaces	30
4.8	Zweite Version des Konfigurationsinterfaces	31
4.9	Veränderung des FOV bei Live-Übertragung	32
5.1	Windows MR Devices [33]	34
5.2	Interaktion mittels MR Controller [60]	35
5.3	Air tap Geste [56]	37
5.4	Bloom Geste [56]	37
5.5	Microsoft HoloLens [10]	38
5.6	Windows Device Portal [45]	39
5.7	Spatial Mapping des Entwicklungsraumes	39
5.8	Vergleich Originalbild zu Spatial Mapping Abbild	40
5.9	Virtueller Trainingsraum in den Applikationen	41
5.10	Erste Version des Menüdesigns	42
5.11	Zweite Version des Menüdesigns	43
5.12	Dritte Version des Menüdesigns	43
5.13	Automatische Interface Positionsanpassung - Seitliches Hindernis	44
5.14	Automatische Interface Positionsanpassung - Oberes Hindernis	45
6.1	Beschleunigungsgraph eines Seitschrittes	47
6.2	Erkennung 10 einer Kniebeuge	50
6.3	Spatial Mapping des verwendeten Steppers	54

Abbildungsverzeichnis

6.4	Spatial Mapping und Realbild der Hände und Füße	56
7.1	Darstellung der HTC Vive mit Controlearn und Basisstationen und Trackern [21]	58
7.2	Darstellung der virtuellen Umgebung	60
7.3	Erkennung einer Kniebeuge	63
8.1	Umgebungserkennung mit ARCore [16]	69
8.2	Samsung Galaxy S8+ [41]	70
8.3	Haori Mirror AR Headset [18]	71
8.4	Konfiguration der AR Kamera für die Smartphone Applikation	71
8.5	Konfiguration der AR Kamera für die Smartphone Applikation	72
9.1	Befestigung der HMDs auf der Messpuppe	78
9.2	Ergebnisse der Messungen HoloLens und HTC Vive	80
9.3	Ergebnisse der Messungen aller Devices	81
9.4	Allgemeines Logging	83
9.5	Zusammenfassung der Ergebnisse	84
9.6	Logging der Microsoft HoloLens Applikation	85
9.7	Logging der HTC Vive Applikation	86
9.8	Logging der Smartphone Applikation	88
10.1	Usability Test - Ergebnis physiotherapeutische Erfahrung	91
10.2	Usability Test - Ergebnis der HoloLens Applikation	92
10.3	Usability Test - Ergebnis der automatischen Menüanpassung	92
10.4	Usability Test - Ergebnis der HTC Applikation	93
10.5	Usability Test - Ergebnis der Smartphone Applikation	94
10.6	Usability Test - Gesamtergebnis	94
10.7	Usability Test - Beste Applikation	95
10.8	Usability Test - Kollisionen in den Anwendungen	96
10.9	Usability Test - Kollisionen pro Hindernis	96

Source-Code Verzeichnis

6.1	Microsoft HoloLens - Erkennung des Bodens	48
6.2	Microsoft HoloLens - Erkennung des Duckens bei Kniebeugen	49
6.3	Microsoft HoloLens - Erkennung des Aufstehens bei Kniebeugen	51
6.4	Microsoft HoloLens - Erkennung von Seitschritten	52
6.5	Microsoft HoloLens - Erkennung von Step-Ups	53
6.6	Microsoft HoloLens - Erkennung von Step-Downs	55
6.7	Microsoft HoloLens - Erkennung von Zehenberührungen	56
7.1	HTC Vive - Erkennung des Bodens	61
7.2	HTC Vive - Erkennung des Duckens bei Kniebeugen	62
7.3	HTC Vive - Erkennung von Seitschritten	64
7.4	HTC Vive - Erkennung von Zehenberührungen	65
8.1	ARCore - Erkennung des Bodens	73

Tabellenverzeichnis

9.1	Detailinformationen des Gerätevergleichs	82
10.1	Detailinformationen der Loggingergebnisse	97

Abkürzungsverzeichnis

AR	Augmented Reality
VR	Virtual Reality
MR	Mixed Reality
NDC	Normalized device coordinates
FOV	Field of view
HMD	Head-Mounted-Display
HUD	Head-up-Display
JSON	JavaScript Object Notation
USB	Universal Serial Bus
XML	Extensible Markup Language

Kurzfassung

Physiotherapie nimmt im Rahmen des täglichen Lebens vieler Menschen bereits eine große Rolle ein. Um diese angenehmer zu gestalten und den Personen den Weg zur Physiotherapie zu ersparen, kann diese auch als Heimphysiotherapie genehmigt werden. Dabei werden die Übungen zu Hause ausgeführt. Die Schwierigkeit hierbei ist, vor allem für ältere Menschen, die Übungen korrekt auszuführen und sich für die Ausführung zu motivieren.

Ziel dieser Arbeit ist es, zu beweisen, dass das physiotherapeutische Training durch den Einsatz von Bewegungserkennung unterstützt werden kann. Außerdem werden zwei High-End Geräte mit einem Smartphone verglichen, um festzustellen, ob das physiotherapeutische Training mittels einem günstigeren Gerät gefördert werden kann.

Im Rahmen dieser Masterarbeit wurden daher drei Applikationen erstellt, welche den üben- den Personen helfen sollen, Motivation zum Üben zu bekommen und vor allem die Übungen richtig auszuführen. Die Anwendungen basieren auf Augmented Reality, Virtual Reality und Mixed Reality. Durch die Durchführung von Usability Tests soll erhoben werden, welche der Anwendungen sich am besten für die Unterstützung von physiotherapeutischem Training eignet.

Das Grundkonzept der Applikationen ist ähnlich, allerdings unterscheidet sich die Art der Bewegungserkennung. Die Bewegungen werden von den Applikationen erkannt und den Nutzern und Nutzerinnen wird sofort visuelles und auditives Feedback gegeben. Um die Bewegungen zu erkennen, wurden mehrere Algorithmen analysiert und einer in dieser Arbeit umgesetzt.

Die Resultate der Usability Tests zeigen, dass die High-End Geräte ähnlich bewertet wurden. Die Erkennung der Übungen und die intuitive Verwendung wurde bei der Smartphone Anwendung nur minimal schlechter bewertet, als bei den anderen Applikationen. Das heißt, die grafische Darstellung der Applikation, sowie die Befestigung des Headsets konnten nicht mit den anderen Anwendungen mithalten. Allerdings ist die Erkennung der Übungen mit dem Smartphone ebenso ausreichend für die Unterstützung von physiotherapeutischem Training. Ebenso wurden im Rahmen der Usability Tests die meisten Stimmen für die beste Unterstützung für die Virtual Reality Anwendung vergeben. Im Gegensatz zu den anderen Applikationen wurden in dieser zusätzliche Sensoren verwendet, um die Erkennungsgenauigkeit zu steigern. Die Nutzer und Nutzerinnen fanden kaum Unterschiede in der Genauigkeit der Erkennung, sondern gaben die Stimmen hauptsächlich aufgrund der erhöhten Immersion ab.

Durch die Umsetzung der Bewegungserkennung und des dauerhaften Feedbacks während den Übungen ist es möglich, die Übungen effizient zu erkennen und den Anwendern und Anwenderinnen ein motivierendes und forderndes Übungserlebnis zu bieten.

Abstract

In the daily lives of many people, physiotherapy already plays a major role. In order to make it more efficient for the people, home physiotherapy can be proposed. If granted, people can perform these physiotherapeutic exercises at home. The difficulty is, especially for the elderly, to do the exercises correctly and to find the motivation to perform them.

The aim of this work is to prove that physiotherapeutic training can be supported by using motion detection. In addition, two high-end devices are compared to a smartphone to determine if physiotherapeutic training can be supported efficiently by using a lower cost device.

As part of this master's thesis, three applications have been created, which should support the physiotherapeutic training. Especially to increase the motivation to perform the exercises as well as to support the correct execution. These applications are based on augmented reality, virtual reality and mixed reality. By conducting usability tests, the application which supports the training best should be selected.

The basic concept of the applications is similar, but the algorithm for the motion detection differs. The person's movement is recognized by the applications and the users are immediately given visual and auditory feedback. In order to detect the movement properly, several algorithms were analyzed.

The results of the usability tests show, that the high-end devices were rated nearly similarly. Compared to the smartphone application, the recognition of the exercises and the intuitive usage was rated only slightly better. This means that the graphical representation of the application, as well as the attachment of the headset could not keep up with the other applications. However, the recognition of the exercises with the smartphone is sufficient to support physiotherapeutic training. In addition, the best application to support the training was chosen. Seventy percent of the users voted for the HTC Vive application. The high immersion was the main reason for this choice.

Concluding, this work states that supporting physiotherapeutic training, by using motion detection and permanent feedback during the exercises, allows to enhance the correctness of the conduction of the exercises and offers the users a motivating and challenging experience.

1 Einleitung

Der Einsatz von Physiotherapie zur Förderung der Regeneration ist ein wichtiger Bestandteil des Alltags geworden. Diese Form der Therapie wird auch als Heimphysiotherapie angeboten, bei der die Übungen von den betroffenen Personen Zuhause ausgeführt werden. Die Schwierigkeiten der Heimphysiotherapie sind vielfältig. Die Hauptprobleme sind allerdings das mangelnde Feedback während des Trainings und die fehlende Motivation die Übungen durchzuführen.

Diese Arbeit fokussiert sich auf die Lösung der Probleme der Heimphysiotherapie. Dafür wurden drei Applikationen erstellt, welche in Mixed Reality (MR), Augmented Reality (AR) und Virtual Reality (VR) eine Basis für das Training bieten sollen. Im Rahmen dieser Applikationen werden die Probleme adressiert, indem die Übungen in eine motivierende und fordernde Umgebung gebracht werden. Zusätzlich zu dieser Funktionalität erkennen die Applikationen die Ausführung der Übungen und geben den Nutzern und Nutzerinnen dauerhaftes Feedback während des Trainings.

Zusätzlich zum visuellen und auditiven Feedback über die Applikationen wurde in dieser Arbeit ein Konfigurationsinterface eingerichtet. Dieses Interface wurde mit Node.js umgesetzt und kann jederzeit von den Therapeuten und Therapeutinnen verwendet werden, um mit den Anwendern und Anwenderinnen der Heimphysiotherapie zu kommunizieren. Dadurch können unter anderem die Schwierigkeiten und die Anzahl der auszuführenden Übungen in Echtzeit angepasst werden. Durch diese Interaktionsmöglichkeit können die Trainingseinheiten im Rahmen der Heimphysiotherapie individuell angepasst werden.

Neben den Problemen der Heimphysiotherapie beschäftigt sich diese Arbeit mit der Frage, ob die Erkennung der physiotherapeutischen Übungen mit den High-End Geräten, Microsoft HoloLens und HTC Vive, sich merkbar von der Erkennung eines herkömmlichen Smartphones unterscheidet. Um dieses Frage beantworten zu können, wurden im Rahmen dieser Arbeit mehrere Messung der Positionserkennung durchgeführt. Außerdem wurde im Rahmen von Usability Tests diese Fragestellung analysiert.

Zusätzlich zu diesen Fragestellungen wird auch das Interfacedesign in MR und AR Applikationen kritisch hinterfragt. Dazu wurde ein Interface entwickelt, welche seine Position dauerhaft anpasst, um im Blickfeld der Nutzer und Nutzerinnen zu sein. Um dieses Ziel zu erreichen, werden diverse Tonquellen benutzt, welche die Position des Interfaces beeinflussen. Das Interface passt seinen Position den Tonquellen an und versucht diese nicht zu verdecken. Die Effektivität dieser Interfaces wird im Rahmen der Usability Tests ausgewertet.

1.1 Aufbau der Arbeit

Aufgrund der Fokussierung auf die Entwicklung der Applikationen werden als erstes diverse Begriffe erklärt. Anschließend wird ein kurzer Einblick in relevante Technologien und Forschungen gegeben. Dabei wird hauptsächlich auf die bereits bestehenden Ansätze der Bewegungserkennung in MR Applikationen eingegangen.

Da für diese Anwendungen bereits im Vorfeld im Kooperationsprojekt Trimotep Fokusgruppen durchgeführt wurden, werden diese relevanten Ergebnisse in Kapitel drei, Fokusgruppen und

1 Einleitung

Applikationsziele, näher beschrieben. Außerdem werden in diesem Kapitel die Zielgruppen und die exakten Ziele der Applikationen definiert.

Kapitel vier, Beschreibung der Applikationen, erläutert die Grundfunktionalität der Applikationen. Zusätzlich werden die erkannten Übungen erklärt und beschrieben, warum diese ausgewählt wurden. Außerdem wird die Umsetzung des Konfigurationsinterfaces näher beschrieben.

In Kapitel fünf und sechs wird auf die Umsetzung der Microsoft HoloLens Applikation, sowie die Bewegungserkennung mit dieser, eingegangen. Dabei werden zuerst die verwendete Hard- und Software beschrieben, bevor das Interfacedesign analysiert wird. Zusätzlich werden mehrere Bewegungserkennungsalgorithmen erläutert und die Erkennung für jede Übung detailliert beschrieben. Außerdem werden die Herausforderungen der Bewegungserkennung dieser Applikation erläutert.

Kapitel sieben geht auf die Bewegungserkennung mit der HTC Vive näher ein. Zuerst wird die verwendete Hard- und Software erläutert, bevor auf die detaillierte Erkennung eingegangen wird. Zusätzlich werden die Herausforderungen der Bewegungserkennung dieses Ansatzes beschrieben.

In Kapitel acht wird die Entwicklung der Letzten der drei Applikationen, die Smartphone Anwendung, erläutert. Dabei wird auf die verwendete Hard- und Software eingegangen sowie die Erstellung der Applikation erklärt. Abschließend werden die Schwierigkeiten der Bewegungserkennung in dieser Anwendung analysiert.

Nach der Beschreibung aller erstellten Applikationen wird in Kapitel neun ein Vergleich der verwendeten Geräte gezogen. Dabei werden Messkriterien und Messverfahren definiert und anschließend die Ergebnisse ausgewertet. Zusätzlich werden die Logdateien der Applikationen dargestellt.

Das zehnte Kapitel beschreibt die Planung, Durchführung und Auswertung der Usability Tests, bevor in Kapitel elf ein Fazit gezogen wird.

1.2 Begriffsdefinitionen

In diesem Unterkapitel werden die elementaren Begriffe dieser Arbeit erläutert. Dabei wird unter anderem auf die Bedeutung, den Ursprung aber auch die aktuelle Verwendung der Ausdrücke eingegangen. Die nachfolgenden Begriffe finden in der gesamten Arbeit Relevanz und werden vermehrt, wie im Abkürzungsverzeichnis gekennzeichnet, abgekürzt.

1.2.1 Realitäts-Virtualitäts-Kontinuum

Das Realitäts-Virtualitäts-Kontinuum beschreibt ein Konzept, welches verwendet wird, um die Verbindung der virtuellen und der realen Welt darzustellen. Wie in Abbildung 1.1 dargestellt, definiert dieses Kontinuum die Abschnitte der echten Welt bis hin zur virtuellen Welt. Dabei beschreibt es Mixed Reality (MR), Augmented Reality (AR) und Virtual Reality (VR) [34].

Augmented Reality (AR)

Das in Abbildung 1.1 dargestellte Realitäts-Virtualitäts-Kontinuum beschreibt unter anderem die augmentierte Realität, welche eine Realität definiert, die um virtuelle Inhalte erweitert wurde. Dadurch erlaubt AR den Nutzern und Nutzerinnen die echte Welt zu sehen und diese nach gegebenen Möglichkeiten zu erweitern. Die Interaktivität in Echtzeit, die Dreidimensionalität und die Kombination von Virtuellem und Realem sind Schlüsselmerkmale von AR.

1 Einleitung

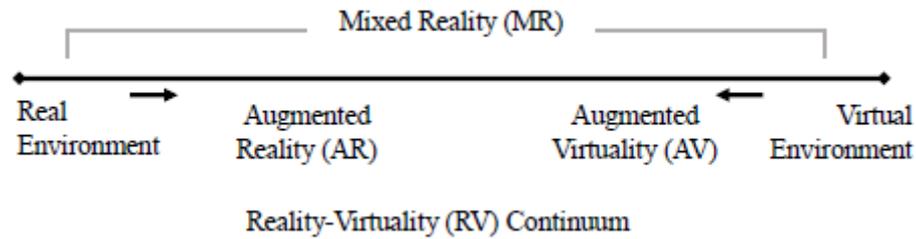


Abbildung 1.1: Realitäts-Virtualitäts-Kontinuum [61]

Kurzum kann AR als Erweiterung der realen Welt durch virtuelle Inhalte gesehen werden. Diese Erweiterungen können in Form von zusätzlich eingeblendeter Informationen oder virtuellen Objekten, mit denen interagiert werden kann, auftreten [3].

Virtual Reality (VR)

Neben der augmentierten Realität beschreibt das in Abbildung 1.1 dargestellte Realitäts-Virtualitäts-Kontinuum auch die virtuelle Realität. Diese bezeichnet im Gegensatz zu AR das komplette Eintauchen in eine virtuelle Welt. In den meisten Fällen geschieht dies mit einem Head-Mounted-Display (HMD). Bei diesem Vorgang wird die reale Welt für die Benutzer und Benutzerinnen ausgeblendet, diese tauchen vollständig in die virtuelle Welt ein. Dadurch kann die Immersion der Nutzer und Nutzerinnen verstärkt werden [43].

Mixed Reality (MR)

Der Begriff MR beschreibt im Gegensatz zu AR und VR nicht ein genaues Stadium des Realitäts-Virtualitäts-Kontinuums, sondern zielt auf die Vermischung und Kombination dieser Realitäten und Virtualitäten ab. Um eine optimale mixed reality Erfahrung erzeugen zu können, ist es notwendig, alle Teilbereiche des Realitäts-Virtualitäts-Kontinuums abzudecken [34]. Das heißt AR und VR decken jeweils einen Bereich des Kontinuums ab, MR hingegen versucht alle Bereiche abzudecken.

Der Bereich der augmentierten Virtualität beschreibt, dass die virtuelle Welt mit Informationen der realen Welt angereichert wird. Der Hauptfokus von MR ist die Kombination aller Bereiche des Realitäts-Virtualitäts-Kontinuums. In der Literatur wird der Begriff AR ähnlich wie MR verwendet, diese Arbeit trennt diese beiden Begriffe allerdings strikt [34].

1.2.2 Degrees of freedom (DoF)

Essenziell für die Einteilung von Devices, welche das Realitäts-Virtualitäts-Kontinuum bedienen, ist die Bestimmung der degrees of freedom. Diese Freiheitsgrade beschreiben die Möglichkeiten, in welchen sich ein Objekt in einem Raum bewegen kann. Um die Einteilung und die Analysen der Devices vornehmen zu können, muss festgestellt werden, wie viele DoF ermöglicht werden [44].

Abbildung 1.2 zeigt die sechs Freiheitsgrade, welche in einem dreidimensionalen Raum existieren. Diese teilen sich in zwei Kategorien auf:

- Rotatorische Bewegungen
- Translationale Bewegungen

1 Einleitung

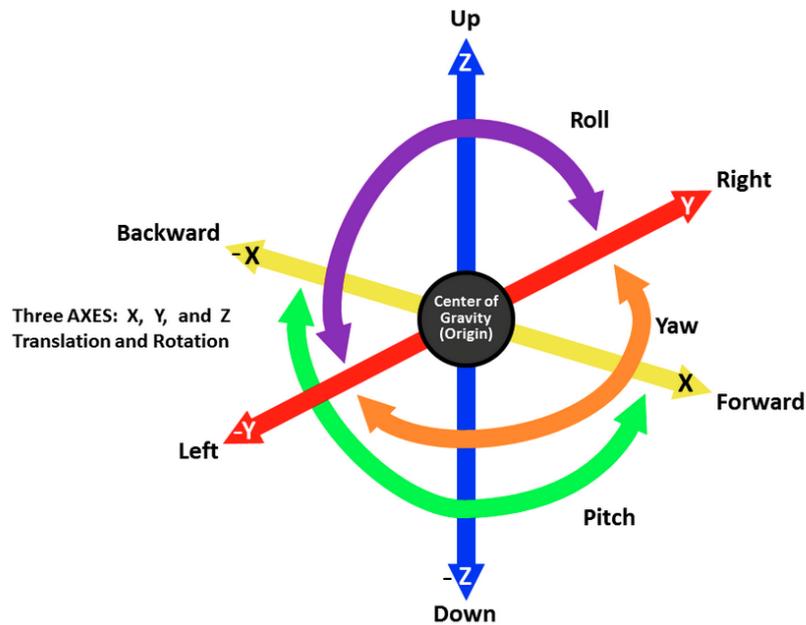


Abbildung 1.2: Sechs Freiheitsgrade [36]

Die rotationalen Bewegungen umfassen das Rollen (roll), Nicken (pitch) und Gieren (yaw). Diese werden meist über On-Board Sensoren, zum Beispiel Akzelerometer oder Gyroskope, von HUDs gemessen. Je nach Kopfbewegung werden diese Messungen aktualisiert [44].

Die translationalen Bewegungen beschreiben Bewegungen nach vor und zurück, auf und ab und links und rechts. Diese Bewegungen werden in den meisten Fällen von einer externen Kamera oder anderen Sensoren analysiert [44].

1.2.3 Field of view (FOV)

Das FOV ist ein Konzept, welches das Feld beschreibt, das von den Nutzern und Nutzerinnen gesehen werden kann. Dieser Bereich bezeichnet die Grenze, indem Informationen über Hinsehen, ohne Bewegen des Kopfes oder der Augen, wahrgenommen werden können [7].

Im Rahmen von MR müssen zwei FOVs berücksichtigt werden. Zum einen was die Nutzer und Nutzerinnen durch ihre eigenen Augen sehen, zum anderen was die Kamera des Geräts sieht. Dieses menschliche Sichtfeld kann nicht verändert werden, daher muss sich die gesamte Anwendung an dieses Feld anpassen. Üblicherweise liegt dieses bei nicht Seh-ingeschränkten Personen horizontal bei 200-220 Grad, wie in Abbildung 1.3 dargestellt wird, und vertikal bei 130-135 Grad [7].

Abbildung 1.3 zeigt das menschliche FOV unterteilt in das Sichtfeld, welches nur mit einem Auge erkennbar ist und das, welches mit beiden Augen wahrnehmbar ist. Gesamt bedeckt dieses Sichtfeld rund 200-220 Grad, die optimale Erkennung gelingt allerdings nur mit beiden Augen, in einem Bereich von rund 114 Grad [25].

Das zweite FOV das berücksichtigt werden muss, ist das des verwendeten Devices. Im Normalfall haben Devices ein geringeres FOV als die Anwender und Anwenderinnen. Diese Diskrepanz kann zu Schwierigkeiten für die Benutzer und Benutzerinnen führen. Die Sichtbarkeit der virtuellen Objekte wird verringert, was zu vermehrter Kopfbewegung der Nutzer und Nutzerinnen führt. Eine weitere Schwierigkeit ist die Überschätzung der Devices, Anwender und Anwenderinnen nehmen an, dass Devices ihr gesamtes FOV unterstützen und werden durch die Diskrepanz enttäuscht [7].

1 Einleitung

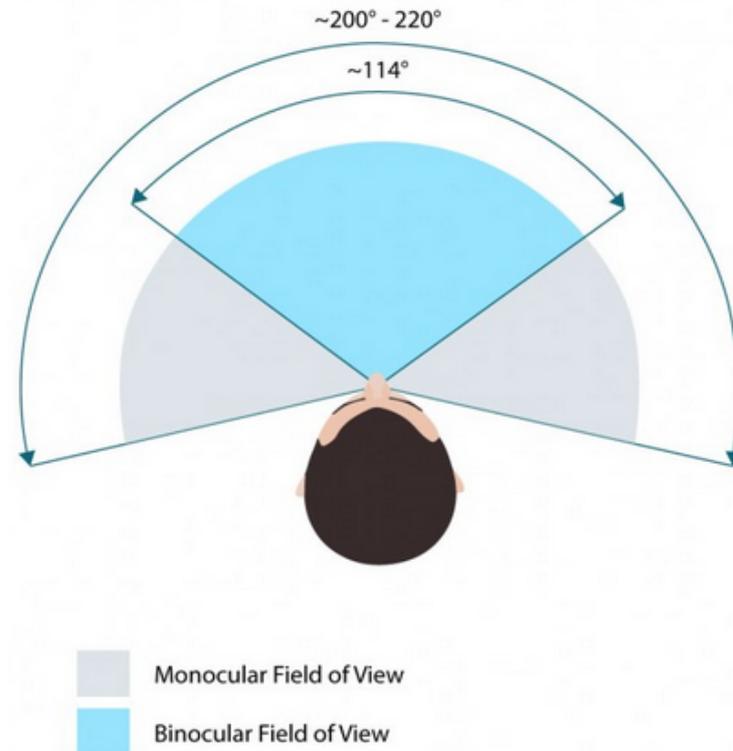


Abbildung 1.3: Menschliches Field of view [25]

Diese Schwierigkeiten werden von MR Devices wahrgenommen und adressiert. Um den zu kleinen und überschätzten FOVs entgegenzuwirken, verwenden MR Devices hauptsächlich drei Optimierungen:

- Spatial audio
- Indikatoren für Aufmerksamkeit
- Tag-along

Mittels Spatial audio, kurz der Lokalisierbarkeit von Audio, eine genauere Definition findet sich in Kapitel vier, Beschreibung der Applikationen, kann die Position und Existenz von Objekten festgestellt werden, ohne dass diese aktuell sichtbar sind [28].

Der Begriff Tag-along beschreibt ein Konzept, welches helfen soll, Inhalt für die Nutzer und Nutzerinnen in MR Anwendung intuitiv darzustellen. Ein Tag-along Objekt versucht, nie aus der Sicht der Anwender und Anwenderinnen zu verschwinden. Dadurch können diese den Inhalt immer nutzen, unabhängig von ihrer Bewegung und Blickrichtung. Abbildung 1.4 zeigt dieses Tag-along anhand einer Bewegung des Kopfes und dem nachfolgenden Menü, welches immer versucht im Sichtfeld zu bleiben [32].

1 Einleitung



Abbildung 1.4: Tag-along Visualisierung [25]

Weitere Optimierungen um die Schwierigkeiten des FOV zu reduzieren sind Indikatoren, welche die Aufmerksamkeit der Nutzer und Nutzerinnen gezielt in die Richtung von Objekten lenken soll. Dadurch soll die Überschätzung des FOVs reduziert werden. Um dem weiter vorzubeugen werden verstärkt Tag-along Mechanismen verwendet, welche Inhalte am Rande des FOVs so positionieren, dass diese mit kleinsten Kopfbewegungen wieder intuitiv sichtbar werden [28].

2 Relevante Technologien und Forschungen

In diesem Kapitel werden diverse Technologien und Forschungen, welche einen enormen Beitrag für diese Arbeit geleistet haben, analysiert und beschrieben. Dabei wird auf die Microsoft HoloLens, eines der neuesten MR Devices, eingegangen. Diese wurde von Evans et. al in einer erstellten Applikation verwendet, um die Neuheiten der ermöglichten Gestenerkennung zu testen. Vielmehr aber wollten sie beweisen, dass über das Spatial Mapping der Microsoft HoloLens eine intuitivere Interaktion möglich wird [10].

Die Microsoft HoloLens ist eine im Jahre 2016 veröffentlichte Hardware, welche ohne Verbindung zu anderer Hardware MR Applikationen unterstützt. Die Brille ermöglicht Nutzern und Nutzerinnen die Interaktion mit virtuellen Inhalten und Hologrammen, welche in die echte Welt integriert sind. Außerdem kann durch dieses Device ein hoher Grad an Realismus erzeugt werden. Dies funktioniert über die mehrdimensionalen Hologramme, welche durch die optische Projektion der Linsen mit sehr geringer Latenz erzeugt und beobachtet werden können. Ein weiteres essenzielles Feature ist die Erkennung der Microsoft HoloLens. Neben Gesten und der Stimme wird auch die Umgebung und die Bewegung der Nutzer und Nutzerinnen mitverfolgt [1]. Eine genauere Beschreibung der Hardware findet sich im Kapitel vier, Beschreibung der Applikationen.

Ein weiteres Schlüsselfeature für diese Arbeit ist die Erkennung von Objekten mittels MR Devices. Diese Objekterkennung wurde von Garon et. al auf der Microsoft HoloLens im Detail untersucht. Dabei stieß deren Arbeit an die Grenzen des Devices. Diese wurden überschritten, indem sie zusätzlich externe Sensoren verwendeten [13].

2.1 Gestenerkennung und Spatial Mapping

Die Gestenerkennung spielt im Rahmen der Interaktion eine wichtige Rolle. Der große Vorteil bei der Verwendung von Gesten als Interaktionsmöglichkeit, ist das Freibleiben der Hände. In vergangenen MR Applikationen mussten die Hände vermehrt die Controller, zum Beispiel Smartphones oder Joysticks, halten, um eine geeignete Interaktion zu gewährleisten [10].

Dabei werden die Gesten der Nutzer und Nutzerinnen über verschiedene Technologien, zum Beispiel Kameras oder Sensoren, erkannt. Ein weiterer Vorteil, neben dem Freibleiben der Hände, ist die Anzahl der möglichen Inputs bei dieser Form der Interaktion. Im Unterschied zur herkömmlichen, begrenzten Inputsituation, sind bei dieser Form unendliche Inputs möglich, da die Vielfältigkeit von Gesten unbegrenzt ist [42].

Die Probleme der Gestenerkennung in der heutigen Zeit sind sehr umfassend. Eine große Rolle spielt die Genauigkeit des Sensors, damit die Gesten optimal erkannt werden können. Hierbei sind die Lichtverhältnisse sowie die Umgebungseigenschaften wichtige Faktoren. Eine weitere Schwierigkeit besteht in der Geschwindigkeit zur Erkennung der Gesten und die Geschwindigkeit der Gesten selbst. [40].

Auch kulturelle Differenzen müssen für die Gestenerkennung überlegt werden. Dies ist essenziell für die Entwicklung von gestengesteuerten Applikationen, da identische Gesten in verschiedenen Ländern und Kulturen verschiedene Bedeutungen haben können [12].

Eine weitere Schlüsseltechnologie im Rahmen dieser Arbeit ist das Spatial Mapping, das ist das Abtasten und Erkennen des Raumes. Durch diese Technologie können reale Inhalte

2 Relevante Technologien und Forschungen

effizient und optimiert in die augmentierte Welt aufgenommen werden. Der Einsatz von Spatial Mapping in dieser Arbeit wird im fünften Kapitel, Beschreibung der Microsoft HoloLens Applikation, näher beschrieben.

Evans et. al haben in ihrer Arbeit eine MR Applikation umgesetzt, welche die Microsoft HoloLens verwendet. Der Fokus der Anwendung liegt auf der intuitiven Nutzung und der Ausreizung der Features der HoloLens. Sie beschreiben einen starken Wandel des Interfacedesigns, welcher mit den Veränderungen der MR Geräte einhergeht. Durch den Wandel von Handheld-Technologien auf HMDs, müssen sich die Interfaces drastisch ändern, um eine intuitive Bedienung der Applikationen beizubehalten [10].

In ihrer Applikation wird die Microsoft HoloLens ausgereizt. Der Hauptgrund für die Auswahl waren die Neuheit und Kabellosigkeit des Gerätes. Das Ziel der Anwendung ist die Auswertung der Kernfeatures der Microsoft HoloLens. Vor allem das Spatial Mapping wird in Ihrer Applikation detailliert analysiert. Im Rahmen dieser Analyse wird das Spatial Mapping dazu verwendet, kleinere Baublöcke zu erkennen [10].

Die Applikation selbst beschreibt eine Baublockanwendung, in der Objekte erkannt werden können und mittels Gestenerkennung neu arrangiert werden. Abbildung 2.1 zeigt das Interface dieser Anwendung, welches von Evans et. al für HMDs optimiert wurde. Die Auswahl der Gegenstände sowie der Menüfeile erfolgt über Gestenerkennung [10].

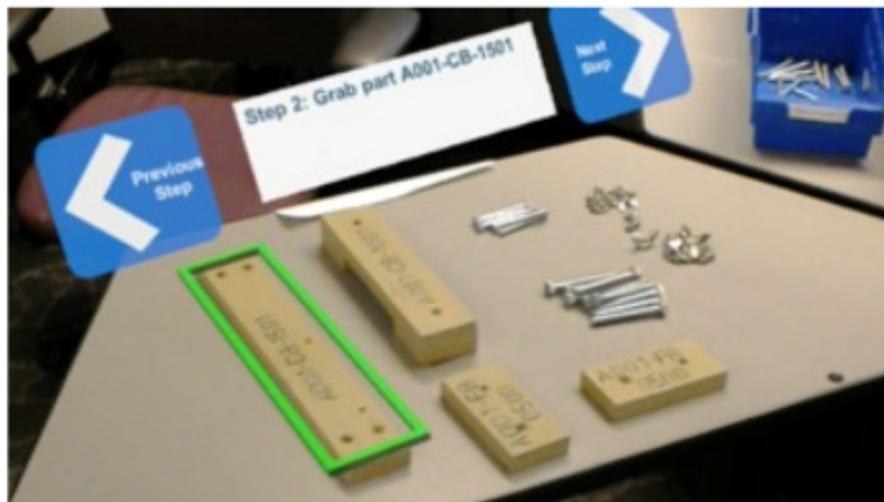


Abbildung 2.1: Interface Design - Baustein Applikation [10]

In ihrer Arbeit beschreiben Evans et. al die Auswahl über Gesten als sehr intuitiv und für HMDs absolut geeignet. Auch die Interaktion mit den virtuellen Menüs werden als einfach und intuitiv empfunden. Darüber hinaus verwenden sie das Spatial Mapping, um die Bauteile zu identifizieren.

Abbildung 2.2 zeigt die Ergebnisse des Spatial Mappings in der Applikation von Evans et. al. Es werden zwar Oberflächen erkannt, allerdings nicht die exakten Blöcke. Für die Anwendung waren diese Ergebnisse unbrauchbar und wurden durch das Tracken von Markern ersetzt. Aus der Arbeit von Evans et. al geht hervor, dass sich das Spatial Mapping der Microsoft HoloLens hervorragend dafür eignet einen Raum oder großflächige Objekte zu tracken, für kleine, präzise Arbeiten sei es allerdings noch zu ungenau [10].

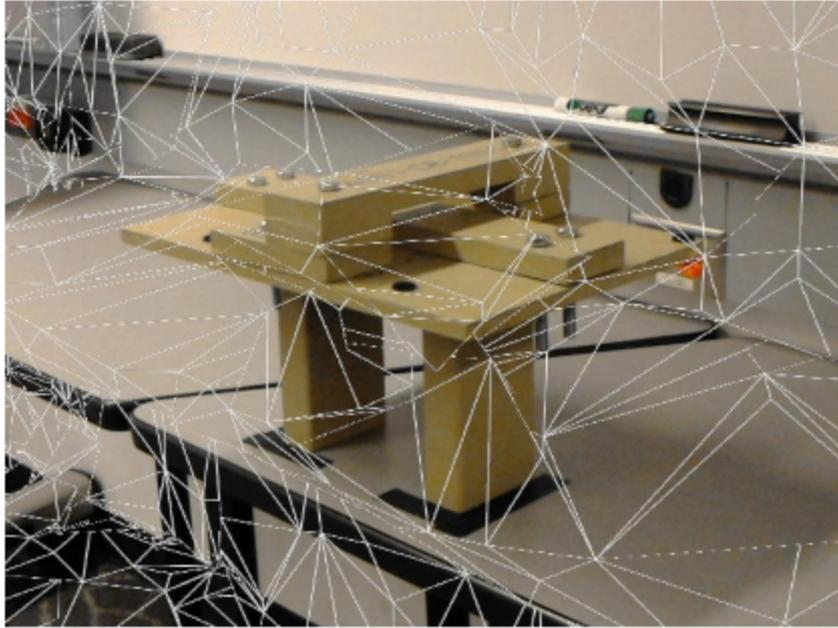


Abbildung 2.2: Spatial Mapping Ergebnisse - Baustein Applikation [10]

2.2 Objekterkennung mit der Microsoft HoloLens

In der Arbeit von Evans et. al wurde die Ungenauigkeit des Spatial Mappings der Microsoft HoloLens angesprochen. Um diesem Problem entgegenzuwirken haben Garon et. al eine Applikation entwickelt, welche diese Erkennung verbessern soll. Das Hauptziel der Applikation war der Vergleich der Objekterkennung von der Microsoft HoloLens und die einer um eine Tiefenkamera erweiterte Variante der HoloLens [13].

Garon et. al beschreiben in ihrer Arbeit die Kabellosigkeit als größten Vorteil der HoloLens, während der fehlende Zugriff auf die Rohdaten der Sensoren für Sie der größte Nachteil ist. Durch die fehlenden Daten gibt es keine Möglichkeit auf die hochauflösenden Tiefeninformationen zuzugreifen und dadurch optimierte und präzise Objekterkennung zu ermöglichen [13].

Um diese präzise Objekterkennung dennoch zu erreichen, verwendeten Garon et. al eine externe Tiefenkamera, welche auf der HoloLens fixiert wurde. Diese Kamera sendet die Informationen an einen Stick, welcher ebenfalls in der HoloLens steckt. Dieser kommuniziert die Daten anschließend mittels kabelloser Netzwerke. Diese Konstruktion ist in Abbildung 2.3 dargestellt.

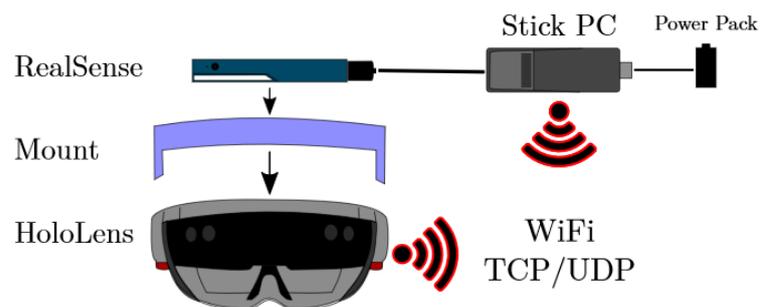


Abbildung 2.3: Hardware Setup - Objekterkennung Applikation [13]

2 Relevante Technologien und Forschungen

Um zu beweisen, dass ihr Ansatz die Objekterkennung präziser und optimierter gestaltet, haben Garon et. al Vergleiche zwischen der unveränderten Aufnahme und ihrer Version gezogen. Für diesen Vergleich wurde eine Szene von dem geänderten Hardwaresetup, in ihrem Fall mit einer RealSense Kamera, und der unveränderten HoloLens aufgenommen. Abbildung 2.4 zeigt die Szene, welche als Referenz für den Vergleich genommen wurde. Die Szene zeigt eine detailreiche Figur und diverse Oberflächen [13].



Abbildung 2.4: Referenzszene - Objekterkennung Applikation [13]

Um einen optimalen Versuch zu gewährleisten, wurde die Szene mit fixierten Kameras aus der gleichen Perspektive gefilmt. Trotz dieser Voraussetzungen ist der Unterschied wesentlich. Ebenso wie Evans et. al konnten auch Garon et. al feststellen, dass das Spatial Mapping der Microsoft HoloLens bei detaillierten, kleinen Objekten noch zu ungenau sei. Die erkannten Objekte spiegelten die Szene nur minimal wieder. Abbildung 2.5 zeigt auf der linken Seite die Aufnahmen mit der unveränderten Microsoft HoloLens [13].

Weiters zeigt Abbildung 2.5. auf der rechten Seite auch die Ergebnisse der modifizierten Version mit der RealSense Kamera. Bei diesem Versuch kamen Garon et. al zu der Erkenntnis, dass die Aufnahme signifikant besser sei und trotzdem noch kabellos. In ihrer Arbeit beschreiben Sie den Fortschritt, welcher erzielt werden konnte, allerdings auch, dass das Ergebnis noch nicht optimal ist und in der Echtzeit-Objekterkennung noch sehr viel Verbesserungspotential zu finden ist.

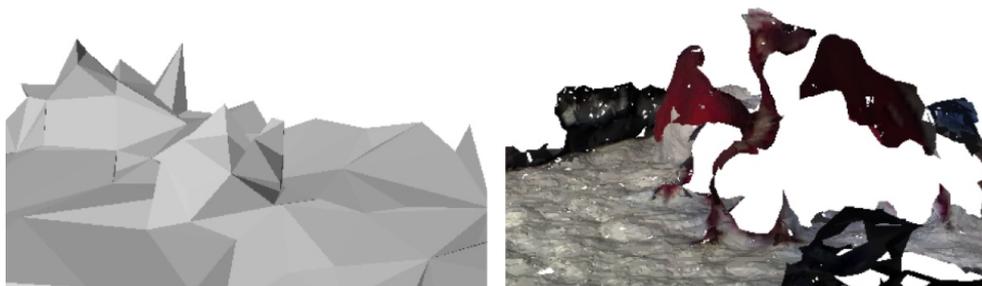


Abbildung 2.5: Vergleichsergebnisse - Objekterkennung Applikation [13]

2.3 Bewegungserkennung in MR Applikationen

Um Bewegungen in MR Applikationen zu erkennen, gibt es diverse Möglichkeiten. Die verschiedenen Methoden bieten unterschiedliche Vorteile und Nachteile. Um diese Ansätze effizient zu verwenden, muss das Ziel der Applikation klar bekannt sein, damit die richtige Wahl getroffen wird. Eine Gemeinsamkeit dieser Methoden ist die intuitive Verwendung für die Nutzer und Nutzerinnen, welche bei der Bewegungserkennung auch im Vordergrund steht.

Bewegungen und Objekte in MR Applikationen genaustens zu erkennen ist essenziell für die intuitive Benutzung der Nutzer und Nutzerinnen. Eines der größten Probleme ist die Positionierung von Objekten in der realen und der virtuellen Welt. Diese müssen übereinstimmen, ansonsten wird die Immersion der User und Userinnen drastisch reduziert. Viele Anwendungen erfordern sogar ein exaktes Erkennen von Objekten und Bewegungen [3].

Im Vergleich zu VR benötigt AR eine viel genauere Erkennung, da die kleinsten Unterschiede in AR den Nutzern und Nutzerinnen augenscheinlich bewusst werden. Wenn Feedback visualisiert wird, sind kleine Unterschiede zwischen der realen und der virtuellen Welt in AR sofort sichtbar. In VR hingegen besteht keine Vergleichsmöglichkeit zur realen Welt und die Unterschiede sind praktisch nicht erkennbar [3].

Unter anderem spielen folgende Faktoren eine signifikante Rolle, um Bewegungen in MR Applikationen effektiv erkennen zu können:

- Optische Verzerrung
- Fehler im Erkennungssystem
- Mechanische Fehleinstellungen

Optische Verzerrungen existieren in den meisten Kamera- und Linsensystemen, sowohl in denen, welche die echte Welt aufnehmen, als auch in den Displays. Diese Verzerrungen sind abhängig vom FOV der Geräte, nahe dem Zentrum sind die Verzerrungen üblicherweise kaum vorhanden. Am Displayrand hingegen können diese verstärkt auftreten. Diese Verzerrungen sind systematische Fehler, das heißt sie können gemessen und erkannt werden. Diese Fehler müssen minimiert werden, um eine effektive Bewegungs- und Objekterkennung zu ermöglichen [3].

Die Fehler im Erkennungssystem sind die verheerendsten der genannten Faktoren, da sie nur schwer zu erkennen und zu messen sind. Diese Abweichungen sind nicht systematisch und schwierig zu definieren. Auch bei diesem Faktor gilt es, die Fehlerquote zu minimieren. Dies wird hauptsächlich durch die erneute Kontrolle der verwendeten Erkennungssysteme erreicht [3].

Der letzte genannte Faktor sind die mechanischen Fehleinstellungen. Diese beschreiben reale physische Abweichungen zwischen der Hardware und den wirklichen Eigenschaften des Systems. Beispielsweise müssen zwei nebeneinander eingebaute Kameras nicht den gleichen Aufnahmewinkel haben. Durch diese Fehleinstellung kann es zu drastischem Mehraufwand kommen, da die Aufnahmen synchronisiert werden müssen. [3].

Ein essenzieller Punkt für die Bewegungserkennung in MR Applikationen ist, dass die Bewegungen der Nutzer und Nutzerinnen nicht eingeschränkt werden. Um ein intuitives und optimales Ergebnis zu erzielen, müssen sich die User und Userinnen vollkommen natürlich bewegen können. Außerdem muss die Erkennung mit minimaler Latenz erfolgen, um die Usability zu steigern. Ebenso führt eine Reduktion der Latenz zu einer effektiven Interaktion [37].

Ein großes Hindernis ist die Genauigkeit der Bewegungserkennung, da bei Fehlerkennungen falsches Feedback an die Benutzer und Benutzerinnen geliefert wird. Diese Abweichung

senkt die Usability und die intuitive Verwendung einer Applikation. Um diese Fehlmessungen vorbeugen zu können, gibt es verschiedene Arten, wie Bewegungen erkannt werden können [37].

Die Verwendung von Positionstrackern, welche an der Kleidung der Nutzer und Nutzerinnen angebracht werden, bieten eine solide Erkennungsbasis. Die festgemachten Sensoren kommunizieren mit einer Hardware und teilen dieser laufend ihre Position mit. Dadurch kann die Position im Raum fast zentimetergenau bestimmt werden. Der große Nachteil dieser Methode ist die aufwendige Befestigung der Sensoren. Um diese effektiv verwenden zu können, müssen diese vor jeder Anwendung individuell bei den Benutzern und Benutzerinnen angebracht werden [37].

Eine weitere Möglichkeit ist die Verwendung von Bilderkennung zur Bewegungserkennung. Im Bereich von MR sind Videobilder aus der Sicht der Nutzer und Nutzerinnen verfügbar. Allerdings muss aus diesen üblicherweise zweidimensionalen Aufnahmen auf dreidimensionale Bewegungen geschlossen werden. Um diese Varianten alle abzudecken und die Bilder effizient zu analysieren, müssen diese korrekt interpretiert werden. Dafür ist ein enormes Maß an Rechenleistung erforderlich. Um diesen Schritt zu vereinfachen, werden für gezielte Anwendungen Referenzbilder angefertigt, welche die benötigte Rechenleistung senken sollen, indem die Ergebnisse effizienter gefunden werden. In einem reinen TestszENARIO, unter optimalen Voraussetzungen, kann eine fast perfekte Erkennung mit diesem Ansatz erreicht werden. Ein großes Hindernis dieses Ansatzes ist allerdings die Instabilität in echten Nutzungsumgebungen. Um die Rechenleistung zu sparen, werden viele Vorhersagen, zum Beispiel auf Kamera- oder User- und Userinnenbewegungen, getroffen. Aufgrund dieser Annahmen sind diese Systeme sehr anfällig für rapide, unvorhersehbare Bewegungsänderungen. Sollten diese Vorhersagen falsch sein und falsch bemessen werden, würde für die Nutzer und Nutzerinnen unzureichendes Feedback gegeben und dadurch die intuitive Verwendung eingeschränkt werden [37].

Ein weiterer Aspekt der Bewegungserkennung in MR Applikationen ist das Erkennen mehrerer parallel ausgeführter Bewegungen. Aktuell ist das Erkennen von Bewegungen einzelner Personen oder Objekte ein wesentlich leichteres Unterfangen als das Erkennen von Bewegungen mehrerer Personen oder Objekte. Grund dafür ist die Leistung und Genauigkeit, welche benötigt werden, um dies effektiv zu tracken. Ein weiteres Hindernis ist die Abgrenzung von den Bewegungsräumen. Es muss festgestellt werden, ob mehrere Bewegungsursprünge vorhanden und unabhängig voneinander sind [38].

Abbildung 2.6 zeigt die Ergebnisse der Erkennung von Park et. al. Es lässt sich erkennen, dass die erkannten Objekte hervorgehoben werden. Aus dieser Abbildung geht die Erkennbarkeit während der Bewegung nicht hervor, allerdings beschreibt die Arbeit eine stabile Erkennung auch während der Bewegung.

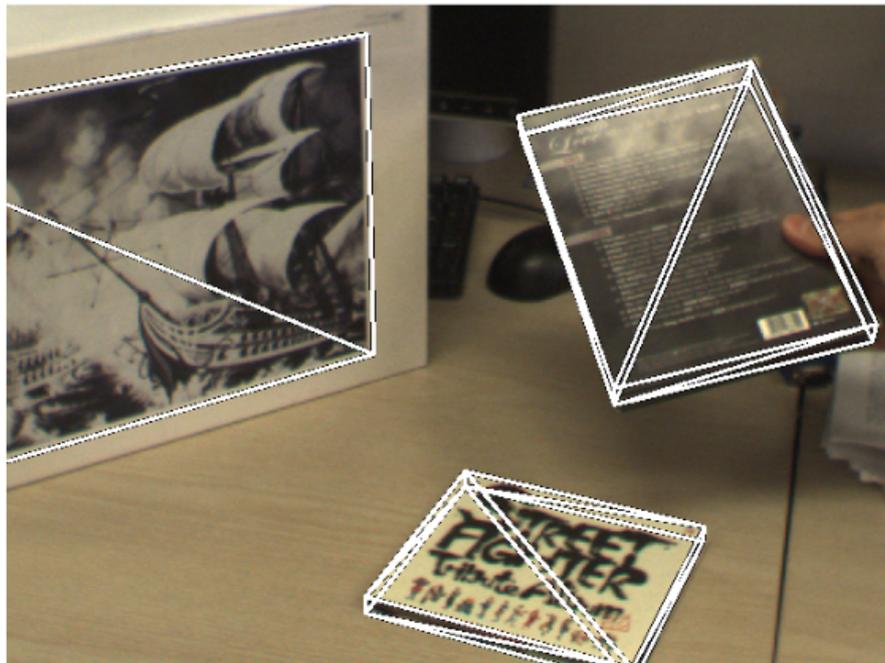


Abbildung 2.6: Multiple Objekterkennung [38]

Um diese Art der Bewegungserkennung umzusetzen, wurde in der Arbeit von Park et. al ein Ansatz verwendet, welcher mittels Feature Extrahierung und Feature Tracking versucht, effizient die Bewegungen zu erkennen. Feature Erkennung ist sehr Zeit- und Leistungsintensiv. Um diese Faktoren zu optimieren, verwenden Park et. al das Feature Tracking, welches weniger Rechenleistung benötigt [38].

Eine weitere Form der Bewegungserkennung ist ein markerbasierter Algorithmus. Diese ähnelt der von Parker et. al verwendeten Form und bietet auch die Möglichkeit mehrerer Trackingziele. Dabei werden vordefinierte Marker auf den realen Trackingobjekten angebracht. Diese werden mittels Feature Erkennung und Feature Tracking erkannt und verfolgt. Durch die Vordefinition der Marker werden die Zeit- und Leistungsintensivität drastisch reduziert. Der Aufwand der Vorbereitung erhöht sich, da die Marker auf den jeweiligen Objekten angebracht werden müssen. Dieser Ansatz ist mit individuellen Menschen nur schwer umsetzbar, da um diese zu tracken, an verschiedenen Körperstellen Marker angebracht werden müssten [30].

2.3.1 Inside-Out Bewegungserkennung

Neben den Ansätzen, die auf Marker basieren, gibt es noch weitere markerlose Ansätze zur Bewegungserkennung und zum Bewegungstracking. Der erste dieser beiden vorgestellten Methoden ist die Inside-Out Methode. Diese Methode wird zur Erfassung der Position innerhalb einer MR Anwendung benutzt. Dabei wird die Position von dem HMDs und den Hardware Controllern permanent verfolgt und gemessen. Der Ursprung des Trackings befindet sich in der Kamera in den HMDs, welche die Nutzer und Nutzerinnen auf den Köpfen tragen. Dieses Tracking misst mittels Sensoren, wie sich die Position in Relation zur Umgebung verändert [46].

Abbildung 2.7 zeigt das Setup für das Inside-Out Tracking. Die Kamera befindet sich auf dem Kopf der Nutzer und Nutzerinnen, wie bereits erläutert, meistens in der Form eines HMDs. Über die eingebauten Sensoren in den Devices, können unterschiedlich viele Freiheitsgrade erreicht werden. Die Microsoft HoloLens erreicht sechs Grade der Freiheit. Die Sensoren

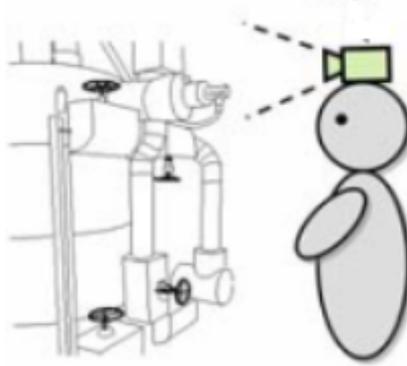


Abbildung 2.7: Inside-Out Tracking [23]

erkennen die Veränderungen in der realen Umgebung, analysieren diese und wandeln diese Information in Positionsdaten um. Auch in dieser Kombination können Marker verwendet werden, um die Erkennung für die Sensoren zu vereinfachen. Im Inside-Out Tracking basiert die gesamte Positions- und Bewegungserkennung auf der Erkennung der Umwelt [46].

Inside-Out Tracking wird von diversen Devices verwendet, zum Beispiel von der:

- Microsoft HoloLens
- Google Cardboard

2.3.2 Outside-In Bewegungserkennung

Das Gegenteil zum Inside-Out Tracking ist die Outside-In Positions- und Bewegungserkennungsmethode. Diese wird zur Erfassung der Position innerhalb einer MR Anwendung benutzt. Dabei wird die Position der Nutzer und Nutzerinnen von externen, im Raum befestigten Sensoren permanent verfolgt und gemessen. Der Ursprung des Trackings befindet sich daher in den externen Sensoren, welche in der realen Umgebung angebracht sind. Dieses Tracking ermittelt die Position der Nutzer und Nutzerinnen durch die Messwerte der Sensoren, welche ausgewertet und die Positionsdaten dadurch berechnet werden [51].

Outside-In Tracking wird von diversen Devices verwendet, zum Beispiel von der:

- Nintendo Wii Remote
- HTC-Vive

Abbildung 2.8 zeigt das Setup für das Outside-In Tracking. Die Kameras befinden sich an fixierten Positionen in der realen Umwelt der Nutzer und Nutzerinnen. Über die benutzten Sensoren für das Tracking können unterschiedlich viele Freiheitsgrade erreicht werden. Die Sensoren erkennen die Veränderungen in der realen Umgebung, analysieren diese und wandeln diese Information in Positionsdaten um. Im Outside-In Tracking basiert die Erkennung der Positionen und Bewegungen auf der Erkennung der Nutzer und Nutzerinnen [51].

Sowohl beim Inside-Out als auch beim Outside-In Tracking werden nicht nur die Kameras der Geräte beziehungsweise der Sensoren verwendet um Informationen über das Gerät zu erlangen, sondern auch die im Gerät integrierten Sensoren selbst. Üblicherweise verfügen diese Geräte über Lagesensoren, welche die Rotation der Geräte wiedergeben und Beschleunigungssensoren, welche die physische Beschleunigung des Gerätes messen.

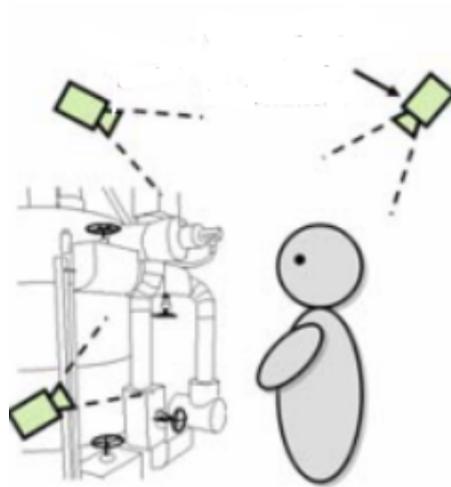


Abbildung 2.8: Outside-In Tracking [23]

2.4 Schwierigkeiten von VR

Im Rahmen dieser Arbeit wurden auch die aktuellen Schwierigkeiten von VR untersucht. Neben dem bereits erwähnten Problem, dass die Positionierung der virtuellen und realen Objekte übereinstimmen muss, gibt es in der virtuellen Welt zusätzlich noch die sogenannte Motion Sickness.

Die Positionierung der virtuellen Objekte ist in der VR Welt essenziell. Durch die gezielte Positionierung kann eine erhöhte Immersion für die Nutzer und Nutzerinnen erzeugt werden. Fehlt diese optimale Ausrichtung der virtuellen Umgebung, wird das Erlebnis nicht mehr überzeugend wahrgenommen [43].

Wenn in einer VR Applikation reale Gegenstände verwendet werden, muss sich die Position der realen Objekte mit denen der virtuellen Objekte decken. Durch die Kombination dieser beiden Faktoren können reale Gegenstände in die virtuelle Welt mit eingebunden werden und steigern dadurch die Immersion [43].

Sollte diese Übereinstimmung nicht vorhanden sein, sehen Nutzer und Nutzerinnen eventuell die Gegenstände an der falschen Position und greifen beziehungsweise treten ins Leere [43]. Dies hat nicht nur den Effekt, dass man die Glaubhaftigkeit der virtuellen Welt in Frage stellt, sondern kann auch gefährlich sein, falls die realen Objekte Hindernisse darstellen.

Eine weitere große Schwierigkeit von VR ist die Motion Sickness. Diese wird im deutschsprachigen Raum auch als VR-Krankheit bezeichnet und steht für eine Form von Übelkeit, welche mit dem Eintauchen beziehungsweise dem Bewegen in der virtuellen Umgebung auftritt [8].

Diese VR-Krankheit entsteht üblicherweise bei virtuellen Bewegungen, welche mit den realen Bewegungen nicht übereinstimmen. Diese Abweichung wird vom menschlichen Körper falsch interpretiert und erzeugt eine physische Abwehrhaltung. Dieser Effekt kann nur von kurzer Dauer sein, aber auch erst einige Stunden nach der Verwendung der VR Applikation auftreten [8].

Die häufigsten Symptome der VR-Krankheit sind [39]:

- Übelkeit
- Schwindel
- Brechreiz
- Schweißausbrüche

Um der VR-Krankheit entgegenzuwirken, ist neben der korrekten Ausführung der Bewegungen im virtuellen Raum auch das richtige Sichtfeld und der Blickwinkel entscheidend. Diese Faktoren müssen für die Nutzer und Nutzerinnen ein optimales Zusammenspiel zwischen Bewegung und Darstellung bieten. Zusätzlich spielt auch die Dauer, welche die User und Userinnen in der VR Applikation verbringen, eine essenzielle Rolle für die VR-Krankheit [39].

2.5 Microsoft HoloLens für Betroffene von Alzheimer

Eine weitere Arbeit, welche für diese Thematik äußerst spannende Ergebnisse aufzeigt, ist das wissenschaftliche Paper von Aruanno et. al. In ihrem vorgestellten Ansatz beschreiben sie eine Applikation, welche auf der Microsoft HoloLens als Hardware basiert. Sie fokussieren sich darauf drei MR Applikationen für eine spezielle Zielgruppe, die von Alzheimer betroffen sind, zu erstellen. Diese Anwendung wurde in Kooperation mit Neurologen erstellt und soll das Kurzzeitgedächtnis und die räumliche Orientierung stärken. Durch die Verbesserungen in diesen beiden Bereichen sollen die Schwierigkeiten der Betroffenen vermindert werden [1].

Der erste Schritt im Entstehungsprozess der Applikationen für Aruanno et. al war es, die Bedürfnisse der Betroffenen und die der Therapeuten und Therapeutinnen zu analysieren. Dafür wurde eine Fokusgruppe organisiert, in welcher die Therapeuten und Therapeutinnen die Microsoft HoloLens anwenden konnten und die Bedürfnisse erarbeitet wurden. Die Vorschläge und Ideen für die Applikationen und Übungen wurden in dieser Fokusgruppe erarbeitet [1].

Einer der wichtigsten Punkte, der in der Fokusgruppe erarbeitet wurde, war die Verständlichkeit der Applikationen für die Nutzer und Nutzerinnen. Da die Zielgruppe durch die Krankheit bereits eingeschränkt ist, muss die Anwendung die Probanden und Probandinnen unterstützen und ein verständliches, für sie intuitives, Interface bieten. Neben dieser Verständlichkeit darf allerdings der Trainingsaspekt nicht vernachlässigt werden. Dennoch muss auf die Schwierigkeiten der älteren Menschen, zum Beispiel die schwächeren Seh- und Gehörsinne, Rücksicht genommen werden [1].

Die Entscheidung für die Microsoft HoloLens wurde von Aruanno et. al getroffen, weil es die Nebeneffekte, welche VR Anwendungen aufweisen können, nicht gibt. Außerdem bleiben die Nutzer und Nutzerinnen in ihrer bekannten Umgebung und können weiterhin mit den Therapeuten und Therapeutinnen interagieren. Ein weiterer Grund war die intuitive Nutzung der Applikationen. In der realen Welt sehen Menschen Dinge üblicherweise an, wenn sie mit diesen interagieren. In den Anwendungen von Aruanno et. al wird der Blickvektor der Nutzer und Nutzerinnen verwendet, um festzustellen, wohin diese ihren Blick in den Applikationen werfen. Abbildung 2.9 zeigt den Ansatz für das simple Interface, um den Blickvektor der Nutzer und Nutzerinnen effektiv darzustellen. Das angesehene Objekt wird zusätzlich eingefärbt, um die Bedienung zu erleichtern [1].

Um das angesehene Objekt auszuwählen, ist die Ausführung einer definierten Geste notwendig. Dabei müssen die Nutzer und Nutzerinnen ihren Zeigefinger heben, schnell nach unten bewegen und wieder anheben. Diese Bewegung muss im FOV der Microsoft HoloLens erfolgen, um erkannt werden zu können. Die zweite Möglichkeit das fokussierte Objekt auszuwählen, ist die Nutzung der Sprachsteuerung. Diese Art der Auswahl wurde von Aruanno et. al. erst nach umfangreichen Nutzertests hinzugefügt, da die älteren Personen Schwierigkeiten bei der Ausführung der Gesten hatten [1].

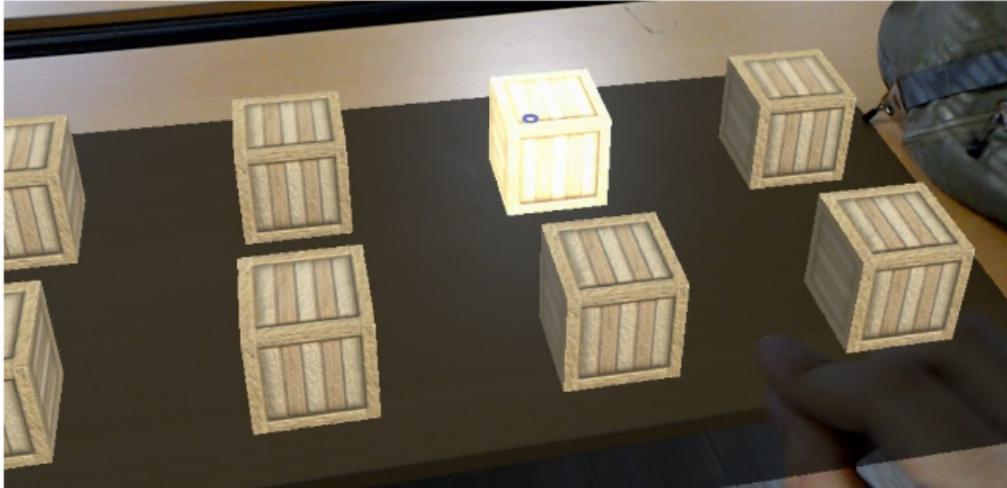


Abbildung 2.9: Visualisierung des Blickvektors [1]

Die erste Applikation

Die erste Anwendung ist die simpelste Anwendung, sie dient dem Aufwärmen und dem Training für das Kurzzeitgedächtnis. Nach erfolgreichem Start wird den Nutzerinnen und Nutzern ein Referenzobjekt gezeigt, welches diese sich merken müssen. Nachdem sich die Anwender und Anwenderinnen dieses eingepägt haben, müssen sie bestätigen, dass sie bereit sind die Übung zu starten. Bei der Übung wird ein Spielfeld, wie in Abbildung 2.10 dargestellt, mit acht Boxen benutzt. Unter jeder dieser Boxen, die mit Auswählen geöffnet werden können, befindet sich ein Objekt. Bei jedem Öffnen müssen die Nutzer und Nutzerinnen beantworten, ob es sich bei dem gefundenen Objekt um das Referenzobjekt handelt. Bei erfolgreichem Abschluss kann zur zweiten Applikation übergegangen werden [1].



Abbildung 2.10: Darstellung des Spielfeldes [1]

Die zweite Applikation

Die zweite Anwendung ist eine Erweiterung der ersten Anwendung, welche sich an dem bekannten Spiel Memory orientiert. Dabei müssen die Nutzer und Nutzerinnen zwei identische Objekte finden, welche sich in den Boxen verstecken. Als Zusatz müssen die Patienten und Patientinnen noch bestätigen, ob sie sicher sind, dass die beiden Objekte übereinstimmen. Für diese Anwendung wird ebenfalls das Spielfeld, welches in Abbildung 2.10 dargestellt ist, verwendet. Ziel dieser Übung ist die Verbesserung des Kurzzeitgedächtnisses [1].

Die dritte Applikation

Die dritte Applikation ist der zweiten sehr ähnlich, allerdings soll diese zusätzlich das räumliche Gedächtnis trainieren. Die Nutzer und Nutzerinnen müssen sich nicht nur die Objekte merken, sondern auch wo diese im Raum um sie herum platziert sind. Die Logik der Übung ist die selbe wie die der zweiten Übung. Die Patienten und Patientinnen müssen zwei identische Objekte finden, indem sie die Boxen öffnen. Allerdings sind die Boxen nicht mehr innerhalb von zwei Reihen aufgestellt, sondern sind im Raum verteilt, wie in Abbildung 2.11 dargestellt. Die Anwender und Anwenderinnen müssen sich im Raum umsehen um die Boxen zu finden. Um die Boxen effektiv im Raum verteilen zu können, wird das Spatial Mapping der Microsoft HoloLens verwendet, welches den Raum um die Nutzer und Nutzerinnen erkennt. Auf dieses Feature wird in Kapitel fünf, Beschreibung der Microsoft HoloLens Applikation, detailliert eingegangen [1].



Abbildung 2.11: Spielfeld der dritten Anwendung [1]

Die Ergebnisse von Aruanno et. al umfassen eine Vielzahl von Erkenntnissen. Während der Entwicklung der Prototypen wurden diverse Probleme gelöst, unter anderem die Neuheit der Microsoft HoloLens. Außerdem war diese Arbeit die erste, welche sich mit dem Interface-design für Betroffene der Alzheimer Krankheit beschäftigt hat. Das Paper von Aruanno et. al kommt zu der Erkenntnis, dass MR Anwendungen in diesem Bereich noch sehr unerforscht sind und bietet Vorschläge für Prototypen, wie diese Anwendungen in Zukunft umgesetzt werden könnten [1].

3 Fokusgruppen und Applikationsziele

In diesem Kapitel werden die Fokusgruppen und die Applikationsziele näher beschrieben. Es wird im Detail darauf eingegangen, welche Nutzer und Nutzerinnen als Zielgruppe definiert werden und die Problemstellungen beschrieben.

3.1 Ziele der Applikationen

Das Ziel der Applikationen ist die Erkennung und Erfassung von definierten physiotherapeutischen Übungen. Diese Übungserkennung dient der Unterstützung der Physiotherapie im Bereich der Heimphysiotherapie. In Kooperation mit mehreren Physiotherapeuten hat sich in der Konzeption dieser Arbeit ergeben, dass der Erfolg der Heimphysiotherapie stark unter der Unsicherheit der betroffenen Personen leidet.

Diese Personen untergehen erst eine herkömmliche Physiotherapie im Physiotherapiezentrum, unter stetiger Begleitung durch einen Physiotherapeuten. Unter diesen Umständen werden sie zur Ausführung angeleitet und bekommen laufend Feedback während den Übungen. Dadurch fühlen sich die übenden Personen sicher und motiviert.

Im Rahmen der Heimphysiotherapie fällt diese laufende Anleitung und das Feedback weg. Dadurch fühlen sich betroffene Personen unsicher und verzichten oftmals auf das Training, aus Angst etwas falsch zu machen. Manche Übende gaben sogar an, sich vor Verletzungen zu fürchten. Den betroffenen Personen mangelt es an laufender Betreuung und Feedback.

Um diesen Problemen entgegenzuwirken wurden die Applikationen, die im Rahmen dieser Arbeit beschrieben werden, erstellt. Das Ziel dieser Arbeit ist die Analyse, inwiefern diese Applikationen die erläuterten Probleme erleichtern oder gar beheben könnten. Um diese Frage zu adressieren, wird im Laufe dieses Papers ein Usability Test durchgeführt. Dieser wird in Kapitel zehn, Usability Tests, ausführlich beschrieben.

Die Applikationen adressieren jeden dieser Ansprüche der betroffenen Personen sehr individuell. Das Problem des fehlenden Feedbacks wird in den Applikationen in Form von mehreren Interfaces realisiert. Über diese Interfaces bekommen die übenden Personen laufend Rückmeldungen zu ihrem Fortschritt, ihrer Übungsgenauigkeit und dem aktuellen Status. Um das Feedback optimal zu gestalten, wird dieses sowohl auditiv als auch visuell übermittelt.

Der Anspruch nach stetiger Betreuung wird von den Applikationen über die Kommunikation mit einem Webserver realisiert. Dabei kommuniziert das Device der übenden Personen mit einem Webserver, welcher direkt von den Therapeuten und Therapeutinnen über eine Webseite gesteuert werden kann. Diese direkte Form der Kommunikation ermöglicht für die Therapeuten und Therapeutinnen eine besser Form der Nachvollziehbarkeit und für die übenden Personen eine bessere Auslastung bei der Übungsdurchführung.

Diese Kommunikation erlaubt den Therapeuten und Therapeutinnen die Trainingssessions zu überwachen und in Echtzeit anzupassen. Die Physiotherapeuten und Physiotherapeutinnen könnten dort unter anderem die Schwierigkeiten der Übungen einstellen und die Zielanzahl konfigurieren. Über diese Konfigurationen kann das Training individuell an die einzelnen Nutzer und Nutzerinnen abgestimmt werden.

Durch diese individuelle Einstellung über die Therapeuten und Therapeutinnen soll die Angst der Nutzer und Nutzerinnen reduziert werden. Außerdem kann über diese dauerhafte

Kommunikation die Nähe zu den Therapeuten und Therapeutinnen simuliert werden.

Die Applikationen fokussieren sich im Besonderen auf folgende Ziele:

- Die Erkennung von definierten Übungen. (Kniebeuge, Seitschritt, Step-Up, Berühren der Zehen)
- Die automatische Anpassung des Interfaces mittels Spatial Audio und Spatial Mapping.
- Das Ermöglichen von Echtzeit-Anpassung durch die Therapeuten und Therapeutinnen.
- Das Bereitstellen von individuellem Feedback nach Abschluss des Trainings.
- Das Wiedergeben von optimiertem visuellem und auditivem Feedback während des Trainings.

3.2 Zusammenhang zur Problemstellung

Diese Applikationen sollen die bereits erklärten Probleme ansprechen und lösen. Die Hauptprobleme sind das fehlende Feedback während den Übungen, die Angst vor Fehlern und die fehlende Betreuung durch die Therapeuten und Therapeutinnen. Über die Applikationen werden diese Hindernisse explizit und individuell angesprochen und versucht diese zu reduzieren.

Die User und Userinnen der Applikationen werden im Rahmen eines Usability Tests zur intuitiven Nutzung als auch zur Verbesserung der erläuterten Problemfaktoren befragt. Dieser Usability Test wird mit den verschiedenen Devices und Applikationen durchgeführt. Dadurch soll ausgewertet werden, welchen Einfluss die Applikationen auf das physiotherapeutische Training der übenden Personen haben.

Durch die eingebauten Hindernisse in den Applikationen bieten diese die Möglichkeit, die Motivation für die Ausübung der Übungen zu erhöhen. Diese Hindernisse erfordern von den übenden Personen Flexibilität und Reaktionsfähigkeit. Durch diese Erhöhung der Schwierigkeit wird die Reaktionsfähigkeit von den betroffenen Personen verbessert.

3.3 Definition und Beschreibung der Zielgruppen

In diesem Unterkapitel werden die Zielgruppen für die Applikationen beschrieben und die Auswahl erläutert. Für die Applikationen wurde eine enge Zusammenarbeit mit Physiotherapeuten und Physiotherapeutinnen gepflegt, um diese für die User und Userinnen optimal gestalten zu können.

Die Zielgruppen für die Applikation sind vielschichtig. Hauptsächlich orientieren sich die Applikationen an Personen zwischen 30 und 50 Jahren, welche Heimphysiotherapie genehmigt bekommen. Dieser Zielgruppe wird bereits eine gewisse Affinität für die Technik zugeschrieben, daher ist die technische Barriere hier nur gering einzuschätzen [26].

Diese Zielgruppe absolviert die Heimphysiotherapie mit dem Ziel innerhalb kurzer Zeit wieder vollkommen regeneriert zu sein. Eine der größten Hindernisse dieser Zielgruppe ist die fehlende, langfristige Motivation für die Übungen. Die Personen der Zielgruppe werden nachlässig mit dem Training und brauchen Einfluss von außen, um die Heimphysiotherapie effektiv auszuführen [26].

Außerdem versuchen die Personen dieser Zielgruppe mit ihren Physiotherapeuten und Physiotherapeutinnen in Kontakt zu bleiben, um den Fortschritt laufend zu kontrollieren. Durch die regelmäßige Fortschrittsbesprechung erhoffen sie sich ein besser abgestimmtes Training und dadurch einen schnelleren Regenerationsprozess. Die Unterstützung des Trainings durch

diverse Hard- und Software stellt für diese Zielgruppe nur eine geringe Herausforderung dar [26].

Eine weitere Zielgruppe sind Personen zwischen 51 und 61 Jahren, welche Heimphysiotherapie genehmigt bekommen. In dieser Zielgruppe ist die Technikaffinität etwas geringer als in der vorher definierten Zielgruppe. Auch in diesem Probandenkreis ist das Ziel in kurzer Zeit vollkommen regeneriert zu sein [26].

Die letzte Zielgruppe umfasst Personen die älter sind als 61 Jahre, welche Heimphysiotherapie genehmigt bekommen. In dieser Zielgruppe ist die Technikaffinität sehr gering. Dadurch ist es für diese Zielgruppe essenziell, dass die Applikationen so intuitiv wie möglich gestaltet werden, um jeglichen Umgang für die Personen zu erleichtern [26].

Diese Zielgruppe zielt nicht auf die Regeneration in kürzester Zeit ab, sondern auf die allgemeine Wiederherstellung der Funktionstüchtigkeit ihres Körpers. Ihr Ziel ist es, sich durch die Heimphysiotherapie fit zu halten und ihrem täglichen Leben gerecht zu werden. Die hauptsächlichen Probleme sind nicht die Übungsgenauigkeit, wie bei den anderen beiden Gruppen, sondern die Angst davor, die Übungen falsch auszuführen und sich zu verletzen [26].

3.4 Beschreibung des Inputs der Zielgruppen

Um den Input der Zielgruppen bezüglich dieser Applikationen zu analysieren, wurden in einem Kooperationsprojekt Trimotep Fokusgruppen durchgeführt. Diese Fokusgruppen hatten das Ziel, die Bedürfnisse der Zielgruppe zu erarbeiten und näher zu betrachten. In diesen durchgeführten Befragungen und Studien wurde auf die Technikfeindlichkeit von älteren Menschen eingegangen und erforscht, ob eine Technikakzeptanz durch User Experience und Usability erzeugt werden kann [26].

Die Befragung zur Erhebung der qualitativen Daten erfolgte in zwei Fokusgruppen:

- Fokusgruppe mit Gesundheitsexperten und Gesundheitsexpertinnen
- Fokusgruppe mit Hüft-Total-Endo-Prothesen Patienten und Patientinnen

Für beide dieser Fokusgruppen ist der Ablauf gleich. Zuerst wird ihnen näher erklärt, welche Thematik behandelt wird, bevor auf allgemeine Fragen, zum Beispiel zur aktuellen Heimphysiotherapiesituation oder zur Übungsmotivation, eingegangen wird. Anschließend wird das Projekt den Teilnehmern und Teilnehmerinnen präsentiert und anschließend auf spezifische Fragen geantwortet. Als letzter Schritt wird den Fokusgruppenmitgliedern ein Ausblick und eine Erklärung weiterer Schritte gegeben [26].

Die Daten der Fokusgruppen wurden im Rahmen dieser Treffen erhoben und verarbeitet. Neben den Fokusgruppen wurden auch zwei Experteninterviews mit Therapeuten und Therapeutinnen durchgeführt. Durch die Gewinnung dieser Daten konnten diese aufgearbeitet und in ein Kategoriensystem eingeteilt werden. Im Rahmen dieser Bearbeitung wurden Personas erstellt, welche auf die Zielgruppe, welche im vorherigen Unterkapitel definiert wurde, explizit zugeschnitten worden sind [26].

Die Ergebnisse der Fokusgruppen im Bereich Heimphysiotherapie weisen, sowohl von den Therapeuten und Therapeutinnen, als auch von den Patienten und Patientinnen befürwortet, eine enorme Wichtigkeit der persönlichen Betreuung auf. Es ist beiderseits gewünscht, dass nach der Entlassung aus der Rehabilitation das Training unter Beaufsichtigung fortgeführt werden kann [26]. Diese Beaufsichtigung gestaltet sich derzeit aus Sicht der Therapeuten und Therapeutinnen äußerst schwierig, da diese die individuelle Betreuung im Rahmen von

3 Fokusgruppen und Applikationsziele

Hausbesuchen nicht anbieten können. In diesem Feld wird von beiden Seiten stark nach einer Optimierung der Kommunikationsmöglichkeiten verlangt.

Die Personen, welche Heimphysiotherapie genehmigt bekommen, verlangen eine Intensivierung des Austausches mit den Physiotherapeuten und Physiotherapeutinnen. Für die Zielgruppe sind die Therapeuten und Therapeutinnen wichtige Ansprechpartner, welche den gesamten Prozess begleiten. [26].

4 Beschreibung der Applikationen

In diesem Kapitel wird die Grundfunktionalität der erstellten Applikationen beschrieben. Diese Funktionalität ähnelt sich in allen drei Applikationen, ist allerdings für jede individuell angepasst.

4.1 Funktionalität der Applikationen

Das Ziel dieser Applikationen ist die Analyse von physiotherapeutischen Übungen von individuellen Nutzern und Nutzerinnen. Dabei tragen die Benutzer und Benutzerinnen die für die jeweilige Applikation definierte Hardware und führen anschließend die definierten Bewegungen aus. Die Übungen, welche erkannt werden können, wurden im Vorhinein definiert und festgelegt.

Im Rahmen dieser Applikation werden die Bewegungen der User und Userinnen über verschiedene Sensoren erkannt. Diese Erkennung ist fortlaufend, um während der Applikation stetig Feedback für die Nutzer und Nutzerinnen zu ermöglichen. Die Form des gegebenen Feedbacks ist sowohl auditiv als auch visuell. Die Kombination dieser Feedbackformen wurde für die Applikation gewählt, um den Benutzern und Benutzerinnen eine möglichst optimierte und allumfassende Form des Feedbacks zu bieten.

Nutzer oder Nutzerinnen werden nach dem Start der Applikation erkannt und kalibriert. In diesem Modus wird die Größe der User und Userinnen ermittelt, um die Bewegungserkennung zu vereinfachen. Diese errechnete Größe wird normalisiert, um die individuelle Berechnung zu erleichtern. Die Kalibration dauert entweder zehn Sekunden oder bis ein gültiger Wert für die Größe der User und Userinnen ermittelt werden kann. Abbildung 4.1 zeigt das Interface, welches nach Abschluss der Kalibration für die Nutzer und Nutzerinnen angezeigt wird.



Abbildung 4.1: Abschluss der Kalibration

Nach Abschluss der Kalibration kann über die Steuerung der jeweiligen Applikation (Ges-tenerkennung oder Controller) die Applikation für Bewegungserkennung gestartet werden.

4 Beschreibung der Applikationen

Die Nutzer und Nutzerinnen finden sich in einem virtuellen Trainingsraum, welcher für die Ausübung der Bewegungen ausgelegt wurde. In diesem Raum haben die User und Userinnen die Möglichkeit, vier physiotherapeutische Übungen auszuführen, welche in diesem Kapitel genauer beschrieben werden, die von der jeweiligen Hardware erkannt werden.

Um die Ausführung der Übungen motivierender zu gestalten, gibt es die Möglichkeit in der Applikation Hindernisse zu aktivieren. Diese Hindernisse erscheinen vor den Nutzern und Nutzerinnen und erfordern immer eine Bewegung, um ihnen auszuweichen. Dabei gibt es fünf verschiedene Hindernisse:

- Hindernis von vorne
- Hindernis von oben
- Hindernis von unten
- Hindernis von links
- Hindernis von rechts

Die seitlichen Hindernisse sowie das Hindernis von vorne erfordern von den Nutzern und Nutzerinnen die Ausführung eines Seitschrittes, um erfolgreich auszuweichen. Beim Hindernis von Oben muss eine Kniebeuge gemacht werden, bei dem Hindernis von Unten muss ein Step-Up gemacht werden. Dieses Ausweichen soll die Motivation bei der Übungsausführung erhöhen, aber auch eine weitere Form des Feedbacks generieren, falls erfolgreich ausgewichen wurde oder die Nutzer und Nutzerinnen mit einem Hindernis kollidiert sind.

Im Raum der Applikation finden sich zwei Interfaces, welche den Benutzern und Benutzerinnen Feedback geben. Eines dieser Interfaces ist in der virtuellen Welt platziert und kann von den Usern und Userinnen verschoben werden. Das zweite Interface befindet sich direkt vor den Augen der Nutzer und Nutzerinnen, es folgt stets ihrem Blick. Diese Formen der Interfaces haben sich im Laufe der Entwicklung der Applikation herauskristallisiert, da sie den Nutzern und Nutzerinnen den intuitivsten Umgang mit der Anwendung ermöglicht haben.

4.2 Beschreibung der ausgewählten Übungen für die Erkennung

Für die Applikationen wurden vier Übungen ausgewählt, welche von den Nutzern und Nutzerinnen im Rahmen des physiotherapeutischen Trainings ausgeführt werden sollen. Diese vier Übungen werden von den Applikationen erkannt. Die Auswahl der Übungen wurde im Vorfeld aufgrund der Fokusgruppen durchgeführt. Die essenziellen Punkte für die Auswahl sind:

- Individuelle Anpassbarkeit
- Alltagsrelevanz
- Koordination der Bewegung
- Für alle Nutzer und Nutzerinnen machbar

Anhand dieser Punkte wurden mehrere Übungen analysiert und einfache, alltägliche Übungen ausgewählt. Die ausgewählten Übungen sind:

- Kniebeugen
- Seitschritte
- Step-Ups
- Berühren der Zehen mit den Händen

Kniebeuge

Die Kniebeuge ist eine allbekannte Übung zur Stärkung der Muskulatur. Dabei werden hauptsächlich die Oberschenkelmuskeln beansprucht, allerdings auch andere Muskeln wie der Gesäß-, Rücken- und Wadenmuskulatur. Dieser große Umfang an abgedeckten Muskeln, sowie der hohe Bekanntheitsgrad waren die Hauptgründe für die Selektion dieser Übung [26].

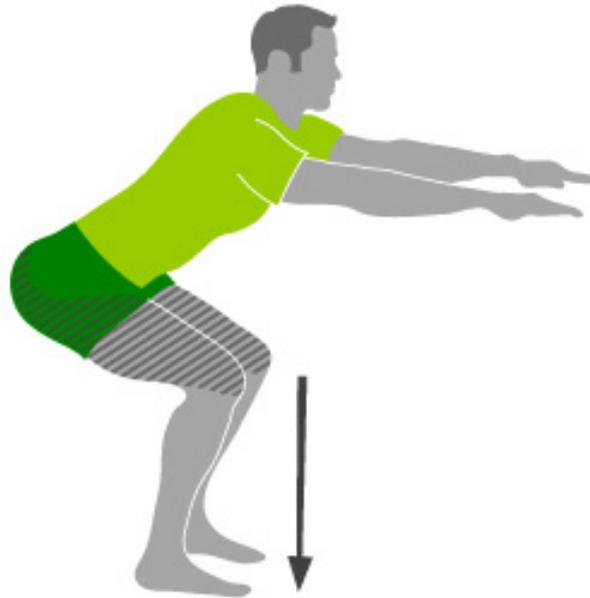


Abbildung 4.2: Darstellung einer Kniebeuge [27]

Bei der Kniebeuge ist die Ausgangsstellung ein symmetrischer Zweibeinstand, wie in Abbildung 4.2 gezeigt. Aus dieser Position muss die durchführende Person in die Hocke gehen, das heißt eine die Sprung-, Hüft- und Kniegelenke beugen. Nach erfolgreichem Erreichen der Hocke, muss wieder in die Ausgangsstellung zurückgekehrt werden.

Ein weiterer Grund für die Auswahl der Kniebeuge ist die Anpassbarkeit der Schwierigkeit. Je nach Tiefe der Beugung und der Wahl des Körperschwerpunktes kann die Übung für die Nutzer und Nutzerinnen individuell angepasst werden [26].

Seitschritt

Der Seitschritt ist eine alltägliche Bewegung, welche mehrfach unbewusst durchgeführt wird. Dabei ist eine Koordination der beiden Beine erforderlich. Aufgrund dieser beiden Faktoren wurde die Übung als eine der Übungen für die Applikation ausgewählt.

Bei dem Seitschritt ist die Ausgangsposition der Stand in breiter Parallelstellung. Das heißt, beide Beine berühren den Boden und stehen parallel zueinander. Außerdem steht die ausführende Person gerade mit durchgestreckten Beinen. Die Durchführung wird in Abbildung 4.3 gezeigt. Dabei wird ein Schritt zur Seite gemacht, der auf einer Linie mit den Beinen sein soll. Unter dieser Bewegung bewegt sich der Oberkörper etwas nach vorne. Die Belastung findet sich hauptsächlich auf einem Bein [26].

Der Seitschritt besitzt ebenfalls eine anpassbare Schwierigkeit. Durch die Schrittlänge sowie die Beugung des Oberkörpers kann die Übung für die Nutzer und Nutzerinnen individuell angepasst werden.



Abbildung 4.3: Darstellung eines Seitschrittes [24]

Step-Up

Die Übung Step-Up basiert auf dem alltäglichen Prinzip des Stufensteigens. Jede gestiegene Stufe kann als ausgeführtes Step-Up gewertet werden. Das heißt, diese Übung repräsentiert eine alltägliche Tätigkeit, welche nicht wegzudenken ist. Außerdem ist bei der Ausführung die Koordination der Beine essenziell.

Die Ausgangsstellung beim Step-Up ist der Stand in breiter Parallelstellung. Das heißt, beide Beine berühren den Boden und stehen parallel zueinander. Außerdem steht die ausführende Person gerade mit durchgestreckten Beinen. Die Durchführung wird in Abbildung 4.4 gezeigt. Dabei wird zuerst ein Bein angehoben und auf den Stepper gestellt, bevor das zweite Bein auf den Stepper gestellt wird. Im Rahmen dieser Applikationen gilt ein Step-Up erst als abgeschlossen, wenn vom Stepper heruntergestiegen wurde. Auch der Vorgang des Heruntersteigens sollte mit jedem Bein einzeln erfolgen.



Abbildung 4.4: Darstellung eines Step-Ups [14]

Der große Vorteil von Step-Ups ist die anpassbare Schwierigkeit. Durch eine einfache Veränderung der Stufenhöhe, kann die Übung individuell angepasst werden. Außerdem kann

durch die Neigung des Oberkörpers während der Ausführung der Übung die Muskelaktivität erhöht werden [26].

Berühren der Zehen mit den Händen

Die letzte ausgewählte Übung ist am schwierigsten auszuführen. Das Berühren der Zehen mit den Händen ist eine Dehnübung, welche die gesamte Rückseite des Körpers dehnt und beansprucht. Trotz dieser Schwierigkeit wurde die Übung ausgewählt, da sie einen hohen Bekanntheitsgrad besitzt und auch an die Verhältnisse der übenden Personen angepasst werden kann.

Die Ausgangsstellung bei dieser Übung ist der Stand in breiter Parallelstellung. Das heißt, beide Beine berühren den Boden und stehen parallel zueinander. Außerdem steht die ausführende Person gerade mit durchgestreckten Beinen. Anschließend wird sich nach vorne gebeugt und die Hände zu den Zehen bewegt. Dabei bleiben die Beine durchgestreckt.

Abbildung 4.5 zeigt die Ausführung der Übung.

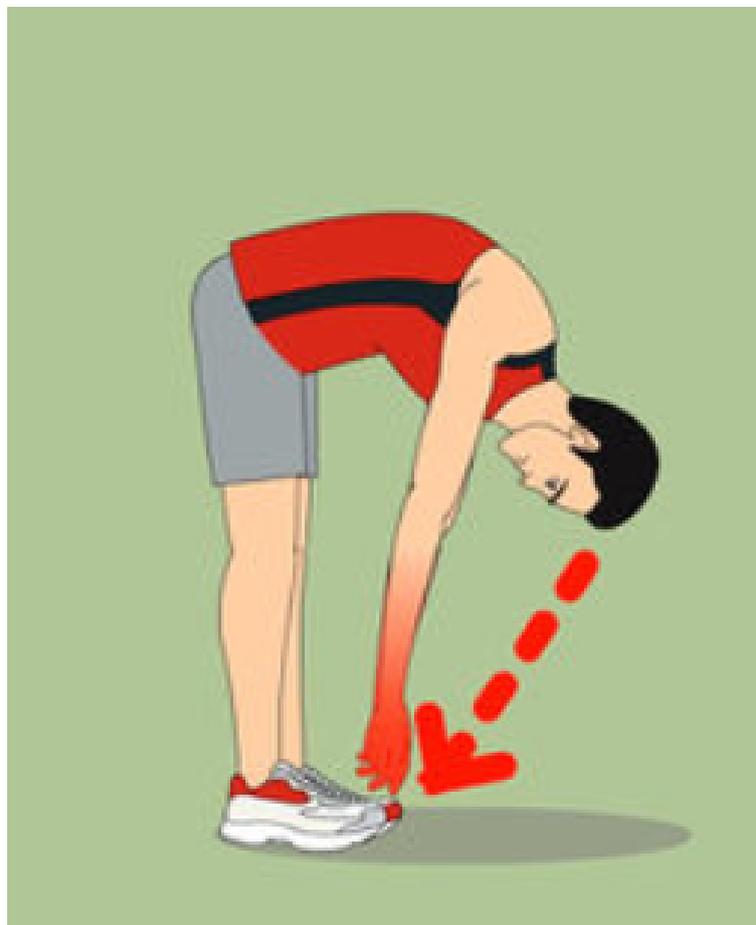


Abbildung 4.5: Darstellung einer Zehenberührung [54]

Diese Übung ist fordernd, allerdings stark individuell anpassbar. Durch leichtes Abwinkeln der Beine oder verringern der Beugung kann die Aktivität individuell angepasst werden. Diese Anpassung wird für die Zielgruppe in einem optimierten Maß in Kooperation mit den Physiotherapeuten und Physiotherapeutinnen vorgenommen.

Ein weiterer Aspekt der Applikationen ist die Möglichkeit der Kommunikation mit einem Webserver. Dieser kann direkt von den Therapeuten und Therapeutinnen über eine Webseite

gesteuert werden. Diese direkte Form der Kommunikation ermöglicht für die Therapeuten und Therapeutinnen eine besser Form der Nachvollziehbarkeit und für die übenden Personen eine bessere Auslastung bei der Übungsdurchführung.

4.3 Konfigurationsinterface für Physiotherapeuten und Physiotherapeutinnen

Im Rahmen dieser Arbeit wurde ein Konfigurationsinterface für Physiotherapeuten und Physiotherapeutinnen erstellt, um die bereits diskutierte Problematik der fehlenden Kommunikation zwischen Nutzern und Nutzerinnen und Therapeuten und Therapeutinnen zu ermöglichen. Dieses Konfigurationsinterface ist für die Physiotherapeuten und Physiotherapeutinnen jederzeit über einen Browser zugänglich, da die Anwendung über einen Webserver gehostet wird. Diese Zugriffsmöglichkeit erlaubt es, dass von jedem beliebigen internetfähigen Gerät, mit den nötigen Berechtigungen, auf die Einstellungen zugegriffen werden kann. Dadurch können die Therapeuten und Therapeutinnen das Training beeinflussen, ohne vor Ort bei den trainierenden Personen zu sein.

4.3.1 Webserver

Der Webserver für das Konfigurationsinterface wurde mit Node.js erstellt. Node.js ist eine öffentlich zugängliche, plattformübergreifende JavaScript Umgebung, welche JavaScript Code serverseitig ausführt. Das Ziel von Node.js ist es, die Entwicklung von Web Applikationen zu vereinfachen, indem nicht mehrere Programmiersprachen verwendet werden müssen, sondern für Server- und Clientseite die gleiche Programmiersprache verwendet werden kann [48].

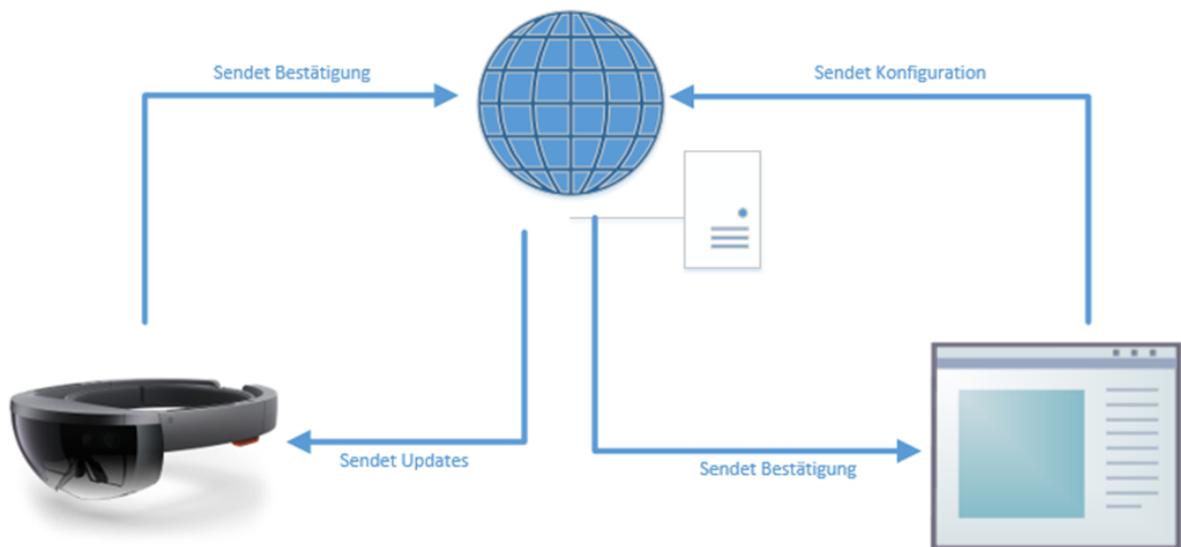


Abbildung 4.6: Darstellung der Webserver-Kommunikation

Abbildung 4.6 zeigt den Ablauf der Kommunikation zwischen dem Konfigurationsinterface (rechts unten) und den Applikationen auf den diversen Devices. In dieser Abbildung wird die Kommunikation zur Microsoft HoloLens (links unten) gezeigt. Das Konfigurationsinterface sendet die Konfigurationen an den Webserver, welcher diese auf Richtigkeit überprüft und anschließend an die Applikationen auf den jeweiligen Devices weiterleitet. Diese Geräte

4 Beschreibung der Applikationen

bestätigen den Erhalt der Informationen, indem sie eine Bestätigungsnachricht an den Webserver senden. Dieser verarbeitet diese Nachricht und leitet die Erfolgsmeldung sowie den aktuellen Status der Applikationen an das Konfigurationsinterface weiter.

Essenziell für die Usability ist die Validierung der Eingaben, sowohl Serverseitig als auch Clientseitig. Auf Seiten des Servers werden die eingehenden Informationen auf Fehler, vor allem fehlerhafte Eingaben, aber auch Eingaben außerhalb des Gültigkeitsbereiches überprüft. Der Server vergewissert sich auch, dass die Informationen vollständig eingelangt sind, bevor diese verarbeitet werden. Diese Überprüfung dient der Sicherstellung, dass keine Teile der Informationen im Prozess verloren gehen.

Auf der Clientseite, das heißt im Konfigurationsinterface, werden die Eingaben ebenfalls überprüft. Durch Eingabeüberprüfungen ob die Daten innerhalb eines erlaubten Bereichs liegen, kann vermieden werden, dass fehlerhafte Daten an den Server gesendet werden. Dies wird während der Eingabe, aber auch beim Absenden der Daten an den Server, überprüft.

4.3.2 Beschreibung des Konfigurationsinterfaces

Das Konfigurationsinterface, welches für die Applikationen erstellt wurde, ermöglicht den Therapeuten und Therapeutinnen die Vornahme von Einstellungen für die diversen Trainingsapplikationen. Diese Einstellungen können in Echtzeit über den Webserver auf die Applikationen angewendet werden. Diese Anpassung in Echtzeit ist ein starker Einflussfaktor auf die Individualität der Trainingseinheiten. Durch diese permanente Anpassungsmöglichkeit von Trainingszielen und Übungsschwierigkeiten kann das Training für jeden User und für jede Userin effektiv gestaltet werden.

Neben dieser Echtzeit Anpassungsmöglichkeiten bietet das Konfigurationsinterface auch die Möglichkeit, das Training Live zu beobachten. Über eine Live-Übertragung des Bildschirms der Microsoft HoloLens können die Physiotherapeuten und Physiotherapeutinnen genau mitverfolgen, in welchem Status sich das Training befindet. Dieser wird in Abbildung 4.7, im unteren Ende, dargestellt.

Abbildung 4.7 zeigt dieses Konfigurationsinterface. Im Konfigurationsinterface können diverse Einstellungen vorgenommen werden. Über Schieberegler können die Schwierigkeit von Kniebeugen und Seitschritten angepasst werden. Bei den Kniebeugen verändert die Distanz, welche zum Boden zurückgelegt werden muss, die Schwierigkeit. Bei den Seitschritten die benötigte Distanz zur Seite.

Neben den Schwierigkeiten kann auch die Zielanzahl der Übungen in Echtzeit angepasst werden. Über numerische Eingabefelder können die Zielanzahl für Kniebeugen, Seitschritte und Step-Ups festgelegt werden.

Über ein Dropdown-Menü kann das Setting, das heißt diverse Modelle, für die Applikation ausgewählt werden. Es kann zwischen einer Sommer-, Winter- und regulären Variante gewählt werden. Diese verändern das Aussehen der Applikation und können auch in Echtzeit angewendet werden. Alle bisher genannten Konfigurationsmöglichkeiten müssen mit dem Drücken auf den Abschicken-Knopf bestätigt werden. Bevor diese abgesendet werden, werden sie überprüft und die Nutzer und Nutzerinnen im Falle eines Eingabefehlers informiert.

Der grüne Knopf, welcher in Abbildung 4.6 dargestellt ist, ermöglicht es den Therapeuten und Therapeutinnen die Hindernisse der Applikation zu starten. Der rote Knopf deaktiviert diese wieder. Der unterste rechte Knopf startet die Live-Übertragung der Microsoft HoloLens, welche anschließend im Bildbereich unten in der Mitte angezeigt wird.

Der Knopf "QR Code erstellen" verweist auf eine Webseite zur Erstellung von QR-Codes, welche

4 Beschreibung der Applikationen

Konfiguration

Schwierigkeit der Kniebeugen (0.6 (schwer) - 0.8 (leicht) (Distanz in % Körperhöhe) = **0.76 %**

Schwierigkeit der Seitschritte (0.30 (leicht) - 0.4 (schwer) (Distanz in m seitlich) = **0.33 m**

Anzahl der Kniebeugen-Teilübungen
Aktuell: 20

Anzahl der Seitschritte
Aktuell: 20

Anzahl der Step-Ups and Downs
Aktuell: 20

Modell für die Applikation auswählen:

Update erfolgreich

Abschicken

Übungsanzahl zurücksetzen

Training Start

Training Stop

QR-Code erstellen

Live Stream Starten

Abbildung 4.7: Erste Version des Konfigurationsinterfaces

für die Applikationen als Registrierung verwendet werden. Mit diesen QR-Codes können Therapeuten und Therapeutinnen verschiedene User und Userinnen in den Applikationen registrieren und diese dadurch personalisieren. Durch diese Registrierung können die Daten der Nutzer und Nutzerinnen gespeichert werden, um abgeschlossene Trainings zu analysieren und zukünftige Trainings zu optimieren.

Um die Übersichtlichkeit des Konfigurationsinterfaces zu erhöhen und dadurch die intuitive Verwendung zu ermöglichen, wurden die Einstellungsmöglichkeiten auf verschiedene Registerkarten verteilt. Abbildung 4.8 zeigt die zweite Version des Konfigurationsinterfaces.

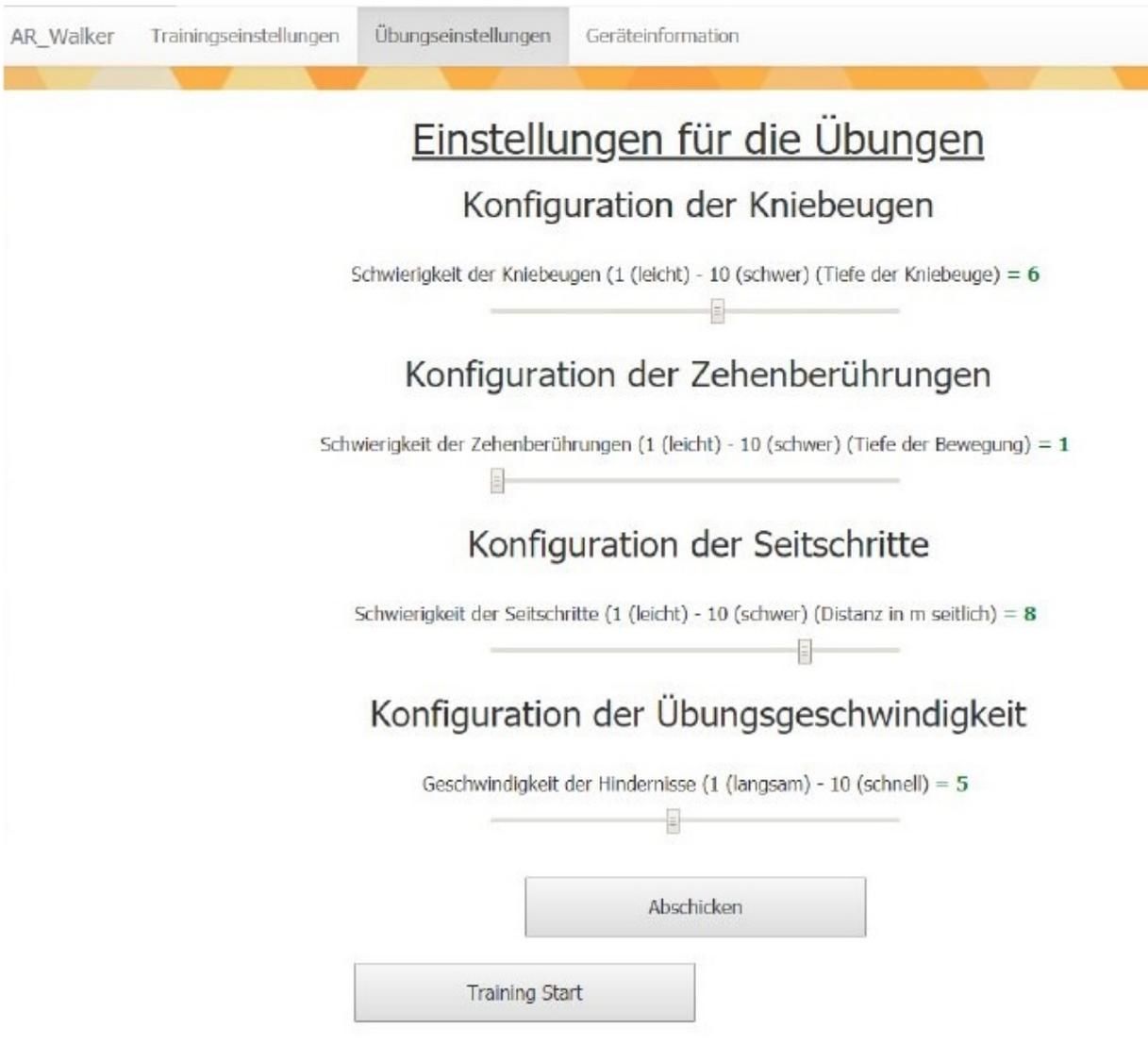


Abbildung 4.8: Zweite Version des Konfigurationsinterfaces

4.3.3 Schwierigkeiten des Konfigurationsinterfaces

Die Live-Übertragung des Bildschirminhaltes der Microsoft HoloLens stellte im Rahmen der Entwicklung des Konfigurationsinterfaces eine große Herausforderung dar. Sobald eine Übertragung des Bildschirminhaltes gestartet wird, verändert sich das FOV der Microsoft HoloLens. Im Normalfall war die Standardeinstellung in Unity bei ungefähr 17 Grad. Nach Start veränderte sich diese auf rund 33 Grad. Durch diese Veränderung konnte die Person, welche die Microsoft HoloLens trug, virtuelle Objekte am Rande des Bildschirms nicht mehr erkennen. Dieses Problem wird in Abbildung 4.8 dargestellt.

Abbildung 4.8 zeigt oben das korrekt dargestellte HUD für die Nutzer und Nutzerinnen. Diese können dort alle eingeblendeten Informationen abrufen. Unten wird das veränderte Interface dargestellt. Dabei ist der komplette linke Rand mit dem Höhenindikator sowie die Kreise mit den aktuellen Übungsanzahlen abgeschnitten. Durch dieses Problem konnte die Live-Übertragung nicht verwendet werden, sofern Nutzer oder Nutzerinnen mit der Applikation interagierten.

Um dieses Problem für die Microsoft HoloLens Applikation zu umgehen, wurden diverse Änderungen an den Unity Canvas vorgenommen. Ursprünglich wurden diese als Screen-Space Canvas konfiguriert, um dauerhaft vor der Kamera angezeigt zu werden. Da bei dieser

4 Beschreibung der Applikationen



Abbildung 4.9: Veränderung des FOV bei Live-Übertragung

Konfiguration der Fehler auftrat, wurde versucht das FOV manuell anzupassen. Allerdings werden die FOV Einstellungen von der Microsoft HoloLens überschrieben und können nicht angepasst werden, da das Device die für sich optimale Konfiguration wählt.

Aus diesem Grund wurde die Einstellung des Canvas darauf gesetzt, an einer definierten Position angezeigt zu werden (World Space). Dadurch ist das Canvas unabhängig von den FOV Einstellungen der Kamera. Allerdings wurde es der Kamera angehängt, um deren Position zu folgen.

5 Beschreibung der Microsoft HoloLens Applikation

In diesem Kapitel wird die erstellte Applikation für die Microsoft HoloLens detailliert beschrieben. Der Aufbau der anderen Applikationen basiert auf dieser Anwendung, daher finden sich dort ähnliche Interfaces und Feedbackformen.

5.1 Beschreibung verwendeter Hard- und Software

In diesem Unterkapitel wird die verwendete Hard- und Software für die Microsoft HoloLens Applikation beschrieben. Die beschriebene Entwicklungsumgebung wird in allen Applikationen verwendet.

5.1.1 Unity 3D

Unity 3D ist eine der bekanntesten plattformübergreifenden Spieleentwicklungssoftware. Unity 3D unterstützt diverse Programmiersprachen, zum Beispiel C# und Javascript und wird hauptsächlich zur Entwicklung von zwei- oder dreidimensionalen Spielen verwendet. Über die Software können Funktionalitäten der erstellten Anwendung durch simples Ziehen und Klicken erstellt werden [52].

Neben der vereinfachten Programmerstellung bietet Unity 3D die Möglichkeit, verschiedene Beleuchtungsmodelle in Form von Shadern zu erstellen. Shader sind Module, welche in Computerprogrammen diverse Anzeigeeffekte erzeugen können [4]. Dabei kann auf bereits vorgefertigte Shader zurückgegriffen werden, oder neue erstellt werden [50].

Außerdem bietet Unity 3D eine bereits integrierte Physikengine, welche es erlaubt Objekte in der virtuellen Welt physikalisch korrekt zu bewegen. Auch die Animation von Charakteren ist innerhalb von Unity 3D möglich. Diese Animationen können ebenfalls aus externen Programmen importiert werden [50].

Das Abspielen von Tönen und Geräuschen wird in Unity 3D mit Audioquellen umgesetzt. Diese können frei platziert und auch dynamisch bewegt werden. Unity 3D bietet die Möglichkeit, die Klänge über diverse Effekte zu verzerren, allerdings bietet Unity 3D ohne Erweiterung keine reale, physikalische Schallausbreitung [50].

Mittels Unity 3D können unter anderem folgende Plattformen angesprochen werden:

- Mobile Plattformen (Android, iOS, Windows Phone)
- Spielkonsolen (Wii U, Nintendo Switch, Playstation 4, Playstation Vita, Nintendo 3DS, Xbox One)
- Betriebssysteme (Mac, Linux, Windows)
- VR / AR / MR Devices (Oculus Rift, Google Cardboard, Daydream, Windows Reality, Microsoft HoloLens, HTC Vive)

5.1.2 Universal Windows Platform

Universal Windows Platform (UWP) ist eine Plattform von Microsoft, welche mit Windows 10 eingeführt wurde. Das Ziel von UWP ist es, universelle Applikationen zu erstellen, welche auf folgenden Geräten installiert werden können [53]:

5 Beschreibung der Microsoft HoloLens Applikation

- Rechner mit Windows 10 Betriebssystem
- Mobile Geräte mit Windows 10 Mobile
- Xbox One
- Windows Mixed Reality Headsets
- Microsoft HoloLens

Diese Plattform zielt darauf ab, Applikationen ohne Umstände auf diesen Geräten benutzen zu können. Sie unterstützt C++, Visual Basic, C Sharp sowie XAML [53]. Im Rahmen dieser Masterarbeit wurde diese Plattform zur Erstellung der Microsoft HoloLens Applikation verwendet.

5.1.3 Windows Mixed Reality

Windows MR, früher auch Windows Holographic genannt, ist eine MR Plattform welche von Microsoft als Teil des Windows 10 Betriebssystems eingeführt wurde. Durch den Einsatz dieser Plattform können MR Inhalte und Applikationen mit diversen HMDs, von verschiedensten Herstellern, erstellt und wahrgenommen werden [33].

Der Hauptanbieter für MR von Microsoft, die Microsoft HoloLens, wird im nächsten Abschnitt näher beschrieben. Mittels Windows MR können Applikationen, welche als Universal Windows Plattform Applikationen erzeugt werden, innerhalb der AR Devices ausgeführt werden. Derzeit unterstützen unter anderem folgende Headsets, wie in Abbildung 5.1 dargestellt, offiziell Windows MR [33].



Abbildung 5.1: Windows MR Devices [33]

Interaktionsmöglichkeiten mit der Applikation

Mit Windows MR Applikationen können die Nutzer und Nutzerinnen über diverse Wege interagieren. Die wichtigsten sind allerdings [57]:

- Steuerung über Bewegung und Controller
- Interaktion über Sprache
- Interaktion mittels Gesten

Steuerung über Bewegung und Controller

Wie bereits in Kapitel zwei, Relevante Technologien und Forschungen, erwähnt, basiert das System der Windows MR auf dem Inside-Out Tracking, das bedeutet, dass die Umwelt über eine Kamera, welche auf dem Kopf der Nutzer und Nutzerinnen befestigt ist, wahrgenommen wird. Durch diese Erkennung der Umwelt können die Positionen der User und Userinnen ermittelt und innerhalb der Anwendungen verwendet werden. Abhängig von den eingebauten Sensoren in den HMDs können auch die Kopfdrotation und die Bewegungsgeschwindigkeit als Interaktionsmöglichkeit mit der Applikation benutzt werden [57].

Neben der Interaktion über die Parameter der Kamera und Sensoren kann auch mittels diversen Controllern über verschiedene Inputs mit der Applikation interagiert werden. Ein Beispielcontroller, in diesem Fall für diverse Windows MR Devices, die Microsoft HoloLens ausgenommen, wird in Abbildung 5.2 dargestellt. Auch für die Microsoft HoloLens gibt es einen externen Mini-Controller [60].



Abbildung 5.2: Interaktion mittels MR Controller [60]

Abbildung 5.2 zeigt die üblichen Interaktionsmöglichkeiten der MR Controller. Der Triggerbutton dient zur Selektion, meist in Kombination mit Blickvektoren. Das heißt, angesehene Objekte werden erkannt und mittels Controller ausgewählt. Über das Touchpad können Funktionen ausgewählt werden und durch das Menü, welches mit dem Menükнопf geöffnet werden kann, navigiert werden. Der Thumbstick ist die Hauptquelle der Bewegung in Applikationen, durch ihn kann sich üblicherweise innerhalb einer Anwendung weiterbewegt werden. Auch die Position der Controller wird vom System erkannt und kann als Interaktionsmöglichkeit für die Applikationen verwendet werden [60].

Interaktion über Sprache

Neben dieser Form der Interaktion bietet Windows MR die Möglichkeit über Sprachkommandos mit Windows MR Devices zu interagieren. Spracherkennung bezeichnet eine Interaktionsmöglichkeit, bei der die Applikation die Worte der Anwender und Anwenderinnen versteht

und anhand dieser Aktionen ausführt. Die Aufnahme dieser gesprochenen Worte erfolgt über ein Mikrofon, während die Ausgabe der Antwort auf verschiedene Arten erfolgen kann. Zum Beispiel kann anhand eines Kommandos auditives oder visuelles Feedback gegeben werden.

Spracherkennungstechnologien basieren darauf einzelne Muster in der Sprache der Nutzer und Nutzerinnen zu erkennen und anhand dieser die richtigen Worte und Sätze zu verstehen. Eine weitere Methode für die Erkennung ist die Erkennung des Klanges der Vokale, um anschließend anhand dieses Klanges auf das gesprochene Wort zu schließen [29].

Spracherkennung baut auf dem so genannten Hidden Markov Model (HMM) auf, welches ein statistisches Modell zur Modellierung von Sequenzen ist [6]. Das HMM dient zur Beschreibung der Verteilung von Wahrscheinlichkeiten über eine unendliche Anzahl von möglichen Sequenzen in einem endlichen Modell. Es wird seit über zwanzig Jahren in der Spracherkennung verwendet [9]. Diese Modelle werden auf die Spracheingaben der Nutzer und Nutzerinnen trainiert, um optimale Ergebnisse zu liefern.

Das Interface der Windows MR Entwicklungsumgebung bietet Entwicklern und Entwicklerinnen eine effiziente Methode Spracherkennung zu verwenden. Über ein individuell steuerbares Programm können Wörter innerhalb der Applikationen definiert werden, welche von dieser Applikation erkannt werden sollen. Somit kann für jede Applikation individuell die Erkennung gewünschter Wörter festgelegt werden.

Interaktion mittels Gesten

Die Interaktion mittels verschiedener Gesten ist eine weitere Form der Interaktion mit Windows MR Applikationen. Dabei erkennt die Applikation die ausgeführten Gesten der Nutzer und Nutzerinnen mittels eingebauter Kameras und Sensoren. Anhand dieser Erkennung werden definierte Aktionen ausgelöst.

Der größte Vorteil der Interaktion mittels Gesten ist die unbegrenzte Anzahl an möglichen Inputs. Bei Interfaces, bei welchen die herkömmliche Interaktion mit Maus und Tastatur erfolgt, ist die Anzahl der Inputs begrenzt. Bei gestenbasierten Interfaces hingegen ist die Anzahl schier unendlich, da auch die Vielfältigkeit der Gesten unendlich ist [42].

Die Probleme der Gestenerkennung in der heutigen Zeit sind sehr umfassend. Eine große Rolle spielt die Genauigkeit des Sensors, damit die Gesten optimal erkannt werden können. Hierbei sind die Lichtverhältnisse sowie die Umgebungseigenschaften wichtige Faktoren. Eine weitere Schwierigkeit ist die Geschwindigkeit der Erkennung der Gesten, diese muss so präzise und schnell wie möglich sein, um die Interaktion benutzerfreundlich zu gestalten [40].

Auch kulturelle Differenzen müssen für die Gestenerkennung bedacht werden. Dies ist essenziell für die Entwicklung von gestengesteuerten Applikationen, da identische Gesten in verschiedenen Ländern und Kulturen verschiedene Bedeutungen haben können [12].

Die Interaktion mittels Gesten wird von Microsoft als Hauptinteraktionsmöglichkeit für die Microsoft HoloLens verwendet. Dabei werden hauptsächlich folgende zwei Gesten verwendet:

- Air tap
- Bloom

Die Air tap Geste ist die Auswahlmöglichkeit auf der Microsoft HoloLens, sofern kein zusätzlicher Controller verwendet wird. Die Ausführung des Air taps hat die gleiche Funktion wie ein Mausklick oder das Drücken des Auswahltriggers. Die Durchführung der Geste wird in

5 Beschreibung der Microsoft HoloLens Applikation

Abbildung 5.3 dargestellt. Die Klickfunktionalität kann auch durch das Sprachkommando Select aufgerufen werden [56].



Abbildung 5.3: Air tap Geste [56]

Die zweite Geste welche essenziell für die Interaktion mit der Microsoft HoloLens ist, ist die Bloom Geste. Diese Geste dient der Rückkehr in das Hauptmenü, oder auch dem Verlassen von gestarteten Applikationen. Um die Geste auszuführen, muss die Hand geschlossen werden, das heißt die Finger müssen einander berühren. Anschließend muss sie rasch geöffnet werden. Der Name der Geste soll auf die Durchführung, welche in Abbildung 5.4 dargestellt ist, schließen lassen, da man die Finger öffnen muss, wie sich eine Blüte einer Blume beim Erblühen öffnet [56].



Abbildung 5.4: Bloom Geste [56]

Für die Microsoft HoloLens Applikation funktioniert die gesamte Steuerung über die genannten Gesten, es wird keine Sprachsteuerung und kein Controller verwendet.

5.1.4 Microsoft HoloLens

Die Microsoft HoloLens, wie in Abbildung 5.5 dargestellt, ist ein MR HMD welches im Jahr 2016 veröffentlicht wurde. Das Device ermöglicht es Usern und Userinnen mittels eines natürlichen und intuitiven, hauptsächlich auf Gesten- und Spracheninteraktion basierten Interfaces mit interaktiven dreidimensionalen Projektionen zu interagieren [2].

Eine weitere Spezialität der Brille ist die Eigenständigkeit, das heißt sie muss nicht mit zusätzlicher Hardware gekoppelt werden, da sie ein konfigurierbares Windows 10 Betriebssystem besitzt, um vollständig zu funktionieren. Das Device besitzt vier Intel Atom x5-Z8100 1.04 Gigahertz Airmont Prozessoren, 64 Gigabyte Flash, 2 Gigabyte Arbeitsspeicher und zwischen

5 Beschreibung der Microsoft HoloLens Applikation



Abbildung 5.5: Microsoft HoloLens [10]

zwei und drei Stunden Akkulaufzeit bei aktiver Benutzung. Der Großteil dieser Prozessorleistung wird verwendet, um die beiden HD Displays der Brille effektiv betreiben zu können [10].

Zusätzlich besitzt das Device die Möglichkeit, die Umgebung rund um sich zu erkennen. Dies wird als Spatial Mapping bezeichnet. Diese Erkennung basiert auf sechs Kameras. Vier Umgebungskameras, einer Farbkamera und einer Tiefenkamera. Neben der Umgebungserkennung wird auch die Stimme über ein Mikrofon aufgenommen und Gesten über die genannten Kameras erkannt [10].

Eines der größten Hindernisse der Microsoft HoloLens ist das etwas eingeschränkte FOV. Bei diesem Device beschränkt sich das Sichtfeld auf ungefähr 30 bis 40 Grad [31]. Die Probleme und Lösungsansätze für FOVs wurden in dieser Arbeit in Kapitel eins, Einleitung, bereits näher behandelt.

Windows Device Portal

Das Windows Device Portal ist ein Entwicklungstool von Microsoft, welches es ermöglicht Informationen zwischen allen Windows 10 Geräten auszutauschen. Dies wird von Microsoft dadurch ermöglicht, dass die Grundfunktionalitäten der HMDs sich kaum unterscheiden, alle aber in gewissen Bereichen Eigenheiten besitzen [45].

Neben der Funktionalität des Datenaustausches bietet das Device Portal die Möglichkeit, das benutzte Device zu konfigurieren und über ein Netzwerk oder eine USB-Verbindung zu bedienen. Dieses Feature wird ermöglicht, indem auf dem jeweiligen Gerät ein Webserver gestartet ist, welcher über den Browser erreichbar ist [11]. Abbildung 5.6 zeigt den Startbildschirm und die Einstellungsmöglichkeiten des Windows Device Portals.

5.1.5 Spatial Mapping

Das Spatial Mapping, also die Raumerkennung, ist eines der umfangreichsten Features der Microsoft HoloLens. Über die eingebauten Kameras und Sensoren kann die Umgebung der Nutzer und Nutzerinnen erkannt werden. Neben der Erkennung kann diese in die MR Applikation mit eingebunden werden. Durch diese Anwendung können virtuelle Objekte im realen Raum platziert werden. [58].

Im Rahmen der Microsoft HoloLens Applikation wurde das Spatial Mapping einerseits verwendet, um die Größe der Nutzer und Nutzerinnen zu analysieren, andererseits, um den

5 Beschreibung der Microsoft HoloLens Applikation



Abbildung 5.6: Windows Device Portal [45]

Stepper zu erkennen und die automatische Interface Anpassung vorzunehmen. Abbildung 5.7 zeigt eine Spatial Mapping Abbildung des Entwicklungsraumes. Der weiße Punkt stellt die Position der Microsoft HoloLens im Raum dar.

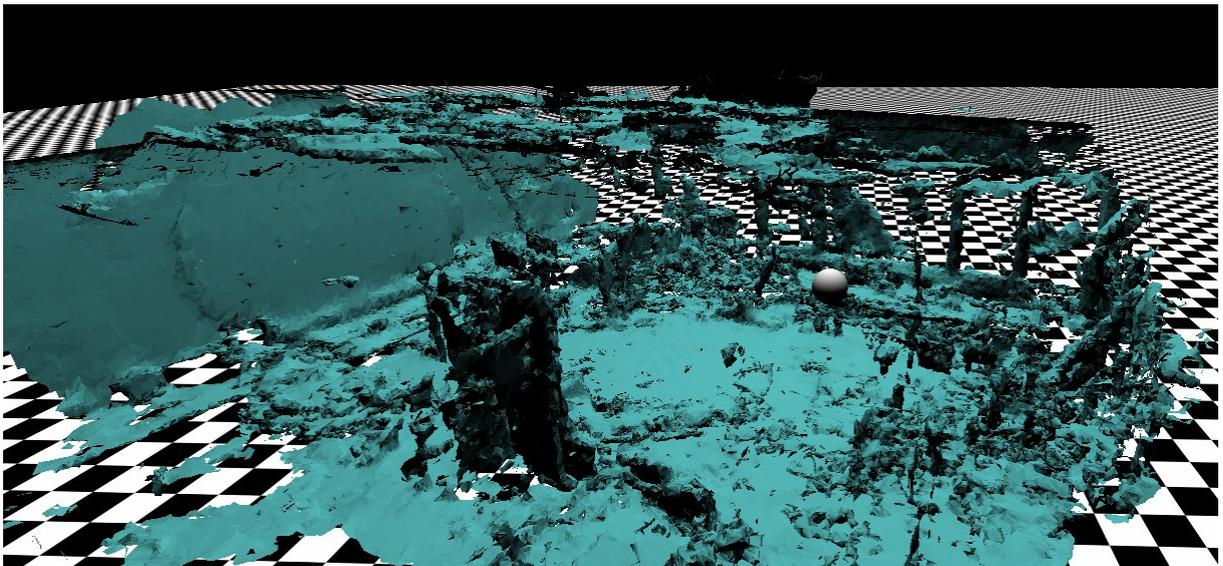


Abbildung 5.7: Spatial Mapping des Entwicklungsraumes

In der Applikation wurde das Spatial Mapping verwendet, um den Boden des realen Raumes zu erkennen. Dafür wurde das HoloToolkit verwendet, welches eine vordefinierte Variante des Spatial Mappings bietet. Mithilfe dieser Variante wird der reale Raum alle drei Sekunden abgetastet und erkannt. Über diese in Abbildung 5.8 dargestellte Erkennung des Raumes wird die Distanz von Microsoft HoloLens beim Start der Applikation und dem erkannten Boden gemessen, um die Größe der Nutzer und Nutzerinnen zu errechnen.

Der große Vorteil bei der Verwendung von Spatial Mapping ist, dass die Applikation auf Informationen über die aktuelle und vergangene Position der Nutzer und Nutzerinnen zugreifen kann. Zusätzlich kann nicht nur die Position gefunden werden, sondern auch Details über die Objekte, welche sich in der Umgebung der User und Userinnen befinden [5].

5 Beschreibung der Microsoft HoloLens Applikation

Die Genauigkeit des Spatial Mappings der Microsoft HoloLens variiert in einem gewissen Spielraum. Wie bereits in Kapitel zwei, Relevante Technologien und Forschungen, erwähnt, hat die Microsoft HoloLens Schwierigkeiten, kleine Objekte exakt zu erkennen. Allerdings gibt es die Möglichkeit, dass das Spatial Mapping die Erkennung der Umgebung verliert, wenn zu wenige Erkennungspunkte und schlechte Lichtverhältnisse für die Kameras und Sensoren vorherrschen [19].

Um die Genauigkeit des Spatial Mappings im Rahmen dieser Arbeit festzustellen, wurde ein Vergleichsbild angefertigt, welches in Abbildung 5.8 dargestellt ist. Dabei wurde eine Aufnahme des Entwicklungsraumes angefertigt, welche ein Originalbild und das zugehörige Spatial Mapping Abbild zeigt.



Abbildung 5.8: Vergleich Originalbild zu Spatial Mapping Abbild

Abbildung 5.8 zeigt wiederum, dass die Microsoft HoloLens große Objekte im Sichtfeld genau erkennen kann. Bei kleineren, eher geraden Oberflächen finden sich vermehrt Schwierigkeiten. Um die Höhe einer Person in der Applikation festzustellen, ist dieses Spatial Mapping Ergebnis allerdings optimal.

5.1.6 Spatial Audio

Spatial Audio beschreibt ein Konzept, bei dem die Nutzer und Nutzerinnen bestimmen können, an welcher Position sich die Tonquelle befindet. Dadurch können, obwohl Objekte nicht gesehen werden, trotzdem der Standort dieser zugeordnet werden [47]. Durch dieses räumliche Hören können die Interfaces für MR Anwendungen effektiver gestaltet werden, da es nicht notwendig ist, alle virtuellen Objekte im Sichtbereich der Nutzer und Nutzerinnen zu halten.

Durch den Einsatz von Spatial Audio und Spatial Mapping erscheint es für die User und Userinnen so, als ob die Soundquelle tatsächlich aus der realen Umgebung kommen würde. Dadurch können die erstellten Hologramme und MR Anwendungen für die Nutzer und Nutzerinnen glaubhafter erstellt werden und die Immersion gesteigert werden [59].

Durch das räumliche Hören kann die Aufmerksamkeit der Nutzer und Nutzerinnen auf Objekte gezogen werden, welche aktuell nicht in deren Sichtfeld liegen. Zum Beispiel kann sich die Lautstärke einer Tonquelle erhöhen, je näher die Nutzer und Nutzerinnen ihr kommen. Das menschliche Gehirn versucht die Distanz und die Richtung zu einer Tonquelle festzustellen, indem es analysiert wie der Ton die Ohren erreicht. Dieses Konzept wird mittels Spatial Audio versucht nachzuahmen, um die Geräusche und Töne den Menschen möglichst natürlich näher zu bringen [59].

Im Rahmen der Microsoft HoloLens Applikation wird Spatial Audio verwendet, um die Nutzer und Nutzerinnen auf die Hindernisse aufmerksam zu machen. Die Tonquelle bewegt sich mit

den Hindernissen mit, um den Usern und Userinnen die Position dieser zu verdeutlichen. So können diese, ohne die Hindernisse sehen zu müssen, wahrnehmen wo sich diese befinden. Die Microsoft HoloLens ist an diverse Speicher- und Leistungslimitationen gebunden. Trotz dieser Limitationen gibt es im Rahmen des Spatial Audio kaum Einschränkungen. Bei der Verwendung von zehn bis zwölf Tonquellen steigt die Auslastung der CPU lediglich auf zwölf Prozent des Maximums an [59].

5.2 Interface der Applikation

Das Interface der Applikation baut auf dem Prinzip des Interfacedesigns auf, welches in Kapitel zwei, Relevante Technologien und Forschungen, angesprochen wurde. Dabei stehen die Nutzer und Nutzerinnen im Mittelpunkt. Für diese soll das Interface möglichst intuitiv und einfach verwendet werden können. Um dieses Ziel erreichen zu können, wurden für die Applikation zwei Interfaces und eine virtuelle Repräsentation eines Trainingsraumes erstellt.

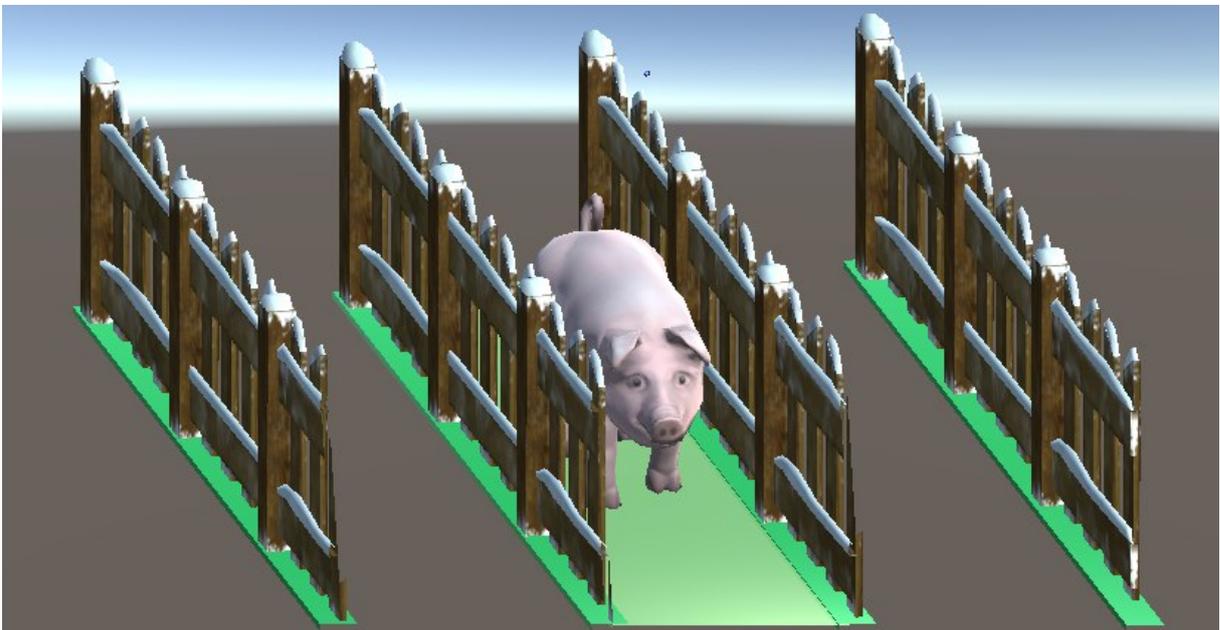


Abbildung 5.9: Virtueller Trainingsraum in den Applikationen

Der Trainingsraum ist in Abbildung 5.9 dargestellt. bei der Erstellung dieses Trainingsraums stand die intuitive Verwendung für Nutzer und Nutzerinnen im Vordergrund. Der Raum besteht aus drei Bahnen, welche mittels Zäunen voneinander getrennt sind. Die Bahnen enden nicht mit dem Zaun, sondern bieten auch davor einen Indikator, auf welcher der Bahnen sich die Nutzer und Nutzerinnen befinden. Die Bahn auf der sich die User und Userinnen aktuell befinden, wird grün hinterlegt. Abbildung 5.9 zeigt diese Hinterlegung auf der mittleren Bahn.

Diese Bahnen repräsentieren die Wege der Hindernisse. Das in Abbildung 5.9 dargestellte Schwein ist ein Hindernis, welchem mittels einem Seitschritt ausgewichen werden muss. Die Zäune dienen dem Zweck, das Gefühl der Nutzer und Nutzerinnen zu verstärken, auf welcher Bahn sie sich befinden. Das heißt, wie weit sie ausweichen müssen, um nicht getroffen zu werden.

Der Trainingsraum ist drei Meter breit, jede der Bahnen ist einen Meter breit, und sechs Meter lang. Nur die Hälfte der Länge ist mit Zäunen abgegrenzt, um zu verhindern, dass

5 Beschreibung der Microsoft HoloLens Applikation

die Nutzer und Nutzerinnen dem Hindernisursprung zu nahe kommen. Natürlicherweise versuchen die User und Userinnen die Übungen außerhalb des Zaunes auszuführen, um nicht mit diesem virtuellen Zaun zu kollidieren.

Es gibt fünf verschiedene Hindernisse:

- Hindernis von vorne
- Hindernis von oben
- Hindernis von unten
- Hindernis von links
- Hindernis von rechts

Für die Hindernisse von vorne, links und rechts ist es notwendig mit einem Seitschritt auszuweichen. Das Hindernis von oben muss mit einer Kniebeuge oder einer Zehenberührung vermieden werden. Dem Hindernis von unten kann nur durch ein Step-Up ausgewichen werden. Die Hindernisrepräsentation kann über das Konfigurationsinterface verändert werden. Im Rahmen der Entwicklung der Applikation hat sich das Interface laufend verändert und die Mehrheit der Informationen wurde auf das HUD verschoben. Der Grund für diese Verschiebung ist die bereits erläuterte Änderung des Interfacesdesigns für MR Devices von Evans, Miller et. al [10]. Zu Beginn der Entwicklung waren die gesamten Übungsinformationen sowie die aktuelle Höhe nur in einem fixierten Menü, welches im Raum platziert wurde, ersichtlich. Das HUD Menü wurde in der ersten Version nur für die Ausgabe von Feedback zur Ausführung von Übungen verwendet. Abbildung 5.10 zeigt die erste Version der beiden Menüs.

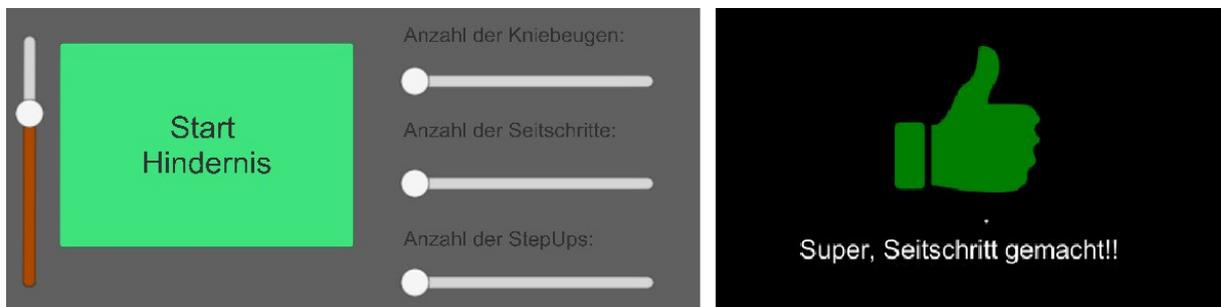


Abbildung 5.10: Erste Version des Menüdesigns

In dieser Abbildung ist links das fixierte Menü ersichtlich, über das der Höhenindikator (linker Balken) eingesehen werden kann. Durch die fixe Position im Raum ist es während der Ausführung von Übungen nicht intuitiv den Kopf in Richtung des Menüs zu bewegen, um den aktuellen Status zu bekommen. Außerdem war die Ausgabe des aktuellen Übungsstatus über die visuelle Schiebeanzeige nicht ausreichend. Das HUD, das in der Abbildung 5.10 auf der rechten Seite dargestellt ist, erfüllte seinen Zweck, allerdings waren hier Probleme mit der Sichtbarkeit, aufgrund mangelnder Transparenz, zu vermerken.

Um diesen Aspekten entgegenzuwirken, wurde das Interface überarbeitet und dem fixen Menü mehr Informationen gegeben. Außerdem wurde dieses fixierte Menü für die Nutzer und Nutzerinnen verschiebbar gemacht, um ihnen die Details leichter zugänglich zu machen. Abbildung 5.11 zeigt die zweite Version des Interfaces. Auf der linken Seite ist die verbesserte Version des fixierten Menüs abgebildet, auf der rechten Seite die überarbeitete Version des HUD Menüs.

Trotz der Anhäufung der Informationen und der erhöhten Transparenz des HUD Menüs konnte nicht die gewünschte intuitive Verwendung der Applikation erreicht werden. Die

5 Beschreibung der Microsoft HoloLens Applikation

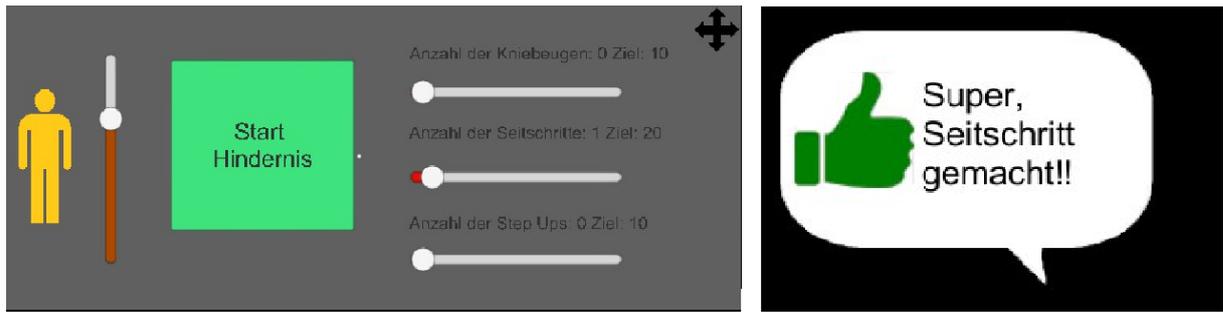


Abbildung 5.11: Zweite Version des Menüdesigns

Möglichkeit der Bewegung des Menüs wurde kaum wahrgenommen und immer der Kopf in Richtung des Menüs bewegt, was die Qualität der Übungsausführung negativ beeinflusste. Um diesem Aspekt entgegen zu wirken, wurde der Großteil der Informationen dupliziert und ebenfalls in das HUD Menü aufgenommen. Die meisten Informationen wurden im fixierten Menü ebenfalls belassen, um einen zusätzlichen Informationspunkt in der Applikation zu bieten. Abbildung 5.12 zeigt die dritte Version des Interfaces. Auf der linken Seite ist die verbesserte Version des fixierten Menüs abgebildet, auf der rechten Seite die überarbeitete Version des HUD Menüs.

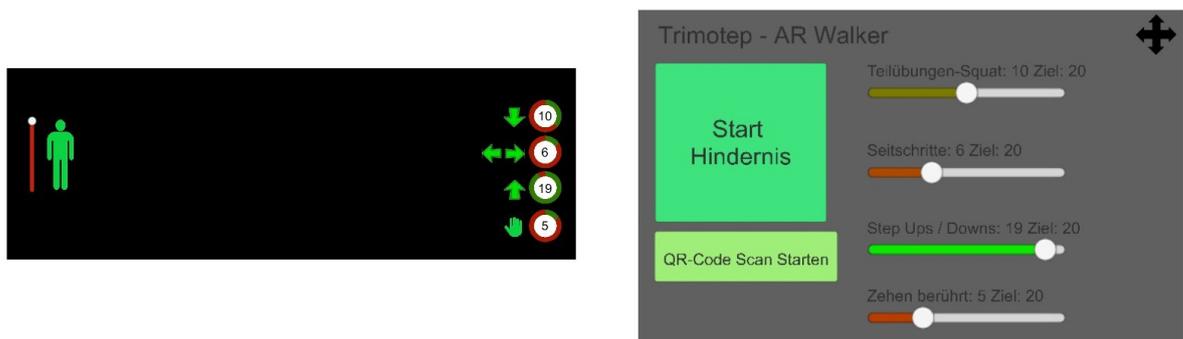


Abbildung 5.12: Dritte Version des Menüdesigns

Wie in Abbildung 5.12 ersichtlich, wurde der Höhenindikator vom fixen Menü auf das HUD Menü verschoben, um die Kopfbewegung zum Status abrufen zu vermeiden. Zusätzlich wird der Übungsfortschritt im HUD Menü angezeigt. Das fixierte Menü besitzt neben der Möglichkeit die Hindernisse zu aktivieren, die Möglichkeit einen QR Code Scan zu starten.

Das Scannen eines QR Codes, welcher über das Konfigurationsmenü erstellt werden kann, ermöglicht es der Applikation, die Daten der Nutzer und Nutzerinnen zu speichern und zu erkennen, welcher User oder welche Userin die Applikation benutzt. Durch diesen Registrierungsvorgang kann die Applikation die Trainingsergebnisse, Konfigurationen und weitere Einstellungen effektiv für die Physiotherapeuten und Physiotherapeutinnen abspeichern. Diese können diese erstellten Daten verwenden, um zukünftige Trainings für Anwender und Anwenderinnen zu planen oder Trainingsverläufe abzurufen. Durch die Analyse dieser Daten soll ein optimiertes, individuelles Training erstellt werden.

Neben dem Höhenindikator und dem Übungsfortschritt wird über das HUD Display auch aktuelles, visuelles Feedback gegeben. Zusätzlich zum visuellen Feedback besitzen die Applikationen auch auditives Feedback. Bei jeder erfolgreichen Ausführung einer Übung werden den Usern und Userinnen zusätzlich zum visuellen Feedback noch bestätigende Geräusche

gesendet. Selbiges gilt für das Ausweichen von Hindernissen. Sollte mit einem Hindernis kollidiert werden, gibt es entsprechendes auditives und visuelles Feedback. Nach Erreichen der definierten Zielanzahlen wird den Usern und Userinnen dieses zusätzlich akustisch mitgeteilt.

5.3 Automatische Interface Positionsanpassung

Eine essenzielle Frage dieser Arbeit ist die Möglichkeit der automatischen Interface Positionsanpassung. Im Rahmen der Entwicklung des Interfaces wurde vermehrt der Kopf nach den positionierten Interfaces ausgerichtet. Um diesem Problem entgegenzuwirken, wurde für diese Applikationen ein HUD Menü verwendet. Allerdings stellt sich diese Arbeit die Frage, ob dieses Problem auch durch die automatische Positionierung des Interfaces mittels Spatial Mapping und Spatial Audio gelöst werden kann.

Dieser Ansatz versucht, im Rahmen der Applikationen zu erahnen, auf welche Position sicher der Blick der Nutzer und Nutzerinnen richten wird. Diese Berechnung basiert auf der Umgebung der User und Userinnen sowie dem räumlichen auditiven Input, den diese erhalten.

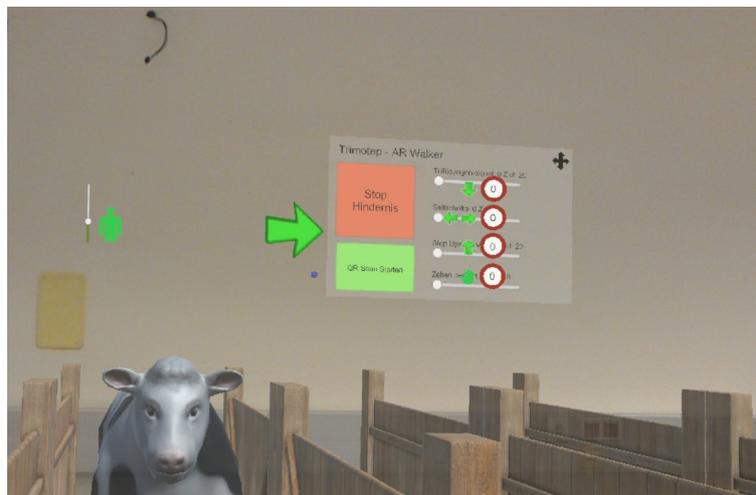


Abbildung 5.13: Automatische Interface Positionsanpassung - Seitliches Hindernis

Anhand dieser Berechnung wird das fixierte Menü der Applikationen so bewegt, dass die Informationen für die Nutzer und Nutzerinnen in ihrem Sichtfeld dargestellt werden. Allerdings soll diese Darstellung sich mit keinen anderen virtuellen Inhalten überlagern, um eine zufriedenstellende Transparenz zu garantieren.

Die Positionierung erfolgt über das Spatial Mapping in der realen Umgebung der Anwender und Anwenderinnen. Üblicherweise ziehen Tonquellen die Aufmerksamkeit von Nutzern und Nutzerinnen auf sich, was die Applikationen benutzen um die Blickpunkte zu analysieren. Abbildung 5.13 zeigt die automatische Anpassung des Interfaces bei einem seitlichen Hindernis. Die übende Person befindet sich auf der linken Bahn, während sich ein Hindernis auf dieser bewegt. Um im Sichtfeld zu bleiben, aber nicht vom Hindernis überdeckt zu werden, bewegt sich das Interface etwas höher und rechts der Nutzer und Nutzerinnen. Da die linke Bahn die äußerste Bahn ist, wird angenommen, dass die nächste Bewegung nach rechts erfolgt.

5 Beschreibung der Microsoft HoloLens Applikation

Abbildung 5.14 zeigt ein weiteres Beispiel der automatischen Positionsanpassung. Die übende Person befindet sich auf der rechten Bahn, während sich das obere Hindernis auf sie zubewegt. Das Menü verschiebt seine Position etwas nach unten, um nicht von dem Hindernis überdeckt zu werden und orientiert sich zwischen der mittleren und der aktuellen Bahn der übenden Person. Diese Anpassung wurde vorgenommen, da in diesem Fall die rechte Bahn die äußerste Bahn ist.

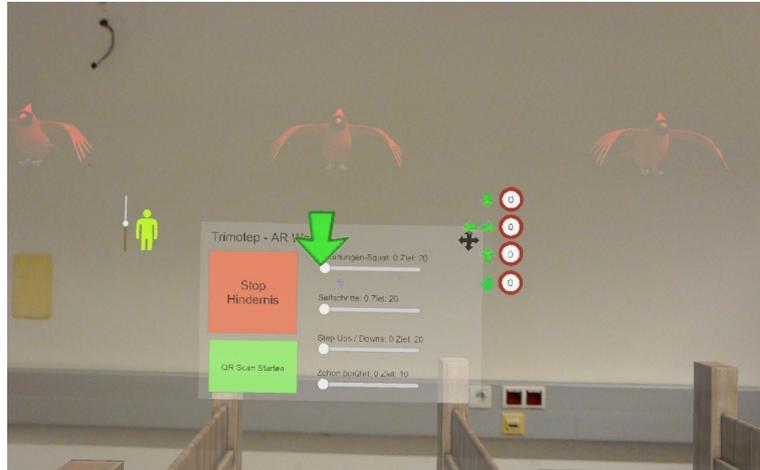


Abbildung 5.14: Automatische Interface Positionsanpassung - Oberes Hindernis

Die automatische Positionsanpassung wird im Rahmen von Usability Tests näher analysiert und in Kapitel zehn, Usability Tests, mit den Ergebnissen beschrieben.

6 Bewegungserkennung mit der Microsoft HoloLens

In diesem Kapitel wird die für die Microsoft HoloLens entwickelte Bewegungserkennung näher beschrieben. Dabei wird auf die exakte Umsetzung der Erkennung von folgenden Übungen eingegangen:

- Kniebeugen
- Seitschritte
- Step-Ups
- Berühren der Zehen mit den Händen

Im Rahmen der Erkennung von Step-Ups wird auch die Erkennung des Steppers mittels Spatial Mapping näher behandelt.

6.1 Beschreibung der analysierten Bewegungserkennungsalgorithmen

Um eine effiziente Bewegungserkennung für die Microsoft HoloLens Applikation zu gewährleisten, wurden im Rahmen dieser Masterarbeit zwei Algorithmen zur Bewegungserkennung detailliert analysiert. Die beiden Ansätze, welche in Betracht gezogen wurden, sind:

- Messen des Abstands zu einer Ursprungsposition
- Messen der Beschleunigung über Zeit zur Errechnung der Bewegung

In der Microsoft HoloLens Applikation wurde der erste Ansatz umgesetzt, da dieser sich im Laufe der Entwicklung als robuster gezeigt hat. Die Erkennung von Bewegungen ist allerdings mit beiden Ansätzen möglich.

6.1.1 Messen des Abstands zu einer Ursprungsposition

Dieser Ansatz misst die Position der Benutzer und Benutzerinnen im Raum während der Ausführung der Applikation mittels dem Tracking der Microsoft HoloLens. Diese Position wird mit einer definierten Referenzposition verglichen und anhand dieser die Abstände und Bewegungen der Anwender und Anwenderinnen gemessen. Die Referenzposition wird bei größeren Positionsveränderungen im Raum aktualisiert und es wird sichergestellt, dass die Übungen im ganzen Raum ausgeführt werden können.

Die exakte Erkennung der Bewegung geschieht immer über ein Messen des Abstands zwischen Referenzposition und aktueller Position. Dabei spielen die Abweichung Links – Rechts, die Höhe und die Abweichung Vorne – Hinten eine essenzielle Rolle. Diese werden alle über die in der HoloLens eingebauten Sensoren getrackt. Durch die dauerhaften Vergleiche kann eine stabile Erkennung garantiert werden.

Die Herausforderung dieses Ansatzes ist die Definition der Referenzposition, denn nur durch eine exakte Positionierung dieser kann ein effektives Tracking erreicht werden. Diese Festlegung der Referenzposition hat enorme Auswirkungen auf die Qualität der Bewegungserkennung dieses Ansatzes.

Dabei kann es hauptsächlich zu folgenden Problemen kommen:

- Die Frequenz der Referenzpositionsaktualisierung ist zu hoch.
- Die Frequenz der Referenzpositionsaktualisierung ist zu niedrig.

Wenn die Frequenz der Aktualisierung der Referenzposition zu hoch ist, wird die Ausführung der Übungen nicht mehr erkannt, da die Vergleichsergebnisse zu klein ausfallen. Die Referenzposition entspricht im schlimmsten Fall permanent der aktuellen Position.

Wenn die Frequenz der Aktualisierung der Referenzposition zu gering ist, wird die Ausführung der Übungen in einem gewissen Radius erkannt, allerdings ist es nicht mehr möglich, die Übungen bei Ausführung im gesamten Raum zu erkennen. Beide dieser Fälle wurden durch vielfache Kalibration optimiert, um eine möglichst genaue Messung zu gewährleisten.

6.1.2 Messen der Beschleunigung über Zeit zur Errechnung der Bewegung

Der zweite analysierte Ansatz beschreibt die Messung der Beschleunigung der Anwender und Anwenderinnen über die Zeit. Dabei wird gemessen, wie schnell sich die User und Userinnen bewegen und anhand der Geschwindigkeit, der Zeit und des Weges die Beschleunigung kalkuliert. Dabei ist nicht die Geschwindigkeit der essenzielle Faktor für diesen Ansatz, sondern die Beschleunigung der Personen.

Um die Bewegungserkennung zu ermöglichen, müssen die Übungen in Beschleunigungsgraphen abgebildet werden. Ein solcher Beschleunigungsgraph zeigt die Übung als Beschleunigungskurve. Zum Beispiel bewegt sich die Person bei einem Seitschritt sehr schnell (hohe Beschleunigung) und anschließend bleibt sie stehen. Abbildung 6.1 zeigt einen Teil eines beispielhaften Beschleunigungsgraphen, wobei t die Zeit und a die Beschleunigung beschreibt.

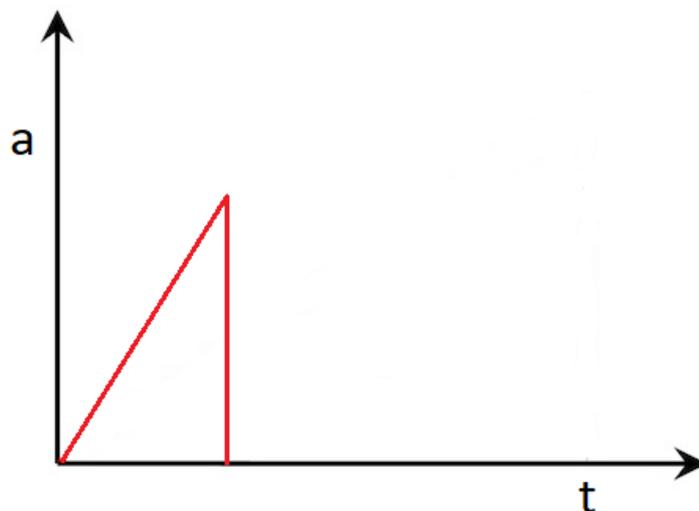


Abbildung 6.1: Beschleunigungsgraph eines Seitschrittes

Die Messung solcher Bewegungsgraphen liefert mit einer Kalibrierung zuverlässige Ergebnisse, allerdings ist es eine Herausforderung den Zeitpunkt für den Beginn der Erkennung zu definieren. Wenn die Beschleunigung dauerhaft gemessen wird, kann es dazu kommen, dass die Bewegungsmuster nicht erkannt werden, da zuvor gemessene Werte das Ergebnis verzerren.

Um den Anfangszeitpunkt der Messung festlegen zu können, müsste man wissen, wann die User und Userinnen mit einer Übung beginnen wollen. Ein theoretisch möglicher Ansatz wäre die Abdeckung über einen Threshold, um vorhergegangene Beschleunigungsmuster

nicht zu werten. Dieser Optimierungsansatz wurde allerdings im Rahmen dieser Arbeit nicht überprüft, da der erste Ansatz für die Erkennung umgesetzt wurde.

6.2 Umsetzung der Bewegungserkennung für die diversen Übungen

Im Rahmen der Microsoft HoloLens Applikation wurde die Bewegungserkennung für die Kniebeugen, Seitschritte, Step-Ups und die Berührung der Zehen mittels oben genanntem Algorithmus umgesetzt. Dafür wurden folgende Daten verwendet:

- Die Höhe des Bodens
- Die Bewegung des Devices
- Die Orientierung des Devices
- Die Position des Devices im Raum

Die Distanz zum Boden wird in der Microsoft HoloLens Applikation über das Spatial Mapping gemessen. Folgender Code-Ausschnitt zeigt den Ablauf dieser Erkennung. Dieser Ausschnitt ist nicht vollkommen funktionstüchtig, er zeigt nur die Vorgehensweise. Code-Ausschnitt 6.1 zeigt die Erkennung des Bodens:

```
1 camera= GameObject.Find("MainCamera").transform.position;
2 ray=new Ray(camera, Vector3.down*5);
3
4 if (Physics.Raycast(ray, out hit))
5 {
6     if (hit.collider.name.Contains("spatial"))
7     {
8         height=hit.collider.gameObject.transform.position.y*-1;
9     }
10 }
```

Source-Code Ausschnitt 6.1: Microsoft HoloLens - Erkennung des Bodens

Die über das Spatial Mapping erkannten Objekte enthalten einen Collider, dessen Name sich aus spatial und einer aufsteigenden Nummer zusammensetzt. Über einen Raycast nach unten wird in Unity erkannt, ob sich unter der Kamera ein Objekt befindet, dessen Name spatial enthält. Wenn ein solches Objekt erkannt wurde, wird die Position dieses Objektes mit -1 multipliziert, da die Position dieser Objekte ein negativer Wert ist. Durch diese Multiplikation erhält man die Größe der Person über das Spatial Mapping der Microsoft HoloLens.

Die Bewegung des Devices wird durch das eingebaute Accelerometer gemessen. Die Orientierung erfolgt über das Gyroskop und die Erkennung der Position erfolgt bei der Microsoft HoloLens über die eingebauten Kameras.

6.2.1 Erkennung von Kniebeugen

Die Erkennung von Kniebeugen erfolgt in zwei Schritten:

- Die Erkennung des Duckens
- Die Erkennung des Aufstehens

Bei der Erkennung des Duckens wird die Höhe des Kopfes, die Rotation der HoloLens, die Größe der Person sowie der momentane Abstand der Anwender und Anwenderinnen zum Boden verwendet. Der definierte offsetX gibt an, wie weit sich die Nutzer und Nutzerinnen mit

ihrem Kopf nach vor und zurück beugen dürfen, damit die Übung noch erkannt wird. Dieser Offset dient zur Sicherstellung, dass Kniebeugen nicht durch Beugen nach vorne erkannt werden.

Der `offsetHeadRotation` ist auf 0.17 definiert. Dieser Wert wird in Unity als Quaternion, eine kompakte Form zur Darstellung von Rotationen, gespeichert. Diese 0.17 entsprechen in etwa einer Rotation von 15 bis 20 Grad. Dieser Offset dient dazu, dass der Kopf während der Ausführung einer Kniebeuge gerade gehalten werden muss, damit diese erkannt wird. Im Rahmen der Kniebeuge wurde nur die Rotation nach oben und unten überprüft, um zwischen einer Zehenberührung und einer Kniebeuge zu unterscheiden. Dieser Wert wurde nach ausgiebigen Tests gewählt, da die Erkennung unter diesen Umständen kaum Fehlererkennung aufzeigte.

Die normalisierte Distanz zum Boden wird errechnet, indem die aktuelle Distanz zum Boden mit der Größe der Person, also dem zuvor kalibrierten Abstand zum Boden, dividiert wird. Dieser Wert liegt zwischen 0 und 1, wenn der Abstand zum Boden nicht größer wird als beim Start der Applikation gemessen.

Um eine Kniebeuge erkennen zu können, wird überprüft, ob sich die Nutzer und Nutzerinnen unterhalb einer definierten Grenze befinden, ihren Kopf gerade halten und diesen nicht zu weit nach vor oder zurück bewegt haben. Wenn dies erfüllt wurde, wird ein Boolean auf `true` gesetzt, welcher indiziert, dass die Person gerade eine Kniebeuge ausführt. Ebenso wird die Anzahl der ausgeführten Kniebeugen erhöht. Der Boolean `kneeBendBool` wird dazu verwendet, dass Kniebeugen erst wieder ausgeführt werden können, nachdem man aufgestanden ist. Dadurch wird verhindert, dass eine Kniebeuge mehrfach gezählt wird.

Die Position `cameraStartingPosition` beschreibt die Referenzposition, welche im vorigen Unterkapitel, Beschreibung der analysierten Bewegungserkennungsalgorithmen, bereits beschreiben wurde. Im Rahmen dieser Arbeit wird die Referenzposition immer aktualisiert, wenn die Nutzer und Nutzerinnen aufrecht stehen. Das heißt, der normalisierte Abstand zum Boden größer ist als 90 Prozent und sich diese mehr als 0.3 Meter von der alten Referenzposition entfernt haben. Code-Ausschnitt 6.2 zeigt diese Erkennung:

```

1 void checkIfSquatDone(Vector3 headPos,Quaternion holoRotation)
2     {
3         float offsetX=0.25f;
4         float offsetHeadRotation=0.17f;
5         float normDistToFloor=distanceToFloor/actDistanceToFloor;
6
7         if(headPos.x > cameraStartingPosition.x - offsetX &&
8             headPos.x < cameraStartingPosition.x + offsetX &&
9             normDistToFloor < float.Parse(squatDif) &&
10            (holoRotation.x < offsetHeadRotation &&
11            holoRotation.x > -offsetHeadRotation))
12            {
13                squating = true;
14                if (kneeBendBool)
15                    {
16                        numberOfSquatsDone++;
17                    }
18                kneeBendBool = false;
19            }
20    }
```

Source-Code Ausschnitt 6.2: Microsoft HoloLens - Erkennung des Duckens bei Kniebeugen

6 Bewegungserkennung mit der Microsoft HoloLens

Der gewählte Threshold für die Erkennung einer Kniebeuge squatdif wird über das Webinterface der Therapeuten und Therapeutinnen konfiguriert und entspricht ebenfalls der normalisierten Distanz zum Boden. Dieser bewegt sich zwischen 60 Prozent und 80 Prozent (0.6 und 0.8). Bei einem Threshold von 70 Prozent muss der Kopf auf 70 Prozent der Größe einer Person gesenkt werden, um als Kniebeuge gewertet zu werden. Bei einer Person von 1.80 Meter muss der Kopf so auf unter rund 1.30 Meter gesenkt werden.

Die Erkennung einer des Duckens einer Kniebeuge, wie im Code-Ausschnitt beschrieben, wird in Abbildung 6.2 dargestellt. Dabei muss der Kopf unter die rote Linie gesenkt werden, welche ungefähr 70 Prozent der Körperhöhe repräsentiert. Außerdem muss der Kopf gerade gehalten, dass heißt die Rotation nach oben und unten nicht mehr als rund 15 Grad abweichen. In diesem Ansatz wird nur die Position und Rotation des Kopfes sowie die gemessene Höhe und die aktuelle Distanz zum Boden verwendet.

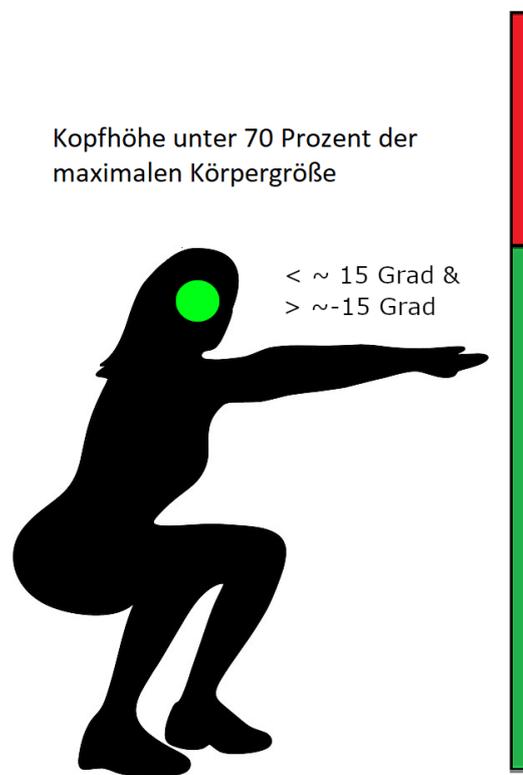


Abbildung 6.2: Erkennung 10 einer Kniebeuge

Um zu Erkennen, ob eine Kniebeuge abgeschlossen wurde, muss festgestellt werden, ob die User und Userinnen wieder in die Startposition zurückkehren. Dies wurde in der Microsoft HoloLens Applikation durch erneutes Errechnen der Höhe des Kopfes in Prozent realisiert. Zuerst wird erneut überprüft, ob der Kopf der Nutzer und Nutzerinnen sich zu weit nach vorne oder zurück bewegt hat. Anschließend wird überprüft, ob die Anwender und Anwenderinnen aufrecht stehen, das heißt ihr Kopf sich über 90 Prozent ihrer maximalen Größe befindet. Außerdem muss die Person zuerst eine Kniebeuge ausgeführt haben, damit das Aufstehen erkannt wird. Code-Ausschnitt 6.3 zeigt diese Erkennung:

```
1 void checkIfStoodUp(Vector3 headPos)
2 {
3     float offsetX=0.25f;
4     float normDistToFloor=distanceToFloor/actDistanceToFloor;
5
6     if (headPos.x > cameraStartingPosition.x - offsetX &&
7         headPos.x < cameraStartingPosition.x + offsetX &&
8         normDistToFloor > 0.9 && squatting)
9     {
10        numberOfSquatsDone++;
11        squatting = false;
12        kneeBendBool = true;
13    }
14 }
```

Source-Code Ausschnitt 6.3: Microsoft HoloLens - Erkennung des Aufstehens bei Kniebeugen

Im Rahmen der Erkennung des Steppers wurde die Erkennung von Kniebeugen so modifiziert, dass diese nur erkannt wurden, wenn die ausführende Person sich nicht auf dem Stepper befindet. Diese Änderung wurde allerdings aufgrund unzureichender Trackingergebnisse wieder rückgängig gemacht. Diese Problematik wird später in diesem Kapitel noch detaillierter erläutert.

6.2.2 Erkennung von Seitschritten

Die Erkennung von Seitschritten in der Microsoft HoloLens Applikation erwies sich als verhältnismäßig simpel, da nur die Kopfposition als Messposition verfügbar ist. Dadurch wurde die Erkennung auf die Kopfbewegung angepasst. Um einen Seitschritt erkennen zu können, müssen sich die Nutzer und Nutzerinnen in einer aufrechten Position befinden und den Kopf gerade halten.

Ein Seitschritt wird erkannt, wenn die übende Person sich eine festgelegte Distanz, *sideDif*, welche im Konfigurationsinterface geändert werden kann, von der Referenzposition entfernt. Dieser Wert wird in Meter angegeben, beschreibt die notwendige seitliche Kopfbewegung und beträgt in allen Applikationen zwischen 0.2 und 0.7 Metern.

Die Rotation wird verwendet um sicherzustellen, dass bei Seitschritten der Kopf gerade gehalten wird. Diese Rotation darf in keine der möglichen Richtungen abweichen. Auch hier wurde ein Boolean definiert, um die mehrfach Erkennungen zu vermeiden. Da es hier keinen Abschluss der Übung wie bei den Kniebeugen gibt, wird nach erfolgreicher Ausführung nach einer Wartezeit von einer Sekunde die nächste Messung eines Seitschrittes ermöglicht.

Neben der Rotation und dem Abstand zur Referenzposition müssen die Nutzer und Nutzerinnen sich in einer aufrechten Position, das heißt Kopf zwischen 90 Prozent und 105 Prozent der Körperhöhe befinden. Diese Einschränkung verhindert, dass Seitschritte in kniender Position oder auf dem Stepper erkannt werden.

Die Überprüfung findet einmal für Seitschritte nach links und einmal für Seitschritte nach rechts statt. Code-Ausschnitt 6.4 zeigt diese Erkennung:

```
1 void detectSideStep(Vector3 headPos,Quaternion holoRotation)
2 {
3     float normDistToFloor=distanceToFloor/actDistanceToFloor;
4     float offsetHeadRotation = 0.17f;
5
6     if (
7         !sideStepped &&
8         cameraStartingPosition.z<headPos.z-float.Parse(sideDif)
9         && normDistToFloor >0.9 && normDistToFloor < 1.05 &&
10        holoRotation.x < offsetHeadRotation &&
11        holoRotation.x > -offsetHeadRotation &&
12        holoRotation.y < offsetHeadRotation &&
13        holoRotation.y > -offsetHeadRotation &&
14        holoRotation.z < offsetHeadRotation &&
15        holoRotation.z > -offsetHeadRotation)
16        ||
17        !sideStepped &&
18        cameraStartingPosition.z>headPos.z+float.Parse(sideDif)
19        && normDistToFloor > 0.9 && normDistToFloor < 1.05 &&
20        holoRotation.x < offsetHeadRotation &&
21        holoRotation.x > -offsetHeadRotation &&
22        holoRotation.y < offsetHeadRotation &&
23        holoRotation.y > -offsetHeadRotation &&
24        holoRotation.z < offsetHeadRotation &&
25        holoRotation.z > -offsetHeadRotation)
26        {
27            numberOfSideStepsDone++;
28            sideStepped = true;
29            StartCoroutine(SetSideSteppedFalse(1f));
30        }
31 }
```

Source-Code Ausschnitt 6.4: Microsoft HoloLens - Erkennung von Seitschritten

In diesem Ansatz wird die Bewegung der Beine nicht mitverfolgt, was die Erkennung anfällig für Ausführungsfehler macht. Diese Erkennung bietet daher keine optimale Voraussetzung zur Erkennung von absolut korrekt ausgeführten Seitschritten.

6.2.3 Erkennung von Step-Ups

Die Erkennung von Step-Ups basiert auf einem ähnlichen Prinzip wie die Erkennung der Kniebeugen. Die Übung wird in zwei Schritte unterteilt:

- Die Erkennung des Step-Ups
- Die Erkennung des Step-Downs

Für die Erkennung des Step-Ups wurde einerseits die Höhe des Kopfes in Prozent zur Körpergröße und andererseits der über das Spatial Mapping erkannte Stepper verwendet. Ob ein Step-Up durchgeführt wurde, ließ sich über die Körpergröße leicht sagen, da der Kopf der Person sich über der maximalen Körpergröße befand. Dies konnte allerdings auch durch einen einfachen Sprung abgebildet werden. Aus diesem Grund wurde die Erkennung des Steppers hinzugezogen.

6 Bewegungserkennung mit der Microsoft HoloLens

Die Erkennung des Step-Ups anhand der Körperhöhe verlief problemlos. Wie in Abbildung 6.3 bereits ersichtlich, leidet allerdings die Genauigkeit des Spatial Mappings bei der Erkennung des Steppers. Es wurden mehrere Aufnahmen aus einer gewählten Perspektive gemacht, bei sieben von zehn Versuchen ähnelte das Ergebnis der oberen Erkennung. Nur bei den anderen drei kam es zu einer akzeptablen Erkennung wie sie unten in Abbildung 6.3 abgebildet ist. Die mittlere Grafik zeigt den verwendeten Stepper. Aufgrund der beschriebenen Gründe wurde die Erkennung des Steppers über das Spatial Mapping entfernt. Code-Ausschnitt 6.5 zeigt diese Erkennung:

```
1 void checkIfSteppedUp()
2 {
3     float normDistToFloor=distanceToFloor/actDistanceToFloor;
4     if (normDistToFloor > 1.10)
5     {
6         steppedUp = true;
7         if (stepUpBool)
8         {
9             numberOfStepUpsDone++;
10        }
11        stepUpBool = false;
12    }
13 }
```

Source-Code Ausschnitt 6.5: Microsoft HoloLens - Erkennung von Step-Ups

Für die Erkennung eines Step-Ups wird in der Microsoft HoloLens Applikation nur der normalisierte Abstand zum Boden verwendet. Ein Step-Up wird erkannt, wenn sich die Person über ihrer maximalen Körpergröße befindet. In diesem getesteten Rahmen muss eine Person diese Größe um 10 Prozent überschreiten. Der für die Testzwecke verwendete Stepper hat eine Höhe von 25 Zentimetern, was diese Grenze für Personen, die nicht größer als 2.5 Meter sind möglich macht.

Abbildung 6.3 zeigt deutlich die Mängel der Spatial Mappings der Microsoft HoloLens. Da das Spatial Mapping für die Applikation jedes Mal ausgeführt wird, ist es essenziell für die Erkennung der Step-Ups, dass der Stepper richtig erkannt wird. Die Anwendung versucht, den Stepper mittels Raycasts zu finden und erlaubt die erfolgreiche Wertung der Übung nur, wenn dieser erkannt wurde. Da das Spatial Mapping nicht von den Anwendern und Anwenderinnen kontrolliert wird, kam es bei mangelhafter Genauigkeit des Spatial Mappings zur Nichterkennung der ausgeführten Step-Ups.

Diese Nichterkennung der ausgeführten Step-Ups führte zu einer gewissen Frustration der User und Userinnen. Auch bei dem Step-Down traten ähnliche Schwierigkeiten mit der Erkennung auf. Die Messung der Höhe des Kopfes in Prozent zur Körpergröße lieferte auch hier zuverlässige Ergebnisse, während diese oft aufgrund fehlender Erkennung des Steppers nicht gewertet wurden.



Abbildung 6.3: Spatial Mapping des verwendeten Steppers

Aus diesem Grund wurde die Stepper Erkennung aus der Microsoft HoloLens Applikation entfernt und die Erkennung nur über die Messung der Höhe des Kopfes in Prozent zur Körpergröße durchgeführt. Auch diese Variante ist nicht der optimale Erkennungsalgorithmus, da die Position und Rotation der Beine vernachlässigt wird.

Die Erkennung des Step-Down auf Basis der Höhe des Kopfes in Prozent der Körperhöhe erwies sich als sehr zuverlässig. Sobald ein Step-Up erkannt wurde und sich die Position des Kopfes anschließend wieder unter der maximalen Körpergröße befand und dieser gerade gehalten wird, wurde die Übung als abgeschlossen gewertet. Code Ausschnitt 6.6 zeigt diese Erkennung:

```
1 void checkIfSteppedDown(Quaternion holoRotation)
2 {
3     float normDistToFloor=distanceToFloor/actDistanceToFloor;
4     float offsetHeadRotation = 0.17f;
5
6     if (normDistToFloor < 1.04 && steppedUp &&
7         holoRotation.x < offsetHeadRotation &&
8         holoRotation.x > -offsetHeadRotation )
9     {
10        steppedUp = false;
11        stepUpBool = true;
12        numberOfStepUpsDone++;
13    }
14 }
```

Source-Code Ausschnitt 6.6: Microsoft HoloLens - Erkennung von Step-Downs

Um ein Step-Down erfolgreich zu erkennen, muss sich der Kopf der übenden Person sich unter dem festgelegten Threshold befinden und dieser darf nicht nach oben oder unten geneigt sein. Diese Neigung wurde zur Erkennung hinzugefügt, da beim Ausweichen der Hindernisse auf den Stepper User und Userinnen vermehrt nach unten geblickt haben, um die Hindernisse zu sehen. In manchen dieser Fälle führte dies zu einer Falscherkennung von Step-Downs. Aus diesem Grund wurde die Rotation zur Erkennung hinzugefügt, um diese Falscherkennungen zu vermeiden.

6.2.4 Erkennung von Zehen berühren mit den Händen

Im Rahmen der Microsoft HoloLens Applikation wurde die Erkennung der Berührung der Zehen mittels bereits erklärtem Algorithmus umgesetzt. Ursprünglich war geplant, die Hände mittels der Kamera der Microsoft HoloLens zu erkennen und diese Position in die Übung miteinzubeziehen. Allerdings erkennt die HoloLens die Hände nur, wenn eine Geste ausgeführt wird. So kann ohne zusätzliche Hardware nicht auf die Handposition geschlossen werden [56]. Um die Applikation frei von zusätzlicher Hardware zu halten, wurde auf die Erkennung der Hände über diese Methode verzichtet.

Als Alternative, um die Handerkennung dennoch einbinden zu können, wurde versucht, die Hände und Füße über das Spatial Mapping zu erkennen. Abbildung 6.4 zeigt die Ergebnisse dieses Versuches und das aktuelle Sichtfeld der Microsoft HoloLens, dargestellt als grüner Rahmen. Aus der Abbildung lässt sich schließen, dass die Erkennung viel zu unpräzise ist, um diesen Ansatz im Rahmen eines therapeutischen Trainings zu verwenden. Aus diesem Grund wurde auch dieser Ansatz für die Applikation nicht verwendet.

Um die Erkennung der Übung trotz dieser Schwierigkeiten umzusetzen, wurde auf die Erkennung der Hände verzichtet und die Position der Beugung des Kopfes nach vorne als Messwert vorgezogen. Die Applikation überprüft die Senkung und die Rotation des Kopfes der User und Userinnen. Wenn diese einen definierten Wert erreichen, gilt die Übung als geschafft. Folgender Code-Ausschnitt beschreibt diese Erkennung genauer.

6 Bewegungserkennung mit der Microsoft HoloLens

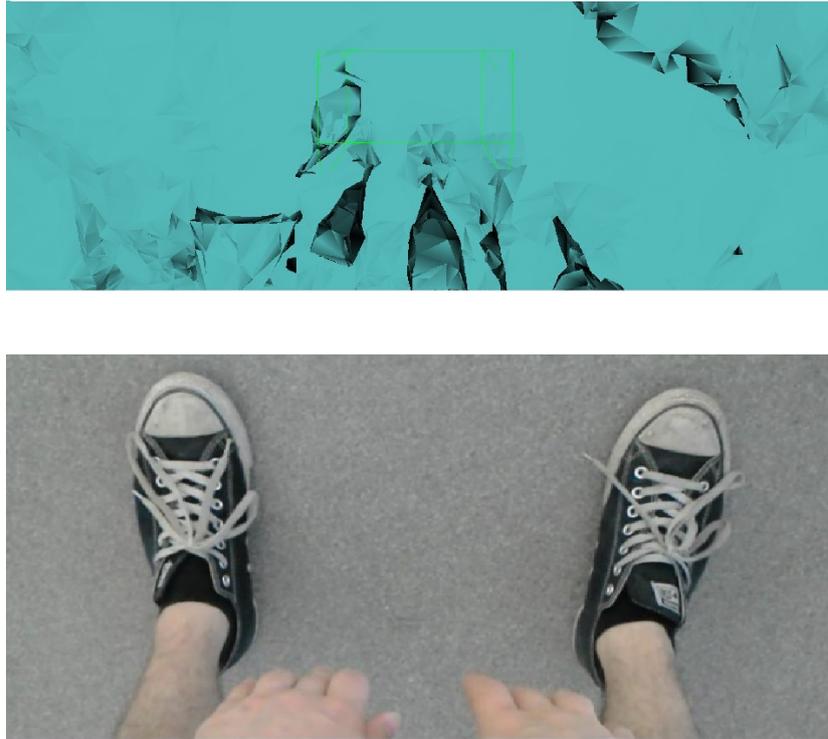


Abbildung 6.4: Spatial Mapping und Realbild der Hände und Füße

Um eine Zehenberührung erfolgreich zu erkennen, muss der Kopf der Nutzer und Nutzerinnen unter einen über das Konfigurationsinterface festgelegten Threshold, `toesDif`, gesenkt werden. Im Rahmen dieser Applikation wurde der Rahmen für diesen Wert zwischen 40 Prozent und 60 Prozent festgelegt, als Standardwert wird 50 Prozent angenommen. Code-Ausschnitt 6.7 zeigt diese Erkennung:

```
1 void checkIfToeTouched(Vector3 headPos,Quaternion headRotation)
2 {
3     float offsetX = 0.25f;
4     float offsetHeadRotation = 0.17f;
5     float normDistToFloor=distanceToFloor/actDistanceToFloor;
6
7     if (headPos.x > cameraStartingPosition.x + offsetX &&
8         normDistToFloor < float.Parse(toesDif) &&
9         (holoRotation.x > offsetHeadRotation ||
10         holoRotation.x < -offsetHeadRotation))
11     {
12         if (toeTouched == false)
13         {
14             numberOfToesDone++;
15             toeTouched = true;
16         }
17     }
18 }
```

Source-Code Ausschnitt 6.7: Microsoft HoloLens - Erkennung von Zehenberührungen

Außerdem muss der Abstand des Kopfes nach vorne größer sein als die Länge zur Referenzposition und einem Offset. Dieser Offset dient dazu, die Distanz festzulegen, welche sich die

Person sich nach vorne beugen muss.

Zusätzlich zur Tiefe der Bewegung und der Bewegung nach vorne, wird die Rotation des Kopfes überprüft. Diese Rotation wird in der Microsoft HoloLens Applikation verwendet, um festzustellen, ob die Nutzer und Nutzerinnen nach unten blicken.

Diese Form der Aufzeichnung vernachlässigt die Handerkennung, allerdings ist die Übung eine sehr bekannte Dehnungsübung, welche üblicherweise intuitiv mit den Händen zum Boden ausgeführt wird. Im Rahmen der Usability-Tests wird genauer analysiert, ob die Erkennung der Hände essenziell für die korrekte Ausführung der Übung ist.

6.2.5 Schwierigkeiten der Bewegungserkennung

Die größten Herausforderungen der Bewegungserkennung waren folgende Punkte:

- Die Definition der Referenzposition
- Die Ungenauigkeit des Spatial Mappings bei Details
- Die Handerkennung

Die Definition der Referenzposition wurde bereits diskutiert, ist aber ein essenzieller Faktor, um diesen Algorithmus zur Bewegungserkennung zu ermöglichen. Wenn die Frequenz der Aktualisierung der Referenzposition zu hoch ist, wird die Ausführung der Übungen nicht mehr erkannt, da die Vergleichsergebnisse zu klein ausfallen. Die Referenzposition entspricht im schlimmsten Fall permanent der aktuellen Position.

Wenn die Frequenz der Aktualisierung der Referenzposition zu gering ist, wird die Ausführung der Übungen in einem gewissen Radius erkannt, allerdings ist es nicht mehr möglich, die Übungen bei Ausführung im gesamten Raum zu erkennen. Um diese effizient definieren zu können, mussten im Rahmen dieser Arbeit zahlreiche Kalibrierungsvorgänge gestartet werden. Durch diese umfangreiche Kalibration konnte eine Referenzposition gefunden werden, welche eine genaue und räumlich unabhängige Übungserkennung garantiert.

Die Ungenauigkeit des Spatial Mappings bei Details war eine große Herausforderung dieser Arbeit. Das Spatial Mapping wurde unter anderem für die Höhenmessung, die Steppererkennung und die Handerkennung verwendet. Im Rahmen der Höhenmessung musste der Boden erkannt werden, was aufgrund des flächendeckenden Spatial Mappings problemlos funktionierte.

Die Erkennung des Steppers zeigte erste Problematiken des Spatial Mappings auf. Abbildung 6.3 zeigt deutlich die Mängel der Spatial Mappings der Microsoft HoloLens. Da das Spatial Mapping für die Applikation jedes Mal ausgeführt wird, ist es essenziell für die Erkennung der Step-Ups, dass der Stepper richtig erkannt wird. Aufgrund dessen, dass das Spatial Mapping den Stepper mangelhaft erkennt, kam es zur Nichterkennung der ausgeführten Step-Ups.

Die Ungenauigkeit bei der Handerkennung basiert auf dem gleichen Problem wie die Steppererkennung. Das Spatial Mapping der Microsoft HoloLens zielt auf großflächige, unbewegliche Gegenstände ab. Die Erkennung von kleineren, beweglichen Dingen schlägt vermehrt fehl.

Ein großes Problemfeld war auch die Erkennung der Hände mit der Microsoft HoloLens. Trotz verfügbarer Kamera und Gestenerkennung, war es im Rahmen dieser Arbeit nicht möglich, die Hände dauerhaft ohne zusätzliche Hardware zu erkennen. Die Erkennung bei Ausführung von Gesten ist ein nützliches Feature, allerdings im Trainingskontext zu ungenau, da bei der Übungsausführung die Hände üblicherweise nicht in Gestenstellung gehalten werden.

7 Bewegungserkennung mit der HTC Vive

In diesem Kapitel wird die Umsetzung der Applikation für die HTC Vive detailliert beschrieben. Der Aufbau der Anwendung basiert auf der bereits beschriebenen Applikation für die Microsoft HoloLens, allerdings wurde die Bewegungserkennung angepasst. Im Rahmen dieser Entwicklung wurden zusätzlich, neben der HTC Vive und den Controllern, noch die HTC Vive Bewegungstracker verwendet.

7.1 HTC Vive

Die HTC Vive ist ein HMD, welches von HTC in Kooperation mit Valve hergestellt wird. Die offizielle Einführung des Devices war im Jahre 2016. Das gesamte Grundausstattungspaket umfasst das Headset, zwei Controller und zwei Basisstationen zur Raumerkennung. Die Grundausstattung wird in Abbildung 7.1 dargestellt. Weiters zeigt Abbildung 7.1 noch die HTC Vive Tracker, welche zur Bewegungserkennung im Rahmen dieser Arbeit verwendet wurden [21].



Abbildung 7.1: Darstellung der HTC Vive mit Controllern und Basisstationen und Trackern [21]

Das HMD hat ein Sichtfeld von ca. 110 Grad, über zwei 1080 mal 1200 Pixel Displays. Diese Pixeldichte ist vergleichbar mit anderen HMDs, wie zum Beispiel der Oculus Rift. Die Erkennung der HTC Vive basiert hauptsächlich auf dem Outside-In Tracking. Das Headset und die Controller werden über die beiden Basisstationen erkannt [35]. Im Rahmen der Entwicklung

der Applikation verlor die HTC Vive vermehrt die Verbindung, da sie von den Basisstationen nicht erkannt wurde. Dabei wurde das Bild grau und es konnte nicht mehr mit dem Device interagiert werden, bis es wieder erkannt wurde.

Virtual Reality

Im Gegensatz zur Microsoft HoloLens ist die Applikation für die HTC Vive keine MR Applikation, sondern eine VR Applikation. Das heißt, die Nutzer und Nutzerinnen tauchen vollständig in die virtuelle Welt ein. Bei diesem Vorgang wird die reale Welt für die Benutzer und Benutzerinnen ausgeblendet [43].

Die Immersion ist im Rahmen von VR Anwendungen essenziell. Immersion beschreibt, wie sehr die Nutzer und Nutzerinnen sich in die virtuelle Welt hineinversetzen können. Bei einer hohen Immersion wird die Wahrnehmung der realen Welt verringert und die User und Userinnen fühlen sich mehr in der virtuellen Umgebung. Die Immersion kann unter anderem durch folgende Faktoren beeinflusst werden [43]:

- Gestaltung der virtuellen Welt
- Intuitivität der Interaktion
- Vielfalt der Möglichkeiten in der virtuellen Welt

Die intuitive Verwendung der Interaktionsmöglichkeiten ist essenziell, um die Immersion zu fördern. Wenn die Aktionen der Nutzer und Nutzerinnen eine Auswirkung auf die virtuelle Welt haben und die User und Userinnen informiert werden, steigt die Immersion. Diese Interaktivität täuscht den Anwendern und Anwenderinnen eine Echtheit ihrer Handlungen vor. Weiters ist es essenziell, die virtuelle Welt so zu gestalten, dass sie für die Nutzer und Nutzerinnen glaubwürdig erscheint [43].

Eine große Herausforderung von VR ist die so genannte Bewegungsübelkeit. Diese entsteht, wenn die realen Bewegungen von der virtuellen Beschleunigung abweichen. Um dies zu vermeiden, ist es notwendig, dass die virtuellen Bewegungen und Beschleunigungen an die realen Bewegungen angepasst werden [43].

HTC Vive Tracker

Die HTC Vive Tracker sind ein Zusatzzubehör für das HMD, welche es ermöglichen, weitere Positionen im Raum zu erkennen und verfolgen. Die Tracker sind in Abbildung 7.1 oben abgebildet. Über die Tracker kann eine Drahtlosverbindung zwischen dem Vive System und der zusätzlichen Positionserkennung hergestellt werden [20].

Die Tracker können verwendet werden, um diverse Gegenstände zu tracken oder auch am Körper befestigt werden, um ein gesamtes Körpertracking zu ermöglichen. Die Tracker können bis zu sechs Stunden verwendet werden, ohne aufgeladen zu werden. Sie sind ca. 10 cm breit und enthalten eine LED Anzeige, welche den Status des Devices widerspiegelt. Der Startknopf ist der in Abbildung 7.1 abgebildete Logoknopf in der Mitte der Geräte. Um die Position der Devices zu erkennen, nutzen die Tracker mehrere Infrarot Sensoren [55].

Der Grund für die Entwicklung der HTC Vive Tracker war die Integration von realen Gegenständen in die virtuelle Welt. Durch die Verbindung eines Trackers mit einem realen Gegenstand, kann die Position in der virtuellen Umgebung verwendet werden [55].

7.2 Applikation für die HTC Vive

Die Applikation, welche für die HTC Vive entwickelt wurde, ist ebenfalls mit Unity 3D umgesetzt worden. In Kombination dazu wurde Steam VR verwendet. Das Ziel der Applikation gleicht dem Ziel der bereits beschriebenen Applikation, eine solide Basis für die Erkennung von physiotherapeutischen Übungen zu bilden. Der große Unterschied zur Microsoft HoloLens ist die Verwendung von VR.

Diese Arbeit hat bereits die Wichtigkeit von Immersion aufgezeigt. Aus diesem Grund war es nicht möglich, einfach die Microsoft HoloLens Applikation zu übernehmen, sondern es musste eine virtuelle Welt gestaltet werden, welche die Nutzer und Nutzerinnen eintauchen lässt. Diese virtuelle Welt muss effektiv gestaltet werden, um eine hohe Immersion und dadurch volle Konzentration für die Übungen zu erreichen.

Die gestaltete Welt wird in Abbildung 7.2 dargestellt.



Abbildung 7.2: Darstellung der virtuellen Umgebung

Essenziell für die Gestaltung der virtuellen Welt ist die visuelle Begrenzung der begehbaren Zone. Diese Begrenzung wird in der HTC Vive Applikation durch den Holzzaun dargestellt. Abbildung 7.2 zeigt den virtuellen Trainingsraum, welcher auf dem Grundkonzept der Microsoft HoloLens basiert.

Das Interface wurde auch an die HTC Vive Applikation angepasst, indem es im virtuellen Raum platziert wurde. Im Rahmen der Microsoft HoloLens, konnte das Menü frei im realen Raum platziert werden, wodurch es im virtuellen Raum in der Luft zu schweben schien. Aus diesem Grund wurde die Hintergrundtafel, welche in Abbildung 7.2 ersichtlich ist, eingeführt, um dem Menü zusätzlichen Kontrast zu geben.

Neben der Gestaltung der virtuellen Welt, war auch die effektive Gestaltung der Interaktion ein wichtiger Punkt dieser Applikation. Im Rahmen der Microsoft HoloLens Applikation erfolgte der Großteil der Interaktion über Gesten. Dieser Ansatz konnte für die HTC Vive

Applikation nicht übernommen werden, da die HTC Vive ohne zusätzliche Hardware keine Gestenerkennung unterstützt.

Die Interaktion der HTC Vive App erfolgt über Körperbewegung und die HTC Vive Controller. Über die Controller können die Knöpfe in der Applikation bedient werden. Dies erfolgt durch das Zielen mittels einer Lasers auf die Knöpfe und das Bestätigen mit der Trigger Taste. Diese Art der Interaktion mittels Controller ist in VR Applikationen gängig.

Über Körperbewegung kann auch mit der Applikation interagiert werden. Hierbei werden die Bewegungen der Nutzer und Nutzerinnen erkannt und ihnen visuelles und auditives Feedback gegeben. Diese Erkennung basiert auf der Positionserkennung durch die Basisstationen sowie die HTC Vive Tracker.

7.2.1 Bewegungserkennung in der Applikation

Im Rahmen der HTC Vive Applikation wurden folgende Übungen erkannt:

- Kniebeugen
- Seitschritte
- Step-Ups
- Berühren der Zehen mit den Händen

Der ursprüngliche Bewegungserkennungsansatz gleicht dem der für die Microsoft HoloLens Applikation verwendet wurde. Dieser wurde allerdings um die Bewegungserkennen mittels der HTC Vive Tracker erweitert. Durch diese Erweiterung konnten die Übungen genauer erkannt und Bewegungen spezifiziert werden. Außerdem konnte die Erkennung der Hände, welche bei der Microsoft HoloLens Applikation nicht möglich war, in dieser Applikation über die HTC Vive Controller umgesetzt werden.

Dieser Ansatz misst die Position der Benutzer und Benutzerinnen im Raum während der Ausführung der Applikation, mittels dem Tracking der HTC Vive Basisstationen. Die Position wird mit einer definierten Referenzposition verglichen und anhand dieser die Abstände und Bewegungen der Anwender und Anwenderinnen gemessen. Die Referenzposition wird bei größeren Positionsveränderungen im Raum aktualisiert und sichergestellt, dass die Übungen im ganzen Raum ausgeführt werden können.

Die exakte Erkennung der Bewegung geschieht immer über ein Messen des Abstands zwischen Referenzposition und aktueller Position. Dabei spielen die Abweichung Links – Rechts, die Höhe und die Abweichung Vorne – Hinten eine essenzielle Rolle. Diese werden alle über die in den Basisstationen eingebauten Sensoren getrackt. Durch die dauerhaften Vergleiche kann eine stabile Erkennung garantiert werden.

Die Distanz zum Boden wird in der HTC Vive Applikation über den Abstand des linken Trackers am Fuß zum HMD gemessen. Folgender Code-Ausschnitt zeigt den Ablauf dieser Erkennung. Dieser Ausschnitt ist nicht vollkommen funktionstüchtig, er zeigt nur die Vorgehensweise:

```
1 camera=GameObject.Find("MainCamera").transform.position;
2
3 if (leftFootTracker)
4 {
5     height=camera.y-leftFootTracker.transform.position.y;
6 }
```

Source-Code Ausschnitt 7.1: HTC Vive - Erkennung des Bodens

Erkennung von Kniebeugen

Die Erkennung von Kniebeugen erfolgt in zwei Schritten:

- Die Erkennung des Duckens
- Die Erkennung des Aufstehens

Bei der Erkennung des Duckens wird die Höhe des Kopfes der Anwender und Anwenderinnen in Prozent zur gemessenen Körpergröße berechnet. Dafür wird die Distanz des Kopfes zum Boden durch die gesetzte Referenzpositionshöhe dividiert. Anschließend wird verglichen, ob die Position der User und Userinnen nicht zu stark von der Referenzposition abweicht (Kniebeugen müssen an einem Standpunkt durchgeführt werden). Zusätzlich wird mittels der HTC Vive Tracker der Abstand zwischen den Trackern, welche auf den Knien positioniert sind und den Trackern, welche auf den Füßen positioniert sind, gemessen. Dieser errechnete Abstand wird verwendet, um erkennen zu können, ob eine wirkliche Kniebewegung ausgeführt wurde, oder nur der Kopf gesenkt wurde, um eine Kniebeuge vorzutäuschen. Die definierten Variablen wurden in Kapitel 6, Bewegungserkennung mit der Microsoft HoloLens, erläutert. Code-Ausschnitt 7.2 zeigt diese Erkennung:

```

1 void checkIfSquatDone(Vector3 headPos, Quaternion headRotation)
2 {
3     float diffLeft;
4     float diffRight;
5     diffLeft = Mathf.Abs(TrackerLeftFoot.transform.position.x)-
6     Mathf.Abs(TrackerLeftKnee.transform.position.x);
7
8     diffRight = Mathf.Abs(TrackerRightFoot.transform.position.x) -
9     Mathf.Abs(TrackerRightKnee.transform.position.x);
10
11     float offsetX = 0.25f;
12     float offsetHeadRotation = 0.17f;
13     float normDistToFloor=distanceToFloor/actDistanceToFloor;
14
15     if (
16         headPos.x > cameraStartingPosition.x - offsetX &&
17         headPosition.x < cameraStartingPosition.x + offsetX &&
18         normDistToFloor < float.Parse(squatDif) &&
19         holoRotation.x < offsetHeadRotation &&
20         holoRotation.x > -offsetHeadRotation &&
21         diffRight < 0.1f &&
22         diffLeft < 0.1f)
23     {
24         squating = true;
25         if (kneeBendBool)
26         {
27             numberOfSquatsDone++;
28         }
29         kneeBendBool = false;
30     }
31 }
```

Source-Code Ausschnitt 7.2: HTC Vive - Erkennung des Duckens bei Kniebeugen

Die im Code-Ausschnitt 7.2 erläuterte Erkennung wird in Abbildung 7.3 detailliert dargestellt.

Dabei muss der Kopf unter die rote Linie gesenkt werden, welche ungefähr 70 Prozent der Körperhöhe repräsentiert. Außerdem muss der Kopf gerade gehalten werden, dass heißt die Rotation nach oben und unten nicht mehr als rund 15 Grad abweichen. Zusätzlich dazu muss der horizontale Abstand der Knie- und Fußtracker kleiner sein als zehn Zentimeter. In diesem Ansatz werden die Position und Rotation des Kopfes sowie die gemessene Höhe und die aktuelle Distanz zum Boden verwendet. Außerdem die Positionen der Knie- und Fußtracker.

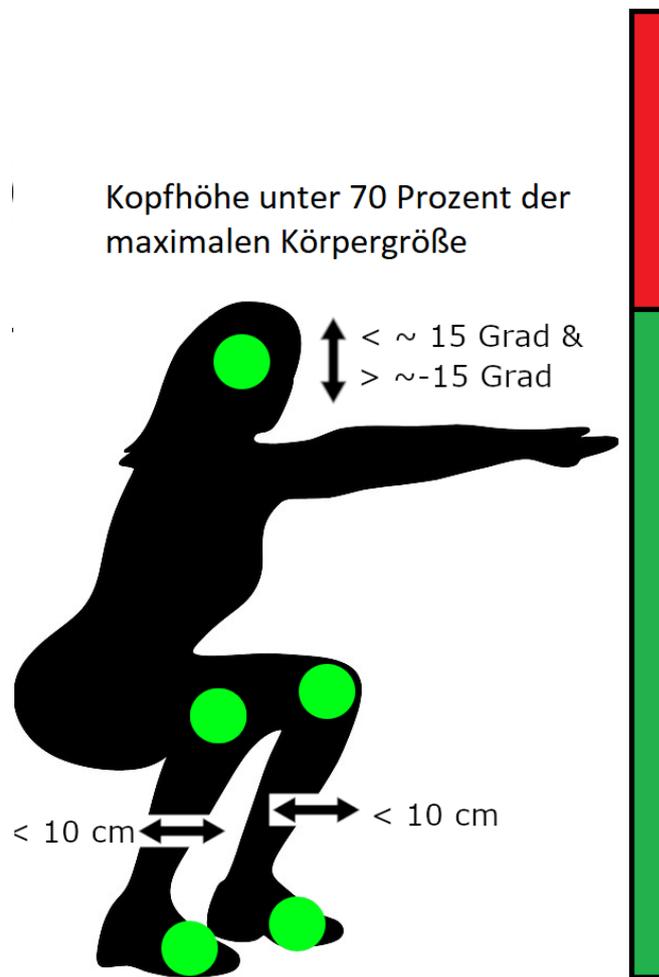


Abbildung 7.3: Erkennung einer Kniebeuge

Die ursprüngliche Erkennung der Kniebeuge basiert auf dem Algorithmus der Microsoft HoloLens Applikation. Diese wurde allerdings um den Abstand zwischen Fuß und Knie erweitert. Dieser Abstand wird im Code als `diffLeft` und `diffRight` bezeichnet. Im Rahmen der HTC Vive Applikation wird überprüft, ob der Abstand vom Knie nach vorne zum Fuß nicht größer ist als zehn Zentimeter. In einer Stillstandposition beträgt dieser Abstand mehr als dieser Wert, bei der korrekten Ausführung einer Kniebeuge verringert sich dieser Abstand und die Ausführung wird erkannt. Dieser Wert wurde durch sorgfältiges Testen überprüft.

Die Erkennung des Aufstehens wurde im Rahmen der HTC Vive Applikation nicht verändert.

Erkennung von Seitschritten

Die Erkennung von Seitschritten in der HTC Vive Applikation ist simpel, da die Kopfposition sowie die Position der Füße und Knie als Messpositionen verfügbar sind. Um einen Seitschritt erkennen zu können, müssen sich die Nutzer und Nutzerinnen in einer aufrechten Position

befinden und den Kopf gerade halten. In der HTC Vive Applikation wird zusätzlich zu der Kopfposition wie in der Microsoft HoloLens Applikation auch die Position der Tracker überprüft, um festzustellen, ob nur der Kopf bewegt wurde. Code-Ausschnitt 7.3 zeigt diese Erkennung:

```

1 void detectSideStep(Vector3 headPos,Quaternion holoRotation)
2     {
3         float normDistToFloor=distanceToFloor/actDistanceToFloor;
4         float offsetHeadRotation = 0.17f;
5         float diff = Mathf.Abs(TrackerLeftFoot.transform.position.z)-
6         Mathf.Abs(TrackerRightFoot.transform.position.z);
7         if (!sideStepped &&
8             cameraStartingPosition.z<headPos.z-
9             float.Parse(sideDif) && diff > 0.10f &&
10            normDistToFloor > 0.9 && normDistToFloor < 1.05 &&
11            holoRotation.x < offsetHeadRotation &&
12            holoRotation.x > -offsetHeadRotation &&
13            holoRotation.y < offsetHeadRotation &&
14            holoRotation.y > -offsetHeadRotation &&
15            holoRotation.z < offsetHeadRotation &&
16            holoRotation.z > -offsetHeadRotation ||
17            !sideStepped && diff > 0.10f &&
18            cameraStartingPosition.z > headPos.z +
19            float.Parse(sideDif) &&
20            normDistToFloor > 0.9 && normDistToFloor < 1.05 &&
21            holoRotation.x < offsetHeadRotation &&
22            holoRotation.x > -offsetHeadRotation &&
23            holoRotation.y < offsetHeadRotation &&
24            holoRotation.y > -offsetHeadRotation &&
25            holoRotation.z < offsetHeadRotation &&
26            holoRotation.z > -offsetHeadRotation)
27         {
28             numberOfSideStepsDone++;
29             sideStepped = true
30             StartCoroutine(SetSideSteppedFalse(1f));
31         }
32     }

```

Source-Code Ausschnitt 7.3: HTC Vive - Erkennung von Seitschritten

Die ursprüngliche Erkennung der Seitschritte basiert auf dem Algorithmus der Microsoft HoloLens Applikation. Diese wurde allerdings um den Abstand zwischen den Füßen erweitert. Der Abstand zwischen den beiden Trackern an den Füßen wird permanent errechnet und zusätzlich zur Kopfbewegung für die Erkennung von Seitschritten verwendet. Dieser wird im Code als diff bezeichnet. Im Rahmen der HTC Vive Applikation muss zusätzlich zu den bereits beschriebenen Faktoren der Abstand zwischen den Füßen mehr als 10 Zentimeter betragen, um einen Seitschritt erfolgreich durchzuführen.

Erkennung von Step-Ups

Die Erkennung von Step-Ups basiert auf einem ähnlichen Prinzip wie die Erkennung der Kniebeugen. Die Übung wird in zwei Schritte unterteilt:

- Die Erkennung des Step-Ups

- Die Erkennung des Step-Downs

Für die Erkennung des Step-Ups und Step-Downs wurde der Algorithmus der Microsoft HoloLens verwendet, die HTC Vive Tracker wurden für diese Übung nicht verwendet. Ob ein Step-Up durchgeführt wurde, ließ sich über die Körpergröße leicht sagen, da der Kopf der Person sich über der maximalen Körpergröße befand.

Die Erkennung des Step-Down auf Basis der Höhe des Kopfes in Prozent der Körperhöhe erwies sich als sehr zuverlässig. Sobald ein Step-Up erkannt wurde und sich die Position des Kopfes anschließend wieder unter der maximalen Körpergröße befand, wurde die Übung als abgeschlossen gewertet.

Erkennung von Zehen berühren mit den Händen

Im Rahmen der HTC Vive Applikation wurde die Erkennung der Berührung der Zehen über die HTC Vive Tracker umgesetzt. Es wird die Position zwischen Füßen und Knien verglichen und daraus errechnet, ob eine Kniebeuge oder eine Dehnung zu den Zehen vorliegt. Zusätzlich zu dieser Positionserkennung wird auch die Position der Hände in die Erkennung miteinbezogen. Um die Übung erfolgreich durchführen zu können, müssen beide Controller von der Kamera des Headsets erkannt werden. Diese Erkennung erfolgt über die Variablen `leftVisible` und `rightVisible`. Diese werden auf `True` gesetzt, wenn sich die Controller im Blickfeld der Nutzer und Nutzerinnen befinden. Code-Ausschnitt 7.4 zeigt diese Erkennung:

```

1 void checkIfToeTouched(Vector3 headPos, Quaternion headRotation)
2 {
3     GameObject left = GameObject.FindTag("ctrlLeft");
4     GameObject right = GameObject.FindTag("ctrlRight");
5     bool leftVisible;
6     bool rightVisible;
7     float diffL=Mathf.Abs(TrackerLeftFoot.transform.position.x)-
8     Mathf.Abs(TrackerLeftKnee.transform.position.x);
9
10    float diffR=Mathf.Abs(TrackerRightFoot.transform.position.x) -
11    Mathf.Abs(TrackerRightKnee.transform.position.x);
12
13    Vector3 screenPointLeft =
14    Camera.main.WorldToViewportPoint(left.transform.position);
15
16    Vector3 screenPointRight =
17    Camera.main.WorldToViewportPoint(right.transform.position);
18
19    if (screenPointLeft.z>0 && screenPointLeft.x>0 &&
20        screenPointLeft.x<1 && screenPointLeft.y>0
21        && screenPointLeft.y<1)
22        {
23            leftVisible = true;
24        }
25    else { leftVisible = false;}
26
27    if (screenPointRight.z>0 && screenPointRight.x>0 &&
28        screenPointRight.x<1 && screenPointRight.y>0 &&
29        screenPointRight.y<1)
30        {

```

```

31         rightVisible = true;
32     }
33     else { rightVisible = false; }
34
35     float offsetX = 0.25f;
36     float offsetHeadRotation = 0.17f;
37     float normDistToFloor=distanceToFloor/actDistanceToFloor;
38
39     if (headPos.x > cameraStartingPosition.x - offsetX &&
40         headPos.x < cameraStartingPosition.x + offsetX &&
41         normDistToFloor < float.Parse(toesDif) &&
42         rightVisible && leftVisible &&
43         diffL > 0.10f && diffR > 0.10f &&
44         (holoRotation.x > offsetHeadRotation ||
45         holoRotation.x < -offsetHeadRotation))
46     {
47         if (toeTouched == false)
48         {
49             numberOfToesDone++;
50             toeTouched = true;
51         }
52     }
53 }

```

Source-Code Ausschnitt 7.4: HTC Vive - Erkennung von Zehenberührungen

Die ursprüngliche Erkennung der Zehenberührungen basiert auf dem Algorithmus der Microsoft HoloLens Applikation. Diese wurde allerdings um den Abstand zwischen Knie und Fuß sowie die Erkennung der Hände erweitert. Dieser wird im Code als diffL und diffR bezeichnet. Die Erkennung der Hände erfolgt über die Controller der HTC Vive, wo über die Position dieser errechnet wird, ob sich diese im Bild befinden. Für die Erkennung der Übung müssen sich die Controller im Bild befinden, das heißt die Nutzer und Nutzerinnen müssen ihre Hände beim Ausführen dieser Übung sehen.

Die Errechnung des Abstandes zwischen Knie und Fuß gleicht der Rechnung bei der Erkennung von Kniebeugen. Im Gegensatz zu dieser muss dieser Abstand in der Erkennung allerdings größer sein als zehn Zentimeter, um eine erfolgreiche Zehenberührung durchzuführen.

7.2.2 Schwierigkeiten der Bewegungserkennung

Die größten Herausforderungen der Bewegungserkennung in der HTC Vive Applikation waren folgende Punkte:

- Die Erkennung der HTC Vive Tracker
- Der geringe Abstand zwischen Knie und Fuß
- Die Platzierung des realen Steppers in der virtuellen Welt

Die initiale Erkennung der HTC Vive Tracker mittels Steam VR verlief problemlos. Diese konnten einfach hinzugefügt werden und wurden in Unity3D erkannt. Die große Schwierigkeit war die Verbindung des Headsets mit den Basisstationen und den Trackern. Im Rahmen dieser Entwicklung verlor das Headset mehrfach die Verbindung während der Anwendung, so konnten Bewegungen nicht mehr erkannt werden. Diese Verbindungsprobleme konnten verbessert werden, indem die Basisstationen mit Kabel verbunden wurden. Allerdings trat das Problem nach wie vor vereinzelt auf.

7 Bewegungserkennung mit der HTC Vive

Eine weitere Schwierigkeit war der geringe Abstand zwischen Knie und Fuß. Für die Bewegungserkennung war es essenziell den Abstand zwischen Knie und Fuß über die HTC Vive Tracker zu ermitteln. In der Regel betrug der Abstand allerdings im geraden Stand nur ca. zehn bis fünfzehn Zentimeter. Dieser geringe Abstand machte die Definition der zu erkennenden Übungen schwierig. Allerdings konnte durch umfangreiches Testen und geeignete Kalibration der Abstand genau genug errechnet werden, um die effektive Erkennung zu gewährleisten. Die letzte der beschriebenen Schwierigkeiten war die Platzierung des realen Steppers in der virtuellen Welt. Die Anwender und Anwenderinnen müssen auch in der HTC Vive Applikation auf einen realen Stepper steigen, um ein Step-Up durchzuführen. Allerdings sehen sie diesen nicht, da sie sich in der virtuellen Umgebung befinden. Aus diesem Grund wurde der Stepper in die virtuelle Welt eingefügt. In Abbildung 7.2 ist dieser rechts unten erkennbar. Dieser wurde so kalibriert, dass, wenn die User und Userinnen auf den virtuellen Stepper steigen möchten, sie automatisch auf den realen Stepper steigen, da die Positionen identisch sind.

8 Bewegungserkennung mithilfe eines Smartphones

In diesem Kapitel wird die Umsetzung der Applikation für ein Smartphone detailliert beschrieben. Der Aufbau der Anwendung basiert auf der bereits beschriebenen Applikation für die Microsoft HoloLens, allerdings wurde die Bewegungs- und Raumerkennung angepasst. Im Rahmen dieser Entwicklung wurden zusätzlich zu einem Smartphone noch ein HMD verwendet, um das Gerät während der Ausführung der Übungen zu fixieren.

Das Ziel der Erstellung dieser Anwendung ist zu zeigen, dass es möglich ist, mittels einem Smartphone die Bewegungserkennung der Microsoft HoloLens zu imitieren und dadurch ein vergleichbares Tracking zu ermöglichen. Für die Entwicklung wurde ein günstiges HMD, das Hao Mirror AR Headset, gewählt um den Vergleich möglichst zu unterstützen.

8.1 Beschreibung der verwendeten Hard- und Software

Für die Entwicklung der Applikation wurde diverse Hard- und Software benutzt. Neben der bereits für die Basisapplikation verwendeten Hard- und Software wurden zusätzliche folgende Komponenten verwendet:

- ARCore
- Samsung Galaxy S8+
- Haori Mirror

8.1.1 AR Core

ARCore ist die Plattform von Google zur Entwicklung von Augmented Reality Applikationen. Über die Verwendung von unterschiedlichen APIs ermöglicht ARCore einem Smartphone, die Umgebung der Nutzer und Nutzerinnen wahrzunehmen [15]. ARCore benutzt, um den virtuellen Inhalt in die reale Welt einzublenden, folgende drei Schlüsselpunkte:

- Tracken der Position des Smartphones
- Verstehen der Umgebung
- Abschätzen des Lichtes

Tracken der Position des Smartphones

ARCore verwendet einen Prozess, welcher als Odometrie bezeichnet wird, um zu verstehen, wo sich das Telefon relativ zur Umgebung befindet. Dafür werden von ARCore visuell markante Punkte im Kamerabild erkannt, welche als Feature-Punkte bezeichnet werden. Diese Punkte werden verwendet, um die Standortänderung zu berechnen. Die visuelle Information wird mit Messungen der Sensoren des Smartphones kombiniert, um die Ausrichtung (Position und Rotation) der Kamera relativ zur Welt über die Zeit zu schätzen [17].

Durch das Ausrichten der Pose der virtuellen Kamera, die ihren 3D-Inhalt mit der von ARCore bereitgestellten Pose der Kamera des Geräts rendert, können virtuelle Inhalte aus der richtigen Perspektive dargestellt werden. Das gerenderte, virtuelle Bild kann über das Bild gelegt werden, das von der Kamera des Geräts erhalten wird, so dass es aussieht, als wäre der virtuelle Inhalt Teil der realen Welt [17].

Verstehen der Umgebung

ARCore verbessert ständig das integrierte Verständnis der realen Umwelt durch die Erkennung von Feature-Punkten und Flächen. Dabei wird nach Ansammlungen von Feature-Points gesucht, welche auf horizontalen oder vertikalen Flächen liegen. Diese erkannten Flächen werden, wie in Abbildung 8.1 gezeigt, für die Applikation verfügbar gemacht. Diese Information kann verwendet werden, um auf der erkannten Fläche, wie in Abbildung 8.1 dargestellt, virtuelle Inhalte zu platzieren [17].

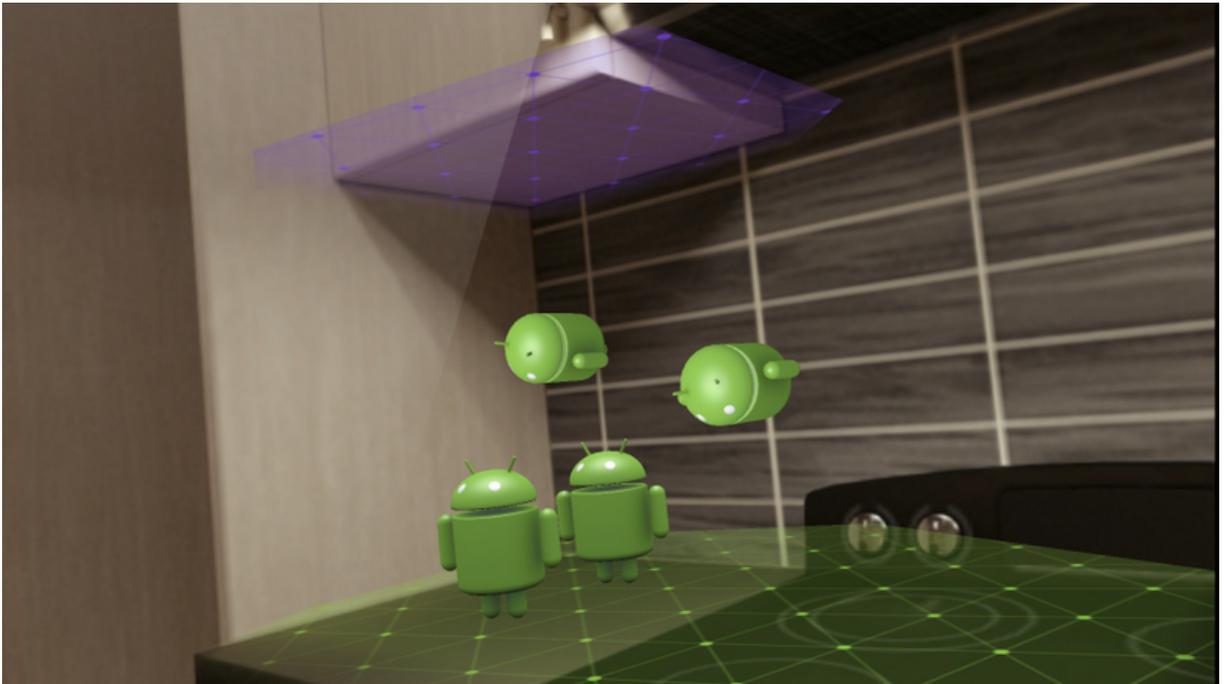


Abbildung 8.1: Umgebungserkennung mit ARCore [16]

Da ARCore Feature-Punkte zum Erkennen von Ebenen verwendet, werden Oberflächen ohne Textur, zum Beispiele eine weiße Wand, möglicherweise nicht richtig erkannt [17].

Abschätzen des Lichtes

ARCore kann Informationen über die Beleuchtung seiner Umgebung erkennen und die durchschnittliche Intensität und Farbkorrektur eines bestimmten Kamerabildes feststellen. Mit dieser Information können die virtuellen Objekte unter den gleichen Bedingungen wie die reale Umgebung beleuchtet werden und so der Realitätssinn erhöht werden [17].

8.1.2 Samsung Galaxy S8+

Das Samsung Galaxy S8+ ist im Rahmen dieser Applikation aufgrund der Kompatibilität mit ARCore verwendet worden. ARCore unterstützt nicht alle Geräte, sondern nur neuere Generationen an Smartphones. Die exakten unterstützten Devices können auf der ARCore Homepage eingesehen werden.

Bildschirm

Das Galaxy S8+ hat leicht nach hinten gebogene Display-Ränder, welche eine größere Fläche des Displays ermöglicht. Dieses ist 6,2 Zoll groß und hat eine Auflösung von 2960 x 1440

8 Bewegungserkennung mithilfe eines Smartphones

Pixel. Abbildung 8.2 zeigt das Smartphone, sowie die Maße des Devices. Durch die fehlenden physischen Knöpfe wurde das Display auch nach unten erweitert. Das Bildschirmverhältnis beträgt 18,5:9 [41].



Abbildung 8.2: Samsung Galaxy S8+ [41]

Kamera

Die eingebaute Rückkamera des Smartphones spielt im Rahmen der Applikation eine essenzielle Rolle, da die Umgebungserkennung auf dieser basiert. Diese hat eine Auflösung von 12 Megapixel. Die Kamera verfügt über einen optischen Bildstabilisator und eine automatische Scharfstellung. Neben der Rückkamera besitzt das Smartphone noch eine Frontkamera mit einer Auflösung von 8 Megapixel, welche allerdings in der Applikation nicht verwendet wird. Die Rückkamera unterstützt FullHD Aufnahmen mit 60 Bildern pro Sekunde, auch eine 4K Aufnahme ist möglich [41].

8.1.3 Haori Mirror

Die Haori Mirror ist ein AR Headset, welches auf der Kamera des Smartphones basiert. Dieses Headset wurde für die Applikation gewählt, da es sehr kostengünstig ist. Dadurch sollte der Vergleich zur Microsoft HoloLens und der HTC Vive verdeutlicht werden. Außerdem bietet die Haori Mirror im Gegensatz zu den üblichen AR Headsets die Möglichkeit, die echte Welt nicht durch die Kamera des Smartphones wahrzunehmen. Dies wird von der Haori Mirror dadurch ermöglicht, dass die Nutzer und Nutzerinnen nicht direkt durch die Kamera des Smartphones blicken, sondern durch eine transparente Fläche, auf die der Bildschirminhalt projiziert wird [18].

Abbildung 8.3 zeigt die Haori Mirror. In der Abbildung lässt sich die transparente Fläche erkennen, auf die das Display des Smartphones projiziert wird. Der abgebildete Nutzer blickt

8 Bewegungserkennung mithilfe eines Smartphones

nicht auf das Smartphone, sondern auf den projizierten Inhalt. Dadurch kann er die reale Umgebung und eingeblendete Inhalte wahrnehmen.



Abbildung 8.3: Haori Mirror AR Headset [18]

8.2 Erstellung der Applikation

Die Applikation, welche für das Smartphone entwickelt wurde, wurde ebenfalls mit Unity 3D entwickelt. In Kombination dazu wurde ARCore verwendet. Das Ziel der Applikation gleicht dem Ziel der bereits beschriebenen Applikationen, eine solide Basis für die Erkennung von physiotherapeutischen Übungen zu bilden. Der große Unterschied zur HTC Vive Applikation ist die Verwendung von AR.

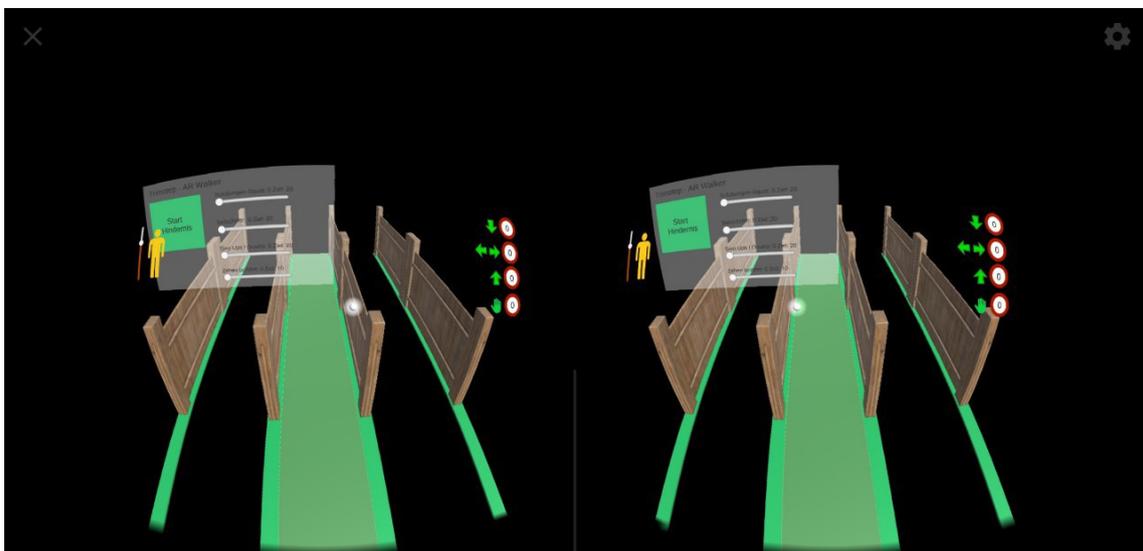


Abbildung 8.4: Konfiguration der AR Kamera für die Smartphone Applikation

8 Bewegungserkennung mithilfe eines Smartphones

Im Rahmen der Erstellung der Applikation wurde das Design der Applikation angepasst, um dem Design der Haori Mirror zu entsprechen. Dafür wurde die AR Kamera von AR Core verwendet, um das Bild für zwei Linsen anzuzeigen. Abbildung 8.4 verdeutlicht diese Konfiguration. Durch die Anpassung der Applikation für die zwei Linsen des AR Headsets, wirkt die Abbildung 8.4 etwas verzogen, vor allem beim Interface, diese Krümmung konnte in der Applikation jedoch nicht nachvollzogen werden.

Für die Interaktion mit der Applikation wurde kein Controller, sondern die Interaktion mittels Blickrichtung, auch Gaze genannt, verwendet. Im Gegensatz zur Microsoft HoloLens Applikation wurde in dieser Applikation auch keine Interaktion mit Gesten verwendet, da die Erkennung nicht von ARCore unterstützt wird.

Gaze ist eine Form der Interaktion mit einer Applikation. Über die Blickrichtung kann festgestellt werden, wohin die Nutzer und Nutzerinnen in der virtuellen Welt blicken. Dadurch können Informationen über ihr Vorhaben gewonnen werden, da üblicherweise Objekte angesehen werden, mit denen Interagiert wird [49].

In dieser Applikation wird nicht die Position der Augen, sondern die Position der Kamera des Smartphones als Ursprungsort genutzt. Wenn sich die Nutzer und Nutzerinnen bewegen, wird die Blickrichtung analysiert und gespeichert. In der Applikation wird die Blickrichtung über einen Cursor, abgebildet als weißer Punkt, wie in Abbildung 8.4, dargestellt. Wenn dieser mit Objekten der virtuellen Welt kollidiert, werden die Interaktionen in dieser Applikation ausgelöst.

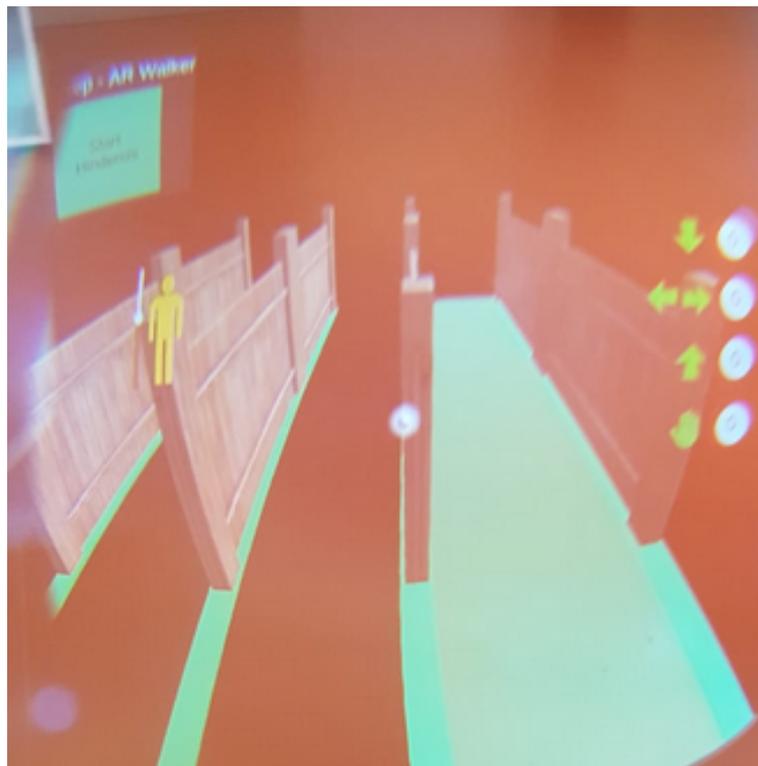


Abbildung 8.5: Konfiguration der AR Kamera für die Smartphone Applikation

Abbildung 8.5 zeigt den Blick der Nutzer und Nutzerinnen durch die Haori Mirror. Es wird das gespiegelte Bild der Anwendung dargestellt.

Im Rahmen dieser Applikation wurde die Interaktion mittels Gaze für das Menü, zum Einblenden der Hindernisse, benutzt. Dafür müssen die Nutzer und Nutzerinnen für 2 Sekunden die Knöpfe des Menüs ansehen, um diese zu betätigen. Im Rahmen der Usability Tests, welche

während der Entwicklung durchgeführt wurden, werden die Ergebnisse zu dieser Form der Interaktion diskutiert.

8.2.1 Umsetzung der Bewegungserkennung

Im Rahmen der Smartphone Applikation werden folgende Übungen erkannt:

- Kniebeugen
- Seitschritte
- Step-Ups
- Berühren der Zehen mit den Händen

Der ursprüngliche Bewegungserkennungsansatz gleicht dem, der für die Microsoft HoloLens Applikation verwendet wurde. Dieser Ansatz misst die Position der Benutzer und Benutzerinnen im Raum, während der Ausführung der Applikation, mittels dem Tracking von ARCore. Die Position des Devices wird in Relation zur Umgebung über die in das Smartphone integrierten Sensoren erkannt. Diese wird anschließend mit einer definierten Referenzposition verglichen und anhand dieser die Abstände und Bewegungen der Anwender und Anwenderinnen gemessen. Die Referenzposition wird bei größeren Positionsveränderungen im Raum aktualisiert und sichergestellt, dass die Übungen im ganzen Raum ausgeführt werden können.

Die exakte Erkennung der Bewegung geschieht immer über ein Messen des Abstands zwischen Referenzposition und aktueller Position. Dabei spielen die Abweichung links – rechts, die Höhe und die Abweichung vorne – hinten eine essenzielle Rolle. Diese werden alle über die in das Smartphone eingebauten Sensoren getrackt. Durch die dauerhaften Vergleiche kann eine stabile Erkennung garantiert werden.

Die Messung der Größe der Nutzer und Nutzerinnen erfolgt in dieser Anwendung über die Verwendung von ARCore. Das ARCore Unity Plugin bietet eine bereits integrierte Variante den Boden zu erkennen und in Unity darzustellen. Diese Funktion wurde im Rahmen der Anwendung genutzt und in Code-Ausschnitt 8.1 dargestellt:

```
1  Session.GetTrackables<TrackedPlane>
2  (m_NewPlanes, TrackableQueryFilter.New);
3
4  for (int i = 0; i < m_NewPlanes.Count; i++)
5  {
6      floorHeight=m_NewPlanes[0].CenterPose.position.y;
7  }
```

Source-Code Ausschnitt 8.1: ARCore - Erkennung des Bodens

Von ARCore wird der Boden mittels Flächen, sogenannten Planes, repräsentiert. Nachdem diese erkannt werden, werden diese in einem Array gespeichert. Im Rahmen dieser Applikation wird angenommen, dass sich der Boden an der Position der ersten gefunden Plane befindet. In den Konfigurationseinstellungen von ARCore wurde eingestellt, dass nur horizontale Flächen erkannt werden, um zu verhindern, dass Wände die Erkennung beeinflussen.

Erkennung von Kniebeugen

Die Erkennung von Kniebeugen erfolgt in zwei Schritten:

- Die Erkennung des Duckens

- Die Erkennung des Aufstehens

Bei der Erkennung des Duckens wird die Höhe des Kopfes der Anwender und Anwenderinnen in Prozent zur gemessenen Körpergröße berechnet. Dafür wird die Distanz des Kopfes zum Boden durch die gesetzte Referenzpositionshöhe dividiert. Anschließend wird verglichen, ob die Position der User und Userinnen nicht zu stark von der Referenzposition abweicht (Kniebeugen müssen an einem Standpunkt durchgeführt werden).

Um zu Erkennen, ob eine Kniebeuge abgeschlossen wurde, muss festgestellt werden, ob die User und Userinnen wieder in die Startposition zurückkehren. Dies wurde wie in der Microsoft HoloLens Applikation umgesetzt. Zuerst wird überprüft, ob der Kopf gerade gehalten wird und ob nicht zu weit vom Ausgangsort der Übung abgewichen wurde. Anschließend wird beim Erreichen einer aufrechten Position, das heißt, der Kopf muss sich auf mindestens 95 Prozent der gemessenen Körpergröße befinden, die Übung als erfolgreich abgeschlossen gewertet.

Die Erkennung von Kniebeugen in der Smartphone Applikation wurde nicht verändert, diese benutzt den Algorithmus der Microsoft HoloLens Applikation.

Erkennung von Seitschritten

Die Erkennung von Seitschritten in der Smartphone Applikation zeigte die gleiche Schwierigkeit wie die Erkennung in der Microsoft HoloLens Applikation. Nur die Kopfposition ist als Messposition verfügbar, dadurch wurde die Erkennung auf die Kopfbewegung angepasst. Um einen Seitschritt erkennen zu können, müssen sich die Nutzer und Nutzerinnen in einer aufrechten Position befinden und den Kopf gerade halten.

Ein Seitschritt wird erkannt, wenn die übende Person sich eine festgelegte Distanz von der Referenzposition entfernt. Dabei wird die Bewegung der Beine nicht mitverfolgt, was die Erkennung anfällig für Ausführungsfehler macht. Diese Erkennung bietet keine optimale Voraussetzung zur Erkennung von absolut korrekt ausgeführten Seitschritten.

Die Erkennung von Seitschritten in der Smartphone Applikation wurde nicht verändert, diese benutzt den Algorithmus der Microsoft HoloLens Applikation.

Erkennung von Step-Ups

Die Erkennung von Step-Ups basiert auf einem ähnlichen Prinzip wie die Erkennung der Kniebeugen. Die Übung wird, wie in der Microsoft HoloLens Applikation, in zwei Schritte unterteilt:

- Die Erkennung des Step-Ups
- Die Erkennung des Step-Downs

Für die Erkennung des Step-Ups wurde die Höhe des Kopfes in Prozent zur Körpergröße verwendet. Ob ein Step-Up durchgeführt wurde, ließ sich über die Körpergröße leicht sagen, da der Kopf der Person sich über der maximalen Körpergröße befand. Dies konnte allerdings auch durch einen einfachen Sprung abgebildet werden und somit die Übung umgangen werden.

Die Erkennung des Step-Down auf Basis der Höhe des Kopfes in Prozent der Körperhöhe erwies sich als sehr zuverlässig. Sobald ein Step-Up erkannt wurde und sich die Position des Kopfes anschließend wieder unter der maximalen Körpergröße befand, wurde die Übung als abgeschlossen gewertet.

Die Erkennung von Step-Ups und Step-Downs in der Smartphone Applikation wurde nicht verändert, diese benutzt den Algorithmus der Microsoft HoloLens Applikation.

Erkennung von Zehen berühren mit den Händen

Im Rahmen der Smartphone Applikation wurde die Erkennung der Berührung der Zehen mittels bereits erklärtem Algorithmus umgesetzt. Die umgesetzte Anwendung bietet keine Möglichkeit die Hände der Nutzer und Nutzerinnen zu erkennen, deshalb wurde, wie in der Microsoft HoloLens Applikation, auf die Erkennung der Hände verzichtet.

Um die Erkennung der Übung trotz dieser Schwierigkeiten umzusetzen, wurde die Position der Beugung des Kopfes nach vorne als Messwert vorgezogen. Die Applikation überprüft die Senkung und die Rotation des Kopfes der User und Userinnen. Wenn diese einen definierten Wert erreichen, gilt die Übung als geschafft.

Die Erkennung von Zehenberührungen in der Smartphone Applikation wurde nicht verändert, diese benutzt den Algorithmus der Microsoft HoloLens Applikation.

Diese Form der Erkennung vernachlässigt die Handerkennung, allerdings ist die Übung eine sehr bekannte Dehnungsübung, welche üblicherweise intuitiv mit den Händen zum Boden ausgeführt wird. Im Rahmen der Usability-Tests wird genauer analysiert, ob die Erkennung der Hände essenziell für die korrekte Ausführung der Übung ist.

8.2.2 Schwierigkeiten der Bewegungserkennung

Im Rahmen der Entwicklung der Applikation für das Smartphone in Kombination mit der Haori Mirror gab es folgende Schwierigkeiten:

- Bodenerkennung mit ARCore
- Akkuverbrauch des Smartphones
- Temperatur des Smartphones und automatische Sperre
- Unterschiedliche Position der Augen und der Kamera

Die Bodenerkennung mit ARCore erwies sich während der Entwicklung als Schwachstelle von ARCore. Die Erkennung an sich funktioniert problemlos und genau, allerdings nicht auf erhöhte Distanzen. Um den Boden erkennen zu können, müssen sich die Nutzer und Nutzerinnen nach unten beugen, da eine Erkennung aus der Standposition nicht möglich ist. Dies beeinflusst die Initialisierung der Applikation immens im Vergleich zur Microsoft HoloLens. Das Spatial Mapping dieser konnte ohne Schwierigkeiten im Stand ausgeführt werden.

Der Akkuverbrauch des Smartphones während der Applikation ist immens. Die Anwendung verbraucht den gesamten Akku des Smartphones innerhalb von 2 Stunden. Diese Laufzeit ist für ein Training derzeit ausreichend, allerdings ist diese Leistung nicht zufriedenstellend um langfristig damit effektiv Arbeiten zu können. Da sich das Smartphone während des Trainings in der Haori Mirror befindet, gibt es keine Möglichkeit den Akku zu laden, ohne die Bewegungsfreiheit der Nutzer und Nutzerinnen einzuschränken.

Neben dem Akkuverbrauch ist auch die Temperatur des Smartphones ein kritischer Faktor. Während den Tests in der Entwicklungsphase erhitze das Telefon stark. Nach rund 15 Minuten Nutzungsdauer forderte das Telefon einen auf, die Applikationen zu schließen und das Telefon nicht zu belasten. Dieser Hinweis konnte durch Bestätigen hingenommen werden, die Anwendung konnte allerdings trotzdem verwendet werden. Trotzdem wurde ein Leistungsabfall des Smartphones erkannt, wenn dieses weiter verwendet wurde. Die Bildrate hielt sich stets über 30 Bildern pro Sekunde, was eine zufriedenstellende Verwendung ermöglichte.

Die letzte Herausforderung der Entwicklung der Smartphone Applikation war die unterschiedliche Position der Augen und der Kamera. Durch die Verwendung der Haori Mirror wird das

8 Bewegungserkennung mithilfe eines Smartphones

Smartphone nicht vor den Augen der Nutzer und Nutzerinnen platziert. Diese sehen zwar den Inhalt der Kamera, dieser wird aber nicht aus der selben Perspektive aufgenommen, wie die Blicke der Anwender und Anwenderinnen.

Durch dieses Problem konnte die Höhenmessung nicht exakt durchgeführt werden, da sich das Smartphone einige Zentimeter über der eigentlichen Augenhöhe befindet. Außerdem entspricht das erwartete Bild der Kamera nicht dem real wahrgenommenen Bild, da es aus einem anderen Winkel aufgenommen wird. Die Spiegelung der Haori Mirror erreicht einen hohen Grad an Realität, ist aber nicht perfekt.

9 Vergleich der verwendeten Devices

In diesem Kapitel werden die beschriebenen Devices anhand diverser Faktoren verglichen. Dabei wird nicht die Genauigkeit der Bewegungserkennung untersucht, sondern die Genauigkeit des Tracking der Devices.

9.1 Definition der Messverfahren und Vergleichskriterien der Applikationen

Im Rahmen dieser Arbeit wurden die HTC Vive mit der Microsoft HoloLens verglichen. Dabei wurden folgende Parameter untersucht:

- Die Genauigkeit der Erkennung der Kopfposition im Raum
- Die Genauigkeit der Erkennung der Körpergröße
- Die Genauigkeit der Erkennung der Distanz zwischen zwei HTC Vive Trackern

9.1.1 Die Genauigkeit der Erkennung der Kopfposition im Raum

Die Genauigkeit der Erkennung der Kopfposition im Raum wurde bei den Devices über die HMDs gemessen. Dafür wurde bei der Microsoft HoloLens nur das Device selbst, bei der HTC Vive hingegen die Basisstationen für das Tracking verwendet. Für diese Messung wurden die HMDs an einer statischen Puppe mit einer Höhe von 1,85 Meter befestigt. Die Augenhöhe, also die Höhe der Devices, beträgt rund 1,80 Meter. Diese Befestigung wird in Abbildung 9.1 dargestellt.

Für diese Erkennung der Kopfposition im Raum wurde die Höhe der jeweiligen HMDs über eine bestimmte Zeit gemessen. Dafür wurden die Devices diesem Test unterzogen und die Ergebnisse anschließend verglichen. Diese werden am Ende dieses Kapitels vorgestellt und diskutiert.

Der Vergleich der Genauigkeit der Erkennung der Kopfposition im Raum hat in dieser Arbeit eine essenzielle Rolle, da die gesamte Bewegungserkennung der Microsoft HoloLens Applikation darauf basiert. Das heißt, dass die gesamte Bewegungserkennung dieser Applikation von der Genauigkeit dieser Erkennung abhängig ist. Um sicherzustellen, dass die Genauigkeit der Erkennung gewährleistet ist, wurde dieses Vergleichskriterium ausgewählt.

9.1.2 Die Genauigkeit der Erkennung der Körpergröße

Das zweite Vergleichskriterium ist die Genauigkeit der Erkennung der Körpergröße. Die Berechnung dieser Kennzahl unterscheidet sich zwischen den Applikationen stark. Im Rahmen der Microsoft HoloLens Applikation wird das Spatial Mapping verwendet, um den Boden zu erkennen und anschließend die Distanz zwischen Boden und HMD errechnet. Die HTC Vive Applikation hingegen verwendet die HTC Vive Tracker um den Boden zu erkennen. Auch in dieser Anwendung wird die Distanz zwischen Boden und HMD errechnet. In der Smartphone Applikation wird die Distanz zum Boden mittels ARCore errechnet.



Abbildung 9.1: Befestigung der HMDs auf der Messpuppe

Für diese Erkennung der Körpergröße wurde die Höhe der jeweiligen HMDs sowie die Höhe des Bodens über eine bestimmte Zeit gemessen. Dafür wurden die Devices diesem Test unterzogen und die Ergebnisse anschließend verglichen.

Auch dieses Kriterium spielt eine essenzielle Rolle, da die Körperhöhe in den Applikationen zur Erkennung der Übungen verwendet wird. Außerdem untersucht die Arbeit die Genauigkeit des Spatial Mappings der Microsoft HoloLens. Um sicherzustellen, dass die Genauigkeit der Erkennung gewährleistet ist, wurde dieses Vergleichskriterium ausgewählt.

9.1.3 Die Genauigkeit der Erkennung der Distanz zwischen zwei HTC Vive Trackern

Das dritte Vergleichskriterium ist die Genauigkeit der Erkennung der Distanz zwischen zwei HTC Vive Trackern. Die Berechnung dieser Kennzahl ist nur in der HTC Vive Applikation möglich. Dafür wurden die beiden Tracker, wie in Abbildung 9.1 dargestellt, an den Füßen der Messpuppe angebracht. Anschließend wurde die Distanz zwischen den Trackern gemessen, um eventuelle Abweichungen feststellen zu können.

Für diese Erkennung der Distanz zwischen zwei HTC Vive Trackern wurde die Position beider Tracker über eine bestimmte Zeit gemessen. Anschließend wurde der Unterschied der Positionen berechnet das Ergebnis analysiert.

Dieses Kriterium spielt eine essenzielle Rolle, da die Distanz zwischen den Trackern in der HTC Vive Applikation für die Erkennung von Kniebeugen, Seitschritten, Step-Ups und Zehenberührungen verwendet wird. Um sicherzustellen, dass die Genauigkeit der Erkennung gewährleistet ist, wurde dieses Vergleichskriterium ausgewählt.

9.1.4 Durchführung der Messungen

Um gleiche Verhältnisse für die Tests herzustellen, wurden die HMDs auf einer Puppe in der gleichen Position platziert. Außerdem wurde die Puppe während der Durchläufe nicht bewegt. Die Puppe wurde fixiert, um die Genauigkeit der Messvorgänge zu garantieren.

Alle Messungen wurden über 120 Sekunden durchgeführt. Dabei wurde jede Sekunde die Position der Tracker oder der HMDs, abhängig vom durchgeführten Test, mitgeloggt. Somit wurden für jede der definierten Kriterien 120 Referenzwerte generiert. Diese wurde in den Applikationen mitgeloggt und gespeichert.

9.2 Auswertung der Messergebnisse

Abbildung 9.2 zeigt die Ergebnisse der Messungen im Vergleich der HTC Vive und der Microsoft HoloLens. Der erste Teil der Abbildung beschreibt die Ergebnisse der Messung der Größe der Anwender und Anwenderinnen. Dabei ist zu erkennen, dass die Messungen der HTC Vive in einem häufigeren Intervall stattfinden, als die Messungen der Microsoft HoloLens. Die Ursache hierfür ist die Berechnung der Größe der Microsoft HoloLens. Dafür wird das Spatial Mapping des Raumes verwendet, welches nicht permanent, sondern nur in größeren Abständen aktualisiert wird. Die HTC Vive Berechnung erfolgt häufiger, da die Trackerposition permanent wiedergegeben wird.

Der Graph der Microsoft HoloLens beginnt, wie in Abbildung 9.2 dargestellt, nicht bei Sekunde null, sondern erst bei Sekunde vier. Der Grund hierfür ist die Dauer der Erkennung des Raumes. Zu Beginn versucht die Microsoft HoloLens den Raum zu erkennen und kann daher die Größe der Nutzer und Nutzerinnen noch nicht bestimmen.

Außerdem zeigt der erste Teil der Abbildung, dass beide Systeme in einem gewissen Rahmen von der Höhe abweichen. Dieser ist allerdings äußerst gering. Die Angaben der Grafik sind in Metern, allerdings lässt sich erkennen, dass die Abweichungen der HTC Vive sich im Millimeterbereich bewegen. Das Minimum der HTC Vive liegt bei rund 1.7736 Metern, das Maximum bei 1.7761. Dabei handelt es sich um eine maximale Abweichung von etwas mehr als zwei Millimeter. Ähnlich gering ist der Unterschied bei der Microsoft HoloLens, wie in Abbildung 9.2 oben ersichtlich.

Der mittlere Teil der Ergebnisse zeigt die Genauigkeit der Erkennung der Position der Geräte im Raum. Hier werden beide Geräte permanent aktualisiert, das heißt, die Positionsdaten werden häufig angepasst. Die HTC Vive hat eine gleichbleibende Erkennung der Position, mit kleinen Schwankungen im Mittelteil. Diese Schwankungen beschränken sich allerdings wieder auf Millimeter.

Die Erkennung der HoloLens zeigt eine etwas höhere Abweichung, welche sich allerdings auch noch in einem Bereich der kleiner ist als fünf Millimeter bewegt. Im Gegensatz zur Erkennung der HTC Vive verwendet die Microsoft HoloLens allerdings keine externen Sensoren um die Position des Gerätes zu erkennen.

Der letzte Teil der Grafik beschreibt eine Messung nur die HTC Vive betreffend. Dabei wurde die Distanz zwischen zwei HTC Vive Trackern analysiert. Die Abbildung zeigt, dass über den Zeitraum der Erkennung eine gewisse Abweichung auftrat. Diese Abweichung erstreckt sich über einen Bereich von rund 3.5 Millimetern.

Abbildung 9.2 zeigt die Ergebnisse des Vergleiches der beiden hochperformanten Geräte. Dieser Vergleich zeigt, dass beide Geräte kleine Abweichungen in der Messgenauigkeit besitzen, allerdings sind diese im Millimeterbereich. Diese kleinen Ungenauigkeiten führen im Rahmen der Applikationen zu keinen Schwierigkeiten.

9 Vergleich der verwendeten Devices

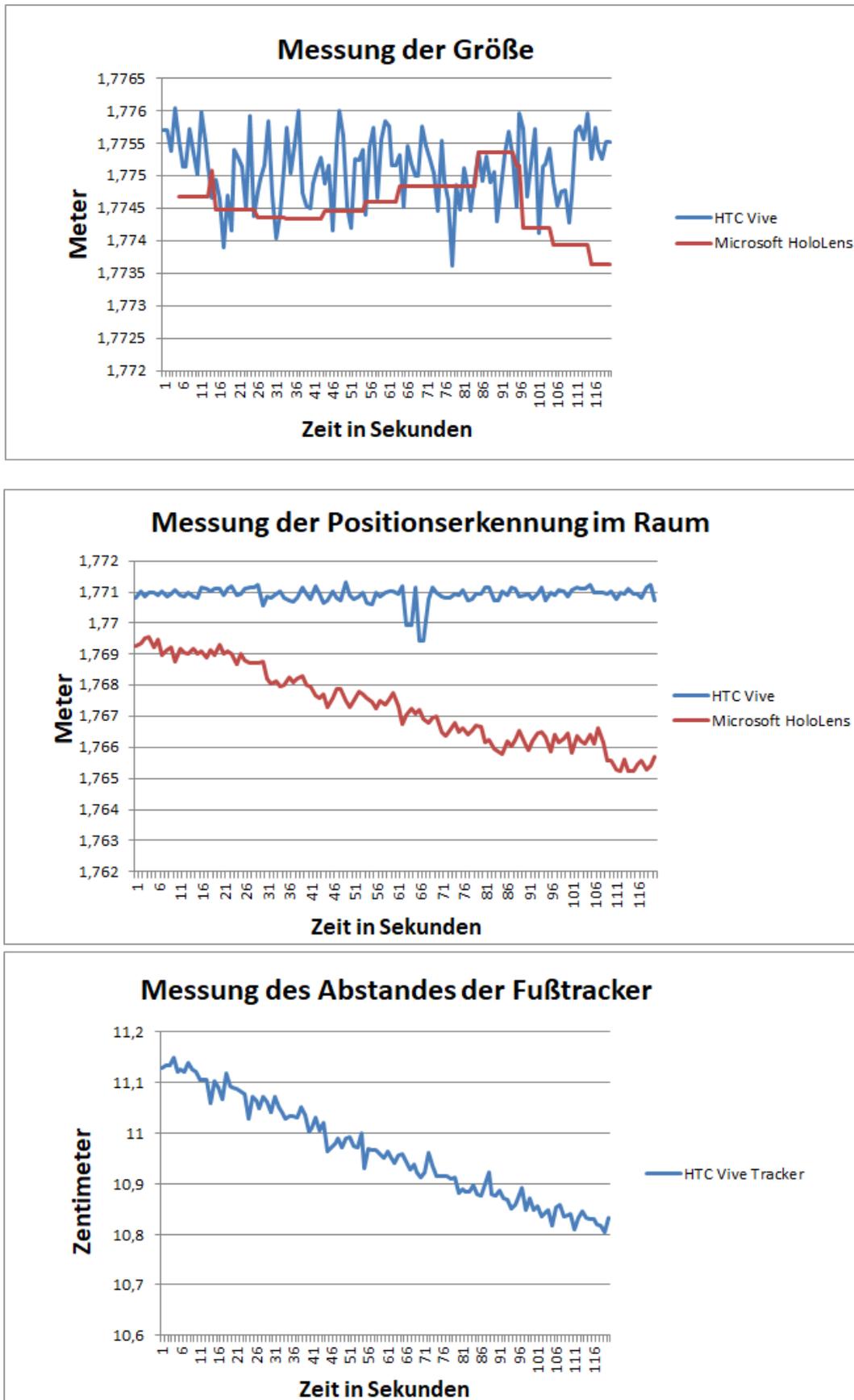
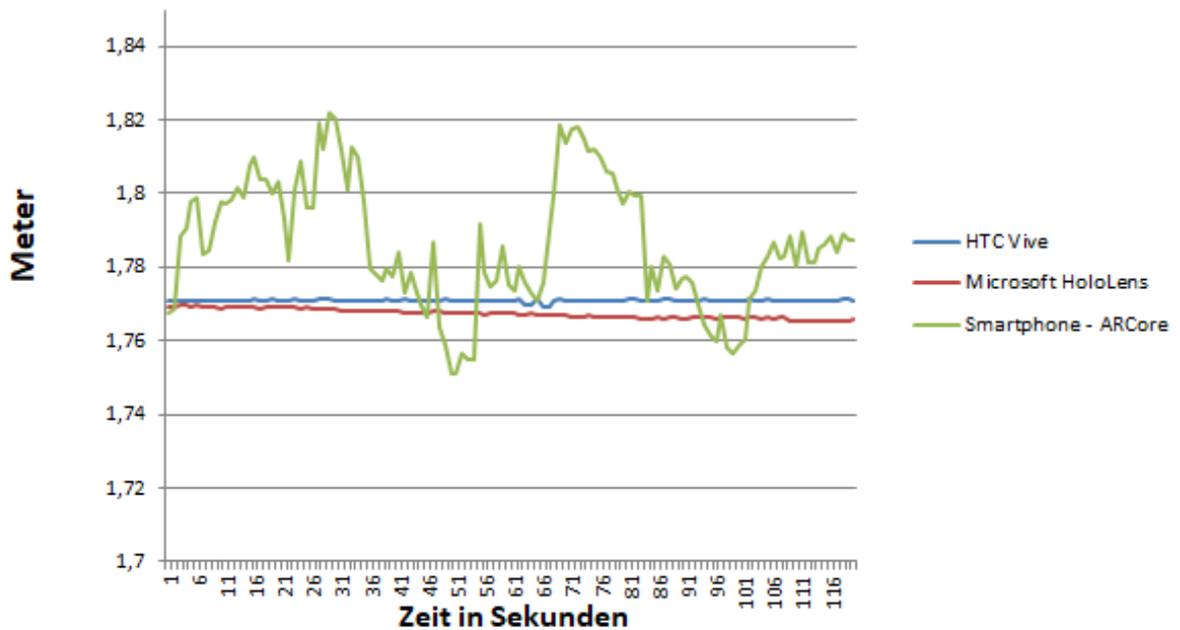


Abbildung 9.2: Ergebnisse der Messungen HoloLens und HTC Vive

9 Vergleich der verwendeten Devices

Neben diesen hochperformanten Devices wurde die Messungen auch mit dem Smartphone durchgeführt. Die Ergebnisse dieser Tests werden in Abbildung 9.3 dargestellt. Die Werte der HoloLens und der HTC Vive entsprechen denen aus Abbildung 9.2.

Messung der Positionserkennung im Raum



Messung der Größe

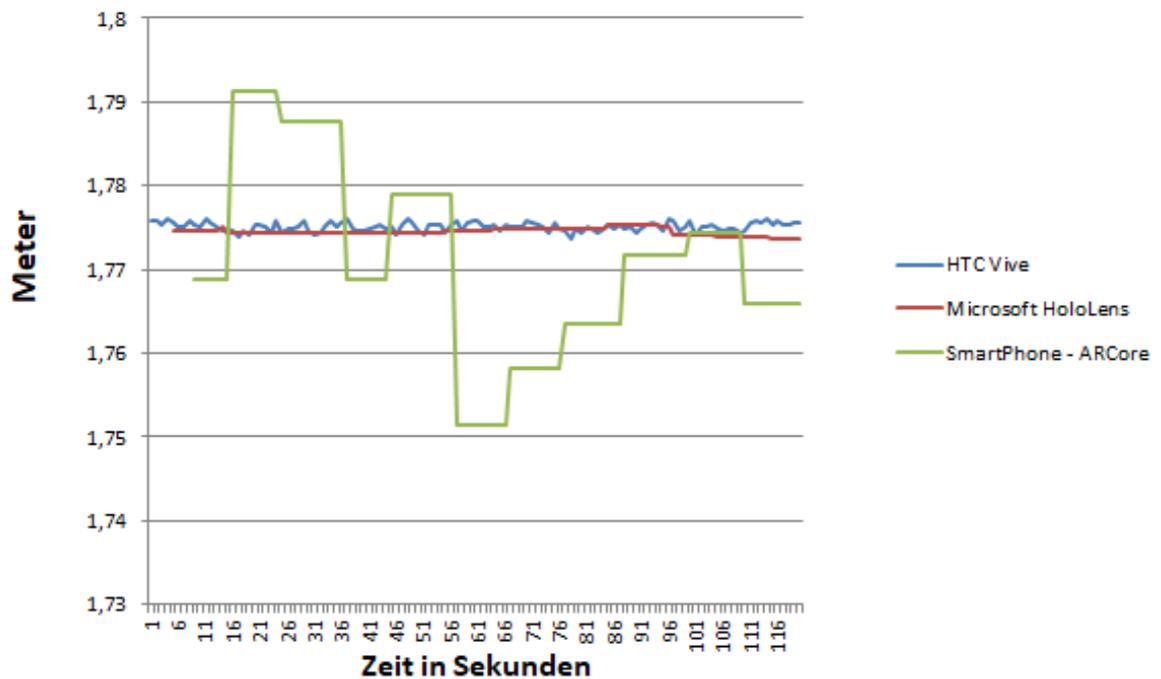


Abbildung 9.3: Ergebnisse der Messungen aller Devices

Der obere Graph von Abbildung 9.3 zeigt die gemessene Position im Raum. Im Vergleich zur Abbildung 9.2 lässt sich erkennen, dass die Abweichung des Smartphones sich deutlich von den anderen unterscheidet. Die Diskrepanz ist nicht mehr in Millimeter, sondern in

9 Vergleich der verwendeten Devices

Geräte	Microsoft HoloLens	HTC Vive	Smartphone
Minimum der Positionshöhe	1.7694	1.7652	1.7509
Maximum der Positionshöhe	1.7713	1.7695	1.8218
Mittelwert der Positionshöhe	1,7709	1,7673	1,7865
Standardabweichung der Positionshöhe	0,0003	0,0013	0,0174
Minimum der Größe	1,7736	1,7726	1,7513
Maximum der Größe	1,7760	1,7754	1,7913
Mittelwert der Größe	1,7751	1,7745	1,7712
Standardabweichung der Größe	0,0005	0,0004	0,0114

Tabelle 9.1: Detailinformationen des Gerätevergleichs

Zentimeter ersichtlich. Bei der Messung der Position im Raum weicht das Smartphone um rund sechs Zentimeter ab. Diese Genauigkeit ist für die Ausführung und die Erkennung der Übungen noch ausreichend, allerdings können durch diese Abweichung in seltenen Fällen Fehlerkennungen stattfinden.

Ebenso zeigt sich diese Abweichung bei der Messung des Bodens mit dem Smartphone. Über die Verwendung von ARCore wird der Boden permanent gemessen. Wie bei der Microsoft HoloLens dauert es zu Beginn, bis der Boden erkannt wird. Deswegen beginnt der Graph erst bei Sekunde acht. Der Boden wird in regelmäßigen Abständen neu getrackt. Trotz der fixierten Position auf dem Kopf der Puppe wird der Boden nicht exakt erkannt. Die maximale Abweichung beträgt ungefähr vier Zentimeter.

Im Vergleich zu den hochperformanten Geräten ist die Genauigkeit des Smartphones definitiv geringer. Dies wird in Tabelle 9.1 genauer dargestellt.

Allerdings handelt es sich bei der Abweichung der Messungen des Smartphones um eine, für diese Applikation akzeptable, Distanz. Für die Übungserkennung hat diese geringe Abweichung kaum merkbare Auswirkungen.

9.3 Vergleich des Loggings

Im Rahmen der Applikationen werden folgende Aktivitäten mitgeloggt:

- Ausführung von Übungen
- Start und Stop der Hindernisse
- Starten der Applikation
- Beenden der Applikation
- Kollision mit Hindernissen
- Erscheinen von Hindernissen

Bei jeder Ausführung der Übungen oder einer Kollision wird die Kopfposition und Kopffrotation der Nutzer und Nutzerinnen mitgeloggt. Alle diese Logs werden als .json Datei gespeichert. Dieses Format wurde aufgrund der Maschinenlesbarkeit gewählt.

Die Logs sind in mehrere Bereiche gegliedert. Zuerst wird das verwendete Device beschrieben, bevor Information über den Startzeitpunkt ausgegeben wird. Anschließend folgt der Bereich, welcher die Zielvorgaben für das Training beschreibt. Abbildung 9.4 zeigt diesen allgemeinen Bereich.

```

"Device": [
  {
    "AR_Walker_Version": "1.0.0.0 ",
    "HoloLens_Name": "HoloLens_AJBHJ",
    "HoloLens_Systemversion": "10.0.14393
    .2248"
  }
],
"TrainingInformation": [
  {
    "Trainee": "ID_1",
    "Training_Start": "2018.06.08
    10:19:14",
    "Size": "1.749533"
  }
],
"Goal": [
  {
    "Squats": 10,
    "Sidesteps": 10,
    "StepUps/StepDowns": 10,
    "ToesTouched": 5
  }
],

```

Abbildung 9.4: Allgemeines Logging

Neben diesem allgemeinen Bereich wird der gesamte Übungsverlauf mitgeloggt. Dieser wird für jedes Device in den folgenden Unterkapiteln dargestellt.

Abschließend wird noch eine Zusammenfassung der Übungsergebnisse geloggt, welche in Abbildung 9.5 ersichtlich ist.

```
"FinalResults": [
  {
    "Squats": 10,
    "Sidesteps": 11,
    "StepUps": 10,
    "ToesTouched": 5,
    "Trainingduration": "03 minutes 03 seconds",
    "AverageDifficulties": [
      {
        "AverageDifficultySquats": "0.76",
        "AverageDifficultySidesteps": "0.33",
        "AverageDifficultyToesTouched": "0.60"
      }
    ]
  }
]
```

Abbildung 9.5: Zusammenfassung der Ergebnisse

9.3.1 Logging der Microsoft HoloLens Applikation

Abbildung 9.6 zeigt einen Auszug aus dem Log der Microsoft HoloLens Applikation. Dieser Auszug zeigt, dass die Hindernisse gestartet wurden. Beim Starten der Hindernisse wird ein neues Set erzeugt, welche die gesamten Informationen sammelt, bis die Hindernisse wieder gestoppt werden. Jedes der Objekte enthält zusätzliche Attribute, welche in der Datei ersichtlich sind.

In der dargestellten Abbildung 9.6 ist erkennbar, dass ein Hindernis entstand und diesem mittels zwei Seitschritten ausgewichen wurde, da keine Kollision erkannt wurde. Das folgende Hindernis war das obere Hindernis, da es den Namen TopBar trägt. Die Höhe dieses Hindernisses befindet sich auf 0, was im Rahmen der Microsoft HoloLens Applikation der Höhe des Kopfes der Benutzer und Benutzerinnen beim Start der Anwendung entspricht. Das heißt, die Hindernisse von oben erscheinen auf der Höhe der zuerst gemessenen Kopfposition.

Für die ausgeführte Übung, in der Microsoft HoloLens Applikation, werden folgende Daten mitgeloggt:

- Anzahl der Übungen dieser Art
- Normalisierte Distanz zum Boden
- Distanz zum Boden
- Eingestellte Schwierigkeit der Übung
- Richtung der Übung
- Kopffrotation
- Kopfposition

- Referenzposition
- Uhrzeit der Ausführung

```

"Device": [
  {
    "AR_Walker_Version": "1.0.0.0 ",
    "HoloLens_Name": "HoloLens_AJBHJ",
    "HoloLens_Systemversion": "10.0.14393.2248"
  }
],
"TrainingInformation": [1],
"Goal": [1],
"Set": [
  {
    "Information": [1],
    "Obstacle": [1],
    "Sidestep": [1],
    "Sidestep": [1],
    "Obstacle": [
      {
        "SpawnTime": "10:19:44",
        "Obstacle": "TopBar",
        "Height": "0"
      }
    ]
  },
  {
    "Squat": [
      {
        "SquatCount": "1",
        "NormalizedDistanceToFloor": "0.7578694",
        "ActualDistanceToFloor": "1.326394",
        "Difficulty": 0.60,
        "Direction": "Down",
        "HeadRotation": "(0.1, 9.2, 359.7)",
        "HeadPosition": "(-0.7799, -0.4281, 0.9636)",
        "ReferencePosition": "(-0.7684, 0.0786, 0.7285)",
        "Time": "10:19:47"
      }
    ]
  },
  {
    "Squat": [1],

```

Abbildung 9.6: Logging der Microsoft HoloLens Applikation

9.3.2 Logging der HTC Vive Applikation

Im Rahmen der HTC Vive Applikation wurde das bereits bestehende Logging angepasst, um die zusätzlichen Erkennungsmöglichkeiten zu integrieren. Abbildung 9.7 zeigt einen Auszug aus dem Logging dieser Anwendung.

9 Vergleich der verwendeten Devices

```
"Device": ⊖ [
  ⊖ {
    "AR_Walker_Version": "1.0.0.0 ",
    "Device_Name": "HTC_Vive"
  }
],
"TrainingInformation": ⊖ [
  ⊕ {...}
],
"Set": ⊖ [
  ⊖ {
    "Information": ⊕ [1],
    "Obstacle": ⊕ [1],
    "Sidestep": ⊕ [1],
    "Obstacle": ⊕ [1],
    "Sidestep": ⊕ [1],
    "Sidestep": ⊕ [1],
    "Sidestep": ⊕ [1],
    "Obstacle": ⊖ [
      ⊖ {
        "SpawnTime": "14:47:07",
        "Obstacle": "MidBar",
        "Position": "(5.8, 0.0, 0.4)"
      }
    ]
  },
  "Sidestep": ⊕ [1],
  "StepUp": ⊖ [
    ⊖ {
      "StepUpCount": "1",
      "Direction": "Up",
      "NormalizedDistanceToFloor": "1.071065",
      "ActualDistanceToFloor": "1.839511",
      "Difficulty": 0.33,
      "HeadRotation": "(6.2, 89.3, 359.0)",
      "HeadPosition": "(0.3869, 1.8127, 0.1603)",
      "ReferencePosition": "(0.3670, 1.5915, 0.1444)",
      "FootLeftPosition": "(0.3462, 0.2883, 0.2461)",
      "FootRightPosition": "(0.0245, 0.3519, 0.0690)",
      "KneeLeftPosition": "(0.4023, 0.7552, 0.2598)",
      "KneeRightPosition": "(0.3534, 0.6965, 0.0487)",
      "Time": "14:47:10"
    }
  ]
]
```

Abbildung 9.7: Logging der HTC Vive Applikation

Diese Abbildung zeigt ein gestartetes Set, bei dem mehreren Hindernissen ausgewichen wurde. Die Details zu einem Hindernis sind in der Abbildung ersichtlich. In der HTC Vive Applikation ist die Höhe 0 nicht die zuerst gemessene Kopfposition, sondern der Boden. Das Hindernis, in diesem Falle das Hindernis in der Mitte, befindet sich also auf dem Boden. Neben den Hindernisinformationen zeigt die Abbildung 9.7 die zusätzlich mitgeloggten Informationen in der HTC Vive Applikation. Diese sind die:

- Position des linken Fußes
- Position des rechten Fußes
- Position des linken Knies
- Position des rechten Knies

Durch dieses genaue Logging kann nachvollzogen werden, wo sich die Person während der Ausführung der Übung befand. Es kann exakt definiert werden, wo sich die Füße, Knie und der Kopf der übenden Person im Raum befinden.

9.3.3 Logging der Smartphone Applikation

Das Logging der Smartphone Applikation entspricht dem Logging der Microsoft HoloLens Applikation. Ein Ausschnitt dieses Loggings wird in Abbildung 9.8 dargestellt.

Der Ausschnitt des Loggings zeigt den Trainingsbeginn sowie die gemessene Größe der übenden Person. Außerdem werden Informationen über das untere Hindernis eingeblendet. Dieses befindet sich auf einer Höhe von -1.38 Meter. Diese Distanz wird von der Größe des Nutzers oder der Nutzerin subtrahiert, um den Abstand des Hindernisses zum Boden zu berechnen. In diesem konkreten Fall ist die übende Person 1.75 Meter groß, das bedeutet, der Mittelpunkt des Hindernisses befindet auf 37 Zentimetern.

Die Informationen welche mitgeloggt werden, entsprechen denen der Microsoft HoloLens Applikation. Die Abweichungen, welche bereits zuvor in diesem Kapitel beschrieben wurden, spielen auch im Rahmen des Loggings eine Rolle. Die Messung der Größe ist essenziell, um eine erfolgreiche Übungserkennung zu gewährleisten. Diese Abweichungen sind auch in den Logdateien merkbar.

```

"Device": [
  {
    "AR_Walker_Version": "1.0.0.0 ",
    "Smartphone_Name": "SamsungGalaxyS8Plus",
    "Android_Version": "8.0.0"
  }
],
"TrainingInformation": [
  {
    "Trainee": "ID_1",
    "Training_Start": "2018.06.11 16:25:15",
    "Size": "1.75"
  }
],
"Set": [
  {
    "Information": [1],
    "Obstacle": [
      {
        "SpawnTime": "16:25:39",
        "Obstacle": "BottomBar",
        "Height": "-1.38"
      }
    ]
  },
  "Sidestep": [1],
  "Sidestep": [1],
  "StepUp": [
    {
      "StepUpCount": "1",
      "NormalizedDistanceToFloor": "1.086183",
      "ActualDistanceToFloor": "1.90082",
      "Direction": "Up",
      "HeadRotation": "(16.9, 355.7, 350.9)",
      "HeadPosition": "(0.1484, 0.3445, 1.5252)",
      "ReferencePosition": "(0.1032, 0.0000, 1.3152)",
      "Time": "16:25:47"
    }
  ]
],

```

Abbildung 9.8: Logging der Smartphone Applikation

10 Usability Tests

In diesem Kapitel werden die durchgeführten Usability Tests beschrieben. Im Rahmen dieser Arbeit wurden zehn Personen befragt und führten die Tests durch. Dieses Kapitel beschreibt die Testplanung, -durchführung und die erhobenen Ergebnisse.

10.1 Testplanung

Um aussagekräftige Informationen über die Applikationen erheben zu können, wurden im Rahmen der Usability Tests zwei Fragebögen erstellt. Der erste dieser beiden Fragebögen wird vor der Durchführung der Tests von den Probanden und Probandinnen beantwortet, um Vorkenntnisse, Erfahrungen und allgemeine Informationen abzudecken. Der zweite der Fragebögen wird nach der Durchführung der Tests beantwortet und beinhaltet spezifische Fragen zu den jeweiligen Applikationen.

10.1.1 Pre-Test Fragebogen

Der Fragebogen, welcher vor den Applikationstests ausgefüllt wird, hat das Ziel, die allgemeinen Informationen, Erfahrungen und Vorkenntnisse der Nutzer und Nutzerinnen zu identifizieren. Folgende Informationen werden mit diesem Fragebogen erhoben:

- Alter der Person
- Geschlecht der Person
- Erfahrung mit Physiotherapie
- Erfahrung mit Heimphysiotherapie
- Erfahrung mit AR und/oder VR
- Besitz eines Smartphones
- Art des Smartphones

Diese Informationen dienen dem Zweck, die Auswertung der Usability Tests zu klassifizieren. Vor allem der Faktor der Erfahrung mit AR und/oder VR spielt im Rahmen der Tests eine tragende Rolle.

Neben diesen Informationen wurde ein zweiter Fragebogen erstellt, welcher nach den spezifischen Tests der Applikationen ausgefüllt wurde.

10.1.2 Post-Test Fragebogen

Um Informationen über die Durchführung der Tests zu erheben, wurde ein Fragebogen erstellt, welcher nach Beendigung der Tests ausgefüllt wird. In diesem werden die einzelnen Applikationen nach folgenden Kriterien bewertet:

- Intuitive Verwendbarkeit
- Erkennung der Übungen
- Interaktion mit der Applikation

- Wohlbefinden während dem Training
- Qualität der Audioausgabe
- Feedback während des Trainings

Die einzelnen Teilbereiche konnten mit Sehr Gut, Gut, Schlecht oder Sehr Schlecht beantwortet werden. Zusätzlich zu diesen Kriterien werden für jede Applikation in einem Antwortfeld die Vor- und Nachteile erhoben. Für die Microsoft HoloLens Applikation wurde die automatische Anpassung des Interfaces ebenfalls als Kriterium hinzugezogen.

Abschließend werden die Probanden und Probandinnen noch aufgefordert, die Applikation zu wählen, welche ihnen die beste Unterstützung für das Training geboten hat. Außerdem müssen sie begründen, warum sie diese Auswahl getroffen haben.

10.2 Testdurchführung

Um die Tests effizient durchführen zu können, wurde das Setup der HTC Vive in einem definierten Raum aufgebaut und fixiert. Alle Usability Tests wurden in diesem Raum, mit dem gleichen Setup, arrangiert. Der Stepper wurde an einer markierten Position abgestellt und für alle Probanden und Probandinnen an dieser belassen.

Für die Durchführung der Tests gab es ein konkretes Konzept. Folgende Schritte wurden für alle Tester und Testerinnen ausgeführt:

- Erklärung der Problemstellung (5 Minuten)
- Ausfüllen des Pre-Test Fragebogens (5 Minuten)
- Erklärung der Durchführung der Tests (5 Minuten)
- Usability Test der Microsoft HoloLens Applikation (10 Minuten)
- Usability Test der HTC Vive Applikation (10 Minuten)
- Usability Test der Smartphone Applikation (10 Minuten)
- Ausfüllen des Post-Test Fragebogens (10 Minuten)
- Abschließendes Gespräch (5 Minuten)

Die angegebene Zeit wurde als Rahmen für die Usability Tests gewählt. Diese wurde nicht immer exakt eingehalten, diente allerdings als Orientierung.

Im Rahmen der Erklärung der Problemstellung wurde den Testern und Testerinnen die Problematik der Heimphysiotherapie und die Ziele der Masterarbeit näher gebracht. Beim Ausfüllen des Pre-Test Fragebogens wurden die allgemeinen Informationen, Erfahrungen und Vorkenntnisse der Probanden und Probandinnen erhoben.

Die Erklärung der Durchführung der Tests enthielt folgende Schritte:

- Erklärung der Testziele
- Erklärung der Übungen
- Erklärung der Hindernisse

Die Nutzer und Nutzerinnen wurden informiert, dass jeweils fünf Kniebeugen, zehn Seit-schritte, fünf Step-Ups und fünf Zehenberührungen durchzuführen sind. Dafür muss die Applikation gestartet und die Hindernisse aktiviert werden. Anschließend müssen die Übungen durchgeführt werden, während den Hindernissen ausgewichen wird.

Zur Erklärung der Übungen wurde den Probanden und Probandinnen eine korrekte Ausführung jeder Übung präsentiert. Die Hindernisse und die Art der Gestaltung (Tiere) wurden erläutert.

Für jede der Applikationen musste die Zielanzahl der Übungen erreicht werden, während den Hindernissen ausgewichen wurde. Nach der erfolgreichen Absolvierung der drei Applikationstests wurde der Post-Test Fragebogen ausgefüllt. Dafür wurde mit den Probanden und Probandinnen alle Punkte besprochen. Die einzelnen Faktoren wurden diesen vor der Beantwortung erklärt.

Im abschließenden Gespräch wurde auf Fragen der Tester und Testerinnen eingegangen. Zusätzlich wurde versucht, weiteres Feedback für die Applikationen zu gewinnen.

Im Rahmen des Tests der HTC Applikation wurde die HTC Vive Pro verwendet. Diese wurde gewählt, da die Auflösung besser ist als auf der älteren Vive. Die Auflösung beträgt 1440 x 1600 Pixel pro Auge. Außerdem hat die HTC Vive Pro integrierte Kopfhörer, was das Setup erleichtert [22]. Dieser Wechsel war möglich, da das Tracking und die Erkennung nicht beeinflusst wurden. Diese Änderung bietet nur den Nutzern und Nutzerinnen optische Vorteile.

10.3 Testauswertung

Anschließend an die Usability Tests wurden die Fragebögen ausgewertet. In dieser Arbeit werden die wichtigsten Ergebnisse der Tests dargestellt.

Es wurden zehn Personen befragt, sieben davon waren zwischen 20 und 30 Jahren alt, die anderen waren zwischen 31 und 40. Ebenso sieben davon waren männliche Teilnehmer, drei waren weibliche Teilnehmerinnen. Acht dieser Teilnehmer und Teilnehmerinnen hatten keine Erfahrung mit physiotherapeutischem Training. Abbildung 10.1 zeigt dieses Ergebnis.

Hatten Sie schon Erfahrung mit physiotherapeutischen Training?

10 Antworten

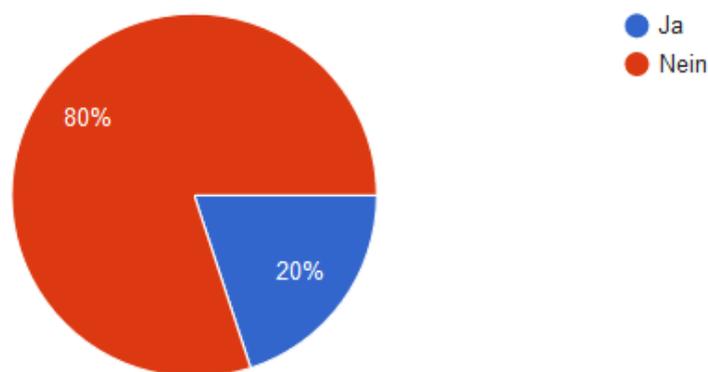


Abbildung 10.1: Usability Test - Ergebnis physiotherapeutische Erfahrung

Die anderen beiden Teilnehmer und Teilnehmerinnen gaben an, bereits mit physiotherapeutischem Training Erfahrung gemacht zu haben, ebenso mit Heimphysiotherapie. Außerdem sagten beide der Personen, dass sie die Heimtherapie Übungen sehr nachlässig ausgeführt haben. Diese wären täglich zu absolvieren gewesen, wurden von diesen beiden Probanden und Probandinnen allerdings nie ausgeführt. Der Grund hierfür war bei beiden die fehlende Motivation, sie sahen keinen Mehrwert in der Ausführung der Übungen.

Von den teilnehmenden Personen haben bereits sechs Erfahrung mit AR oder VR gemacht, die anderen vier noch nicht. Ein Smartphone benutzten alle Teilnehmer und Teilnehmerinnen.

Die durchschnittlichen Kosten der Smartphones der Nutzer und Nutzerinnen sind ungefähr 320€.

10.3.1 Microsoft HoloLens Applikation

Für die Microsoft HoloLens Applikation wurde die Intuitive Verwendbarkeit vier Mal mit Sehr Gut, vier Mal mit Gut und ein Mal mit Schlecht bewertet. Die Erkennung der Übungen wurde mit acht Mal Sehr Gut und zwei Mal Gut bewertet. Die Interaktion mit der Applikation wurde sechs Mal mit Sehr Gut und vier Mal mit Gut bewertet. Beim Wohlbefinden während der Übung gaben die Hälfte der Teilnehmer und Teilnehmerinnen Sehr Gut und die andere Hälfte Gut an. Der Tragekomfort wurde von sechs Probanden und Probandinnen als Sehr Gut bewertet. Die anderen Stimmen verteilten sich, jeweils zwei davon, auf Gut und Schlecht. Der Faktor der Verständlichkeit der Audioausgabe wurde mit acht Mal Sehr Gut und zwei Mal Gut bewertet. Die Qualität der Audioausgabe wurde von sieben Teilnehmern und Teilnehmerinnen als Sehr gut bewertet, die anderen drei vergaben ein Gut. Das Feedback während des Trainings wurde sieben Mal mit Sehr Gut, zwei Mal mit Gut und ein Mal mit Schlecht bewertet.

Abbildung 10.2 zeigt eine Zusammenfassung dieser Ergebnisse. In der Abbildung sind die Ergebnisse in der Reihenfolge, welche in Unterkapitel 10.1.2, Post-Test Fragebogen, beschrieben wurde.

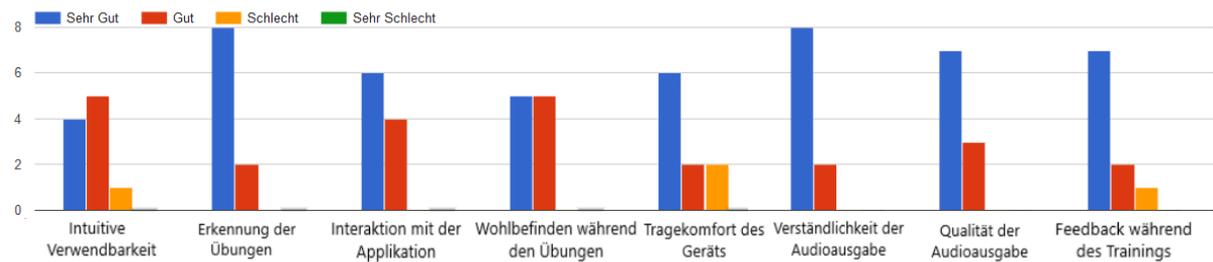


Abbildung 10.2: Usability Test - Ergebnis der HoloLens Applikation

Zusätzlich wurde bei der Microsoft HoloLens Applikation die automatische Anpassung der Menüposition hinterfragt. Die Auswertung der Usability Tests, bezogen auf dieses Kriterium, ist in Abbildung 10.3 dargestellt.

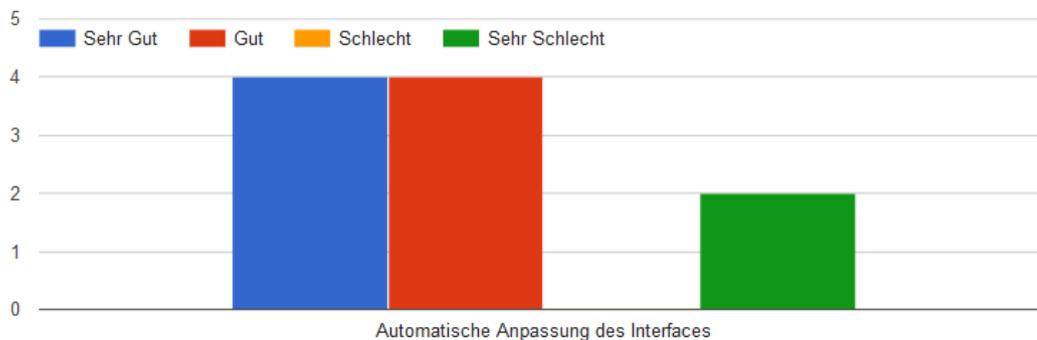


Abbildung 10.3: Usability Test - Ergebnis der automatischen Menüanpassung

Dieses Kriterium wurde von vier Personen als Sehr Gut, von vier Personen als Gut und von zwei als Sehr Schlecht bewertet.

Als Vorteile der Microsoft HoloLens wurde die Sichtbarkeit des Menüs genannt, welches sich immer im Sichtfeld befand. Ein weiterer Vorteil welcher von vier Probanden und Probandinnen genannt wurde, war die Sichtbarkeit der realen Umgebung. Der einzige erhobene Nachteil war das geringe Sichtfeld, wodurch Hindernisse teilweise nicht im Blickfeld lagen.

10.3.2 HTC Vive Applikation

Für die HTC Vive Applikation wurde die Intuitive Verwendbarkeit fünf Mal mit Sehr Gut und fünf Mal mit Gut bewertet. Die Erkennung der Übungen wurde mit neun Mal Sehr Gut und ein Mal Gut bewertet. Die Interaktion mit der Applikation wurde neun Mal mit Sehr Gut und ein Mal mit Gut bewertet. Beim Wohlbefinden während der Übung vergaben vier der Teilnehmer und Teilnehmerinnen Sehr Gut, fünf Gut und eine Person Sehr schlecht. Der Tragekomfort wurde von fünf Probanden und Probandinnen als Sehr Gut bewertet. Die anderen Stimmen verteilten sich, vier auf Gut und eine auf Schlecht. Der Faktor der Verständlichkeit der Audioausgabe wurde mit zehn Mal Sehr Gut benotet. Die Qualität der Audioausgabe wurde von neun Teilnehmern und Teilnehmerinnen als Sehr gut bewertet, die andere Person vergab ein Gut. Das Feedback während des Trainings wurde acht Mal mit Sehr Gut, ein Mal mit Gut und ein Mal mit Schlecht bewertet.

Abbildung 10.4 zeigt eine Zusammenfassung dieser Ergebnisse. In der Abbildung sind die Ergebnisse in der Reihenfolge, welche in Unterkapitel 10.1.2, Post-Test Fragebogen, beschrieben wurde.

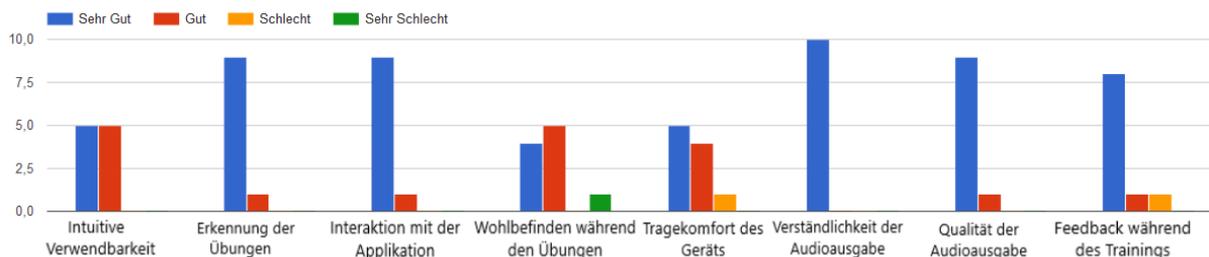


Abbildung 10.4: Usability Test - Ergebnis der HTC Applikation

Vorteile der HTC Vive Applikationen, welche im Rahmen der Usability Tests erhoben wurden, waren die sehr intuitive Verwendung der Applikation sowie die Darstellung des virtuellen Raumes. Die Nachteile waren das umfangreiche Einrichten vor Beginn der Übungen. Die Teilnehmer und Teilnehmerinnen fiel es negativ auf, dass sie extra Sensoren anlegen mussten. Außerdem gaben sie an, sich in der realen Umgebung sicherer zu fühlen. Vor allem das Steigen auf den virtuellen Stepper wurde von zwei Testern und Testerinnen bemängelt.

10.3.3 Smartphone Applikation

Für die Smartphone Applikation wurde die Intuitive Verwendbarkeit vier Mal mit Sehr Gut und sechs Mal mit Gut bewertet. Die Erkennung der Übungen wurde mit 3 Mal Sehr Gut und sieben Mal Gut bewertet. Die Interaktion mit der Applikation wurde vier Mal mit Sehr Gut und sechs Mal mit Gut bewertet. Beim Wohlbefinden während der Übung vergab eine Person Sehr Gut, vier Gut, vier Schlecht und eine Person Sehr schlecht. Der Tragekomfort wurde von zwei Probanden und Probandinnen als Sehr Gut bewertet. Die anderen Stimmen verteilten sich, vier auf Gut, vier auf Schlecht und eine auf Sehr Schlecht. Der Faktor der Verständlichkeit der Audioausgabe wurde mit vier Mal Sehr Gut, fünf Mal Gut und ein Mal schlecht benotet.

10 Usability Tests

Die Qualität der Audioausgabe wurde von vier Teilnehmern und Teilnehmerinnen als Sehr gut bewertet, die anderen Personen vergaben vier Mal Gut und zwei Mal Schlecht. Das Feedback während des Trainings wurde vier Mal mit Sehr Gut, vier Mal mit Gut und zwei Mal mit Schlecht bewertet.

Abbildung 10.5 zeigt eine Zusammenfassung dieser Ergebnisse. In der Abbildung sind die Ergebnisse in der Reihenfolge, welche in Unterkapitel 10.1.2, Post-Test Fragebogen, beschrieben wurde.

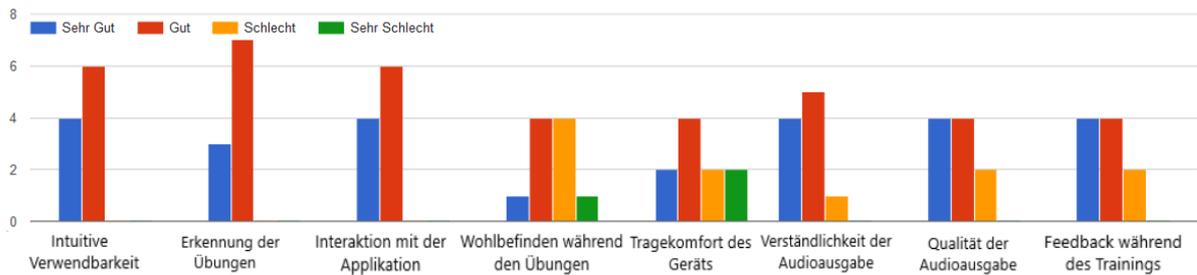
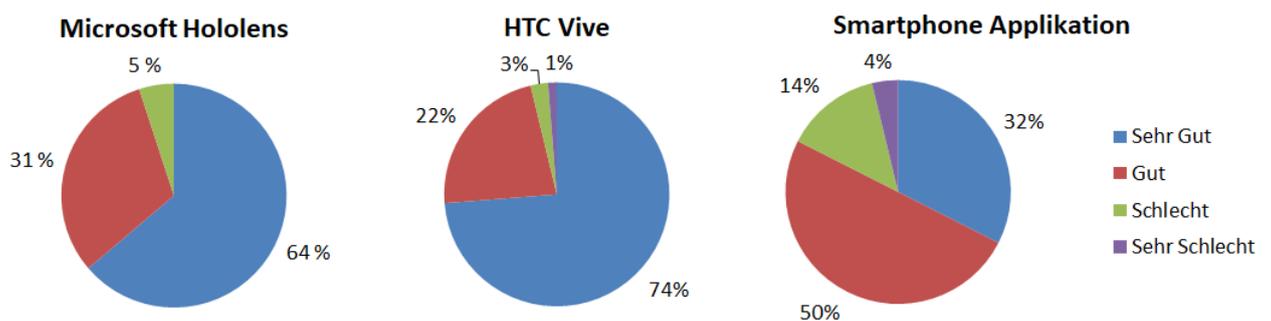


Abbildung 10.5: Usability Test - Ergebnis der Smartphone Applikation

Für die Smartphone Applikation wurden in den Gesprächen keine Vorteile erhoben. Lediglich der Nachteil der Befestigung am Kopf wurde explizit erwähnt. Die Befestigung mit dem HMD sei nicht ausreichend für diese Übungen, da es immer verrutscht.

10.3.4 Gesamtauswertung

Im Rahmen der Usability Tests wurden die drei Applikationen in acht Kategorien bewertet. Das heißt, für jede der Applikationen wurden 80 Stimmen vergeben. Die Aufteilung dieser Stimmen auf die Bewertungsskala wird in Abbildung 10.6 dargestellt.



Gesamt: 80 Stimmen pro Applikation.

Abbildung 10.6: Usability Test - Gesamtergebnis

Die Abbildung zeigt den Anteil der prozentuellen Bewertungen der Devices anhand der Bewertungsskala. Die Microsoft HoloLens und die HTC Vive konnten beide über 60 Prozent aller Bewertungen mit Sehr Gut abschließen. Dieser Anteil ist bei der Smartphone Applikation hingegen unter 35 Prozent. Die Microsoft HoloLens Applikation wurde als einzige nie mit Sehr Schlecht bewertet. Die einzige Sehr Schlechte Bewertung bei der HTC Vive Applikation wurde für das Wohlbefinden im Raum vergeben. Beim Smartphone verteilten sich die drei Sehr Schlecht auf das Wohlbefinden im Raum und den Tragekomfort des Geräts auf. Mit nur vier Prozent negativen Bewertungen schnitt die HTC Vive deutlich besser ab als die Smartphone

Applikation, kann sich aber hauptsächlich durch die Sehr Guts von der Microsoft HoloLens Applikation distanzieren.

Weiters wurden die Probanden und Probandinnen gebeten, die Applikation auszuwählen, welche ihnen die beste Unterstützung für das Training geboten hat. Die HTC Vive Applikation wurde von sieben Teilnehmern und Teilnehmerinnen ausgewählt, die Microsoft HoloLens Applikation von drei Nutzern und Nutzerinnen. Abbildung 10.7 zeigt diese Auswertung.

Welche Applikation hat Ihnen die beste Unterstützung für das Training geboten?

10 Antworten

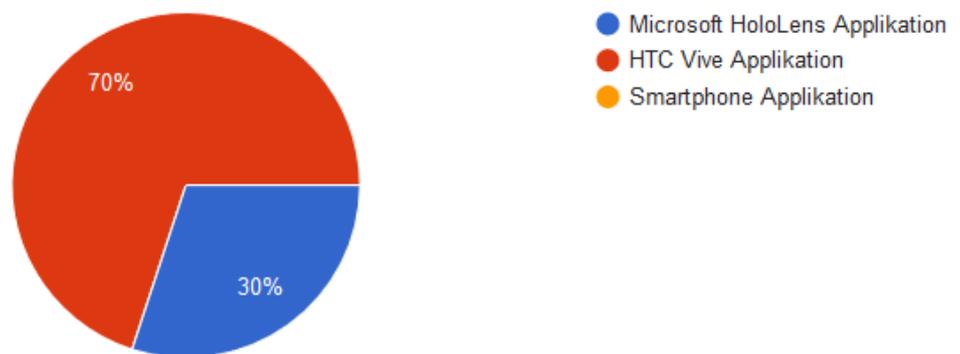


Abbildung 10.7: Usability Test - Beste Applikation

Die Gründe für die Auswahl der HTC Vive waren folgende:

- Höchste Immersion
- Genaueste Erkennung
- Bestes Erlebnis
- Bestes Raumgefühl

Die drei Personen, welche sich für die Microsoft HoloLens Applikation entschieden, gaben an, die hohe Sicherheit während dem Training zu schätzen. Außerdem lobten diese die Darstellung im realen Raum, sowie die Bedienung des Geräts.

Personen, welche keine Erfahrung mit AR / VR hatten, tendierten dazu die Microsoft HoloLens Applikation zu wählen. Der Grund dafür war, das fehlende Vertrauen in die Technik. Das Steigen auf den virtuellen Stepper kostete diese Personen eine enorme Überwindung und beeinflusste das Training negativ, da die Übung nicht flüssig ausgeführt werden konnte. Diese Schwierigkeit trat auf, obwohl die virtuelle und die reale Position des Steppers übereinstimmten.

Die Usability Tests wurden von allen Teilnehmern und Teilnehmerinnen bis zum Ende durchgeführt. Es gab keine vorzeitigen Abbrüche.

Neben den Ergebnissen der Fragebögen wurde auch das Logging der Usability Tests ausgewertet. Im Rahmen der Tests wurden pro Applikation fünf Kniebeugen, zehn Seitschritte, fünf Step-Ups und fünf Zehenberührungen ausgeführt. Das heißt, 75 Übungen pro Person in allen drei Übungen. Gesamt beläuft sich die Anzahl der ausgeführten Übungen aller Tests auf etwas mehr als diese 750, da manche Teilnehmer und Teilnehmerinnen zusätzliche Übungen

ausgeführt haben. Im Durchschnitt dauerte der Test von dem Start der Microsoft HoloLens Applikation bis zum Ende des Smartphones Tests rund 27 Minuten.

Durch das Logging wurden vor allem die Kollisionen mit den Hindernissen genaustens analysiert. In der Microsoft HoloLens Applikation gab es 26 Kollisionen, während es in der HTC Vive Anwendung nur 10 Kollisionen gab. Zwischen den beiden lag die Smartphone Applikation, mit 22 Kollisionen. Abbildung 10.8 zeigt diese Verteilung.

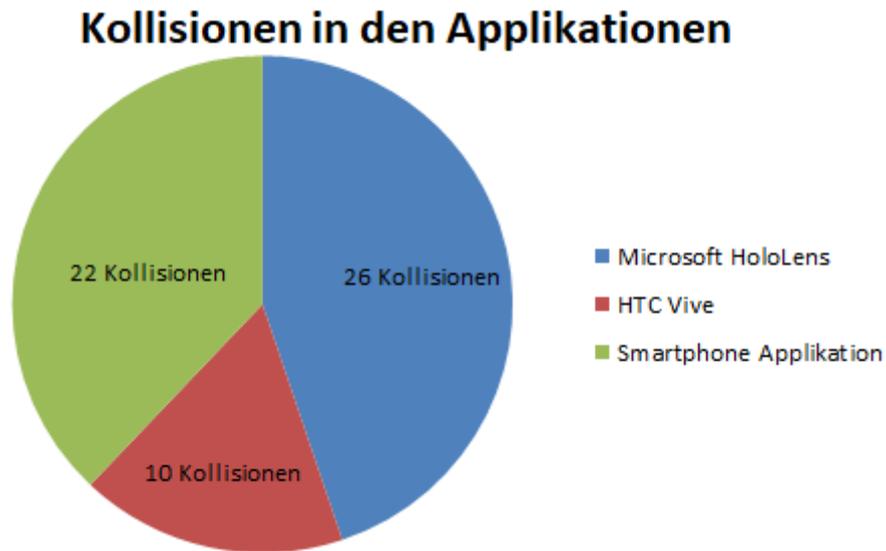


Abbildung 10.8: Usability Test - Kollisionen in den Anwendungen

Neben diesen Ergebnissen wurde auch die Anzahl der Kollisionen mit den jeweiligen Hindernissen gemessen. Am häufigsten wurde mit den oberen Hindernis kollidiert. Mit dem rechten Hindernis kollidierte keine der teilnehmenden Personen. Abbildung 10.9 zeigt diese Darstellung.

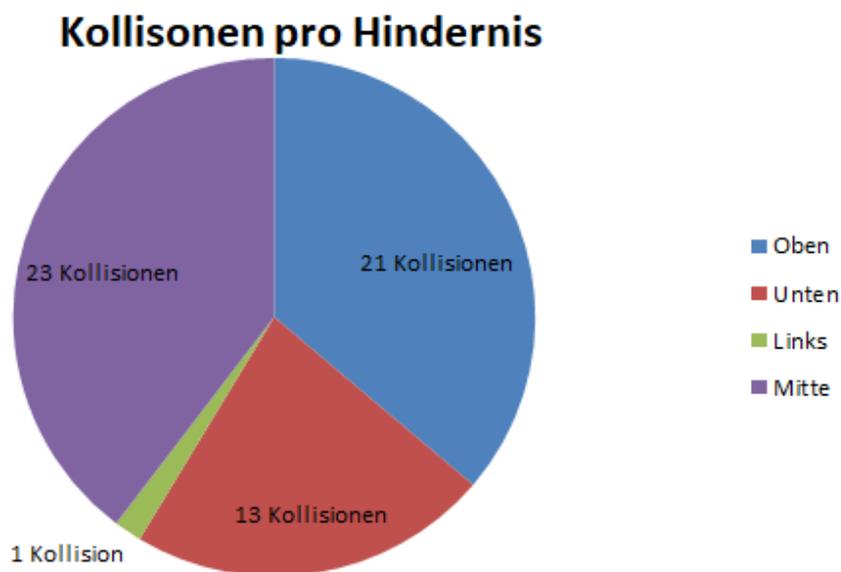


Abbildung 10.9: Usability Test - Kollisionen pro Hindernis

Die wichtigsten Daten des Loggings werden in Tabelle 10.1 dargestellt.

Geräte	Microsoft HoloLens	HTC Vive	Smartphone
Kollisionen	26	10	22
Kollisionen oben	11	3	7
Kollisionen unten	5	2	6
Kollisionen links	0	1	0
Kollisionen rechts	0	0	0
Kollisionen mitte	10	4	9
Maximale Kollisionen	5	3	6
Minimale Kollisionen	0	0	0
Durchschnittliche Kollisionen	2.6	1.0	2.2

Tabelle 10.1: Detailinformationen der Loggingergebnisse

Tabelle 10.1 beschreibt die Detailinformationen der Loggingergebnisse. In der Tabelle sind die Informationen über die Kollisionen ersichtlich. Die geringste Anzahl an Kollisionen hat die HTC Vive Applikation, wobei die höchste Anzahl an Hinderniskontakten die Microsoft HoloLens Anwendung vorzuweisen hat.

10.4 Diskussion der Ergebnisse

In diesem Unterkapitel werden die Ergebnisse der Usability Tests analysiert und diskutiert. Im Rahmen der Befragung gaben zwei Personen an, bereits Erfahrung mit Physiotherapie und Heimphysiotherapie gemacht zu haben. Diese erklärten, dass sie die Übungen aufgrund fehlender Motivation zu Hause nicht ausgeführt haben. Diese beiden Aussagen unterstreichen die Problemstellung der Arbeit. Obwohl diese Personen jünger sind als die Hauptzielgruppe, zeigen sich bei diesen bereits Probleme der Heimphysiotherapie.

Dieses Ergebnis zeigt deutlich, dass es für die Heimphysiotherapie notwendig ist, eine motivierende Hilfestellung zur Verfügung zu stellen. Diese beiden Teilnehmer und Teilnehmerinnen gaben ebenso an, dass Feedback während den Übungen ihnen helfen würde, sich für die Ausführung der Physiotherapie zu motivieren.

Der Unterschied zwischen den Personen mit Erfahrung in AR oder VR und den Personen ohne diese Erfahrung zeigte sich bei der Evaluierung der Ergebnisse am deutlichsten bei der Auswahl der besten Applikation. Die Microsoft HoloLens wurde nur von Personen zum besten Gerät gewählt, welche noch keine Erfahrung mit AR oder VR hatten. Der Grund hierfür ist das mangelnde Vertrauen zur Technik, vor allem beim Steigen auf den Stepper.

Es fiel diesen Personen im Rahmen der Tests schwer auf den Stepper zu steigen, ohne sich unwohl zu fühlen. Vor jedem Step-Up tasteten diese erst nach dem Stepper, was dazu führte, dass die Übung nicht flüssig ausgeführt wurde. Zusätzlich gaben diese Personen an, sich ohne die reale Umgebung bei den Bewegungen nicht wohl zu fühlen. Dieser fehlende Bezug zur echten Welt stellt für unerfahrene Personen ein Hindernis dar, welches adressiert werden muss, um die Anwendung effektiv für diese Personen einsetzen zu können.

Die Auswertung der Microsoft HoloLens Applikation war solide. Die Erkennung der ausgeführten Übungen war nur knapp hinter der HTC Vive, obwohl diese mit den zusätzlichen Sensoren deutlich genauer war. Allerdings fiel dieser Unterschied den Teilnehmern und Teilnehmerinnen kaum auf.

Im Gesamtvergleich der beiden High-End Geräte sind sich diese ziemlich ähnlich. Der Hauptunterschied lag bei der Interaktion mit der Applikation. Diese wurde bei der Microsoft HoloLens über die Gestensteuerung, bei der HTC Vive über die Controller umgesetzt. Die Steuerung

über die Controller wurde als intuitiver empfunden, als die Steuerung über die Gesten. Um die Gesten effektiv einsetzen zu können, mussten die Probanden und Probandinnen diese einige mal üben, bevor sie die Interaktion verstanden. Der Umgang mit Controllern war allen Teilnehmern und Teilnehmerinnen sofort geläufig.

Das Wohlbefinden während den Übungen wurde durchschnittlich für die Microsoft HoloLens besser bewertet als für die HTC Vive. Eine Person vergab bei der HTC Vive sogar ein Sehr Schlecht, da sie so große Angst vor den Step-Ups hatte. Die Bewegungen im virtuellen Raum auszuführen beunruhigt die Teilnehmer und Teilnehmerinnen, da sie weder ihre Beine, noch die reale Welt sehen können. Dieser Aspekt wurde bei der Microsoft HoloLens sehr positiv bewertet. Die Einblendung fügt sich in die reale Welt ein und unterstützt das Training.

Die Loggingergebnisse sprechen für die HTC Vive. In dieser Applikation wurde am wenigsten mit den Hindernissen kollidiert. Diese wenigen Kollisionen können auf die erhöhte Immersion zurückgeführt werden. Wie die Probanden und Probandinnen anmerkten, fühlten diese sich in dieser Anwendung am meisten in die Welt hineinversetzt. Sie konnten die Position der Hindernisse am exaktesten zuordnen.

Das Ergebnis der Microsoft HoloLens mit den meisten Kollisionen ist nicht nur auf die Positionierung im Raum zurückzuführen. Neben diesem Faktor spielte auch die Reihenfolge der Anwendungen im Test eine tragende Rolle. Diese Applikation wurde zuerst von den Testpersonen verwendet. Durch diese Vorreiterrolle mussten sich die Anwender und Anwenderinnen erst an die Applikation gewöhnen, während sie bei den anderen beiden Applikationen diese bereit kannten.

Es wurde in den Anwendungen am meisten mit dem mittleren und dem oberen Hindernis kollidiert. Die Kollision mit dem oberen Hindernis war in den meisten Fällen, da die Probanden und Probandinnen die Kniebeuge oder Zehenberührung zu früh verließen. Die Teilnehmer und Teilnehmerinnen warteten nicht auf das Signal, dass das Hindernis vorbei sei und sahen auch nicht hoch, sie standen einfach auf, da sie dachten es sei vorbei. In diesem Fall würde eine verbesserte Form des Interfaces helfen, diese Kollisionen zu vermeiden. Zusätzlich zum auditiven Feedback müsste es eine visuelle Komponente geben, die permanent signalisiert, wie lange eine Übung ausgeführt werden muss.

Die hohe Häufigkeit der Kollision mit dem mittleren Hindernis kann im Rahmen dieser Arbeit nicht erklärt werden. Während die Anzahl der Kollisionen mit dem oberen Hindernis sehr von den Personen abhängig ist, kollidierten in der Microsoft HoloLens Applikation sieben von zehn Teilnehmern und Teilnehmerinnen mit dem mittleren Hindernis. Bei dem oberen Hindernis waren es lediglich fünf.

Die HTC Vive wurde als beste Unterstützung für das Training qualifiziert. Diese Auswahl lässt sich auf folgende Faktoren zurückführen:

- Höchste Immersion
- Genaueste Erkennung
- Bestes Erlebnis
- Bestes Raumgefühl

Diese Arbeit kommt zu dem Schluss, dass die Auswahl der HTC im Rahmen dieser Usability Tests auf der hohen Anzahl an Personen mit Vorerfahrung in AR oder VR basiert. Diese hatten die Probleme des Wohlbefindens und des Vertrauens in die Technik nicht und konnten die Applikation ausreizen. Durch diese Vorerfahrung wurde die genaueste Bewegungserkennung und die Immersion optimal genutzt. Ohne diese Vorerfahrung fühlen sich die Menschen in der realen Welt wohler und können die Trainingsunterstützung besser ausnutzen.

Neben dem Vergleich der beiden High-End Geräte, wurde auch noch die Smartphone Applikation getestet. Diese schnitt im Vergleich zu den anderen Geräten nur mäßig ab. Die intuitive

Verwendung sowie die Erkennung der Bewegungen sind vergleichbar mit den anderen Applikationen. Allerdings fiel die Applikation in folgenden Punkten ab:

- Wohlbefinden während der Übung
- Tragekomfort des Geräts
- Qualität der Audioausgabe

Durch das gespiegelte Bild des Smartphones wurde einigen Teilnehmern und Teilnehmerinnen während der Ausführung der Übungen etwas unwohl. Auch gaben diese an, das Bild sei nicht absolut scharf. Dieses Kriterium fiel im Rahmen dieser Applikation wesentlich schlechter aus als bei den anderen beiden Anwendungen.

Der Tragekomfort des Gerätes lässt sich hauptsächlich durch den Preisunterschied erklären. Während die Microsoft HoloLens und die HTC Vive Pro über ein Verstellrad fixiert werden, musste die Haori Mirror mittels Klettbandern festgemacht werden. Bei zwei der Teilnehmer und Teilnehmerinnen gelang es nicht, diese so zu fixieren, dass sie während den Tests nicht verrutschte. Die Qualität der Audioausgabe wurde zwei Mal mit Schlecht bewertet, allerdings gaben diese Teilnehmer und Teilnehmerinnen an, dass die Ausgabe in Ordnung war, aber die von den anderen Geräten einfach besser. Sowohl die Microsoft HoloLens als auch die HTC Vive besitzen mehrere Audioausgabequellen, während das Smartphone nur einen Lautsprecher hat.

Gesamt lässt sich allerdings behaupten, dass die Smartphone Applikation durchaus für eine solche Unterstützung des Trainings tauglich ist. Im Vergleich zu den High-End Geräten musste mit Abstufungen gerechnet werden, allerdings ist der wichtigste Punkt, die Erkennung der Übungen, in dieser Applikation nur mit Sehr Gut und Gut bewertet worden.

Diese Arbeit kommt zu dem Schluss, dass die Verwendung eines anderen HMDs große Auswirkungen auf das Feedback der Smartphone Applikation hat. Das negative Feedback für das Wohlbefinden und die Befestigung können auf die Verwendung der Haori Mirror zurückgeführt werden.

In dieser Arbeit wurde ebenso die automatische Anpassung des Interfaces erläutert. Diese wurde im Rahmen der Microsoft HoloLens Applikation angewendet und getestet. Die Usability Tests zeigen, dass diese Anpassung hauptsächlich positiv aufgenommen wurde. Vier Personen bewerteten diese mit Sehr Gut, vier mit Gut und zwei mit Sehr Schlecht. Die Teilnehmer und Teilnehmerinnen, welche negative Bewertungen vergaben, hätten sich eine fixierte Menüposition gewünscht. Diese sagten, dass die dauerhafte Menübewegung sie von den Hindernissen ablenkt.

Als positives Feedback zur automatischen Anpassung des Interfaces wurden folgende Punkte genannt:

- Menü immer gut sichtbar
- Keine Verdeckung der Hindernisse
- Angenehme Bewegungsgeschwindigkeit

Aufgrund dieser Analyse kommt diese Arbeit zu dem Schluss, dass die automatische Anpassung des Interfaces eine positive Bereicherung für die Anwendungen ist. Allerdings ist es notwendig, das Feedback für die Nutzer und Nutzerinnen auch an einem fixierten Ort darzustellen. Um eine effiziente Lösung für alle Personen zu ermöglichen, könnten Detailinformationen an einen fixen Standpunkt versetzt werden. Die wichtigsten Informationen wären so immer automatisch im Bild, während die Detailinformationen fixiert sind. Dadurch haben die Teilnehmer und Teilnehmerinnen immer die notwendigsten Informationen, können die Details allerdings bei Bedarf abrufen.

11 Conclusio

Im Rahmen dieser Arbeit wurden mehrere Applikationen zur Erkennung von physiotherapeutischen Übungen detailliert beschrieben. Die gezeigten Applikationen umfassten AR, VR und MR und wurden verwendet, um den Schwierigkeiten der Heimphysiotherapie entgegenzuwirken. Die größten Herausforderungen der Heimphysiotherapie sind es, die Personen zum Training zu motivieren und diesen Feedback während des Trainings zu geben. Diese wurden von dieser Arbeit adressiert, indem die Applikationen die Übungen in eine fordernde Umgebung verpackten. Dadurch wurde versucht, die Motivation zu steigern. Zusätzlich bekamen die Teilnehmer und Teilnehmerinnen über die Applikationen ständiges Feedback zu ihrem aktuellen Trainingsfortschritt, sowie zur Korrektheit der ausgeführten Übungen.

Neben den Problemen der Heimphysiotherapie wurden auch die Hindernisse des Spatial Mappings der Microsoft HoloLens analysiert. Dieses wurde verwendet, um den Boden und den Stepper zu erkennen. Die Erkennung des Bodens in der Applikation verlief problemlos, während die detaillierte und dauerhafte Detektion des Steppers nicht möglich war. Wie in dieser Arbeit beschrieben, erkannte die Microsoft HoloLens den Stepper nicht zuverlässig. Diese Ungenauigkeit der Erkennung führte dazu, dass falsche Ergebnisse erkannt wurden. Um diesem Problem entgegenzuwirken, müsste die Detailerkennung der Microsoft HoloLens verbessert werden.

Wie in dieser Arbeit erklärt wurde, musste für die HTC Vive Applikation eine virtuelle Umgebung erstellt werden. Diese erwies sich im Rahmen der Usability Tests als äußerst immersionsfördernd. Die große Herausforderung der VR Anwendung ist die Platzierung des Steppers in der realen und der virtuellen Welt. Obwohl die Positionen übereinstimmten, fiel es vielen Teilnehmern und Teilnehmerinnen schwer, der Technologie zu vertrauen. Dadurch wurden die Step-Ups nicht flüssig ausgeführt, da erst der Stepper ertastet wurde.

Im Rahmen dieser Arbeit wurden Usability Tests mit zehn Personen durchgeführt, bei denen die Mehrheit der Probanden und Probandinnen die HTC Vive Applikation als beste Unterstützung für das Training gewählt haben. Dieses Ergebnis beruht auf der genauesten Erkennung sowie auf der höchsten Immersion. Obwohl das oben genannte Problem der Positionierung aufgetreten ist, fühlte sich die Mehrheit der Teilnehmer und Teilnehmerinnen in der VR Anwendung am besten unterstützt.

Ein weiterer essentieller Punkt, der von dieser Arbeit analysiert wurde, war die automatische Anpassung von Interfaces. In der Microsoft HoloLens Applikation passte das Interface die Position so an, dass es von keinen Hindernissen verdeckt wurde und immer für die Nutzer und Nutzerinnen sichtbar war. Diese Anpassung wurde in den Usability Tests hinterfragt und als äußerst positiv empfunden. Die Mehrheit der Probanden und Probandinnen empfand es als hilfreich, die wichtigsten Informationen immer im Blickfeld zu sehen.

Neben diesen Erhebungen wurde in dieser Arbeit noch ein Vergleich zwischen den beiden High-End Devices, der Microsoft HoloLens und der HTC Vive, und einem Smartphone mit günstigem HMD gezogen. Im Rahmen dieser Analyse stellte sich heraus, dass die Smartphone Applikation in einigen Punkten nicht mit den teureren Geräten mithalten konnte, allerdings durchaus einsetzbar ist. Da der Vergleich der Applikationen unmittelbar nacheinander erfolgte, hatte das Smartphone gegenüber den anderen Geräten klare Nachteile. Allerdings wurden die wichtigsten Punkte, die Genauigkeit der Erkennung der Übungen und die intuitive Verwendung, bei der Smartphone Applikation nur gering schlechter bewertet als bei den anderen beiden Geräten.

Diese Arbeit kommt zu dem Schluss, dass die Erkennung von physiotherapeutischen Übungen mit AR, VR und MR Applikationen möglich ist. Diese können erkannt und getrackt werden. Zusätzlich können diese effizient ausgewertet werden, um den Nutzern und Nutzerinnen Feedback während des Trainings zu bieten. Die Genauigkeit der Erkennung basiert auf der Anzahl der verwendeten Sensoren. Im Rahmen der Microsoft HoloLens Anwendung und der Smartphone Applikation wurde nur die Kopfposition und der Boden verwendet, um vier Übungen abzudecken. Die HTC Vive Applikation verwendete zusätzliche Sensoren für die Knie- und Fußposition. Dadurch konnten die Bewegungen genauer erkannt werden. Dieser definitive Unterschied in der Genauigkeit wurde von den Nutzern und Nutzerinnen allerdings kaum bemerkt. Das heißt, alle drei Anwendungen sind für die Erkennung solcher Übungen geeignet und können dafür verwendet werden.

In dieser Arbeit wurden nur vier physiotherapeutische Übungen erkannt. Über die Verwendung von Kopfposition und Boden als einzige Messwerte ist es sehr komplex, schwierigere Übungen zu erkennen. Über die Verwendung der Sensoren der HTC Vive ist es zukünftig möglich, auch weitere Übungen zu erkennen, welche im Rahmen von physiotherapeutischem Training relevant sind.

Literaturverzeichnis

- [1] Aruanno, B., Garzotto, F., and Rodriguez, M. C. (2017). Hololens-based mixed reality experiences for subjects with alzheimer's disease. In Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter, pages 1–15. ACM.
- [2] Avila, L. and Bailey, M. (2016). Augment your reality. *IEEE computer graphics and applications*, 36(1):6–7.
- [3] Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4):355–385.
- [4] Berteig, R., Driemeyer, T., LeFrancois, M.-K., and Herken, R. (2009). Computer graphics shader systems and methods. US Patent 7,548,238.
- [5] Bleck, B. M., Katko, B. J., Trujillo, J. B., Harden, T. A., Farrar, C. R., Wysong, A. R., and Mascarenas, D. D. L. (2017). Augmented reality tools for the development of smart nuclear facilities. Technical report, Los Alamos National Laboratory (LANL).
- [6] Blunsom, P. (2004). Hidden markov models. Lecture notes, August, 15:18–19.
- [7] Clay, O. J., Wadley, V. G., Edwards, J. D., Roth, D. L., Roenker, D. L., and Ball, K. K. (2005). Cumulative meta-analysis of the relationship between useful field of view and driving performance in older adults: Current and future implications. *Optometry and vision science*, 82(8):724–731.
- [8] Cruz-Neira, C., Sandin, D. J., and DeFanti, T. A. (1993). Surround-screen projection-based virtual reality: the design and implementation of the cave. In Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pages 135–142. ACM.
- [9] Eddy, S. R. (1996). Hidden markov models. *Current opinion in structural biology*, 6(3):361–365.
- [10] Evans, G., Miller, J., Pena, M. I., MacAllister, A., and Winer, E. (2017). Evaluating the Microsoft Hololens through an augmented reality assembly application. In *Degraded Environments: Sensing, Processing, and Display 2017*, volume 10197. International Society for Optics and Photonics.
- [11] Farley, P., Satran, M., Matteson, M., and Jacobs, M. (2017). Windows device portal overview. website. Windows Dev Center, <https://docs.microsoft.com/en-us/windows/iot-core/manage-your-device/deviceportal>, Besucht am: 2018-04-18.
- [12] Galanti, G.-A. (2004). *Caring for patients from different cultures*. University of Pennsylvania Press.
- [13] Garon, M., Boulet, P.-O., Doironz, J.-P., Beaulieu, L., and Lalonde, J.-F. (2016). Real-time high resolution 3d data on the hololens. In *Mixed and Augmented Reality (ISMAR-Adjunct)*, 2016 IEEE International Symposium on, pages 189–191. IEEE.

- [14] Get-Healthy (2018). How to do up up down down. website. <https://gethealthy.com/exercise/up-up-down-down/>, Besucht am: 2018-04-27.
- [15] Google (2018a). Arcore overview. website. <https://developers.google.com/ar/discover/>, Besucht am: 2018-05-15.
- [16] Google (2018b). Explore the helloar sample app code. website. <https://developers.google.com/ar/develop/unity/tutorials/hello-ar-sample>, Besucht am: 2018-05-15.
- [17] Google (2018c). Fundamental concepts. website. <https://developers.google.com/ar/discover/concepts>, Besucht am: 2018-05-15.
- [18] Haori-Technologies (2018). Product details. website. <http://haoritechnology.com/en-col.jsp?id=109>, Besucht am: 2018-05-15.
- [19] Hockett, P. and Ingleby, T. (2016). Augmented reality with hololens: Experiential architectures embedded in the real world. arXiv preprint arXiv:1610.04281.
- [20] HTC-Corporation (2018a). Gehe über deine controller hinaus. website. <https://www.vive.com/de/vive-tracker/>, Besucht am: 2018-04-29.
- [21] HTC-Corporation (2018b). Htc vive. website. <https://www.vive.com/de/>, Besucht am: 2018-04-29.
- [22] HTC-Corporation (2018c). Htc vive pro. website. <https://www.vive.com/eu/product/vive-pro/>, Besucht am: 2018-06-13.
- [23] Ishii, H. (2010). Augmented reality. fundamentals and nuclear related applications. International Electronic Journal of Nuclear safety and simulation, 1(4):316–327.
- [24] Jansenberger, H. (2011). Sturzprävention in Therapie und Training. Georg Thieme Verlag.
- [25] Jay (2016). Field of view for virtual reality headsets explained. website. VR Lens Lab, <https://vr-lens-lab.com/field-of-view-for-virtual-reality-headsets/>, Besucht am: 2018-04-05.
- [26] Jocham, B. (2017). Erhebung der Bedürfnisse der Zielgruppe - Definition der Nutzungsanforderungen. In Trainingsunterstützung im Alltag nach Rehabilitation Hüft-TEP (TRIMOTEP) -eine Pilotstudie, pages 1–28. FH Joanneum.
- [27] Joggen-Online (2018). Kniebeuge. website. Kniebeugen – einfach und effektiv, <https://www.joggen-online.de/Krafttraining/Bauch,-Beine-und-Po/Kniebeuge-711>, Besucht am: 2018-04-27.
- [28] johnHenry (2016). Field of view limitations. website. Windows Mixed Reality Developer Forum, <https://forums.hololens.com/discussion/192/field-of-view-limitations>, Besucht am: 2018-07-04.
- [29] Juang, B.-H. and Rabiner, L. R. (2005). Automatic speech recognition—a brief history of the technology development. Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara, 1:67.
- [30] Kato, H. and Billinghurst, M. (1999). Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on, pages 85–94. IEEE.

- [31] Koelle, M. and Kranz, M. (2015). The mind behind the glass: human factors in the design of collaborative ar environments. In Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia, pages 445–449. ACM.
- [32] Linville, A., Zeller, M., and Bray, B. (2018). Billboarding and tag-along. website. Windows Dev Center, <https://docs.microsoft.com/en-us/windows/mixed-reality/billboarding-and-tag-along>, Besucht am: 2018-04-12.
- [33] Microsoft (2018). Introducing windows mixed reality. website. Microsoft - Windows Mixed Reality, <https://www.microsoft.com/en-us/windows/windows-mixed-reality>, Besucht am: 2018-04-16.
- [34] Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329.
- [35] Niehorster, D. C., Li, L., and Lappe, M. (2017). The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research. *i-Perception*, 8(3).
- [36] Niklas, D. (2016). Skydiving coordinate system part 2. website. Skydivemag, <https://www.smallteaser.com/@niklasdaniel-com/article/skydiving-coordinate-system-part-2?category=Besucht> am: 2018-04-12.
- [37] Ohshima, T. and Satoh, K. (2005). Game apparatus for mixed reality space, image processing method thereof, and program storage medium. US Patent 6,951,515.
- [38] Park, Y., Lepetit, V., and Woo, W. (2008). Multiple 3d object tracking for augmented reality. In Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, pages 117–120. IEEE Computer Society.
- [39] Ruddle, R. A. (2004). The effect of environment characteristics and user interaction on levels of virtual environment sickness. In *Virtual Reality, 2004. Proceedings.* IEEE, pages 141–285. IEEE.
- [40] Saffer, D. (2008). *Designing gestural interfaces: touchscreens and interactive devices.* O'Reilly Media, Inc."
- [41] Samsung (2017). Specifications. website. <http://www.samsung.com/global/galaxy/galaxy-s8/specs/>, Besucht am: 2018-05-15.
- [42] Schlömer, T., Poppinga, B., Henze, N., and Boll, S. (2008). Gesture recognition with a wii controller. In Proceedings of the 2nd international conference on Tangible and embedded interaction, pages 11–14. ACM.
- [43] Steuer, J. (1992). Defining virtual reality: Dimensions determining telepresence. *Journal of communication*, 42(4):73–93.
- [44] Stewart, D. (1965). A platform with six degrees of freedom. *Proceedings of the institution of mechanical engineers*, 180(1):371–386.
- [45] Sudo, M. and Namrata, K. (2017). Windows device portal. website. Windows Dev Center, <https://docs.microsoft.com/en-us/windows/iot-core/manage-your-device/deviceportal>, Besucht am: 2018-04-18.

- [46] Sun, L., Klank, U., and Beetz, M. (2009). Eyewatchme—3d hand and object tracking for inside out activity analysis. In Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on, pages 9–16. IEEE.
- [47] Taylor, A. G. (2016). Develop Microsoft HoloLens Apps Now. Springer.
- [48] Tilkov, S. and Vinoski, S. (2010). Node.js: Using javascript to build high-performance network programs. IEEE Internet Computing, 14(6):80–83.
- [49] Turner, A., Zeller, M., and Bray, B. (2018). Gaze. website. Windows Dev Center, <https://docs.microsoft.com/en-us/windows/mixed-reality/gaze>, Besucht am: 2018-05-16.
- [50] Unity3D (2018). Unity user manual. website. Unity Manual, <https://docs.unity3d.com/Manual/index.html>, Besucht am: 2018-04-15.
- [51] Wagner, D. and Schmalstieg, D. (2003). First steps towards handheld augmented reality. IEEE.
- [52] Wang, S., Mao, Z., Zeng, C., Gong, H., Li, S., and Chen, B. (2010). A new method of virtual reality based on unity3d. In Geoinformatics, 2010 18th International Conference on, pages 1–5. IEEE.
- [53] Whitney, T., Jacobs, M., Satran, M., and Weston, S. (2018). What’s a universal windows platform (uwp) app? website. Windows Dev Center, <https://docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide> , Besucht am: 2018-06-13.
- [54] Wiki-How (2017). Deine zehen berühren. website. <https://de.wikihow.com/Deine-Zehen-beruehren> , Besucht am: 2018-06-26.
- [55] Xinreality (2017). Vive tracker. website. https://xinreality.com/wiki/Vive_Tracker, Besucht am: 2018-04-29.
- [56] Zeller, M. and Bray, B. (2018a). Gestures. website. Windows Dev Center, <https://docs.microsoft.com/en-us/windows/mixed-reality/gestures>, Besucht am: 2018-04-16.
- [57] Zeller, M. and Bray, B. (2018b). Interaction fundamentals. website. Windows Dev Center, <https://docs.microsoft.com/en-us/windows/mixed-reality/interaction-fundamentals>, Besucht am: 2018-04-16.
- [58] Zeller, M., Bray, B., and Baker, K. (2018a). Spatial mapping. website. Windows Dev Center, <https://docs.microsoft.com/en-us/windows/mixed-reality/spatial-mapping>, Besucht am: 2018-04-19.
- [59] Zeller, M., Bray, B., and Baker, K. (2018b). Spatial sound. website. Windows Dev Center, <https://docs.microsoft.com/en-us/windows/mixed-reality/spatial-sound>, Besucht am: 2018-04-19.
- [60] Zeller, M., Bray, B., Wojciakowski, M., and Cowley, E. (2018c). Motion controllers. website. Windows Dev Center, <https://docs.microsoft.com/en-us/windows/mixed-reality/motion-controllers>, Besucht am: 2018-04-16.
- [61] Zinn, B., Guo, Q., and Sari, D. (2016). Entwicklung und Evaluation der virtuellen Lern- und Arbeitsumgebung VILA. Journal of Technical Education (JOTED), 4(1).