

FAKULTÄT FÜR INFORMATIK

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelorarbeit in Informatik: Games Engineering

**Virtual Reality Wiederverkörperung einer
Hand mittels simulierter Robotik**

Tassilo Johannes Walto Steinmann

FAKULTÄT FÜR INFORMATIK

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelorarbeit in Informatik: Games Engineering

**Virtual Reality Re-Embodiment of a Hand
using simulated Robotics**

**Virtual Reality Wiederverkörperung einer
Hand mittels simulierter Robotik**

Autor:	Tassilo Johannes Walto Steinmann
Aufgabenstellerin:	Prof. Gudrun Klinker
Betreuer:	M.Sc. Sandro Weber
Abgabedatum:	13.3.2020

Ich versichere, dass ich diese Bachelorarbeit in Informatik: Games Engineering selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, 13.3.2020

Tassilo Johannes Walto Steinmann

Danksagung

Zuerst möchte ich mich bei allen Mitarbeitern des Lehrstuhls für Augmented Reality bedanken, insbesondere bei Sandro Weber und Professorin Gudrun Klinker, die diese Bachelorarbeit ermöglichten, für Fragen offen waren, mich bei der Umsetzung dieser Arbeit unterstützten und mir einen Raum mit der für diese Arbeit nötigen Ausstattung zu Verfügung gestellt haben. Insbesondere danke ich Sandro Weber für seine Geduld und Unterstützung dieser Arbeit. Ich danke Adnane Jadid für hilfreiche Tipps, Professorin Gudrun Klinker und Linda Rudolph für die Teilnahme an der Nutzerstudie und allen anderen Teilnehmern.

Weitere Danksagungen bekunde ich meiner Familie, die mich bei dieser Arbeit begleitete, sowie Maximilian H., Marie O., Severin S. und Melina W. für das Bereitstellen einer idealen Schreibatmosphäre.

Abstract

Virtuelle Realität (VR) übernimmt eine immer größer Rolle in unserer Gesellschaft, sei es auf spieltechnischem, wie auch auf industriellem oder medizinischem Sektor. So ist es sowohl derzeit möglich, sich über VR Brillen bei Computerspielen durch virtuelle Gegner zu kämpfen, als auch Simulationen, wie bei einem chirurgischen Eingriff mit optischen Erweiterungen durch virtuelle Realität zu erleben. Aufgrund dieser virtuellen Möglichkeiten stellt sich die Frage nach Wiederverkörperung. Zielsetzung dieser Arbeit ist es, die Wiederverkörperung der eigenen Hände durch virtuelle Roboterhände im Bereich der Virtuellen Realität in einem Programm zu implementieren, indem der Unity 3D Client für die Neurorobotics Platform um Fingertracking mit den Valve Index Controllern erweitert wird. Dabei wird zunächst das Programm für das Fingertracking geschrieben. Dieses Programm wird anschließend mit der Manipulation eines Würfels durch Hand- und Fingertracking erweitert. Mit Hilfe von Versuchsreihen mit verschiedenen Testpersonen wird auf die implementierte Wiederverkörperung und das Fingertracking getestet und ausgewertet.

Inhalt

Danksagung	iii
Abstract	iv
1. Einleitung	1
1.1. Motivation	1
1.2. Virtuelle Realität	1
1.3. Wiederverkörperung	2
2. Verwandte Arbeiten	3
2.1. Gummihand Illusion	3
2.2. VR Re-Embodiment in the Neurorobotics Platform	4
2.3. Virtual Embodiment: Umgang mit Diskrepanzen zwischen dem virtuellen und dem realen Körper	4
2.4. Virtuelle Verkörperung von menschlichen Füßen in der Neurorobotics Platform	5
2.5. PID-Tuning Framework für ferngesteuerte humanoide Roboter	5
3. Hintergrundwissen	6
3.1. Verkörperung und Wiederverkörperung	6
3.1.1. Gefühl der Selbstlokalisierung (sense of self-localization)	6
3.1.2. Gefühl des Handlungsbewusstseins (sense of agency)	7
3.1.3. Gefühl des Körperbesitzes (sense of body ownership)	8
3.1.4. Virtuelle Wiederverkörperung	9
3.1.5. Robotische Wiederverkörperung	10
3.2. Hardware	11
3.2.1. Basisstationen	11
3.2.2. Headset	12
3.2.3. Controller	12
3.2.4. Mögliche andere Hardware	13
3.3. Software	14
3.3.1. The Human Brain Project und die Neurorobotics Platform	14
3.3.2. SteamVR Unity Plugin	15

3.3.3. Unity 3D Client	15
3.4. Simulierte Robotik und PID Regler	15
4. Vorgeschlagener Ansatz	17
5. Implementierung	18
5.1. SteamVR Input System	18
5.2. Implementierung des Handtrackings in Unity	18
5.3. Server-Client Kommunikation	20
5.4. Implementierung des haptischen Feedbacks	25
5.5. Aufbau der Umgebung für eine spezifische Nutzerstudie	29
6. Nutzerstudie und Ergebnisse	31
6.1. Aufbau und Durchführung der Nutzerstudie	31
6.2. Auswertung	32
7. Fazit und Ausblick	40
7.1. Fazit	40
7.2. Zukünftige Arbeit	41
Verzeichnis der Abbildungen	43
Verzeichnis der Tabellen	44
Literaturverzeichnis	45
A. Anhang	48
A.1. Programmcode	48
A.2. Diagramme der Nutzerstudie	48

1. Einleitung

Dieses Kapitel befasst sich mit der grundlegenden Motivation dieser Arbeit, der virtuellen Realität und gibt einen kurzen Einblick in den Begriff der Wiederverkörperung.

1.1. Motivation

Um einen Roboter zu steuern bedarf es entweder einer Software mit künstlicher Intelligenz, oder eines Benutzers, der den Roboter ferngesteuert kontrolliert. Diese Arbeit befasst sich damit, einem potentiellen Benutzer eine möglichst intuitive Fernsteuerung eines virtuellen Roboters zu Verfügung zu stellen, sodass es für eine große Bandbreite von Personen ohne viel Vorwissen möglich ist, diesen Roboter zu steuern. Hierzu dient die Neurorobotics Platform, die es erlaubt, Experimente mit virtuellen, oder sogar realen Robotern durchzuführen. Der Fokus dieser Arbeit liegt auf den Händen des Benutzers, der die virtuellen Hände des Roboters mit dem sogenannten Valve Index Controller steuern soll. Dieser Controller ermöglicht offenbar die intuitive Bedienung durch Handtracking, ist darüber hinaus im Vergleich zu anderer Handtrackinghardware relativ kostengünstig und kann vor allem in privatem Umfeld effizient genutzt werden.

1.2. Virtuelle Realität

Virtuelle Realität gewinnt, wie vorausgeschickt, immer mehr an Bedeutung. Durch virtuelle Reality-Headsets kann der Benutzer in die Welt des Computers eintauchen und interagieren. Um dem Spieler eine immersive Spielerfahrung zu verschaffen wird bei Computerspielen oft auf die Egoperspektive zurückgegriffen. Durch die Technik der virtuellen Realität wird das Spielgeschehen als nahezu real empfunden. Dabei wird ein virtueller Avatar als eigener Körper wahrgenommen. Das Gefühl, sich in einen virtuellen Körper hineinzusetzen, wird in der Literatur als Wiederverkörperung beschrieben. Auf diese Wiederverkörperung wird im Verlauf dieser Arbeit näher eingegangen.

Bei der Kontrolle eines Roboters in einer virtuellen Realitätsumgebung schlüpft der

Benutzer in die Rolle des Roboters. Durch die intuitive Steuerung der Roboterhände mit dem Index Controller wird vermutet, dass auch hier ein Gefühl der Wiederverkörperung auftritt.

1.3. Wiederverkörperung

Die Wiederverkörperung, die in 3.1. näher erleutert wird, ist ein Gefühl, sich in einen künstlichen Körper hineinzusetzen, also ihn wahrzunehmen, als wäre es der eigene Körper. Dieses Gefühl kann bei einer Interaktion zwischen Benutzer und einem Roboter, oder in der virtuellen Realität auftreten, wenn ein Benutzer einen Körper, oder einzelne Gliedmaßen steuert. Dazu verweist diese Arbeit auf den Versuch der Gummihand Illusion, die später in 2.1. beschrieben wird.

Der Begriff "Wiederverkörperung" setzt allerdings den Begriff "Verkörperung" voraus. Der Unterschied zwischen Verkörperung und Wiederverkörperung liegt in der Definition dieser Begriffe und in der Wahl des Mediums, also ob eine künstliche Gummihand, eine virtuelle Simulation der Hand oder ein Roboter verwendet wird. Sie befassen sich aber grundlegend mit der Wahrnehmung einer Person eines künstlichen Gegenstandes und diesen als realen Teil des eigenen Körpers zu empfinden. Näheres dazu wird in 3.1. erläutert.

2. Verwandte Arbeiten

Dieses Kapitel befasst sich mit einigen Arbeiten, in denen die Thematik der Verkörperung und des Bewegungstrackings behandelt wird. Vier dieser Arbeiten verwenden die Neurorobotics Plattform und den Unity 3D Client und sind dadurch Grundlage für diese Arbeit.

2.1. Gummihand Illusion

Die Gummihand Illusion, wie unter anderem in der Arbeit von Ye Yuan und Anthony Steed beschrieben [YS10], ist ein wesentliches Experiment für die Verkörperung und damit ein wichtiger Hintergrund für diese Arbeit. Das Experiment, wie eine Gummihand Illusion hervorgerufen wird, benötigt eine Testperson, einen Tester und eine Gummihand. Die Testperson sieht nur die Gummihand, ihre reale Hand befindet sich hinter einem Sichtschutz. Sowohl künstliche, als auch reale Hand werden taktil stimuliert, beispielsweise mit einer Feder gestreichelt. Wichtig bei diesem Experiment ist, dass die taktile Stimulation an den korrespondierenden Stellen an Gummihand und realer Hand erfolgt. Dadurch entsteht nach kurzer Zeit das Gefühl, die Gummihand sei die reale Hand. In manchen Versuchsreihen wird vom Tester unerwartet mit einem Hammer auf die Gummihand eingeschlagen, was zur Folge hat, dass die Testperson zusammensuckt und versucht, die als Teil des eigenen Körpers empfundene Gummihand vor dem Schlag wegzuziehen.

Die Arbeit "Is the Rubber Hand Illusion Induced by Immersive Virtual Reality?" von Ye Yuan und Anthony Steed geht der Frage nach, ob die Gummihand Illusion auch in immersiver virtueller Realität stattfindet. Die Autoren stellen eine Beziehung zwischen immersiver virtueller Realitätserfahrung und der Gummihand Illusion her, indem sie behaupten, dass eine ähnliche Illusion, wie die bei der Gummihand auch erzeugt werden kann, wenn ein virtueller Körper aktiv genutzt wird. Um dies zu beweisen, verwenden sie die gleichen Evaluierungen, wie die für die Gummihand Illusion. Sie stellen zwei Thesen auf, die in einem Experiment überprüft werden. Die Testpersonen assoziieren den virtuellen Arm als ihren eigenen, wenn eine visuelle und propriozeptive Übereinstimmung mit virtuellem und realem Arm stattfindet und diese Assoziation nicht von Abweichungen der Tracker gestört wird. Dieses Experiment wird durch Fragebögen und der sogenannten galvanischen Hautreaktion evaluiert. [YS10]

2.2. VR Re-Embodiment in the Neurorobotics Platform

In der Arbeit von Sandro Weber und Gudrun Klinker ist die Wiederverkörperung in der Neurorobotics Platform wegweisend für weitere Arbeiten.

Die Neurorobotics Platform erlaubt demnach Benutzern, einen Roboter zu simulieren und dadurch Wiederverkörperungsexperimente durchzuführen. Ziel der Arbeit ist die Antwort auf die Frage, wie man Personen mit Hilfe von virtuellen Realitätsgeräten, wie Headsets oder Controllern, in die Neurorobotics Platform einbringen kann, sodass Experimente für die Verkörperung, beziehungsweise Wiederverkörperung möglich sind. Die Idee dabei ist, einen humanoiden Roboter, den Avatar, zu erzeugen, der von einem Benutzer kontrolliert wird. Der Benutzer soll sich in diesem Avatar wiederverkörpert fühlen. Anschließend wird der Avatar in eine virtuelle, simulierte Umgebung der Neurorobotics Platform eingesetzt, mit der er physikalisch interagieren kann. Die Arbeit von Sandro Weber und Gudrun Klinker befasst sich auch mit dem Begriff der Wiederverkörperung und der Implementierung des Avatars, sowie der Kontrolle des Benutzers. Besonders hervorzuheben ist, dass die verwendete Software öffentlich und frei zugänglich ist und die verwendete Hardware, die HTC Vive, zu relativ günstigen Preisen von Benutzern gekauft werden kann, um das Ergebnis des Projekts weiterverfolgen, oder für spätere Anwendungen nutzen zu können. [WK19]

2.3. Virtual Embodiment: Umgang mit Diskrepanzen zwischen dem virtuellen und dem realen Körper

Die Bachelorarbeit von Jonathan Haudenschild beschreibt Diskrepanzen, die in der virtuellen Realität aufkommen können. Das Ziel seiner Arbeit ist, die virtuelle Verkörperung in immersiver Umgebung zu verbessern. Er fokussiert sich auf die Abweichungen, die in Bewegungsabläufen an Armen, Händen und Kopf auftreten können, die beispielsweise durch Hindernisse verursacht werden, die in der virtuellen Umgebung vorkommen, nicht jedoch in der Realität existieren, sodass durch sie durchgegriffen werden kann. Um das Gefühl der Verkörperung nicht zu verlieren, werden diese Diskrepanzen virtuell durch verschiedene Mechanismen ausgeglichen. Die Diskrepanzen an Händen und Armen behandelt der Autor mit haptischem Feedback, Audio-Feedback und visuellen Indikatoren. Die Diskrepanzen am Kopf versieht er mit visuellen Effekten. Dies wird zusammenfassend in einer abschließenden Nutzerstudie evaluiert. Für diese Arbeit wurde auch die HTC Vive inklusive der zwei zu dem virtuellen Reality-Headset gehörenden Controller verwendet. Das Projekt wurde in Unity entwickelt und die Modelle in Blender erstellt. [Hau18]

2.4. Virtuelle Verkörperung von menschlichen Füßen in der Neurorobotics Plattform

Jason Janse van Rensburg befasste sich in seiner Bachelorarbeit "Virtuelle Verkörperung von menschlichen Füßen in der Neurorobotics Plattform" mit der Implementierung von Fußtracking in der Neurorobotics Plattform, dem Unity 3D Client und der damit einhergehenden Verkörperung des Avatars. Dafür wurden Feedbackmechanismen sowohl für Füße, als auch Hände und Kopf entwickelt. Anschließend wurde eine Nutzerstudie für diese Mechanismen durchgeführt. [Ren19]

2.5. PID-Tuning Framework für ferngesteuerte humanoide Roboter

Die Masterarbeit von Markus Webel fokussiert sich auf die Einstellung des PID Reglers in der Neurorobotics Plattform, der für die Gelenkbewegungen eines virtuellen Roboters verwendet wird. Um diese zu bewerkstelligen wurde ein Framework entwickelt, welches aus Tools, APIs und UIs besteht. Dadurch sollen sowohl erfahrene, als auch unerfahrene Benutzer Tools zu Verfügung gestellt bekommen, die das Einstellen des PID Reglers erleichtern. Zudem erleichtert das Framework sowohl das manuelle, als auch das automatische Einstellen der Reglerparameter. Es bietet außerdem Funktionalität für Datenerfassung, Bewegungswiedergabe und Leistungsbewertung. Für die automatische Einstellung wurden verschiedene Heuristiken verwendet, die in vier verschiedenen Robotersetups getestet wurden. [Web19]

3. Hintergrundwissen

Dieses Kapitel beschreibt das Hintergrundwissen für die Wiederverkörperung und die für die Implementierung verwendete Hardware und Software, auf die diese Arbeit zurückgreift. Die Zitate der englischen Fachliteratur wurden für diese Arbeit übersetzt.

3.1. Verkörperung und Wiederverkörperung

"Sich hier und dort vertreten zu fühlen", also an mehreren Orten virtuell, oder real, ist nach Robert Rosenberger das sogenannte abgeänderte Präsenzgefühl. Es wird von Wissenschaftlern, die phänomenologischen Analysen von menschlichem Austausch mit Technologie nachgehen, als Wiederverkörperung bezeichnet. Im folgenden Abschnitt wird auf zwei Varianten der technologischen Wiederverkörperung eingegangen: die virtuelle Wiederverkörperung und die robotische Wiederverkörperung. [Ros15]

Um die Wiederverkörperung in der virtuellen Realität zu verstehen, muss erst das Gefühl der Verkörperung näher beschrieben werden. Das Gefühl der Verkörperung unterliegt der Fragestellung, ob es für eine Person möglich ist, einen virtuellen Körper in einer virtuellen Umgebung genau so, wie den physischen Körper zu spüren. Dabei untergliedert man dieses Gefühl in drei Komponenten:

- Gefühl der Selbstlokalisierung
- Gefühl des Handlungsbewusstseins
- Gefühl des Körperbesitzes

[KGS12]

3.1.1. Gefühl der Selbstlokalisierung (sense of self-localization)

Nach Kiltner et al. (2012) ist "Selbstlokalisierung ein begrenztes Volumen im realen Raum, in dem man sich geortet fühlt". Die Autoren erklären, dass im Normalfall Selbstlokalisierung und leiblicher Raum in dem Gefühl zusammenfallen, dass man sich selbst innerhalb eines physischen Körpers befindet. Allerdings kann dieses Gefühl bei außerkörperlichen Erfahrungen, also wenn man sich außerhalb des physischen Körpers

wahrnimmt, zusammenbrechen. [KGS12]

Das Gefühl der Selbstlokalisierung ist die eigene räumliche Erfahrung, in einem Körper zu sein. Das Erleben der eigenen Anwesenheit in einer Umgebung ist jedoch nicht mit dem Gefühl der Selbstlokalisierung gleichzusetzen. Selbstlokalisierung und Anwesenheit werden von Kilteni et al. (2012) als psychologische Zustände benannt. Während Selbstlokalisierung für die Beziehung zwischen sich selbst und dem eigenen Körper zuständig ist, beschreibt die Anwesenheit die Beziehung zwischen sich selbst und der Umgebung. Bei der Wahrnehmung einer Representation eines Körpers stellt sich die Frage nach der Beziehung zwischen dem Körper und der Umgebung. Um diese Unterscheidung besser zu veranschaulichen, kann man sich die Selbstlokalisierung als Gefühl vorstellen, sich im eigenen biologischen Körper, aber auch in einem Körper eines Avatars verankert zu fühlen. Das Gefühl der Anwesenheit beschreibt, sich in einem physischen oder virtuellen Raum zu lokalisieren, ohne dass eine Darstellung des Körpers benötigt wird. Obwohl beide Gefühle verschiedene Wahrnehmungen eines Selbst widerspiegeln, ergänzen sie sich zur räumlichen Representation einer Person. Kilteni et al. (2012) beschreiben, dass Selbstlokalisierung stark von der visuellräumlichen Perspektive des "individuellen Selbst" bestimmt wird. Studien haben gezeigt, dass physiologische Reaktionen zum Beispiel auf Bedrohungen durch einen künstlichen Körper in der Egoperspektive ausschlaggebender waren, als in der Third-Person Perspektive. Auch vestibuläre, also den Gleichgewichtssinn betreffende Signale, spielen eine wesentliche Rolle für das Gefühl der Selbstlokalisierung, da sie Informationen hinsichtlich der Verschiebung und Rotation des zu representierenden Körpers zusätzlich zur Orientierung in Bezug auf die Schwerkraft enthalten. Taktile Anregungen beeinflussen ebenfalls die Selbstlokalisierung. Die Grenze zwischen Umgebung und Körper erfühlt sich über die Haut. Eine Studie von Lenggenhager et al. (2009) hat gezeigt, dass beobachtete taktile Stimulation begleitet von deckungsgleicher physikalischer Stimulation die visuelle Perspektive dominieren und dadurch die Selbstlokalisierung bestimmen. [KGS12] Das bedeutet, dass wenn beispielsweise bei der Gummihand Illusion eine künstliche und reale Hand der Testperson gleichzeitig stimuliert werden, die Gummihand zwar als künstlich wahrgenommen wird, aber die Illusion entsteht, ein Gefühl in dieser Hand zu spüren, sodass dadurch das Gefühl der Selbstlokalisierung auf die Gummihand übertragen wird.

3.1.2. Gefühl des Handlungsbewusstseins (sense of agency)

Das Gefühl des Handlungsbewusstseins ist nach Kilteni et al. (2012) das Gefühl, ganzheitliche, motorische Kontrolle zu haben, einschließlich der subjektiven Erfahrung von Aktion, Steuerung, Intention, Bewegungskontrolle und der bewussten Erkenntnis des Willens. Ein Beispiel für eine Störung dieses Gefühls zeigt das Alien-Hand-

Syndrom. Betroffene Personen können einzelne Gliedmaßen nicht mehr bewusst steuern und haben die Empfindung, dass diese Gliedmaßen verselbstständigt handeln. Das Gefühl des Handlungsbewusstseins resultiert aus dem Vergleich zwischen den prognostizierten, sensorischen Konsequenzen des eigenen Handelns ausgehend von der Efferenzkopie und der tatsächlichen Auswirkung der Sinneswahrnehmung. [KGS12] Dies ist ein mentaler Vorgang. Efferenz ist der Begriff zur Klassifizierung der Leitungsrichtung von Nervenfasern. Eine efferente Nervenfasern übermitteln hierbei Informationen vom Zentralnervensystem an die Erfolgsorgane. [Spe00a] Gewebe, Drüsen und Organe, die von solchen efferenten Nervenfasern angesteuert werden, nennt man Erfolgsorgane. [Spe00b] Die Efferenzkopie ist eine hypothetische Vorstellung einer Kopie der nach außen abgegebenen Efferenzen, mit der die Rückmeldungen über die vollzogenen Handlungen verglichen werden. [Spe00a] Wenn eine vorhergesehene Auswirkung einer Tat mit der tatsächlichen Auswirkung der Tat übereinstimmt, also synchrone, visuomotorische Korrelation unter aktiver Bewegung, empfindet man sich als Handlungsinitiator dieser Tat. Als Beispiel sei die Augen-Hand-Koordination genannt. Wenn ein Werkzeug unter der Kontrolle eines Initiators steht, kann es zu einer Verkörperung dieses Werkzeuges kommen. [KGS12] Ein Beispiel solch einer Verkörperung wird in dieser Arbeit näher beschrieben.

In der Nutzerstudie wird darauf getestet, ob ein Gefühl entsteht, Herr über die Roboterhände zu sein, also diese Hände mit dem eigenen Willen kontrollieren zu können. Einige Studien aus der Arbeit von Kilteni et al. (2012) haben gezeigt, dass Abweichungen zwischen visuellem Feedback einer Aktion und der tatsächlichen Bewegung das Handlungsbewusstsein negativ beeinflussen. Bereits eine Abweichung von 150 Millisekunden stört dieses Bewusstsein. [KGS12] Daraus lässt sich vermuten, dass im virtuellen Bereich diese Störung der zeitlichen Korrelation ein Gefühl der Verkörperung zwischen virtuellem Avatar und realer Person nicht zulässt.

3.1.3. Gefühl des Körperbesitzes (sense of body ownership)

Nach Kilteni et al. ist Körperbesitz die Zuordnung eines Selbst zu seinem eigenen Körper. [KGS12] Der Körper wird als Quelle der erlebten Empfindung verstanden. Eine Störung dieses Gefühls kann durch "Somatoparaphrenie" hervorgerufen werden. Dabei verspüren betroffene Personen ihre Körperteile als fremd. Normalerweise entsteht das Gefühl des Körperbesitzes aus einer Kombination von Bottom-Up- und Top-Down-Informationen. Bottom-Up-Informationen sind afferente, sensorische Informationen, die von den Sinnesorganen zum Gehirn führen, wie beispielsweise visuelle, oder taktile Informationen, das bedeutet Informationen, die über das Auge oder die Haut aufgenommen werden. Top-Down-Informationen bestehen aus kognitiven Prozessen, die die Verarbeitung von Sinnesreizen regulieren. [KGS12] Werner Stangl beschreibt

Top-Down-Informationen, beziehungsweise Top-Down-Verarbeitung in der Psychologie als die absteigende, konzeptgesteuerte Informationsverarbeitung, die durch höhere mentale Prozesse gesteuert wird, etwa wenn Menschen Wahrnehmung auf Grund ihrer eigenen Erfahrungen und Erwartungen interpretieren. [Sta20] Diese Kombination aus beiden Informationswegen lässt sich gut durch ein Beispiel veranschaulichen: Bei einer blinden Person mit ihrem Blindenstock entsteht ein Gefühl des Körperbesitzes, wenn der Blindenstock ständig verwendet wird. Der Stock wird zu einer Verlängerung des Armes und übermittelt taktile Informationen über den Arm an das Gehirn. Das sind Bottom-Up-Informationen. Im Gedächtnis werden Erinnerungen für die Verwendung des Stocks gespeichert, um sie erneut abrufen zu können. Das sind Top-Down-Informationen. Beide Informationen bestimmen im täglichen Leben jeden Handlungsablauf.

Ein weiteres Beispiel für das Gefühl des Körperbesitzes wird besonders deutlich im Experiment der Gummihand Illusion (künstliche, menschenähnliche Körperteile, wie eine Gummihand, können als Teil des eigenen Körpers wahrgenommen werden). Wie bereits oben beschrieben, sieht die Person bei der Illusion nur die künstliche Gummihand, während sowohl künstliche, als auch reale Hand zeitgleich stimuliert werden. Dies führt dazu, dass die Person denkt, die Gummihand sei Teil ihres Körpers.

Das Gefühl des Körperbesitzes kann bei Experimenten, wie bei der Gummihand Illusion, gestört werden, "wenn das externe Objekt den realen Körperteilen nicht gleichkommt, oder in einer anderen räumlichen Konfiguration steht." Das bedeutet, wenn das zu verkörpernde Objekt nicht so, wie das eigene Körperteil aussieht und räumlich zu weit vom eigenen Körper entfernt ist, kann man sich nicht ausreichend in das Objekt hineinversetzen. "Deshalb ist eine grundlegende, morphologische Ähnlichkeit von externem Objekt und realem Körperteil erforderlich, um ein "besitzergreifendes" Gefühl zu erzeugen." [KGS12] Grob gesagt sollte eine Hand also als eine Hand abgebildet werden und nicht als ein Fuß.

3.1.4. Virtuelle Wiederverkörperung

Virtuelle Wiederverkörperung wird in Videospielen besonders deutlich. Der Spieler schlüpft nach einer kurzen Einfühlungszeit in die Rolle der selbstgestalteten, oder vorgegebenen Spielfigur. In Rollenspielen, wie "The Elder Scrolls 5: Skyrim" ist es möglich, die eigene Spielfigur optisch nach eigenen Vorlieben zu gestalten, sodass eine Verbindung zwischen realer Person und virtuellem Avatar aufgebaut wird. Dabei ist die Spielerperspektive relevant. Im Spiel "The Witcher 3: Wild Hunt" wird die Spielfigur, Geralt von Riva, in der Third-Person-Perspektive gesteuert. Aber dies hält den Spieler nicht davon ab, sich mit dem Protagonisten zu identifizieren. Deutlicher wird das Gefühl der virtuellen Wiederverkörperung in Virtual Reality Spielen, da die

Spieler durch ein Virtual Reality Headset direkt in die virtuelle Umgebung eintauchen und meistens in der Egoperspektive spielen. Im Spiel Boneworks kann man den Protagonisten des Spiels mit dem Valve Index Controller steuern, der für eine realistische Handsimulation sorgt. Die Kombination aus Egoperspektive und akkuratem Bewegungstracking führt zum eindrucklichen Gefühl, selbst in das Spiel integriert zu sein. In solchen Spielen fungieren die virtuellen Avatare für den Spieler als Ort der Wahrnehmung und des Handelns in der jeweiligen virtuellen Umgebung. Die Spieler passen sich an diese virtuelle Umgebung an, indem sie eigene reale Elemente der Verkörperung und Raumempfindungen in diese übertragen. Die Sozialpsychologen Jeremy Bailenson und Jim Blascovich haben in verschiedenen Studien zusammengefasst, dass Menschen ihre Avatare und die Avatare von anderen nicht als bloße Darstellungen oder leere Animationen betrachten. Viel mehr nehmen sie ihre Avatare wie ihre Körper wahr und das innerhalb von Minuten der Steuerung des Avatars. [Ros15] Demnach scheint es eine starke Identifikation mit dem Körper des angeeigneten Avatars zu geben, aber auch eine Projektion von räumlichen Zwängen, wenn man beispielsweise erkennt, nicht durch eine Öffnung zu passen. Nach Robert Rosenberger (2015) fließen auch die Möglichkeiten der realen Verkörperung, sich psychisch in den Avatar hineinzusetzen und gesellschaftliche Normen und Werte, wie man in beispielsweise Onlinespielen mit anderen Spielern umgeht, mit ein. Diese Kombination solcher Erscheinungsformen kann als virtuelle Wiederverkörperung bezeichnet werden. [Ros15]

3.1.5. Robotische Wiederverkörperung

Der wesentliche Unterschied zwischen der virtuellen und der robotischen Wiederverkörperung liegt in der Wahl des Mediums, das vom Benutzer verwendet wird. So wird von Rosenberger et. al. die robotische Wiederverkörperung als das Gefühl beschrieben, in einem Roboter zu sein, ihn sozusagen zu bewohnen. [Ros15] Das bedeutet, der Körper des Roboters, beziehungsweise seine Körperteile verkörpern die realen, fleischlichen Körperteile des Benutzers. In einem Artikel von Jonathan Cole, Oliver Sacks und Ian Waterman, der im Buch "Postphenomenological Investigations: Essays on Human–Technology Relations" zitiert wird, wird von einer Erfahrung bei dem Gebrauch von einem Teleroboter-System beim Johnson Space Center in Houston, Texas berichtet. Dort sahen und steuerten Versuchspersonen einen Roboterarm ohne peripheres Feedback zu erhalten. Werkzeuge wurden mittels einer realen Hand von einer Roboterhand in die andere genommen, Eier aufgehoben und Knoten gebunden. Nach einigen Minuten entstand bei den Benutzern das Gefühl einer Wiederverkörperung, also den Roboterkörper als Teil des realen Körpers zu spüren. Dies zeigte sich, als eine der Personen sehr vorsichtig auf einen Schraubenschlüssel achtete, aus Angst, er könne ihr auf den Fuß fallen, obwohl lediglich die Roboterarme gesehen und bewegt wurden.

Robotische Wiederverkörperung kommt in unterschiedlichem Maße in Teleroboter-Systemen vor, wie beispielsweise beim entfernten Bedienen von Fahrzeugen, oder Telechirurgie. Trotz wichtiger Unterschiede sind drei Ähnlichkeiten zwischen virtueller und robotischer Wiederverkörperung entscheidend. Beide Wiederverkörperungsarten erfordern eine Schnittstellenausstattung, wie beispielsweise Joysticks oder Tastaturen, visuellen Zugang zu und Rückmeldung von der entfernten Umgebung und die Fähigkeit, dort aktiv und effektiv zu sein. [Ros15] Das bedeutet, dass der Benutzer die Möglichkeit hat, einen Roboter durch technische Geräte in einer entfernten Umgebung zu steuern. Durch beispielsweise einen Bildschirm, oder einen Beamer wird der Roboter und seine Umgebung für den Benutzer dargestellt. Dies ist der visuelle Zugang und die Rückmeldung des Roboters. Der Benutzer muss dazu noch die Fähigkeit haben, den Roboter in der jeweiligen Umgebung steuern zu können, also aktiv verschiedene Aufgaben zu erledigen und darin effektiv zu sein. Beispiel für eine Anwendung ist ein Bombenentschärfungsroboter. Dieser wird aktiv und effektiv vom jeweiligen Benutzer durch einen Controller gesteuert und liefert über eine Kamera visuelle Rückmeldung.

3.2. Hardware

3.2.1. Basisstationen

Zwei Basisstationen, auch Lighthouses genannt, sind diagonal gegenüberliegend an der Zimmerdecke angebracht. Mit ihnen ist es möglich, die räumliche Position des Headsets und der Controller zu erfassen. Dabei wird ein Bereich von 3,5 m x 3,5 m abgedeckt. [Vivnaa]

Diese Lighthouses emittieren 60 mal pro Sekunde Infrarotlichtimpulse, die von den Empfängern, Headset oder Controller, über Photosensoren empfangen werden. Indem die Lichtstrahlen der Lighthouses auf die Photosensoren des Empfängergerätes treffen, wird die Position des Empfangsgerätes im Raum abhängig von den Positionen der Lighthouses mathematisch berechnet. [Buc15]



Abb. 3.1.: HTC Vive Basisstationen [HTC20]

3.2.2. Headset

Das HTC Vive Headset ist ein am Kopf befestigter Bildschirm. Es hat zwei AMOLED 3.6" Bildschirme mit einer Auflösung von 1200×1800 pixel pro Auge und eine Aktualisierungsrate von 90 Hz, wodurch ein Sichtfeld von 110° ermöglicht wird. [Vivnab] Das Headset ist mit einem Gyroskop, einem Accelerometer und einem Laser Positionssensor ausgestattet, um die Rotation des Kopfes zu verfolgen. [Mai15] Um die räumliche Position des Headsets zu berechnen, sind darin diverse Photosensoren eingebaut, die die Infrarotlichtimpulse der Basisstationen empfangen können. [Buc15] Das Headset verfügt über eine Front-Facing Kamera, HDMI, USB 2.0, ein stereo 3.5 mm Kopfhörereingang, Bluetooth und ein integriertes Mikrofon. [Vivnab]



Abb. 3.2.: HTC Vive Headset [HTC20]

3.2.3. Controller

Die während ihrer Entwicklung Knuckles genannten Index Controller wurden von Valve Cooperation zusammen mit einem eigenen Virtual Reality System, inklusive Headset und Basisstationen entwickelt. Sie sind kompatibel mit Basisstationen für SteamVR 1.0 und 2.0, damit auch mit dem HTC Vive System. Anders als die Controller der Vive wurden die Index Controller für spezielle Handbewegungen, wie Objekte greifen, oder werfen, konzipiert. Die Handschlaufen ermöglichen Interaktionen mit offenen Händen, ohne die Gefahr, den Controller fallen zu lassen. Durch 87 Griffkraft- und Bewegungssensoren werden Handposition, Fingerpositionen, Bewegung und Grifffestigkeit verfolgt. Dazu besitzen die Controller A-, B- und Systemtasten, Trigger, Analogstick und eine in der Mitte des Controllers liegende Track-Taste mit Kraftsensor. Nach einer Akkulaufzeit von mehr als 7 Stunden können die Controller über einen USB-C Anschluss wieder aufgeladen werden. [Valna]



Abb. 3.3.: Valve Index Controller [Str18]

3.2.4. Mögliche andere Hardware

Für die Umsetzung des Handtrackings könnte auch auf andere Hardware zurückgegriffen werden. Ein Cyberglove zum Beispiel hat den Vorteil, virtuell existierende Hindernisse durch haptisches und taktiles Feedback zu realisieren. CyberGlove Systems bietet verschiedene Produkte für dieses Anwendungsgebiet an. Der kabellose CyberGlove II benutzt proprietäre, resistive Biegesensor-Technologie, um die Hand- und Fingerbewegungen in Echtzeit in digitale Gelenkwinkeldaten zu verwandeln. Für das taktile Feedback sorgt das CyberTouch II System durch vibrotaktile Stimulatoren an jedem Finger. Ein Exoskelett, wie das CyberGrasp System, setzt das Force-Feedback um. Dadurch ist es möglich, die Masse und Form eines virtuellen Objekts zu spüren. [Sys17] Der wesentliche Nachteil an dem CyberGlove System ist die Anzahl an Hardwarekomponenten, die man benötigt, um haptisches, taktiles und Force-Feedback nutzen zu können. DextaRobotics bietet im Gegensatz zum CyberGlove ein reines Exoskelett (das sogenannte Dexmo) mit Motion Capture und Force-Feedback an. Dieses Dexmo Exoskelett System, ist leicht, kabellos und ist für Anwendungsgebiete, wie Bildung, Gaming, Medizin, Simulation, oder Luft- und Raumfahrt entworfen. [Dex18] Der Preis für das Exoskelett ist sehr hoch und industrieorientiert und damit für den Privatgebrauch ungeeignet. [Ska19]

Eine weitere Möglichkeit für das Hand- und Fingertracking ist ein auf einer Kamera basierender Ansatz, wie die Leap Motion oder die ZED Mini. Der Leap Motion Controller ist ein optisches Handtracking Modul, das die Bewegungen der Hand erfasst und in ein Skelettmodell umsetzt. Der Hersteller verspricht eine geringe erforderliche Rechenleistung, ein weites Sichtfeld und eine Latenz nahe bei null. [ulna] Die ZED Mini ist eine Tiefenkamera, optimiert für erweiterte Realität. Sie besitzt visuelle Trägheitsbewegungsverfolgung und verbesserte Tiefenmessung. Um präzises Bewe-

gungstracking zu gewährleisten, sind in der ZED Mini Gyroskop und Accelerometer integriert. Mit Next-Generation visuell-träger Odometrietechnologie und visuellen Daten können Bewegungen mit erhöhter Genauigkeit erfasst werden. [ZED18] Der Nachteil an kamerabasiertem Handtracking ist das fehlende haptische Feedback. Das von Thalmic Labs entwickelte Myo Armband bietet eine weitere Alternative. Durch Elektromyographiesensoren werden elektronische Impulse, die von den Muskeln bei Bewegung abgegeben werden, erfasst und aufgezeichnet. Dadurch können Gesten erkannt werden. Allerdings ist dieses Armband nicht mehr erhältlich. [Osb18]

3.3. Software

3.3.1. The Human Brain Project und die Neurorobotics Platform

Innerhalb des Human Brain Projects (HBP) wird eine Forschungsinfrastruktur gebaut, um die Bereiche Neurowissenschaften, Medizin und Informatik zu verbessern. Es ist eines der vier größten wissenschaftlichen, von der europäischen Union finanzierten Projekte und hat einen 10 Jahresplan. Das Projekt wurde 2013 gestartet und beschäftigt seit Beginn 500 Wissenschaftler an mehr als 100 Universitäten, Lehrkrankenhäusern und Forschungszentren in Europa. Der Kern des HBPs besteht aus sechs Informations- und Kommunikationstechnikforschungsplattformen. Eine dieser Plattformen ist die Neurorobotics Plattform (NRP). [Pro17b]

Diese ist der Teil des HBPs, in dem virtuelle Gehirnmodelle auf reale oder simulierte Roboter treffen und verknüpft werden. Die NRP erlaubt es Forschern, jedem simulierten Gehirnmodell einen virtuellen oder sogar einen realen Körper zuzuweisen. Dadurch kann ermittelt werden, wie Roboter Bewegungen kontrollieren, wie sie auf Stimulierung reagieren und wie sie in einer virtuellen Umgebung lernen. Die Plattform ist öffentlich und online verfügbar. [Pro17a] Die NRP kann entweder über ein Webinterface oder lokal auf dem eigenen Rechner benutzt werden. [Pla18]

Die Plattform wird unter anderem vom Robot Operating System (ROS) und Gazebo betrieben. ROS ist ein flexibles Framework zur Entwicklung von Roboter Software. Es besteht aus einer Sammlung von Tools, Bibliotheken und Konventionen zur Vereinfachung von Aufgaben für die Schaffung von komplexem und robustem Roboterverhalten über eine Vielfalt von Roboterplattformen. [Robna] Gazebo ist für die Robotersimulation zuständig. Es bietet die Funktionalität, die Gesamtheit der Roboter in komplexen inneren und äußeren Umgebungen zu simulieren. Hierfür stehen eine Physik Engine, hohe Qualitätsgrafik, praktische Programmierung und ein grafisches Interface zu Verfügung. Gazebo ist kostenlos und open-source. [Fou14]

3.3.2. SteamVR Unity Plugin

Valve verwaltet ein Unity Plugin für eine reibungslose Schnittstelle zwischen SteamVR und Unity. SteamVR ist eine Anwendungsprogrammierschnittstelle (API), mit der sich populäre Virtual Reality Headsets verbinden können. Das SteamVR Unity Plugin betreibt dabei drei Hauptmechanismen für Entwickler. Es lädt ein 3D Modell der Controller, handhabt die Eingabe dieser Controller und ermittelt die Darstellung der Hände während der Verwendung. Darüber hinaus wird ein Beispiel für ein Interaktionssystem bereitgestellt, um Entwicklern bei ihren Applikationen zu helfen. Zur Verwendung des Plugins muss SteamVR über die Plattform Steam installiert sein und ausgeführt werden. [Corna]

3.3.3. Unity 3D Client

Unity ist eine Echtzeit 3D Entwicklungsplattform für die Betriebssysteme Windows, Mac und Linux. Mit ihr können im Unity Editor 2D oder 3D Szenen inklusive Animationen und Filmsequenzen entwickelt werden. [Tec20]

Der Unity 3D Client ermöglicht es Entwicklern, einen humanoiden Roboter mit Funktionalität für die NRP zu erweitern. Die von dem Headset, Trackern und Controllern gelieferten Bewegungsdaten werden durch Forward und Inverse Kinematics auf einen lokalen, transparent dargestellten Avatar übertragen. Dadurch können Entwickler das Tracking der Geräte auf Korrektheit evaluieren. Die Positionsdaten des lokalen Avatars werden an die NRP geschickt, die die physikalischen Berechnungen für den virtuellen Roboter übernimmt und seine Bewegungen an die Positionen des lokalen Avatars approximiert. Der Unity 3D Client ist die Schnittstelle für die Server-Client Kommunikation.

3.4. Simulierte Robotik und PID Regler

Die Neurorobotics Plattform übernimmt die physikalischen Berechnungen des virtuellen Roboters und simuliert dadurch ein realistisches Verhalten der Bewegungen des Roboters. Diese Berechnungen basieren auf einem PID Regler, dessen Parameter sich im Unity 3D Client einstellen lassen und die Bewegung der Gelenke des Roboters kontrollieren. Ein PID Regler ist ein closed-Loop Regler, der in diesem Kontext die Ist-Gelenkpositionen an die Soll-Gelenkpositionen approximiert. Der Regler hat drei Parameter, den P-Anteil, den I-Anteil und den D-Anteil, die in Abhängigkeit voneinander stehen und sich auf die Reglerreaktion auswirken. Dabei kann die Gewichtung dieser Parameter konfiguriert werden. Als Eingabewert bekommt der Regler einen Fehlerwert, der aus Stabilitätsgründen entweder verkleinert, oder zumindest konstant

bleiben soll. Der Ausgabewert ist der Reglerwert für die Gelenke, beziehungsweise simulierte Gelenkmotoren, die den Roboter bewegen. [Web19]

Der P-Anteil skaliert proportional mit dem Fehler und trägt zum Großteil der Reglerausgabe bei. Er überbrückt die Lücke zwischen dem Ist- und dem Soll-Wert der Gelenkpositionen. Den P-Anteil kann man allerdings nicht alleine zur Berechnung verwenden, da oft ein gewisser Restfehler besteht, der die Reglerausgabe zu stark beeinflusst. Um diesen Fehler zu minimieren wird der I-Anteil verwendet, der mit dem Integral des Fehlerwertes mit der Zeit wächst. Dadurch sollen Fehlerwerte ausgeglichen werden. Der Nachteil des I-Anteils ist, dass er mit vergangenen Fehlerwerten rechnet, sodass der Kontrollmechanismus zu einer Übersteuerung führen kann, selbst wenn der Ist- und Soll-Wert der Gelenkpositionen übereinstimmen. Der D-Anteil soll dieser Übersteuerung entgegenwirken und skaliert mit der Ableitung des Fehlerwertes mit zeitlichem Bezug. Da er allerdings auf einer Ableitung basiert, kann der D-Anteil gerade bei Rauschen oder Störungen einen hohen Rückgabewert liefern. Deshalb wird die Ableitung vor der finalen Berechnung oft geegnet, oder begrenzt. [Web19]

4. Vorgeschlagener Ansatz

Zur Realisierung des Handtrackings dieser Arbeit wird auf von SteamVR bereitgestellte Skripte zurückgegriffen. Durch das SteamVR Unity Plugin stehen für zwei virtuelle Hände Verhaltensskripte zu Verfügung, die die Benutzereingaben des Valve Index Controllers für die virtuelle Umgebung umsetzen. Da sie allerdings eine bestimmte Objekthierarchie voraussetzen, können sie nicht direkt für den Avatar des Unity 3D Clients verwendet werden. Für das Handtracking werden zwei von SteamVR vorbereitete Hände als Referenzobjekt eingesetzt, um die Gelenke der Hand des Avatars zu approximieren. Die dadurch generierten Gelenkpositions- und Rotationsdaten des lokalen Avatars werden an die Neurorobotics Plattform geschickt, um den entfernten Avatar auf Serverseite steuern zu können. Ein PID-Regler stellt die physikalischen Berechnungen für eine physikalisch korrekte Bewegung des entfernten Avatars sicher, sodass sich dieser wie ein humanoider Roboter verhält. Die Parameter des PID-Reglers müssen dabei noch für die Hände im Unity 3D Client angepasst werden. Zusätzlich muss das Spatial Data File (sdf) für das Handmodell des Avatars mit den Fingergelenken der Hand erweitert werden.

5. Implementierung

In diesem Kapitel wird detailliert die Implementierung für das Hand- und Fingertracking beschrieben. Dabei wird näher auf das SteamVR Input System, das Tracking und die Server-Client Kommunikation eingegangen. Abschließend wird die Umgebung für die Nutzerstudie beschrieben.

5.1. SteamVR Input System

Beim Handtracking stützt sich diese Arbeit auf das SteamVR Input System des SteamVR Unity Plugins. Dieses System übernimmt die gerätespezifischen Teile des Programmcodes, sodass man sich bei der Entwicklung komplett auf die Aktionen konzentrieren kann (Aktionen sind der wesentliche Bestandteil des Systems). Darunter fallen unter anderem neben Boolean und Vector3 Aktionen auch die für die Implementierung wichtige Skeleton Aktion. Diese verwendet den SteamVR Skeleton Input, um eine genaue Finger- ausrichtung zu berechnen, während die Controller verwendet werden. Dazu gehört das SteamVR_Behaviour_Skeleton Skript, eine wichtige Komponente für das Handtracking in Unity. Dieses Skript handhabt die vom Controller gesammelten Handdaten und setzt sie für ein virtuelles Handmodell um, sodass die Hand und einzelne Finger getrackt werden. Das Skript verlangt lediglich eine Skelettwurzel des Handmodells. Allerdings muss das Handmodell eine bestimmte Knochenhierarchie aufweisen, um verwendet werden zu können. Dies ist bei dem Handmodell des Avatars des Unity 3D Clients jedoch nicht unmittelbar möglich, sodass das Skript nicht direkt auf dieses Handmodell angewendet werden kann. Es bedarf deshalb eines Zwischenschrittes, der im folgenden Abschnitt beschrieben wird. Eine umfangreiche Dokumentation zum SteamVR Unity Plugin findet sich online. [Corna]

5.2. Implementierung des Handtrackings in Unity

Um das Handtracking im Unity 3D Client zu implementieren, werden zuerst vom SteamVR Unity Plugin schon existierende Handmodelle zusammen mit dem SteamVR Behaviour Skeleton Skript übernommen. Diese Handmodelle verfügen durch das Skript über präzises Hand- und Fingertracking.

Die Herausforderung bei der Implementierung bestand darin, die Positionen und Rotationen der einzelnen Fingergelenke des Handmodells aus dem Plugin auf die Fingergelenke des Avatars des Unity 3D Clients zu übertragen und anzuwenden.

Im Rahmen dieser Arbeit wurde folgendes implementiert:

Zuständig für die Übertragung der Positions- und Rotationsdaten ist ein leeres Gameobjekt namens FKManager, das das TrackingFKManager Skript beinhaltet. Am Anfang dieses Skripts werden Variablen für den TrackingIKTargetManager, der durch inverse Kinematik für das Tracking des Körpers zuständig ist, definiert. Zusätzlich werden die Fingergelenke der Hand des Plugins und die Ziele für die Handgelenke des Avatars des Unity 3D Clients in weiteren Variablen festgelegt. Pro Finger existieren vier Gelenke, drei für die Finger und ein Endgelenk an der Fingerkuppe, die übertragen werden müssen. Während für die Fingergelenke der Hand aus dem Plugin nur die Transform-Komponente benötigt wird, werden für die Fingergelenke des Avatars neue Gameobjekte initialisiert. In der Start-Methode werden die Transform-Komponenten der Hand aus dem Plugin den entsprechenden Variablen zugewiesen und eine Methode, "Setup-Targets" aufgerufen. Diese wiederum ruft für jedes Fingergelenk eine entsprechende Setupmethode auf, die ein neues Gameobjekt mit entsprechendem Fingergelenknamen erzeugt und sowohl die Position, als auch die Rotation des dazugehörigen Fingergelenks der Hand aus dem Plugin übernimmt. Der folgende Code-Auszug soll dies veranschaulichen.

```
private void SetupTargetIndex1(Transform trackingTarget)
{
    TargetIndex1 = new GameObject("Left Target Index1");
    TargetIndex1.transform.parent = trackingTarget;
    TargetIndex1.transform.localPosition = trackingTarget.position;
    TargetIndex1.transform.localRotation = trackingTarget.rotation;

    TargetIndex1.transform.SetPositionAndRotation(
        virtualIndex1.transform.position, virtualIndex1.transform.rotation);
}
```

Als nächstes werden in der Update-Methode des Skripts die Position und Rotation der eben erstellten Gameobjekte, der sogenannten Targets, pro Frame mit der SetPositionAndRotation-Methode und der Position und Rotation der Fingergelenke der Hand aus dem Plugin aktualisiert, wenn der trackingIKTargetManager bereit ist. Das heißt, sobald alle Targets initialisiert wurden. Get-Methoden geben die Transform-Komponente der Targets zurück.

Im UserAvatarIKControl Skript repräsentiert eine Variable den trackingFKManager. In der Startmethode dieses Skripts werden für jedes Fingergelenk die dazugehörigen

Fingergelenke des Modells zugewiesen. Diese werden in einer LateUpdate-Methode um die Fingerpositionen der Targets aktualisiert.

```
private void LateUpdate()
{
    getFingerTargetLeft();
    getFingerTargetRight();

    updateFingerTargetLeft();
    updateFingerTargetRight();
}

private void updateFingerTargetLeft()
{
    [...]
    lModelIndex1.transform.rotation = leftIndex1.rotation;
    [...]
}
```

Im Anschluss rufen die beiden getFingerTarget-Methoden beim trackingFKManager Skript die Get-Methoden für jedes Fingergelenk auf und weisen sie einer Variablen zu. Die updateFingerTarget-Methoden weisen dem entsprechenden Modellfingergelenk die passende Rotation des Targets zu. Bei diesem Vorgang reicht es aus, nur die Rotation zu übergeben. Das hat den wichtigen Effekt, dass die Finger der Hand an dem Handmodell genauer dargestellt werden, als wenn die Transform-Komponente mit übergeben würde. Denn bei der Übergabe der Transform-Komponenten würden die Finger an andere Positionen der Hand verschoben werden, als die für die Finger vom Modell vorgesehenen Positionen.

Um die Hände in der korrekten räumlichen Position, wie die der realen Hände darzustellen, wurden in dem TrackingIKTargetManager Skript die Setup-Methoden der Hände leicht abgeändert. Es wurden analog wie im trackingFKManager für die Fingergelenke Targets für die Hände erstellt. Die Handgelenke der Handmodelle des SteamVR Unity Plugins werden als Referenz für die Rotation und Position verwendet. Die Rotation wird um eine Verschiebung um 90 Grad an der Z-Achse angepasst und die Positionen leicht korrigiert. Damit werden die Hände korrekt dargestellt.

5.3. Server-Client Kommunikation

Zuerst musste die Modelldatei auf einem separaten Rechner, der nur für den Server zuständig ist, um die Fingergelenke erweitert werden. Hierfür wurden in der model.sdf

Datei zuerst Links für jedes Gelenk definiert. Diese Links beinhalten Informationen über den Namen, Position, Darstellung und physikalische Eigenschaften des Gelenks, wie Masse, Kollision und Gravitation. Ein Fingergelenk hat den Massenwert von 0,05. Dazu wurden sowohl für die jeweilige Hand, als auch für die ersten Daumengelenke zwei sogenannte JointLinks definiert (unsichtbare Objekte, die für Rotation in drei Koordinatenachsen zuständig sind). Diese Links werden in Joints verbunden. Wichtig ist die Reihenfolge, in der sie verbunden werden. Die JointLinks für die Daumengelenke werden zwischen Handgelenk, Jointlink und Daumengelenk eingesetzt. Damit die Gelenke sich bewegen können, wird ihr Typ auf "revolute" gesetzt. Der effort-Wert legt die maximale Kraft fest, die von diesem Gelenk ausgeübt werden kann. Denn ohne diesen Wert an den Fingergelenken kommt es zu einem Rückschlag (Kickback), sobald man die Faust ballt, oder sie aus der Faust heraus wieder streckt. Der folgende Codeauszug ist ein Beispiel für die Definition eines Joints aus der model.sdf.

```
<joint name="mixamorig_LeftHandIndex1" type="revolute">
  <parent>mixamorig_LeftHand</parent>
  <child>mixamorig_LeftHandIndex1</child>
  <axis>
    <xyz>0.0 1.0 0.0</xyz>
    <limit>
      <lower>-1.5707963267948966</lower>
      <upper>1.5707963267948966</upper>
      <effort>200.0</effort>
    </limit>
  </axis>
</joint>
```

Der Unity 3D Client veröffentlicht nun für jeden Link die passenden Positionen und Rotationen an den Server. Das RigAngleTracker Skript erstellt für jedes Target ein Topic, welches im UserAvatarService Skript an den Server geschickt wird. Dazu werden am Anfang des RigAngleTracker Skripts Variablen für die Namen der Gelenke des Modells definiert. Zusätzlich wird ein Array erstellt, das die Namen der Hand- und Fingergelenke beinhaltet, um später für diese Gelenke andere PID-Parameter, als die für den Körper des Modells an den Server schicken zu können. Wichtig sind die Methoden CreateMappingForRemoteAvatar und CreateMappingForLocalAvatar. Mittels dieser Methoden wird ein JointMapping für den Server erzeugt. Das JointMapping besteht aus vier Komponenten: Zwei Transform-Komponenten, dem Elternobjekt und dem Kindobjekt, einer booleschen Variable, die angibt, ob die Rotation invertiert werden soll und dem MappedEulerAngle, der Achse, um die rotiert werden soll. Es folgt ein zusammengefasster Auszug dieses Codes aus dem RigAngleTracker Skript für die

Methoden für das erste Daumengelenk und das Zeigefingergelenk.

```
private void CreateMapingForRemoteAvatar()
{
    const string L_HAND_THUMB1_JLINK1_NAME =
        "mixamorig_LeftHandThumb1_JointLink1";
    const string L_HAND_THUMB1_JLINK2_NAME =
        "mixamorig_LeftHandThumb1_JointLink2";

    Transform leftThumb1 =
        FindChildTransformRecursive(transform, L_HAND_THUMB1_NAME);
    Transform leftThumb1_jlink1 =
        FindChildTransformRecursive(transform, L_HAND_THUMB1_JLINK1_NAME);
    Transform leftThumb1_jlink2 =
        FindChildTransformRecursive(transform, L_HAND_THUMB1_JLINK2_NAME);
    Transform leftIndex1 =
        FindChildTransformRecursive(transform, L_HAND_INDEX1_NAME);

    _jointMappings[L_HAND_THUMB1_NAME + "_x"] =
        new JointMapping(leftThumb1_jlink1, leftThumb1_jlink2,
            true, MappedEulerAngle.X);
    _jointMappings[L_HAND_THUMB1_NAME + "_y"] =
        new JointMapping(leftThumb1_jlink2, leftThumb1,
            true, MappedEulerAngle.Y);
    _jointMappings[L_HAND_THUMB1_NAME + "_z"] =
        new JointMapping(leftHand, leftThumb1_jlink1,
            true, MappedEulerAngle.InvertedZ);

    _jointMappings[L_HAND_INDEX1_NAME] =
        new JointMapping(leftHand, leftIndex1, true, MappedEulerAngle.Y);
}
```

Anhand dieses Codes wird deutlich, wie die Rotation in mehreren Achsen umgesetzt wird.

In der `model.sdf` Datei werden `JointLinks` definiert, auf die im obigen Codeauszug zugegriffen wird. Sie verbinden die Hand mit dem ersten Daumengelenk und greifen auf unterschiedliche Achsen zurück. Dadurch lässt sich der Daumen in alle drei Achsen bewegen. Bei den Zeigefingergelenken ist dies nicht notwendig, deshalb wird das `JointMapping` von der Hand direkt an das erste Gelenk gebunden und im Array gespeichert.

Die `CreateMappingForLocalAvatar`-Methode ist ähnlich zu der Methode für den Remote Avatar, mit dem Unterschied, dass die `JointLinks` nicht mehr erzeugt werden müssen, da in Unity Rotationen um mehrere Achsen möglich sind. Des Weiteren wird die boolsche Variable für die Invertierung der Rotation auf "false" gesetzt, da Unity andere Koordinatenachsenorientierungen hat. Zuletzt wird bei den Achsen für das erste Daumengelenk die X- und Z-Achse vertauscht, ebenfalls aufgrund der Achsenorientierung von Unity. Die Z-Achse wird invertiert.

Nun werden im `UserAvatarService` Skript in der `Update`-Methode in einer im Programm festgelegten Frequenz die Targets in Topics zum Server mit den PID-Parametern geschickt. Hierfür wurde die `PublishJointPIDParams`-Methode angepasst. In einer If-Abfrage unterscheidet die Methode zwischen Hand- beziehungsweise Fingergelenken und den restlichen Joints des Modells. Für die Hand- und Fingergelenke werden andere PID-Parameter an den Server geschickt, als die für den Körper des Modells. Diese betragen für den P-Anteil 1000, für den I-Anteil 300 und den D-Anteil 500. Sie wurden durch Versuche ermittelt. Im folgenden UML-Diagramm werden die Beziehungen zwischen den Klassen verdeutlicht. In diesem UML-Diagramm wurden die vordefinierten Objekte gelb markiert, die selbst entwickelten Klasse grün und die angepassten Klassen blau.

5. Implementierung

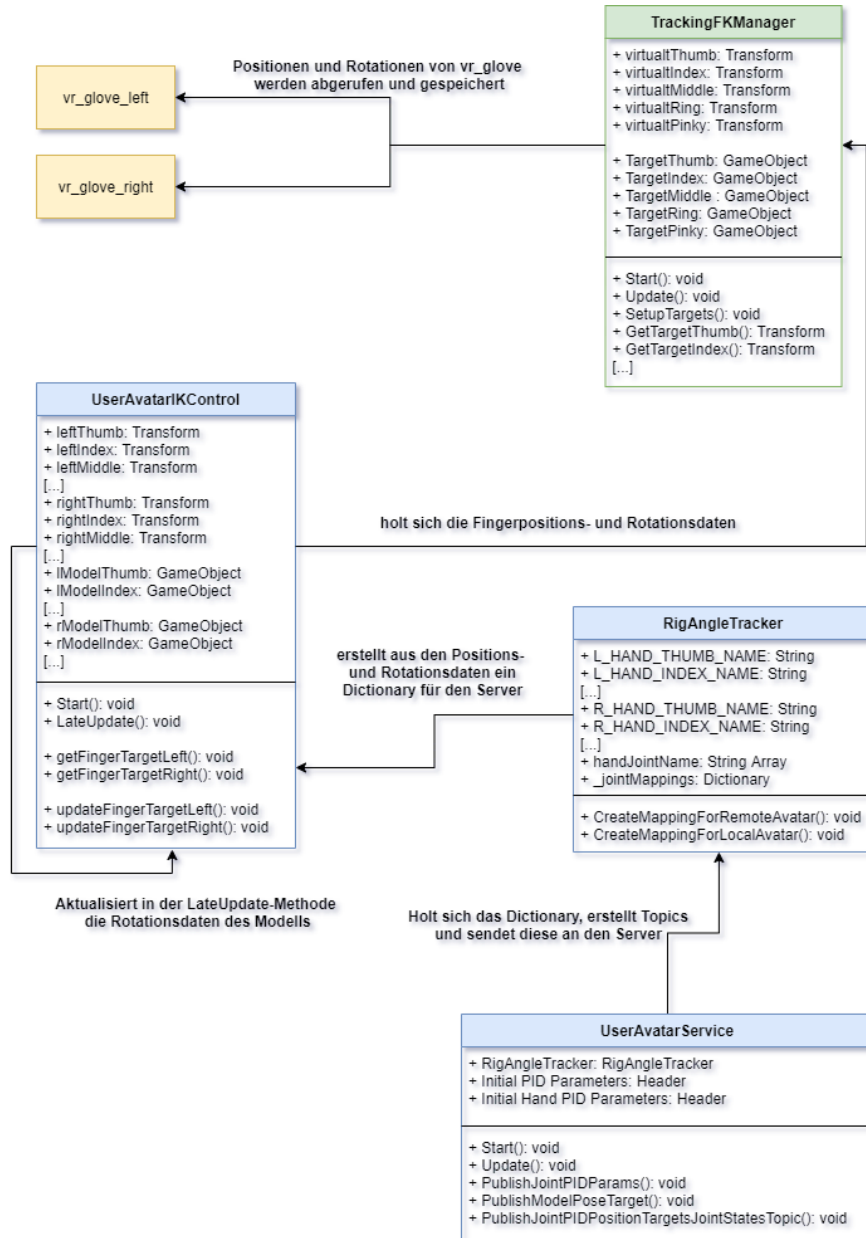


Abb. 5.1.: UML-Diagramm zur Veranschaulichung der Zusammenarbeit der Klassen

5.4. Implementierung des haptischen Feedbacks

Zur Vermittlung eines deutlicheren Gefühls, wann ein Objekt gegriffen wird, wurde haptisches Feedback in Form von Vibration der Controller implementiert. Es wurde ein leeres Gameobjekt namens `VibrateControllers` mit zwei Kindobjekten namens `VibrateLeft` und `VibrateRight` erstellt. Der Ansatz war, die Controller zum vibrieren zu bringen, sobald eine Kollision zwischen den Finger- und Handoberflächen der Hände des remote Avatars und eines Objekts stattfindet. In der Nutzerstudie werden fünf Würfel bereitgestellt. Bei vier von den Würfeln sollen die Controller vibrieren, sobald diese berührt werden, also eine Kollision zwischen Händen und Würfel stattfindet. Der erste Schritt zur Umsetzung dieses Ansatzes ist, die nötigen Gameobjekte mit Collidern zu versehen. Hierfür wurde ein Skript `AssignMeshCollider` geschrieben, das zur Laufzeit einen Meshcollider zu den Fingeroberflächen der Hände des remote Avatars hinzufügt. Zusätzlich wurden die Würfel mit `BoxCollider`, `Rigidbody` und einem weiteren Skript `CollisionHandler` versehen. Die `Rigidbody`-Komponente ist notwendig, da sonst eine entsprechende Methode im Skript `CollisionHandler` nicht aufgerufen wird. Dies geschieht in der `Update`-Methode, allerdings nur, wenn die boolesche Variable `initialized` des Skripts auf `"false"` gesetzt ist und der `UserAvatarService` den remote Avatar gespawnt hat. Sobald die Variable `initialized` auf `"true"` gesetzt ist, werden die Positionen und Rotationen des Würfels mit den Positionen und Rotationen des Würfels, der vom Server in den Unity 3D Client geladen wird, aktualisiert. Dies ist notwendig, da sonst die Collider des Würfels sich nach physikalischer Berechnung vom Unity 3D Client bewegen und aber nicht so, wie die Würfel auf Serverseite. Aus diesem Grund wird auch die Gravitation in Unity deaktiviert. Zu Referenzzwecken bildet der folgende Codeabschnitt einen Teil der `Update`-Methode ab.

```
void Update () {
    if (initialized && UserAvatarService.Instance.IsRemoteAvatarPresent)
    {
        lHand_vib = GameObject.Find(lHand)
        lHand_vib.AddComponent<MeshCollider>();

        box1 = GameObject.Find(
            "box_0_0::link::collision_COLLISION_VISUAL_/Cube");
        box1reference = GameObject.Find("box_0_0").transform;

        string ScriptName = "CollisionHandler";
        System.Type MyScriptType = System.Type.GetType(ScriptName +
            ", Assembly-CSharp");
    }
}
```



```
        box1.AddComponent(MyScriptType);
        box1.AddComponent<BoxCollider>();
        box1.AddComponent<Rigidbody>();
        box1.GetComponent<Rigidbody>().useGravity = false;

        initialized = true;
    }
    if (initialized)
    {
        box1.transform.SetPositionAndRotation(box1reference.position,
                                              box1reference.rotation);
    }
}
```

Nun befindet sich das CollisionHandler Skript auf jedem Würfel, der die Controller zum vibrieren bringen soll. Zu Beginn des Skripts werden Strings mit Namen der Finger- beziehungsweise Handoberflächenobjekte der Hände des remote Avatars deklariert. Zusätzlich werden drei Stringarrays instanziiert, die in der Start-Methode gefüllt werden. Das Stringarray `collisionNames` enthält alle Namen, das Stringarray `collisionNamesLeft` nur die Namen der Fingeroberflächenobjekte der linken Hand und das Stringarray `collisionNamesRight` die der rechten Hand. Die zwei wesentlichen Methoden des Skripts sind die `OnCollisionEnter`- und `OnCollisionExit`-Methode. Erstere Methode wird aufgerufen, sobald ein Collider den Collider des Würfels berührt. Die zweite Methode wird aufgerufen, sobald diese Kollision aufgehoben wird. Die `OnCollisionEnter` Methode sucht im `collisionNames`-Array nach dem Objektname des Objekts, zwischen dem und dem Würfel es zur Kollision kommt. Wird ein passender Name gefunden, wird eine boolesche Variable `vibrate` auf "true" gesetzt und der Name des zu kollidierenden Objekts im String `activeCollision` gespeichert. Anschließend wird in den beiden anderen Arrays nacheinander gesucht, ob das kollidierende Objekt zu der linken oder rechten Hand gehört. Das Resultat wird in einem weiteren String `side` unter `LeftHand` oder `RightHand` gespeichert. Die `OnCollisionExit`-Methode setzt die Variable `vibrate` auf "false", sobald die Kollision nicht mehr stattfindet.

```
public void OnCollisionEnter(Collision collision)
{
    for (int i = 0; i < collisionNames.Length; i++)
    {
        if (collision.gameObject.name == collisionNames[i])
        {
            vibrate = true;
        }
    }
}
```

```
        activeCollision = collision.gameObject.Name;
        for (int j = 0; j < collisionNamesLeft.Length; j++)
        {
            if (activeCollision == collisionNamesLeft[j])
            {
                side = "LeftHand";
            }
        }
        for (int j = 0; j < collisionNamesRight.Length; j++)
        {
            if (activeCollision == collisionNamesRight[j])
            {
                side = "RightHand";
            }
        }
    }
}

public void OnCollisionExit(Collision collision)
{
    if (collision.gameObject.name == activeCollision)
    {
        vibrate = false;
    }
}
```

Das HapticsHand Skript sorgt schließlich dafür, dass die Controller vibrieren. In der Update-Methode werden Referenzen der Würfel angelegt und in ein Array geschrieben, wenn der remote Avatar gespawnt ist und die dem Skript zugehörige boolsche Variable initialized auf "false" gesetzt ist. Wenn nun die Variable initialized dieses Skripts und des AssignMeshCollider Skripts auf "true" gesetzt ist, wird in einer For-Schleife überprüft, bei welchen der Würfel eine Kollision stattgefunden hat und welche Hand involviert war, die rechte oder die linke Hand. Je nachdem, welche Hand eine Kollision mit einem der Würfel ausgelöst hat, beginnt der entsprechende Controller in der linken oder rechten Hand zu vibrieren. Dies erfolgt durch eine von dem SteamVR Unity Plugin bereitgestellte Aktion. Allerdings muss zu Beginn des Skripts die Eingabequelle definiert werden. In diesem Fall ist es der Controller in der linken oder der rechten Hand. Zusätzlich dazu wird die Vibrationsaktion deklariert. Um diese Aktion nutzen

zu können, wird die Methode `TriggerHapticPulse` verwendet, die drei Floatwerte als Eingabeparameter verlangt: Die Dauer, die Frequenz und die Amplitude der Vibration. Diese Methode ruft für die Aktion eine vom SteamVR Unity Plugin bereitgestellte Methode `Execute` auf, die die Parameter der `TriggerHapticPulse`-Methode übernimmt und zusätzlich die Eingabequelle festlegt, bei der die Vibration stattfinden soll. Die `Execute`-Methode besitzt auch einen weiteren Parameter, um die Vibration zeitverzögert zu starten. Dieser Parameter wurde allerdings auf 0 gesetzt, da die Vibration sofort bei Kollision eintreten soll. Die Werte für die Dauer, Frequenz und Amplitude wurden so angepasst, dass die Vibration für den Nutzer nicht als störend wahrgenommen wird, aber ihm trotzdem ein haptisches Feedback vermittelt, sobald ein Würfel gegriffen wird. Das Skript ist als Komponente an den Kindobjekten `VibrateLeft` und `VibrateRight` angehängt.

```
public SteamVR_Input_Sources handType;
public SteamVR_Action_Vibration hapticAction =
SteamVR_Input.GetAction<SteamVR_Action_Vibration>("Haptic");
[SerializeField] public AssignMeshCollider assign;

void Update () {
    if (UserAvatarService.Instance.IsRemoteAvatarPresent && !initialized)
    {
        box1 = GameObject.Find(
            "box_0_0::link::collision_COLLISION_VISUAL_/Cube");
        [...]
        boxArray = new GameObject[]{box1, box2, box3, box4};

        initialized = true;
    }
    if (initialized && assign.GetInitialized())
    {
        for (int i = 0; i < boxArray.Length; i++)
        {
            if (boxArray[i].GetComponent<CollisionHandler>().GetVibrate())
            {
                if (this.handType.ToString() ==
                    boxArray[i].GetComponent<CollisionHandler>().GetSide())
                {
                    TriggerHapticPulse(0.2f, 10f, 30f);
                    break;
                }
            }
        }
    }
}
```

```
        }
    }
}

public void TriggerHapticPulse (float duration, float frequency,
                                float amplitude)
{
    hapticAction.Execute(0, duration, frequency, amplitude, handType);
}
```

5.5. Aufbau der Umgebung für eine spezifische Nutzerstudie

Auf Client Seite wurden Anpassungen am Avatar vorgenommen: Ein neues Gameobjekt, `RemoteAvatarVisuals`, das das Skript `RemoteAvatarVisuals` enthält, sorgt dafür, dass der Torso des Remote Avatars (das Avatarmodell des Servers) transparent erscheint, damit die Sicht für den Benutzer in der Nutzerstudie nicht blockiert wird, wenn sich der Avatar vor die Kamera schiebt. In diesem Skript werden Hüften- und Wirbelsäulengameobjekte, also der gesamte Torso identifiziert und deren Material in der Rendererkomponente durch eine Undurchsichtigkeitsvariable von 0.1 transparent gemacht. Sobald der Remote Avatar in dem `UserAvatarService` Skript erzeugt wird, wird das `RemoteAvatarVisuals` Skript aufgerufen und der Torso des Avatars erscheint transparent.

Die Hand des lokalen Avatars reagiert schneller, als die des Remote Avatars, da Latenz auf Serverseite besteht, wodurch die Testpersonen versuchen könnten, mit der lokalen Avatarhand die Studie durchzuführen. Allerdings sind Physikberechnungen auf der Serverseite nur für den Remote Avatar implementiert. Deshalb wurde eine Möglichkeit geschaffen, den Avatar nahezu unsichtbar zu machen, indem im `UserAvatarVisuals` Skript eine weitere Anpassung für die Nutzerstudie vorgenommen wurde. In der Update-Methode wird überprüft, ob die m- oder die n-Taste der Tastatur gedrückt wird. Die Betätigung der m-Taste setzt die Undurchsichtigkeitsvariable des Skripts auf 0, wodurch der lokale Avatar nahezu unsichtbar wird. Hätte man alternativ den Renderer abgeschaltet, um den lokalen Avatar komplett unsichtbar zu machen, hätte dies das Hand- und Fingertracking außer Kraft gesetzt. Das Drücken der n-Taste setzt den Wert der Variablen wieder auf 0.5, wodurch der lokale Avatar wieder transparent und dadurch sichtbar wird.

5. Implementierung

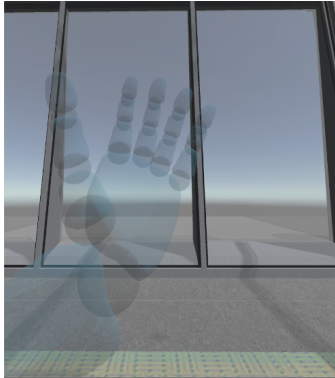


Abb. 5.2.: Transparenter lokaler Avatar

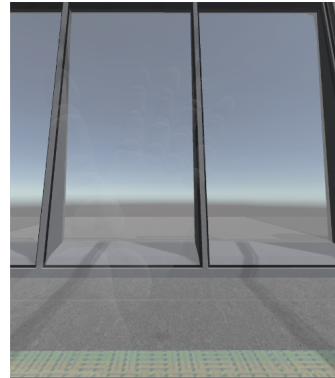


Abb. 5.3.: Unsichtbarer lokaler Avatar

Für die Nutzerstudie wurde in der Neurorobotics Platform ein neues Experiment angelegt. Dafür wurde das Template eines kleinen Raumes verwendet und dort vorhandene, aber für die Nutzerstudie unnötige Objekte in der Mitte des Raumes entfernt. Mit dem Environment Editor wurde stattdessen ein Holztisch in den Raum eingefügt. Auf diesem Tisch wurden fünf Würfel platziert. Zwei rote, zwei blaue und ein grauer statischer Würfel, der sich nicht bewegen lässt. Zusätzlich wurde die Größe aller Würfel in der SDF-Datei angepasst, sowie Gewicht und Reibung der Oberfläche. Dadurch lassen sich die Würfel besser anheben, ohne sofort aus der virtuellen Hand zu rutschen. Der graue Würfel wurde etwas größer dimensioniert, als die anderen.

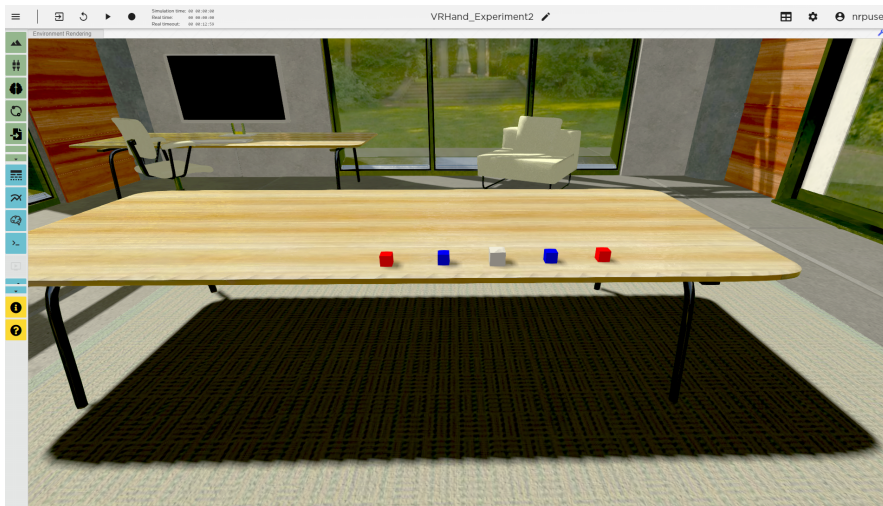


Abb. 5.4.: Neurorobotics Platform Experiment für die Nutzerstudie

6. Nutzerstudie und Ergebnisse

In diesem Kapitel wird der Aufbau, der Ablauf, die Auswertung und damit die Ergebnisse der entwickelten Nutzerstudie dargestellt. Für die Auswertung wurde ein begrenztes Maß an Fragen ausgewählt. Die komplette Nutzerstudie befindet sich im Anhang.

6.1. Aufbau und Durchführung der Nutzerstudie

Vorausgeschickt wird, dass die Testpersonen für diese Nutzerstudie anonym mitgearbeitet haben. Die Nutzerstudie ist aus fünf demografischen, zehn System Usability Scale (SUS) und sieben speziellen Fragen aufgebaut. Die demografischen Fragen dokumentieren allgemeine Informationen über die Testperson, wie Alter, Geschlecht, dem derzeitigen Ausbildungsabschnitt, Erfahrung mit virtueller Realität und Erfahrung mit dem Index Controller. Die zehn SUS Fragen wurden aus der Arbeit von John Brook entnommen und für die Anwendung in der Nutzerstudie angepasst. [Bro96] Die sieben speziellen Fragen sollen auf die Wiederverkörperung testen, die während des Experiments auftreten soll und wurden teils selbst entwickelt, teils aus der bereits erwähnten Arbeit von Ye Yuan und Anthony Steed abgeleitet, die sich mit dem Thema der Gummihand Illusion in der virtuellen Realität auseinandersetzt. [YS10]

Für die Nutzerstudie wurde eine ungefähre Testzeit von 30 Minuten angesetzt, damit die Testpersonen die einzelnen Phasen des Experiments komplett und ohne zeitlichen Druck durchführen und den anschließenden Fragebogen beantworten konnten. Das Experiment bestand aus sechs Phasen. In der ersten Phase sollten sich die Testpersonen mit dem Index Controller und dem Handtracking vertraut machen, indem jeder virtuelle Finger einzeln bewegt werden und Gesten ausgeführt werden sollten, zum Beispiel die Finger zu einer Faust zu krümmen, die Hand wieder zu öffnen und nur Zeige- und kleinen Finger auszustrecken. In der zweiten Phase des Experiments sollte in der virtuellen Testumgebung ein Würfel vom Tisch angehoben werden. Diese beiden Phasen waren als Eingewöhnungsphase gedacht. In der dritten Phase sollte nun ein Würfel angehoben und auf dem bereits beschriebenen grauen Würfel auf dem Tisch platziert werden. Die vierte Phase bestand darin, einen Würfel mit nur drei Fingern anzuheben und diesen auf dem grauen Würfel zu platzieren. In der fünften Phase sollte man die selbe Aufgabe nur mit zwei Fingern durchführen. In der abschließenden

sechsten Phase sollte ein Würfel auf reale Kopfhöhe gehoben und losgelassen werden, um ihn mit der anderen Hand aufzufangen. Für die Phasen drei bis sechs wurde die benötigte Zeit für die Lösung der Aufgaben gestoppt und notiert. Des Weiteren wurde die Simulationszeit und die reale Zeit des Experiments auch notiert. Die Simulationszeit ist die Zeit, die in der Simulation der Neurorobotics Plattform gezählt wird. Diese verläuft langsamer als die reale Zeit, da der Server für Berechnungen der Physik und Grafik Zeit benötigt. Anschließend wurde der Fragebogen, der mit Google Formulare erstellt wurde, von den Testpersonen ausgefüllt.

6.2. Auswertung

An der Nutzerstudie haben insgesamt 30 Personen teilgenommen, davon waren 10 Teilnehmer weiblich, 20 männlich. Die meisten Teilnehmer waren zwischen 24 und 27 Jahren alt. Die Testpersonen befanden sich in unterschiedlichen Ausbildungsabschnitten, wobei die Meisten Bachelor und Master Studenten waren. 20 der Testpersonen hatten bereits Erfahrung mit virtueller Realität, aber nur 5 der Testpersonen hatten Erfahrung mit dem Index Controller. Das folgende Diagramm veranschaulicht die Verteilung der Testpersonen auf die Ausbildungsabschnitte. Dabei ist zu erkennen, dass nicht nur Studenten, die mit der Thematik vertraut sind, an der Nutzerstudie teilgenommen haben. Damit wurde ein allgemeiner Querschnitt erreicht.

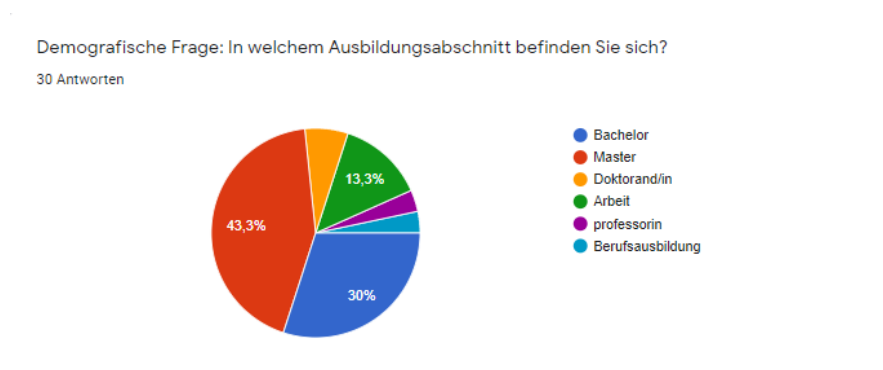


Abb. 6.1.: Diagramm zum Ausbildungsabschnitt der Teilnehmer

Die Antworten für die SUS Fragen fielen sehr unterschiedlich aus. Aber ein Trend zur einfachen Nutzung der Index Controller ist erkennbar. Auch wurde das Hand- und Fingertracking, sowie das haptische Feedback in Form von Vibration der Controller als gut integriert empfunden, wobei es bei der Frage nach Inkonsistenzen im Handtracking fast eine Gleichverteilung zwischen positivem und negativem Feedback gab. Die folgenden Diagramme eben genannter SUS Fragensauswertungen veranschaulichen

6. Nutzerstudie und Ergebnisse

dies. Für die SUS Fragen wurde eine Bewertungsskala von 1 bis 5 verwendet, mit der die Testpersonen ihre Antworten auf die Fragen klassifizieren sollten. 1 bedeutete, gar nicht zuzustimmen, 5 bedeutete, sehr zuzustimmen.

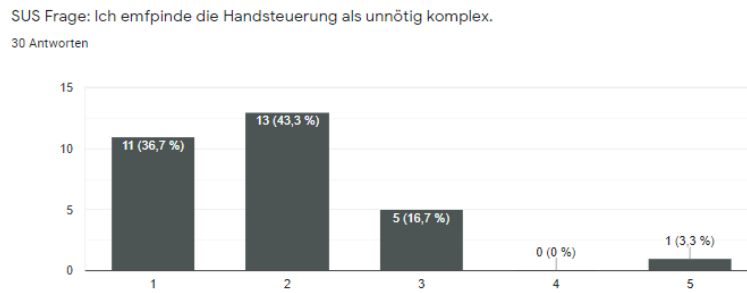


Abb. 6.2.: Diagramm zur SUS Frage nach der Komplexität der Bedienung



Abb. 6.3.: Diagramm zur SUS Frage nach der Schwierigkeit der Bedienung

6. Nutzerstudie und Ergebnisse

SUS Frage: Ich empfinde die Bedienung als sehr umständlich.

30 Antworten

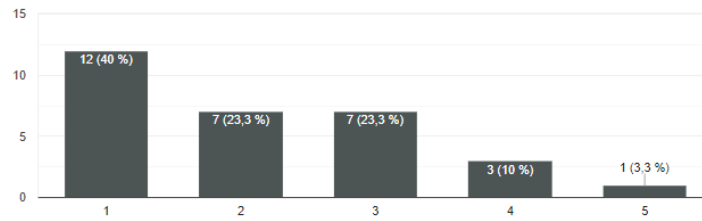


Abb. 6.4.: Diagramm zur SUS Frage nach der Bedienung

SUS Frage: Ich finde, dass die verschiedenen Funktionen der Handsteuerung (Handtracking und haptisches Feedback) gut integriert sind.

30 Antworten

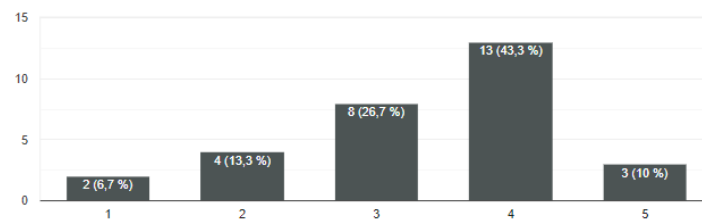


Abb. 6.5.: Diagramm zur SUS Frage nach der Integration

SUS Frage: Ich finde, dass es bei der Handsteuerung zu viele Inkonsistenzen gibt.

30 Antworten

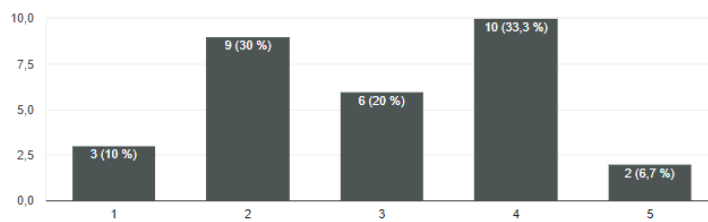


Abb. 6.6.: Diagramm zur SUS Frage nach den Inkonsistenzen

Den Diagrammen ist zu entnehmen, dass die Handsteuerung relativ intuitiv verwendbar ist. Allerdings haben sich in dieser Versuchsanordnung Inkonsistenzen gezeigt, wie etwa bei der räumlichen Verschiebung von realer und virtueller Hand und der zeitlichen Korrelation zwischen der Bewegung des virtuellen Avatars und des Probanden.

Für jeden der 30 Probanden wurde für die SUS Antworten der individuelle SUS Punktestand berechnet und ein Durchschnitt ermittelt. Dieses Ergebnis des SUS Punktestandes beträgt gerundet 63,1667. Dem folgenden Diagramm aus dem Artikel "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale" von Bangor et al. [BKM09] ist zu entnehmen, dass der ermittelte Punktestand im Grenzbereich der Verwendbarkeit liegt. Das bedeutet für das System, dass es im Hinblick auf die Nutzbarkeit in einem größeren Rahmen ausgebaut werden muss.

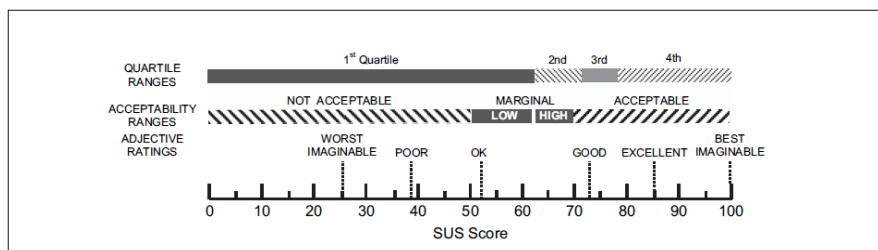


Abb. 6.7.: Diagramm zur Auswertung des SUS Punktestands [BKM09]

Die meisten Probanden gaben mündliches Feedback zur Nutzerstudie ab. Neben der räumlichen Verschiebung von virtuellem und realem Arm bemängelten sie die Trägheit der Würfel und eine als zu gering wahrgenommene Reibung zwischen Würfeln und Fingern. Beide Faktoren erschwerten das Anheben der Würfel. Die Latenz des Servers, die zu einer zeitlichen Korrelation zwischen den Bewegungen der realen Hand und dem virtuellen Avatar geführt hat, wurde allerdings hingenommen und die Probanden passten ihre Bewegungen an diese Latenz an. Die räumliche Verschiebung behinderte die Probanden bei der Ausführung der unterschiedlichen Phasen des Experiments geringfügig. Sie störte eher das Gefühl der Wiederverkörperung, also anstatt der virtuellen Hände die realen Hände zu steuern.

Auf die zu beantwortende, spezielle Frage, welche Phase mit der dazugehörigen Aufgabe des Experiments am einfachsten durchzuführen war, wählten 12 Probanden, also die Mehrheit der Testpersonen Phase 2. Wie bereits erwähnt bestand die Aufgabe darin, einen der Würfel mit beliebig vielen Fingern anzuheben. Bei den anderen Teilnehmern der Studie gab es eine gleiche Verteilung auf die Phasen 4 und 6 mit fünf Probanden und auf die Phasen 3 und 5 mit jeweils vier Probanden. Bei der speziellen Frage, bei welcher Phase des Experiments das Gefühl, seine eigenen Hände zu steuern, am

höchsten war, stimmten 8 Probanden für Phase 5, die Phase mit der Aufgabe, einen der Würfel mit zwei Fingern anzuheben und ihn auf dem grauen Würfel zu platzieren. Die restlichen Probanden teilten sich auf die anderen Phasen auf. Dies könnte so bewertet werden, dass man für diese fünfte Phase des Experiments eine höhere Konzentration aufbringen und Feinmotorik mit den Fingern ausführen muss, um den Würfel mit zwei virtuellen Fingern anzuheben. Durch diese beiden Faktoren verstärkt sich das Gefühl, die eigenen Hände anstatt der virtuellen zu steuern. Für diese Phase haben nur 8 der insgesamt 30 Probanden gestimmt. Deshalb müsste die These überprüft und weiter untersucht werden, wann das Gefühl der Wiederverkörperung ausgelöst wird. Die restlichen Antworten, die dem nachfolgenden Diagramm entnommen werden können, variieren.

Spezielle Frage: Bei welchem Experiment (1-6) war das Gefühl, seine eigenen Hände zu steuern, am höchsten?
30 Antworten

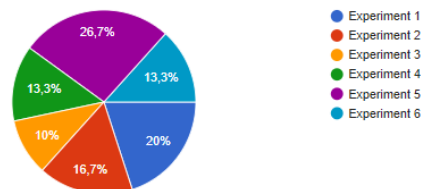


Abb. 6.8.: Diagramm zur speziellen Frage nach dem Gefühl, bei welcher Phase die virtuellen Hände als eigene empfunden werden

Um die zeitliche Korrelation zwischen virtueller und realer Hand zu untersuchen, wurden zeitbezogene Analysen aufgestellt. Die Simulationszeit, die von der Neurorobotics Plattform bestimmt wird und die real verstrichene Zeit wurden beim Experiment mitgestoppt. Bei einigen Probanden war es nötig, die Experimentumgebung neuzustarten, um die Würfel an ihre Ausgangsposition zu platzieren. Demnach variieren die Simulationszeiten und die realen Zeiten. Je länger die Simulation dauerte, desto größer wurde die Differenz zwischen beiden Zeiten, da die Simulationszeit wie bereits erwähnt langsamer läuft, als die reale Zeit. Das begründet sich dadurch, dass die Neurorobotics Plattform die physikalischen Berechnungen und das Rendern für das Webfrontend übernimmt. Diese Rechenoperationen benötigen Zeit und verlangsamen dadurch die Simulationszeit. Bei einer realen Zeit von unter 3 Minuten beträgt die Differenz der beiden Zeiten ungefähr 30 Sekunden. Die durchschnittliche Differenz aller gemessenen Simulations- und realen Zeiten beträgt hingegen gerundet 47 Sekunden. Der Server liegt mit seiner Zeitrechnung also immer etwas hinter der realen Zeit. Das erklärt die Latenz im Experiment. Da diese Latenz von den Probanden aber als nicht störend

wahrgenommen wurde, scheint die Differenz von 47 Sekunden noch akzeptabel zu sein.

Für die Phasen 3 bis 6 wurde die benötigte Zeit zum Ausführen der Aufgaben auch gestoppt. Der folgenden Tabelle lassen sich die durchschnittlich benötigten Zeiten für die verschiedenen Aufgaben entnehmen.

	Phase 3	Phase 4	Phase 5	Phase 6
Durchschnittliche Zeit in Sekunden	80	73	126	131
Durchschnittliche Zeit in Minuten	1,20	1,13	2,06	2,11

Tab. 6.1.: Durchschnittlich benötigte Zeit für die Bewältigung der verschiedenen Aufgaben

Der Tabelle ist zu entnehmen, dass die sechste Phase die meiste Zeit benötigte. In dieser Phase sollte die Aufgabe ausgeführt werden, einen Würfel auf Kopfhöhe zu heben und mit der anderen Hand wieder aufzufangen. Diese Phase erfordert komplexere Konzentration, Feinmotorik und Reaktionsgeschwindigkeit. Mit 5 Sekunden Zeitabstand folgt die zweite Phase des Experiments, das Anheben des Würfels mit zwei Fingern und seine Platzierung auf dem grauen Würfel. Auch wenn diese Phase im Vergleich zu den anderen Phasen relativ viel Zeit zur Ausführung in Anspruch genommen hat, war das Gefühl, die virtuellen als die eigenen Hände zu steuern am höchsten, da auch hier ein erhöhtes Maß an Konzentration aufgebracht werden musste. Also je höher die Konzentration, desto eher stellt sich das Gefühl der Wiederverkörperung ein. Der zeitliche Unterschied zwischen Phase 3 und 4 lässt sich so erklären, dass die Probanden einen Lerneffekt hatten, nachdem sie einen der Würfel auf den grauen Würfel platziert hatten und es ihnen damit schneller und leichter gefallen ist, die Aufgabe mit nur drei Fingern auszuführen.

Dennoch beantworteten 19 der Probanden, dass sie während der Dauer des Experiments zu keinem Zeitpunkt die virtuellen Hände als ihre eigenen wahrgenommen haben.

6. Nutzerstudie und Ergebnisse

Spezielle Frage: Gab es während des gesamten Experiments einen Zeitpunkt, indem die virtuellen Hände als die eigenen wahrgenommen wurden?

30 Antworten

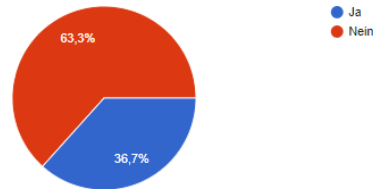


Abb. 6.9.: Diagramm zur speziellen Frage nach der Wiederverkörperung der Hände

Der ausschlaggebende Grund dafür war die räumliche Verschiebung zwischen realer und virtueller Hand. Von den Probanden gesehen befindet sich die virtuelle Hand etwas über der realen. Bei der speziellen Frage, ob diese Verschiebung das Gefühl der Verkörperung gestört hat, haben 27 Probanden mit "Ja" gestimmt. Auch wurde im mündlichen Feedback die Kameraeinstellung bemängelt, die sich im virtuellen Avatar auf Brusthöhe befunden hat, sodass die virtuellen Arme von oben her agierten, und damit zusätzlich die räumliche Verschiebung zwischen realem und virtuellem Arm verstärkten. Die Visualisierung musste allerdings im Rahmen dieser Arbeit vorgegeben bleiben.

Spezielle Frage: Hat die räumliche Verschiebung zwischen realer Hand und virtuellem Avatar das Gefühl, der Avatar zu sein, gestört?

30 Antworten

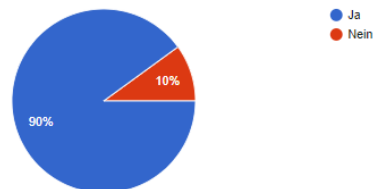


Abb. 6.10.: Diagramm zur speziellen Frage nach der räumlichen Verschiebung

Ein weiterer Störfaktor für die Wiederverkörperung der realen Hände in die virtuellen wurde in der Nutzerstudie im sogenannte "Kickback" vermutet. Das bedeutet, wenn entweder die komplette Hand zur Faust geballt wird, oder die Finger von der Faust wieder ausgestreckt werden, kommt es zu einem Rückstoß im Handgelenk. Dieser Rückstoß bewegt die Hand impulsartig nach oben oder unten, je nachdem, ob die Faust geballt, oder ausgestreckt wird, tritt aber nur auf, sobald alle Finger entsprechend

6. Nutzerstudie und Ergebnisse

bewegt werden. Dieser Rückstoß wurde aber von den wenigsten Probanden überhaupt wahrgenommen und deshalb bei dem Experiment größtenteils als nicht störend empfunden, wie das folgende Diagramm zu dieser speziellen Frage zeigt.

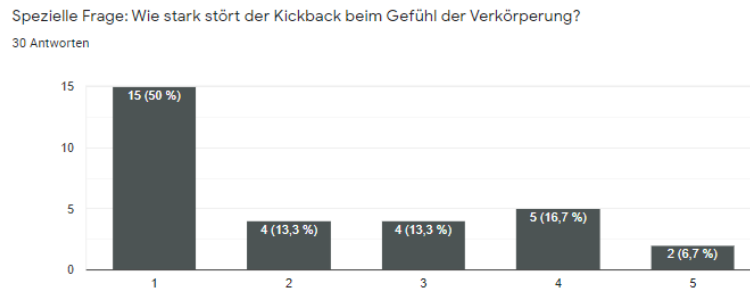


Abb. 6.11.: Diagramm zur speziellen Frage nach dem "Kickback" (Rückstoß)

Im Verlauf dieses Kapitels wurden die wichtigsten Fragen der Nutzerstudie aus Übersichts- und Relevanzgründen ausgewertet, sowie Zeitanalysen betrachtet. Zur Vervollständigung befinden sich im Anhang alle Diagramme zu jeder Frage der Nutzerstudie.

7. Fazit und Ausblick

Dieses Kapitel zieht ein abschließendes Fazit dieser Bachelorarbeit und gibt einen Ausblick für zukünftige Arbeiten.

7.1. Fazit

Auch wenn die Nutzerstudie bei verhältnismäßig wenigen Probanden zum Gefühl der Wiederverkörperung geführt hat, wurde die Implementierung des Hand- und Fingertrackings als gut umgesetzt bewertet. Die meisten Probanden lernten schnell, mit dem Index Controller umzugehen und die Würfel zu greifen. Die Hauptaspekte, die störten, waren die räumliche Verschiebung zwischen virtuellem und realem Arm, sowie die Platzierung der Kamera im Körper des virtuellen Avatars. Die Kameraführung sorgte dafür, dass die Oberarme neben, beziehungsweise knapp über dem Blickfeld empfunden wurden. Dadurch agierten die virtuellen Arme mit den Händen von überhalb des Blickfeldes herab und nicht auf gleicher Ebene eines gerade ausgerichteten Blickes bis zum unteren Rand des Blickfeldes, wie es bei realen Armen der Fall ist. Dieser Aspekt irritierte die Probanden so, dass der Fokus der Empfindung bei einigen nicht auf die Hände gelegt werden konnte. Trotzdem führte das Experiment der Nutzerstudie bei 11 der 30 Probanden zu einem Gefühl der Wiederverkörperung, die virtuellen Hände als ihre realen Hände zu empfinden. Aus Zeitgründen konnte die störende räumliche Verschiebung und ungünstige Kamerapositionierung im Rahmen dieser Arbeit nicht mehr behoben werden.

Die Entscheidung für die zwei Index Controller für das Hand- und Fingertracking für die Neurorobotics Plattform hat sich bewährt. Denn sie sind im Vergleich zu Exoskeletten oder Cybergloves relativ preiswert für den Benutzer zu erwerben und verfügen über solides Hand- und Fingertracking für Bewegungsabläufe. So sind Aktionen wie alle Finger einzeln, oder insgesamt zu krümmen, oder die Faust zu ballen und Objekte zu greifen, durch das SteamVR Unity Plugin schnell zu implementieren. Diese Controller haben allerdings den Nachteil, dass sie nur vibrieren können, um ein haptisches Feedback zu liefern. Ein Exoskelett dagegen blockiert eine Fingerbewegung, wenn der Finger entsprechend auf ein Hindernis trifft. Dies könnte das Gefühl der Wiederverkörperung wahrscheinlich verstärken. Allerdings ist hier die Frage, inwiefern ein Exoskelett als alternative Hardware in Unity einsetzbar ist. Es könnte zu erheblichem Aufwand für

die Implementierung kommen und es ist auch fraglich, wie akkurat das Fingertracking des Exoskeletts umgesetzt werden kann. Hiermit sind Faktoren genannt, die bei der Wahl einer anderen Hardware für haptisches Feedback berücksichtigt werden müssen. Zusammenfassend erzeugte das implementierte Programm durch intuitive Steuerung mit den Index Controllern und haptisches Feedback, welches durch Vibration der Controller umgesetzt wurde, ein Gefühl der Wiederverkörperung bei einigen Probanden. Tatsächlich findet also eine Wiederverkörperung statt.

7.2. Zukünftige Arbeit

Auch wenn der Rückstoß, beziehungsweise Kickback von den Probanden als nicht störend wahrgenommen wurde, sollten die PID Parameter in zukünftigen Arbeiten so angepasst werden, dass dieser Kickback reduziert wird. Um zukünftig bei mehr Probanden das Gefühl der Wiederverkörperung hervorzurufen, sollte auch die Kameraeinstellung überarbeitet werden, sodass die Oberarme als natürlich am Oberkörper platziert empfunden werden. Zusätzlich könnte die Knochenhierarchie des Modells so angepasst werden, dass das SteamVR Behaviour Skeleton Skript direkt auf die Hände des virtuellen Modells angewendet werden kann. Dadurch könnte der Code in Bezug auf die Laufzeit performanter gestaltet werden, wodurch die Latenz ausgeglichen werden könnte. Aus der Nutzerstudie wurde die These abgeleitet, dass das Gefühl der Wiederverkörperung der Hände bei erhöhter Konzentration auftritt. Diese These könnte in zukünftigen Arbeiten mit weiterentwickelten Experimenten überprüft werden, indem Versuchsanordnungen aufgestellt werden, für die ein noch höheres Maß an Konzentration aufgebracht werden muss. Es wäre ebenfalls interessant zu testen, wie haptisches Feedback das Gefühl der Wiederverkörperung beeinflusst. Bei der Gummihand Illusion werden sowohl künstliche, als auch reale Hand gleichzeitig stimuliert. Demnach wäre es möglich, auf andere Hardware, wie schon angemerkt, wie ein Exoskelett zurückzugreifen. Wenn ein virtueller Würfel gegriffen wird, blockiert das Exoskelett die Finger, sodass ein Gefühl entstehen könnte, den virtuellen Würfel in der realen Hand zu halten. Dieses alternative Feedback könnte das Gefühl der Wiederverkörperung verstärken.

Daraus folgend bieten sich weitere Anwendungsmöglichkeiten. Beispielsweise könnte man versuchen, einen weiteren virtuellen, humanoiden Roboter in die Neurorobotics Plattform einzubinden und ihn durch die passende Hardware mit der Steuerung eines weiteren Benutzers kombinieren. Das bedeutet, zwei Benutzer könnten durch virtuelle Roboter interagieren. Dies bietet ein Spektrum neuer Versuchsreihen. Beispielsweise könnten Versuche durchgeführt werden, wie sich Objekte zuzuwerfen, diese aufzufangen, oder abzuwehren, ähnlich wie ein Tennisspiel. Die Neurorobotics Plattform bietet

auch, wie bereits erwähnt, die Möglichkeit, reale Roboter zu unterstützen, also auch das Hand- und Fingertracking für einen realen Roboter umzusetzen. Dieser Roboter könnte in unterschiedlichen Umgebungen gesteuert werden, die menschenunzugänglich sind. Im medizinischen Bereich könnten Roboterhände für Telechirurgie bei höchst ansteckenden Patienten eingesetzt werden, um Infektionen zu vermeiden. Wichtige Reparaturen könnten in lebensbedrohlichen, wie verstrahlten Gebieten ausgeführt werden. Auch im Weltall könnte sich so ein Steuerungsmechanismus als nützlich erweisen. Teilweise ist das auch schon Realität: denn ein Beispiel für ein solches, schon existierendes System ist der "Tactile Telerobot", Roboterhände, die über eine Art Cyberglove mit Exoskelett gesteuert werden. Der "Tactile Telerobot" kann laut Website in verschiedenen Bereichen, unter anderem industrieller Produktion, Bombenentschärfung, Biomedizin und nukleareren Stilllegungen eingesetzt werden. [Com20] Dieses System ist allerdings großindustriell ausgerichtet und für Benutzer im privaten Bereich finanziell nicht erschwinglich. Im Gegensatz dazu steht diese Arbeit, die einen für Verbraucher erschwinglichen Ansatz verfolgt. Dennoch zeigen sowohl das eben genannte Beispiel, als auch das Ergebnis dieser Arbeit, dass der Ansatz, Roboterhände zu steuern und damit einhergehend das Gefühl der Wiederverkörperung hervorzurufen, sowohl für Wissenschaft, Industrie, als auch im privaten Bereich sehr vielversprechend für die Zukunft ist.

Verzeichnis der Abbildungen

3.1. HTC Vive Basisstationen [HTC20]	11
3.2. HTC Vive Headset [HTC20]	12
3.3. Valve Index Controller [Str18]	13
5.1. UML-Diagramm zur Veranschaulichung der Zusammenarbeit der Klassen	24
5.2. Transparenter lokaler Avatar	30
5.3. Unsichtbarer lokaler Avatar	30
5.4. Neurorobotics Platform Experiment für die Nutzerstudie	30
6.1. Diagramm zum Ausbildungsabschnitt der Teilnehmer	32
6.2. Diagramm zur SUS Frage nach der Komplexität der Bedienung	33
6.3. Diagramm zur SUS Frage nach der Schwierigkeit der Bedienung	33
6.4. Diagramm zur SUS Frage nach der Bedienung	34
6.5. Diagramm zur SUS Frage nach der Integration	34
6.6. Diagramm zur SUS Frage nach den Inkonsistenzen	34
6.7. Diagramm zur Auswertung des SUS Punktestands [BKM09]	35
6.8. Diagramm zur speziellen Frage nach dem Gefühl, bei welcher Phase die virtuellen Hände als eigene empfunden werden	36
6.9. Diagramm zur speziellen Frage nach der Wiederverkörperung der Hände	38
6.10. Diagramm zur speziellen Frage nach der räumlichen Verschiebung	38
6.11. Diagramm zur speziellen Frage nach dem "Kickback" (Rückstoß)	39

Verzeichnis der Tabellen

6.1. Durchschnittlich benötigte Zeit für die Bewältigung der verschiedenen Aufgaben	37
-----------------------------------------------------------------------------------------------	----

Literaturverzeichnis

- [BKM09] A. Bangor, P. Kortum, and J. Miller. "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale." In: *J. Usability Stud.* 4 (Apr. 2009), pp. 114–123.
- [Bro96] J. Brooke. *"SUS-A quick and dirty usability scale."* *Usability evaluation in industry*. ISBN: 9780748404605. CRC Press, June 1996.
- [Buc15] S. Buckley. *This Is How Valve's Amazing Lighthouse Tracking Technology Works*. besucht am 12.01.2020. Mai 2015. URL: <https://gizmodo.com/this-is-how-valve-s-amazing-lighthouse-tracking-technol-1705356768>.
- [Com20] S. R. Company. *Tactile Telerobot The World's First Haptic Telerobot Hand*. besucht am 10.03.2020. n.a. 2020. URL: <https://www.shadowrobot.com/telerobots/>.
- [Corna] V. Corporation. *SteamVR Unity Plugin*. besucht am 18.01.2020. n.a. n.a. URL: https://valvesoftware.github.io/steamvr_unity_plugin/.
- [Dex18] DextaRobotics. *The virtual world lies within your grasp*. besucht am 15.01.2020. n.a. 2018. URL: <https://origin.dextarobotics.com/en-us/>.
- [Fou14] O. S. R. Foundation. *Why Gazebo?* besucht am 18.01.2020. n.a. 2014. URL: <http://gazebosim.org/>.
- [Hau18] J. Haudenschild. "Virtual Embodiment: Umgang mit Diskrepanzen zwischen dem virtuellen und dem realen Körper." Technische Universität München, Oct. 2018.
- [HTC20] HTC. *Hardware*. besucht am 10.02.2020. n.a. 2020. URL: <https://www.vive.com/de/product/>.
- [KGS12] K. Kilteni, R. Groten, and M. Slater. "The Sense of Embodiment in Virtual Reality." In: *Presence Teleoperators and Virtual Environments* 21.4 (Nov. 2012), pp. 373–387. DOI: 10.1162/PRES_a_00124.
- [Mai15] E. Maiberg. *Valve and HTC Reveal Vive VR Headset*. besucht am 13.01.2020. Mai 2015. URL: <https://www.gamespot.com/articles/valve-and-htc-reveal-vive-vr-headset/1100-6425606/>.

- [Osb18] C. Osborne. *Thalmic Labs shuts sales of Myo gesture control armband*. besucht am 15.01.2020. Oktober 2018. URL: <https://www.zdnet.com/article/thalmic-labs-shuts-down-myo-gesture-control-armband-project/>.
- [Pla18] N. Platform. *HBP Neurorobotics Platform Brains for bodies and bodies for brains*. besucht am 15.01.2020. n.a. 2018. URL: <https://neurorobotics.net/>.
- [Pro17a] H. B. Project. *Bodies for brains*. besucht am 15.01.2020. n.a. 2017. URL: <https://www.humanbrainproject.eu/en/robots/>.
- [Pro17b] H. B. Project. *Short Overview of the Human Brain Project*. besucht am 15.01.2020. n.a. 2017. URL: <https://www.humanbrainproject.eu/en/about/overview/>.
- [Ren19] J. J. van Rensburg. "Virtuelle Verkörperung von menschlichen Füßen in der Neurorobotics Platform." Technische Universität München, Aug. 2019.
- [Robna] O. Robotics. *About ROS*. besucht am 18.01.2020. n.a. n.a. URL: <https://www.ros.org/about-ros/>.
- [Ros15] R. Rosenberger. *Postphenomenological Investigations: Essays on Human-Technology Relations*. English. Ed. by P. Verbeek. Postphenomenology and the Philosophy of Technology. Lexington Books, 2015. ISBN: 978-0739194362.
- [Ska19] Skarredghost. *Price*. besucht am 15.01.2020. July 2019. URL: <https://skarredghost.com/2019/07/12/dexmo-dexta-robotics-vr-price-specs-review/>.
- [Spe00a] Spektrum. *Efferenzkopie*. besucht am 20.01.2020. n.a. 2000. URL: <https://www.spektrum.de/lexikon/neurowissenschaft/efferenzkopie/3219>.
- [Spe00b] Spektrum. *Erfolgsorgan*. besucht am 24.01.2020. n.a. 2000. URL: <https://www.spektrum.de/lexikon/neurowissenschaft/erfolgsorgan/3646>.
- [Sta20] W. Stangl. *Top-down-Verarbeitung*. besucht am 25.01.2020. n.a. 2020. URL: <https://lexikon.stangl.eu/4755/top-down-verarbeitung/>.
- [Str18] G. Streeter. *Valve Announces Knuckles EV3 And Here Are The Upgrades*. besucht am 10.02.2020. Sept. 2018. URL: <https://www.vrandfun.com/valve-announces-knuckles-ev3-and-here-are-the-upgrades/>.
- [Sys17] C. Systems. *Featured Hardware*. besucht am 15.01.2020. n.a. 2017. URL: <http://www.cyberglovesystems.com/>.
- [Tec20] U. Technologies. *Unleash your creativity*. besucht am 18.01.2020. n.a. 2020. URL: <https://unity.com/products/core-platform>.
- [ultra] ultraleap. *Small. Fast. Accurate. World-class hand tracking for anyone, anywhere*. besucht am 15.01.2020. n.a. n.a. URL: <https://www.ultraleap.com/product/leap-motion-controller/>.

- [Valna] Valve. *Controller*. besucht am 13.01.2020. n.a. n.a. URL: <https://www.valvesoftware.com/de/index/controllers>.
- [Vivnaa] H. Vive. *Basisstation*. besucht am 12.01.2020. n.a. n.a. URL: <https://www.vive.com/de/accessory/base-station/>.
- [Vivnab] H. Vive. *Vive Specs*. besucht am 13.01.2020. n.a. n.a. URL: <https://www.vive.com/us/product/vive-virtual-reality-system/>.
- [Web19] M. Webel. "PID-Tuning Framework for Remotely Operated Humanoid Robots." MA thesis. Technische Universität München, Dec. 2019.
- [WK19] S. Weber and G. Klinker. "VR Re-Embodiment in the Neurorobotics Platform." In: *Mensch und Computer 2019 - Workshopband*. Bonn: Gesellschaft für Informatik e.V., 2019. DOI: 10.18420/muc2019-ws-585.
- [YS10] Y. Yuan and A. Steed. "Is the Rubber Hand Illusion Induced by Immersive Virtual Reality?" In: *2010 IEEE Virtual Real. Conf* 95–102 (Apr. 2010), pp. 95–102. DOI: 10.1109/VR.2010.5444807.
- [ZED18] the ZED Team. *Introducing ZED Mini*. besucht am 15.01.2020. n.a. 2018. URL: <https://www.stereolabs.com/blog/introducing-zed-mini/>.

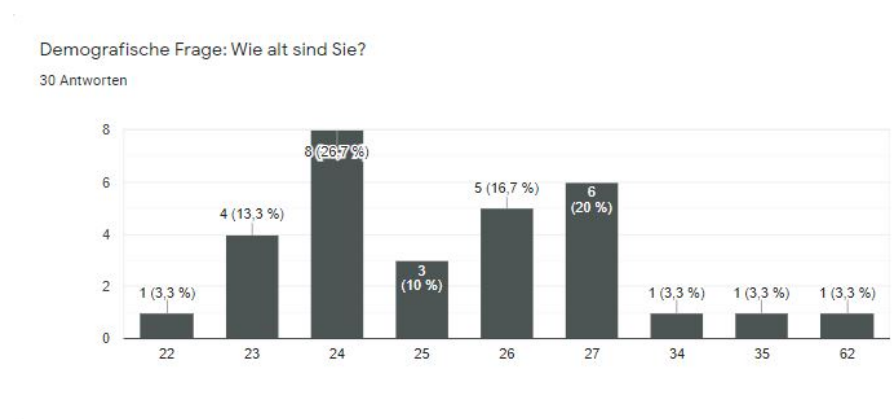
A. Anhang

A.1. Programmcode

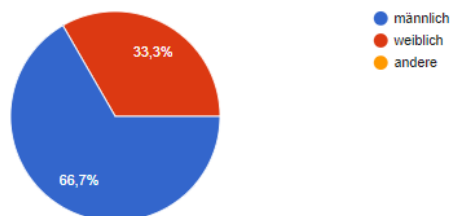
Der gesamte Programmcode dieser Arbeit befindet sich auf GitHub unter folgender URL:

https://github.com/HBPNeurorobotics/Unity3D-Client/tree/steinmann_hand

A.2. Diagramme der Nutzerstudie



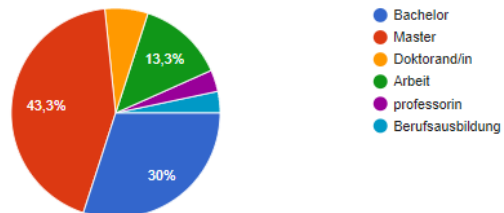
Demografische Frage: Welchem Geschlecht fühlen Sie sich zugehörig?
30 Antworten



A. Anhang

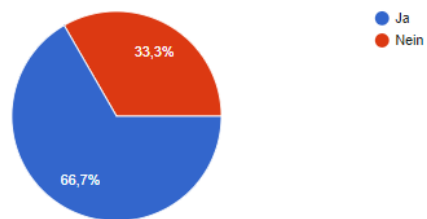
Demografische Frage: In welchem Ausbildungsabschnitt befinden Sie sich?

30 Antworten



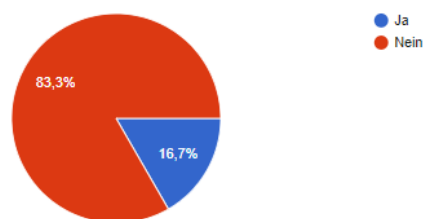
Demografische Frage: Haben Sie bereits Erfahrung mit virtueller Realität?

30 Antworten



Demografische Frage: Haben Sie bereits Erfahrung mit dem Valve Index Controller?

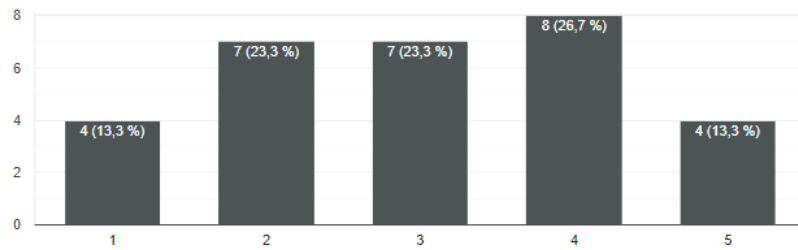
30 Antworten



A. Anhang

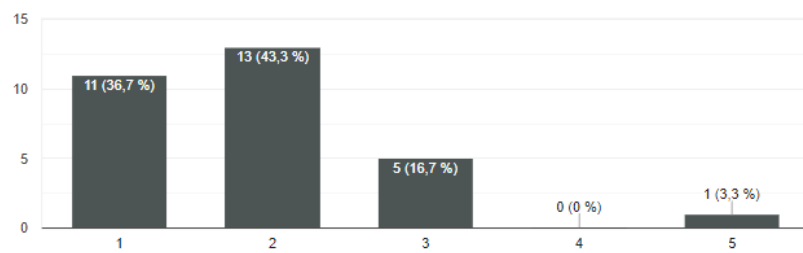
SUS Frage: Ich kann mir sehr gut vorstellen, das Virtual Reality System der Handsteuerung mit dem Index Controller regelmäßig zu nutzen.

30 Antworten



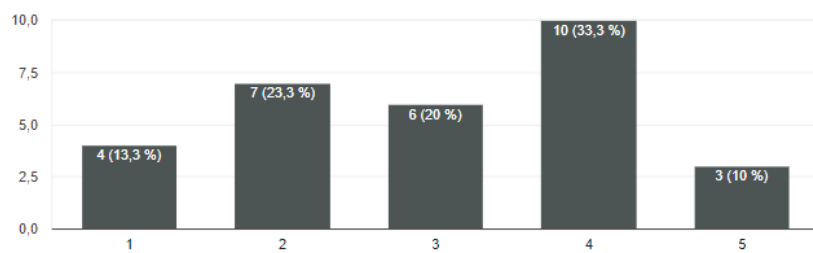
SUS Frage: Ich empfinde die Handsteuerung als unnötig komplex.

30 Antworten



SUS Frage: Ich empfinde die Handsteuerung als einfach zu nutzen.

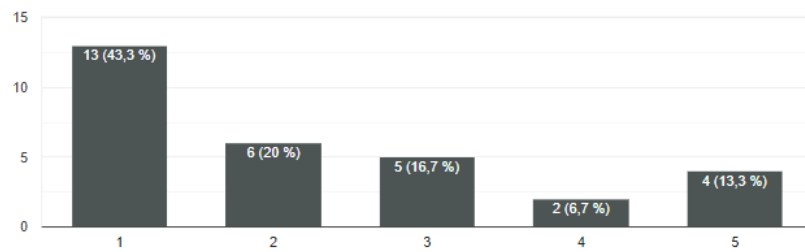
30 Antworten



A. Anhang

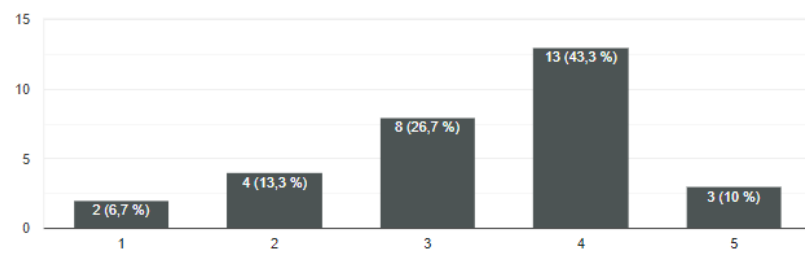
SUS Frage: Ich denke, dass ich technischen Support brauchen würde, um das VR System der Handsteuerung zu nutzen.

30 Antworten



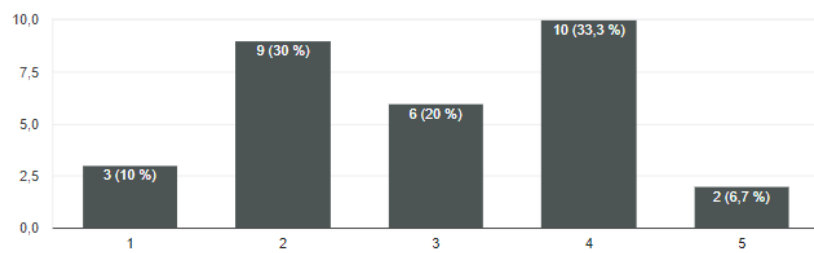
SUS Frage: Ich finde, dass die verschiedenen Funktionen der Handsteuerung (Handtracking und haptisches Feedback) gut integriert sind.

30 Antworten



SUS Frage: Ich finde, dass es bei der Handsteuerung zu viele Inkonsistenzen gibt.

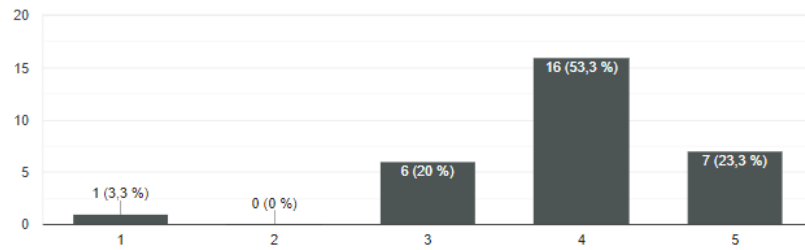
30 Antworten



A. Anhang

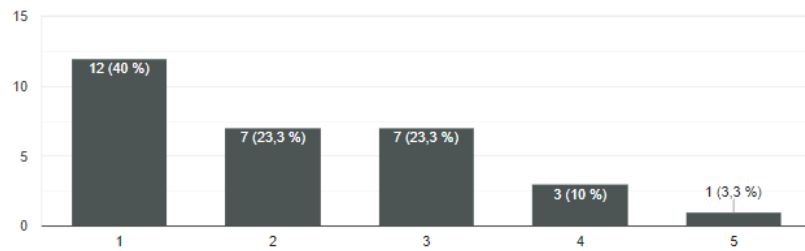
SUS Frage: Ich kann mir vorstellen, dass die meisten Leute die Handsteuerung schnell zu beherrschen lernen.

30 Antworten



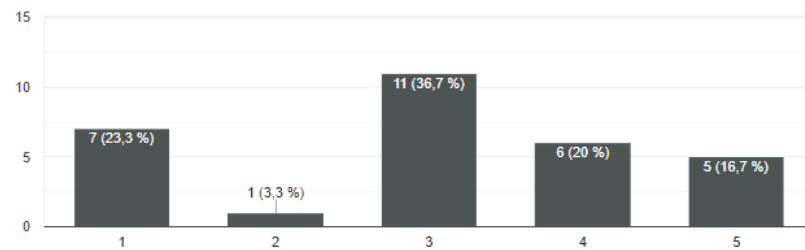
SUS Frage: Ich empfinde die Bedienung als sehr umständlich.

30 Antworten



SUS Frage: Ich habe mich bei der Nutzung der Handsteuerung sehr sicher gefühlt.

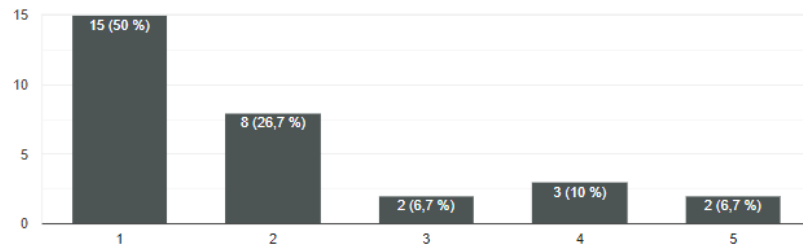
30 Antworten



A. Anhang

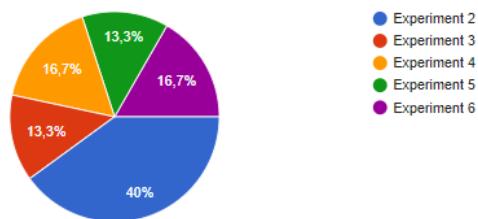
SUS Frage: Ich musste eine Menge Dinge lernen, bevor ich mit der Handsteuerung arbeiten konnte.

30 Antworten



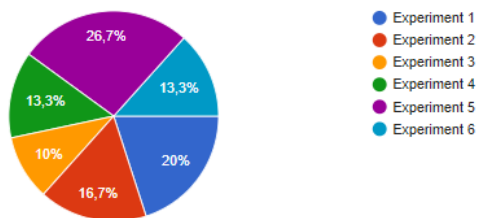
Spezielle Frage: Welches Experiment (2-6) war am einfachsten durchzuführen?

30 Antworten



Spezielle Frage: Bei welchem Experiment (1-6) war das Gefühl, seine eigenen Hände zu steuern, am höchsten?

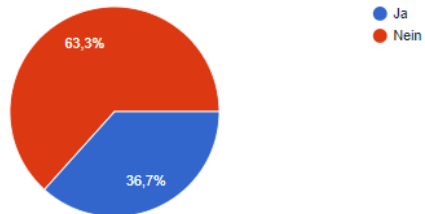
30 Antworten



A. Anhang

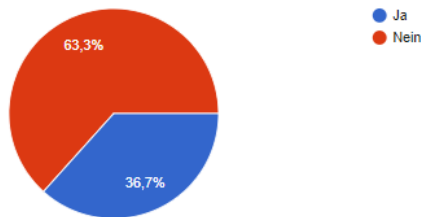
Spezielle Frage: Gab es während des gesamten Experiments einen Zeitpunkt, indem die virtuellen Hände als die eigenen wahrgenommen wurden?

30 Antworten



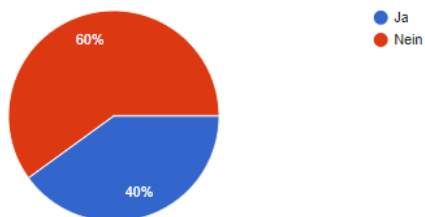
Spezielle Frage: Trat das Gefühl zu einem Zeitpunkt des gesamten Experimentes auf, indem Sie das Gefühl hatten, mehr als 2 reale Arme zu haben?

30 Antworten



Spezielle Frage: Hatten Sie das Gefühl, einen virtuellen Würfel in der realen Hand zu halten?

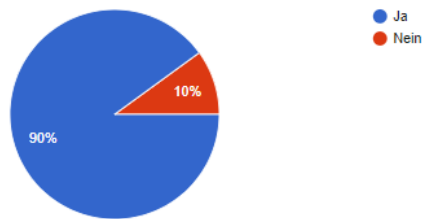
30 Antworten



A. Anhang

Spezielle Frage: Hat die räumliche Verschiebung zwischen realer Hand und virtuellem Avatar das Gefühl, der Avatar zu sein, gestört?

30 Antworten



Spezielle Frage: Wie stark stört der Kickback beim Gefühl der Verkörperung?

30 Antworten

