



Hochschule für Angewandte
Wissenschaften Hamburg
Hamburg University of Applied Sciences

Technische Analyse eines Full-Body immersiven
VR-Systems realisiert mit Motion Capturing, Unity 3D
und Oculus Rift

Fakultät Design, Medien und Informatik
Department Medientechnik
der Hochschule für Angewandte Wissenschaft Hamburg

Abschlussarbeit

zur Erlangung des akademischen Grades
Bachelor of Science

vorgelegt von

Thomas Fischer

Matrikelnummer: 2065364

Erstprüfer: Prof. Dr. R. Greule

Zweitprüfer: Dipl. Ing. (FH) M. Kuhr

Hamburg, 29.02.2016

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Bachelor-Thesis mit dem Titel:
Technische Analyse eines Full-Body immersiven VR-Systems realisiert mit Motion Cap-
turing, Unity 3D und Oculus Rift
selbstständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die
ich wörtlich aus der Literatur oder aus anderen Quellen wie z.B. Internetseiten übernom-
men habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

Datum: _____ Unterschrift: _____

Abstract

In der folgenden Bachelor Abschluss Thesis geht es um den Aufbau eines immersiven VR-Systems. Es werden Grundlagen und Grundbegriffe zum Thema Virtual Reality näher erläutert und erklärt. Außerdem wird der Versuchsaufbau unter technischen Gesichtspunkten und ihrer Funktion untersucht und bewertet.

Inhaltsverzeichnis

Eidesstattliche Erklärung	2
Abstract	3
1 Einleitung	5
2 Grundlagen	6
2.1 Motion Capture	6
2.1.1 Perception Neuron	9
2.1.2 Xsens	12
2.2 Game Engine	15
2.2.1 Unreal	16
2.2.2 Unity 3D	17
2.3 VR Brillen	19
2.3.1 Oculus Rift DK2	20
2.3.2 Project Morpheus	21
2.3.3 HTC Vive	22
3 Versuchsaufbau	25
4 Versuchsdurchführung	31
4.1 Inbetriebnahme	31
4.2 Versuch	34
5 Ergebnisse	36
5.1 Axis Neuron Pro	36
5.2 VR Setup auf einer Ebene	39
5.3 VR Setup auf der Treppe	42
6 Fazit	44
6.1 Zusammenfassende Bewertung	44
6.2 Ausblick	45

1 Einleitung

Virtual Reality (VR) ist eine künstlich geschaffene Welt, die der Fantasie oder der Wirklichkeit entspricht. Die Idee gibt es schon seit vielen Jahren. Bestes Beispiel ist die Science Fiction Serie Star Trek, in der mittels Hologramme Räume erzeugt werden, in denen Personen miteinander interagieren können. Jedoch bedeutet VR nicht nur die Darstellung von dreidimensionalen Bildern. Es geht im Kern darum, die gesamte eigene Wahrnehmung zu täuschen.

In der folgenden Arbeit wird eine besondere Form der Virtual Reality, nämlich die sogenannte Mixed Reality, technisch analysiert.

Mixed Reality (MR) gibt es in zweierlei Hinsicht. Einmal als Augmented Reality (AR), wobei hier mittels einer halbdurchlässigen Virtual Reality Brille virtuelle Informationen, Animationen oder Objekte der Realität hinzugefügt werden. Desweiteren gibt es die Augmented Virtuality (AV). Hierbei handelt es sich um eine virtuelle Umgebung mit realen Objekten mit denen man interagiert. Diese Form der MR wird in dieser Thesis Anwendung finden.

Die Arbeit beginnt mit den technischen Grundlagen und der Erklärung einzelner Technologien, die zum Aufbau eines VR-Systems nötig sind. In den folgenden Kapiteln wird der Versuchsaufbau sowie die Versuchsdurchführung näher erläutert. Es folgt die Auswertung der Ergebnisse des Experiments sowie ein kurzes Fazit.

Ziel des Versuchs ist es zum einen, mögliche technischen Komplikationen im Aufbau sowie bei der Durchführung herauszufinden. Zum anderen um festzustellen, ob die Darstellung des Anwenders/der Anwenderin¹ in der virtuellen Welt wirklich frei von Trackingfehlern ist, wie es in diversen Videos im Internet suggeriert wird und welche Rolle dabei externe Umgebungsfaktoren wie metallische Gegenstände oder elektromagnetische Störfelder spielen. Bei den Tests wird sich der Anwender auf einer ebenen Fläche und auf einer Treppe bewegen.

¹im weiteren Verlauf der Arbeit immer als "der Anwender" bezeichnet

2 Grundlagen

Ein AV²-System besteht aus drei technischen Bestandteilen die in Echtzeit miteinander kommunizieren müssen. Dem Tracken, mittels Motion Capturing, dem Erstellen der virtuellen Welt mithilfe einer 3D Game Engine und der Benutzung einer geschlossenen VR-Brille um in die erweiterte Virtualität einzutauchen. Dadurch, dass die Brille geschlossen ist, sieht der Anwender eine komplett virtuelle Welt. Allerdings kann er typisch AV mit real existierenden Objekten interagieren. Grundvoraussetzung dafür ist, dass diese der Game Engine beigelegt sind.

2.1 Motion Capture

Motion Capture ist ein Tracking-Verfahren, welches entwickelt wurde, um Bewegungen unterschiedlichster Art zu digitalisieren. Mittels eines Computers können diese Daten ausgelesen, analysiert und weiterverarbeitet werden. Die häufigste Anwendung dieses Verfahrens ist die Übertragung menschlicher Bewegungen auf ein, in einer Grafik Engine generiertes, Drahtmodell. Diese reichen von der Bewegung einzelner Finger bis hin zur Gesichtserfassung, also Mimik und Gestik.

Grundsätzlich wird bei den Tracking-Verfahren zwischen optischer und nicht optischer Bewegungserkennung unterschieden.

Bei den optischen Verfahren wird wiederum zwischen Tracken mit Markern und ohne Markern unterschieden. Beim Tracken mit Markern wird die Triangulation genutzt. Triangulation ist eine Entfernungsmessung, bei dem der Abstand eines Objektpunktes über ein Dreieck und dessen Innenwinkel berechnet wird. Die Markerbewegung einzelner Kamerabilder werden dann in 3D berechnet. Beim Tracken ohne Markern wird eine Muster bzw. Silhouetterkennung, beispielsweise mittels einer Kinect, genutzt. Dieses System funktioniert, indem es die Silhouette des Spielers aus dem Raum extrahiert und die Bewegung auf ein virtuelles 3D Modell adaptiert.

Nicht optische Trackingverfahren werden in inertielle, mechanische und magnetische Sys-

²AV: Augmented Virtuality

teme unterteilt.

Bei dem Inertialsystem messen Sensoren die Beschleunigung und die Winkelgeschwindigkeit bestimmter Punkte. Die Bewegungsdaten werden dann an einen Computer gesendet und mit Hilfe einer Software ausgewertet oder aufgenommen. Das Problem bei diesem Verfahren ist, dass beim Tracken der Bewegungen es schnell zu nicht eindeutigen Signalen kommen kann und diese zu falsche Bewegungen des digitalen Avatars führen. Um diese Fehler zu vermeiden, bieten beispielsweise die Entwickler von 3D Motion Tracking Systemen Xsens(Kapitel 2.1.2) und Perception Neuron (Kapitel.2.1.1) vorgefertigte menschliche Modelle in ihrer jeweiligen Software an. Diese Modelle kann man sich wie ein menschliches Skelett vorstellen. Die einzelnen Gelenke des menschlichen Körpers sind fest vorgegeben, um möglichst realistische Bewegungen darstellen zu können (siehe Abbildung 2.1).

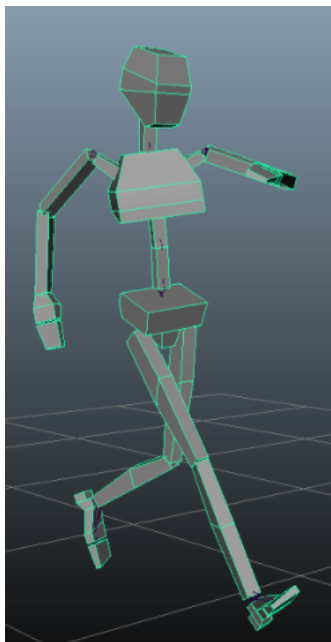


Abbildung 2.1: biomechanisches Softwaremodell

Um mit einem solchen Modell arbeiten zu können, ist eine vorherige Kalibrierung nötig, sowie die Angabe von Bodymaßen, wie zum Beispiel Körpergröße, Armlänge oder Schuhgröße, in der jeweiligen Software. Dies hängt mit der späteren Sensorposition zusammen. Bei richtiger Positionierung der Sensoren aber falscher Körperangaben kann es zu Abbildungsfehlern kommen, da die Abstände zwischen Sensor und Gelenk bzw. nächstem Body-Segment nicht übereinstimmen (siehe Abbildung 2.2).

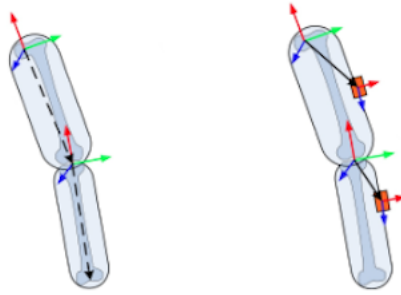


Abbildung 2.2: korrekte Sensorposition

Um die Bewegung des Sensors zu messen, dient ein Algorithmus, der Beschleunigung und Winkelgeschwindigkeit miteinander kombiniert, sodass die Position des Sensor berechnet werden kann. Ein Sensor kann den Abstand zum Boden sowie die kleinste Winkeländerungen exakt berechnen und an die Software weitergeben. Die Sensoren arbeiten mit einer sehr hohen Framerate, sodass jede kleinste Bewegungsänderung registriert wird. Nachteile dieses Verfahrens sind, dass der Benutzer in manchen Bewegungen wie eine Marionette aussieht, eine etwas geringere Positiongenauigkeit und dass es zum Drift³ kommen kann. Anzüge, die mit diesem System arbeiten, kosten heutzutage zwischen 1.000€ und 20.000€, wobei die günstigeren Motion Capturing Anzüge meist neuere und kleinere Chips verwenden und nahezu die gleiche Performance liefern wie die Teuren.

Beim mechanischen Motion Capturing wird die Bewegung des Benutzers direkt auf ein mechanisches Außenskelett übertragen (siehe Abbildung 2.3). Das Skelett besteht meist aus starrem Material wie Metall oder Plastik. Die einzelnen Teile sind über Gelenke, wie bei einem Menschen, verbunden. Somit reagieren diese Anzüge exakt auf die Bewegungen des Benutzers. Solche Systeme arbeiten in Echtzeit und sind ortsunabhängig, da man keine Kamera oder einen Signalreceiver benötigt.

Magnetische Motion Capturing Sensoren berechnen ihre Position in einem Magnetfeld, welches von einem Transmitter erzeugt wird und eine sehr niedrige Frequenz besitzt. In dem Transmitter befinden sich zueinander rechtwinklige Spulen, die die Richtung des Magnetfeldes bestimmen. Man unterscheidet zwischen zwei Systemen: AC und DC. Handelt es sich um AC besitzt die Frequenz die Form eines Sinussignals und bei DC besitzt es die Form eines Rechtecksignals. In der Regel besteht so ein System aus sechs bis elf Sensoren pro Gelenk. Jeder dieser Sensoren stellt Messungen bezüglich Position und Rotation an, um die Bewegung akkurat darzustellen. Ein großer Nachteil dieses Systems ist die hohe Anfälligkeit bei elektromagnetischen Störungen durch Metall in der Umgebung und ande-

³Drift: trotz Stillstand bewegt sich das 3D Modell langsam in der virtuellen Welt



Abbildung 2.3: mechanischer MoCap-Anzug "Gypsy 7" der Firma "Meta Motion"

re magnetische Felder, erzeugt durch Stromkabel, Monitore usw. Außerdem verhindert die aufwändige Verkabelung der vielen Sensoren schnelle Bewegungen. Jedoch findet Motion Capturing mit magnetischen Sensoren aufgrund der hohen Exaktheit der Bewegungen im Militär und in der Medizin großen Anklang.

Mit der Entwicklung nichtoptischer Tracking-Anzüge wurden inertielle und magnetische Sensoren miteinander kombiniert.

2.1.1 Perception Neuron

Der Motion Capturing Anzug Perception Neuron ist ein Produkt der Firma Noitom Limited. Der Firmenname entstand aus dem einfachen Wort "Motion" welches rückwärts geschrieben "Noitom" ergibt. Sie wurde im Jahr 2011 gegründet und hat ihren Sitz in Peking. Das Unternehmen hat sich der Weiterentwicklung bzw. dem Neudefinieren von Motion Capturing verschrieben. Ihr Hauptaugenmerk liegt in der Untersuchung des menschlichen Körpers und dessen Bewegungen sowie der Interaktion mit Gegenständen. Noitom wirbt für seine Motion Capture Technologie nicht nur im Film- und Gaming-Bereich, sondern soll auch in der Wissenschaft und der Medizin Anwendung finden. Im Augenblick hat Noitom drei Motion Capture Produkte auf dem Markt:

- Perception Neuron
- Perception Legacy
- MySwing Golf



Abbildung 2.4: Perception Neuron Full-Body Anzug

Der Perception Neuron Anzug besteht aus 32 Sensoren, die über dünne Kabel miteinander verbunden sind. Man kann diese nach belieben an den jeweiligen Punkten einsetzen. Je nachdem, ob man den kompletten Arm inklusive Hand, den Oberkörper oder den ganzen Körper tracken möchte. Zur Fixation dienen Klettverschlüsse, die man an den vorgegeben Stellen befestigen muss (siehe Abbildung 2.4).



Abbildung 2.5: Neuron Sensor

Die Sensoren, wie in Abbildung 2.5 zu sehen, sind mit den Maßen 12.5mm x 13.1mm x 4.3mm die kleinsten, die an einem MoCap Anzug je verbaut wurden. Sie arbeiten nach dem Inertialsystem, welches jedoch den Nachteil besitzt, dass es wie bereits erwähnt zum

sogenanntem Drift kommen kann. Um dies zu vermeiden hat Noitom die Inertialsensorik insofern verbessert, als das sie in jedem Sensor ein 3D Gyroskope⁴, ein 3D Beschleunigungssensor und ein 3D Magnetometer verbaut haben. Durch die Hinzunahme des Magnetsensors, wird eine höhere Stabilität in den horizontalen Bewegungen erreicht. Der Sensor tastet den Erdmagnetismus ab und arbeitet somit ähnlich wie ein Kompass. Die vertikalen Bewegungsänderungen werden von dem Beschleunigungssensor durch abtasten der Beschleunigung im Verhältnis zur Erdanziehungskraft berechnet. Mit Hilfe dieser sich ergänzenden Sensoren und deren komplementären Daten, sowie eines besonderen Algorithmus zur Verarbeitung und Auswertung der korrespondierenden Daten, lässt sich das Modell relativ akkurat darstellen. Zum Senden der durch die Sensoren ermittelten Daten dient der "Neuron Hub"(siehe Abbildung 2.6). An ihm laufen die Datenströme zusammen. Mittels USB Anschluss kann man den Hub mit dem Computer verbinden, bzw. Bewegungen mittels SD Karte aufzeichnen oder den Datenaustausch über Wifi konfigurieren. Abhängig von der Anzahl der angeschlossenen Sensoren sendet der Hub mit einer Bildrate von 60fps bei maximal 32 Neuronen, bzw. 120fps bei maximal 18 angeschlossenen Neuronen. An ihm befindet sich ebenfalls der Stromanschluss zur Versorgung aller Sensoren über einen externen Akku.



Abbildung 2.6: Neuron Hub

Um die gesendeten Daten des Anzugs auf dem Computer auszuwerten dient die von Noitom ebenfalls entwickelte Software "Axis Neuron" bzw. "Axis Neuron Pro" (siehe Abbildung 2.7). Diese nutzt wie im vorherigen Absatz erwähnt ein biomechanisches Modell. So dienen vorprogrammierte Segmente und Gelenke der höchstmöglichen und realisten

⁴Sensor zur Messung der Winkelgeschwindigkeit

Darstellung menschlicher Bewegungen. Diese können dann aufgezeichnet sowie in Echtzeit an eine Game Engine (Kapitel 2.2) übertragen werden. In der "Pro" Version können maximal fünf Personen mit einem MoCap Anzug dargestellt werden, in der freien Version lediglich zwei. Außerdem werden weitere Einstellungen zur genaueren Abbildung und Aufnahme der Bewegungen sowie eine professionellere Auswertung der Daten ermöglicht. Die "Pro" Version kostet 499\$, ist aber laut aktuellem Stand noch nicht als fertige Version auf dem Markt.

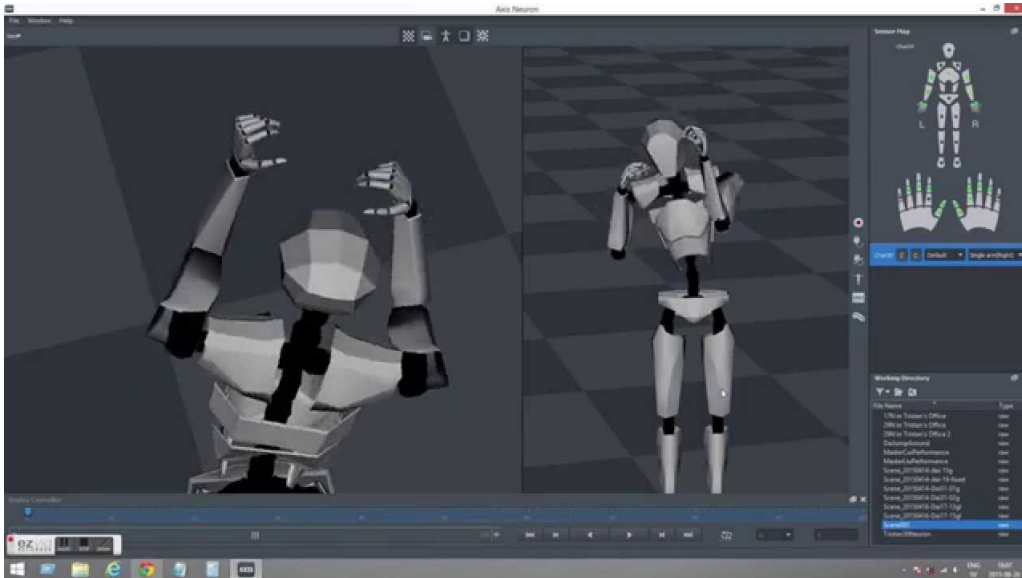


Abbildung 2.7: Axis Neuron

2.1.2 Xsens

Xsens Technologies B.V. ist eine Firma aus den Niederlanden mit ihrem Hauptsitz in Enschede und ihrer Tochterfirma XSENS North America in Los Angeles. Sie haben sich der Entwicklung von 3D Motion Capture Technologien verschrieben und arbeiten mit inertialen MEMS⁵ Sensoren, kombiniert mit weiteren Technologien wie GPS, RF positioning⁶, sowie biomechanischen Modellen.

Die Firma wurde im Jahr 2000 von Casper Peeters und Per Slycke gegründet, und entwickelte Einheiten zur Inertialmessungen (Messung der Trägheit) sowie zur Messung menschlicher Bewegungen. Im Jahr 2007 brachte XSENS ein inertial Motion Capture System mit dem Namen "Moven" auf den Markt. Zwei Jahre später wurde dieses System dann in Xsens MVN umbenannt.

Der Xsens MVN Motion Capture Anzug basiert ebenfalls auf Inertialsensoren, die über

⁵MEMS: Microelectromechanical systems

⁶RF (radio frequency) positioning: Ein Ortungssystem, welches mit einem 10GHz RF Signal die Positionierung eines MoCap Sensors mit einer maximalen Abweichung von 3cm ermöglicht

zwei Receiver kabellos mit dem PC und der dazugehörigen Software kommunizieren. Auch Xsens arbeitet mit der Kombination aus einem 3D Gyroskopsensor, einem 3D Beschleunigungssensor und einem 3D Magnetometer, um Drift zu vermeiden. Die von Xsens entwickelten MVN Fusion Engine verarbeitet dann die Daten der Sensoren und wertet diese aus.

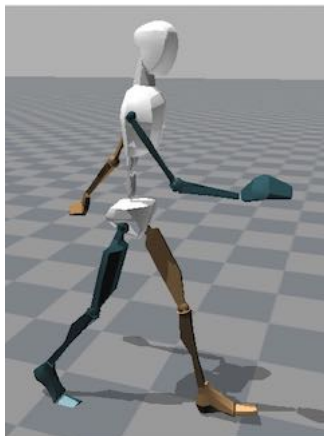


Abbildung 2.8: menschliches Modell der Xsens Software "MVN Studio"

Das menschliche Körpermodell mit dem der MVN Anzug arbeitet ist in 23 Segmenten und 22 Gelenken gegliedert (siehe Abbildung 2.8). So hat man die Möglichkeit, nicht nur den kompletten Oberkörper oder den kompletten Arm zu bewegen, sondern auch komplexe Bewegungsabläufe wie Laufen realistisch darzustellen.

Das MVN System besteht wie in Abbildung 2.9 zu sehen aus einem Anzug, dessen Stoff aus Elasthan besteht, zwei Handschuhen und einem Stirnband bzw Mütze. In diesem kompletten Anzug sind 16 Sensoren integriert und über Kabel, die im Anzug eingebettet sind, miteinander verbunden. Dabei wird der Anzug in links und rechts aufgeteilt. Pro Körperhälfte ist einen Xbus Master vorhanden, die sich in Taschen auf dem Rücken befinden. Die Aufgabe des Xbus Masters besteht darin, die Informationen der Sensoren, an den am Computer angeschlossenen Receiver, zu senden.

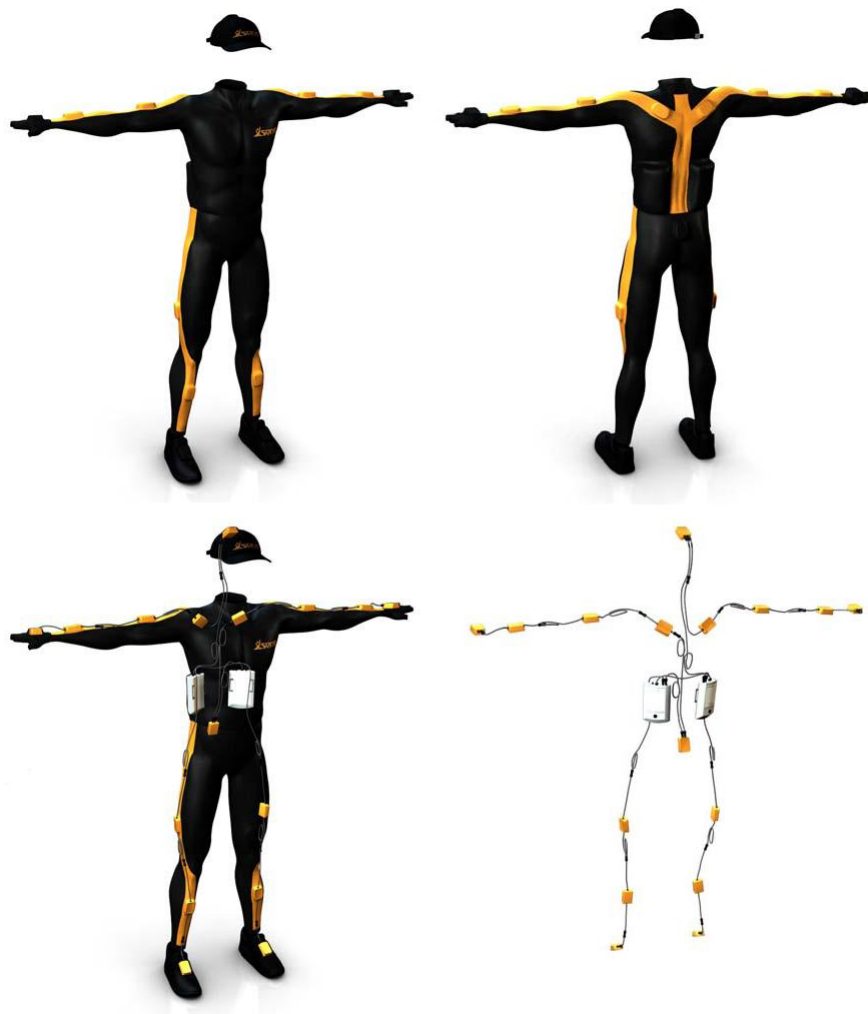


Abbildung 2.9: Xsens MVN Anzug

Aufgrund der schnellen Installation sowie Kalibrierung, der einfachen Handhabung und dem Verzicht auf Markern bzw. zusätzlichen Kameras, findet das System in der 3D Animation großen Anklang. Beste Beispiele wären die Filme "Mr. Go", "The Leviathan" und "TED" (siehe Abbildung 2.10) sowie diverse Spieleanwendungen. Aber nicht nur im Entertainmentbereich bedient man sich der Xsens Technologie. Aufgrund der niedrigen Latenz und der geschmeidigen Bewegungsdarstellung ist es möglich, mehrere Benutzer in ein VR System einzubetten. Dies findet vor allem bei virtuellem Training und Simulationen Anwendung wie beim Militär oder Sport. In Kombination mit der Computertechnik ist es ebenfalls möglich Augment Reality⁷ Systeme aufzubauen. AR wird im Ingenieurwesen, in der Sozialforschung und im Gamingbereich angewendet. Doch auch in der Medizin

⁷AR: Augmented Reality, computergenerierte erweiterte Realität

bzw. in der Rehaforschung könne mit Hilfe der Xsens Sensoren, im Zusammenspiel mit biomechanischen Gelenken, Prothesen sowie Roboter gesteuert werden.



Abbildung 2.10: "Ted" aus dem gleichnamigen Film gespielt von Seth MacFarlane

2.2 Game Engine

Die Game Engine ist das Fundament eines Spiels. Es umfasst nicht nur die Darstellung der Anwendung sondern auch den Sound, die Menülayouts, das Gameplay bis hin zur Zustandsspeicherung im Spiel. Im Game Bereich werden überwiegend die CryEngine, die Unreal Engine[Kapitel 2.2.1] und die Unity Engine[Kapitel 2.2.2] als Entwicklungsumgebungen genutzt.

Eine Game Engine ist also die Basis eines Spiels und übernimmt, abhängig von der Anwendung, diverse Funktionen. Sie beinhaltet die Grafik Engine, dem Programmcode welcher als Schnittstelle zwischen Hard und Software dient. Außerdem umfasst die Game Engine die komplette Physik in einem Spiel, mit deren Hilfe dieser sich ein höherer Realismus erreichen lässt. In Unity3D kann man zum Beispiel einem Objekt die Rigid-Body-Physik zuweisen. Mit dieser Physik verhalten sich die Objekte nach den Gesetzen Newtons. Wenn beispielsweise ein Ball auf einer schrägen Ebene liegt, rollt dieser die Ebene, abhängig von dem Winkel der Schräge, schnell herunter. Virtuelle Charaktere erhalten über die Ragdoll-Physik, ein Teil der Rigid-Body-Physik, physikalisch korrekte Attribute. Nicht feste Objekte wie Wasser, Rauch oder diverse Textilien können ebenfalls durch die Physik-Engine mit einem realistischen Verhalten dargestellt werden, zum Beispiel das Wehen einer Fahne bei Wind.

Der Sound eines Spiels ist mit den Jahren immer wichtiger geworden. Gerade wenn es darum geht, mögliche Gegner akustisch zu orten oder dass durch die Veränderung des

Klangbildes in verschiedenen Räumen wie Badezimmer oder große Hallen ein höherer räumlicher Eindruck entsteht, ist ein dementsprechendes Sounddesign unabdingbar. Mit den heutigen Game Engines ist dies ohne Weiteres umsetzbar. Sie unterstützen Raumklänge wie 5.1 oder auch 7.1 sowie Technologien wie EAX⁸. Außerdem bestimmt man in der Game Engine die Steuerung eines Spiels, also ob die Eingabe über ein Controller oder ganz klassisch über Maus und Tastatur erfolgt. In der Engine kann dann der Avatar mit einem Skript belegt werden, der beschreibt, welche Art der Steuerungssysteme genutzt werden soll und welche Tasten welche Funktion bedienen. Im Allgemeinen dienen Skripte der Programmierung der Spiellogik eines Spiels. Die Skriptsprache wird von der jeweiligen Game Engine bestimmt. Unreal arbeitet beispielsweise mit *UnrealScript* und Unity 3D mit *C++*. Zusätzlich zur Umgebungsgestaltung, Steuerung und Programmierung verfügt eine Game Engine über weitere Funktionen wie das Laden und Speichern von Datenpaketen. Diese Pakete können Spielstände sein oder Objekte und deren Zustände. Um mehrere Spieler an einem Spiel teilhaben zu lassen, kann mit Hilfe der Game Engine der Netzcode bestimmt werden. Das bedeutet, es wird festgelegt, wieviele Spieler simultan an einer Session teilnehmen können und was für eine Übertragungsbandbreite nötig ist. Außerdem hat der Entwickler die Wahl zwischen den Netzwerkprotokollen UDP und TCP. UDP ist ein verbindungsloses Protokoll und ermöglicht eine sehr schnelle Datenübertragung, jedoch ohne Rücksicht auf die Übertragungssicherung. Einzelne Datenpakete können sehr schnell fehlerhaft, doppelt, in falscher Reihenfolge übertragen werden oder komplett entfallen. TCP hingegen steht für eine langsamere aber dafür sichere und fehlerfreie Datenübertragung, da diese verbindungsorientiert arbeitet.

Hauptsächlich werden UDP und TCP zeitgleich betrieben. Somit ist TCP für die sichere Übertragung wichtiger Daten zuständig und UDP für den schnellen Transfer der kleinen und unbeständigen Daten.

2.2.1 Unreal

Epic Games hat im Jahr 1998 die Game Engine Unreal auf den Markt gebracht. Es beinhaltet die Grafik Engine, die Skriptsprache *UnrealScript* und weiterer Unterstützungssoftware wie zum Beispiel ein Leveleditor mit dem Namen *UnrealEd*. Das besondere an der Unreal Engine ist, dass diese bausteinartig aufgebaut ist. Das bedeutet, dass Bauteile der Engine zwar neu geschrieben werden, aber die Engine an sich bleibt gleich. Somit spricht man bei Unreal nicht von verschiedenen Versionsnummern, sondern von verschiedenen Generationen.

⁸EAX: Environmental Audio Extensions ist ein System, welches Umgebungsgeräusche in Spielen real erscheinen lässt, indem der Klang eines Tons, abhängig von dem Raum in dem sich der Spieler befindet, berechnet wird

Die erste Genreation wurde 1998 veröffentlicht. Als Grafikdemo für die Engine fungierte der PC Ego-Shooter Unreal. Vier Jahre später erschien die Unreal Engine 2 für PCs und die Konsolen GameCube, Playstation 2 und Xbox. Sie war erstmals auf dem freien Markt für 350.000 Dollar käuflich zu erwerben und für den nichtkommerziellen Gebrauch sowie zu Lehrzwecken kostenlos verfügbar. Jedoch wurde bei der kostenlosen Version Funktionen wie das Ändern von Quellcodes nicht unterstützt.

Die Spiele *RoboBlitz* und *Gears of War* (siehe Abbildung 2.11) erschienen im November 2006 und waren die ersten Anwendung in denen die Unreal Engine 3 zum Einsatz kam.



Abbildung 2.11: *Gears of War* erschien im 12. November 2006

Seit dem Jahr 2003 arbeiteten Mitarbeiter von Epic Games an der vierten Generation, welche dann unter dem Namen Unreal Engine 4 am 19. März 2014 erschien. Anfangs mussten Entwickler für die Nutzung monatliche Abonnementkosten zahlen. Inzwischen ist es für Schulungszwecke sowie für den kommerziellen Gebrauch kostenfrei. Mit der neuesten Engine werden nicht nur Spiele entwickelt. Sie wird desweiteren für die Umsetzung von Architekturanwendungen, Filmen und Virtual Reality Umgebungen genutzt.

2.2.2 Unity 3D

Die Firma *Unity Technology* wurde in Kopenhagen von einem Dänen, Isländer und Deutschen gegründet und hat ihren Hauptsitz in San Francisco. Mit der Engine ist die Erschaffung von 3D Grafik Applikationen wie Computerspielen sowie VR- und AR- Umgebungen, für diverse Systeme wie Windows, OS X, Spielekonsolen und mobilen Geräten möglich. Zur Erstellung von Scripte werden die Programmiersprachen C#, Unity Script, Boo und C++ genutzt. Grundsätzlich dienen die Scripte zur Ausführung des Spielablaufs und der

Spiellogik. Als Entwicklungsumgebung nutzt Unity in der Regel "MonoDevelop. Objekte die diffizil und nicht einfach zu beschreiben sind, werden in der Engine als *Assets* betitelt. Zu diesen Assets gehören beispielsweise Texturen, komplexe 3D-Objekte, Animationen oder besondere Klänge. Diese können im Programm im sogenannten "Asset Store" heruntergeladen werden.

Die Grafik Engine beruht auf OpenGL oder Direct3D. Die heutigen Beleuchtungsmodelle wie beispielsweise Bumpmapping, Enviroment Mapping, Umgebungsverdeckung und dynamische Schatten, finden zur Darstellung von Licht und Schatten Anwendung. Zusätzlich kann man die vorgegebenen Lichteffekte durch eigens erarbeitete Shader Effekte erweitern.

Um Charaktere zu animieren, nutzt Unity die "Skin and Bones" Technik. Das bedeutet dass eine Figur in eine Vielzahl von Polygone unterteilt wird. Diese Polygone liegen wie ein Netz über einem unsichtbaren Skelett. Auf diesem Skelett findet die eigentliche Animation statt und die Polygone ändern ihre Position abhängig von der Bewegung des Skeletts. Diese Technik wird oft bei der Animaton von Menschen und im Allgemeinen bei der Animation von organischen Modellen angewandt. Um eine realistische Darstellung von gasförmigen Elementen wie Feuer und Rauch zu realisieren wird ein Partikelsystem genutzt. Das bedeutet, das die einzelnen Partikel aus geometrische Objekte wie Kugeln oder Würfel bestehen. Ein sogenanntes "Emmitter Partikel" welches mit den verschiedensten Eigenschaften wie Geschwindigkeit, Lebensdauer, Anzahl der Partikel, Material usw. belegt werden kann, bestimmt die Bewegung und Eigenschaften aller Partikel in dem System. Der Unity-Editor kann außerdem mittels Plug-ins um weitere Features und Assets ergänzt werden, wie zum Beispiel:

- Terrain Former
- Vegetation Elements Pack
- Big Enviroment Pack Vol. 2
- Particle Playground
- Perception Neuron Integration Package

Unity ist mit 45% Weltmarktanteil führende Software in der Entwicklung von Spielen. Im Laufe der letzten vier Jahren hat sich die Zahl der Benutzer von knapp 500.000 auf 4.000.000 erhöht. Das liegt vor allem an der Entwicklung von Spielen für Mobilgeräte. Gerade in China und Japan nutzen über 70% der Spielehersteller Unity3D. In Zukunft wollen sich die Entwickler weiter verstärkt auf die Umsetzung von Mobile Games konzentrieren und weiter auf diesem Markt expandieren.

2.3 VR Brillen

VR-Brillen gehören zu der Gruppe der HMDs (Head-Mounted Display). Es sind Anzeigedisplays, die am Kopf befestigt werden. Man unterscheidet diese in vier verschiedene Klassen: monokular, binokular, see-through und non-see-through. Ob ein HMD ein mono- oder binokular Display ist, hängt davon ab, ob ein Display vor einem oder beiden Augen vorhanden ist. Die meisten Anwendungen von monokularen HMDs finden in der Industrie statt, da die Betätigungen in der Realität im Vordergrund stehen.

Bei einem see-through HMD ist das Display durchsichtig. Somit kann man sich dahinter befindende Umgebung sehen. Diese Technologie finden vor allem dann Anwendung, wenn die reale Welt mit Zusatzinformation bestückt werden soll, das nennt man Augmented Reality. HMDs mit einem non-see-through Display sind hingegen geschlossen. Im Games-Bereich und im VR-Bereich werden hauptsächlich binokulare, non-see-through HMDs genutzt. So wird der Anwender komplett in eine andere, virtuelle Welt gezogen und die Immersion ist am größten.

Die einfachste Form eines binokularen, geschlossenen HMDs ist die Videobrille. Sie wird ausschließlich zur Wiedergabe von Videoinhalten genutzt, wie zum Beispiel das Ansehen von DVDs im privaten Umfeld oder für Videospiele. Sie besitzen keinerlei Sensorik wie VR-Brillen. Diese besitzen Head Trackers, um die Videoinhalte in Abhängigkeit der Kopfposition wiederzugeben. Der Träger der VR-Brille erhält somit das Gefühl sich aktiv in der Umgebung zu bewegen und kann so komplett in eine andere Welt eindringen.

Anwendungsbereiche:

- immersives virtuelles Umfeld
- Augmented Reality bzw. Augmented Virtuality
- virtuelle Begehungen
- Prüfen von Räumen, Flugzeugcockpits, Autos, etc.
- Übungsszenarien des Militärs oder in der Medizin

Ein Leistungsparameter binokulare non-see-through HMDs ist unter anderem die Option auf einem bzw. zwei Bildschirmen verschieden Bilder darzustellen. Somit besteht die Möglichkeit, mittels Stereoskopie, Bildern den Eindruck einer räumlichen Tiefe zu verleihen. Außerdem spielt der Pupillenabstand eine wichtige Rolle, den man mit Hilfe von einer Konfigurationssoftware der Brillen einstellen kann bzw. die Brille erkennt den Abstand automatisch.

Die Größe des sichtbaren Bildausschnitts ist ein weiterer wichtiger Parameter. Das menschliche Auge hat ein 3D Sichtfeld von 120° in der Horizontalen und 135° in der Vertikalen. Dazu kommen noch jeweils 30° für das nicht dreidimensionale periphere Sichtfeld. Mit der Information, dass ein menschliches Auge 60 Pixel pro Grad Blickfeld sieht, hat AMD festgestellt, dass eine Auflösung von 16k nötig wäre, um eine optisch perfekte Immersion herzustellen. Dieser Wert errechnet sich, indem die Produkte aus horizontalem Sichtfeld mal 60 und vertikalem Sichtfeld mal 60 miteinander multipliziert werden und dieser Wert wiederum mal zwei genommen wird, da der Mensch zwei Augen besitzt. Das Ergebnis sind 116 Megapixel. Diese Auflösung können lediglich Displays mit 16k, also 128 Megapixel realisieren.

Moderne HMDs besitzen eine deutlich geringere Auflösung und liegen mit 10 bis 20 Pixel pro Grad deutlich unter dem des menschlichen Auges. Allerdings kann mit einem speziellen Rendering Verfahren eine Fokussierung von Objekten simuliert werden. In Kombination mit einer großen Gradzahl der binokulare Überdeckung, der Bereich in dem beide Augen sehen, der Mensch hat eine Überdeckung der Augen von 110 Grad, 55 Grad rechts und 55 Grad links von der Nase, entsteht trotz geringerer Auflösung ein hoher Grad an Immersion.

2.3.1 Oculus Rift DK2

Die Oculus Rift (siehe Abbildung 2.12) ist der berühmteste Vertreter der Virtual Reality Brillen. Palmer Luckey, Erfinder der Oculus Rift, war schon in jungen Jahren leidenschaftlicher Bastler und VR-Liebhaber. Allerdings war er mit der Qualität der damals auf dem Markt sich befindenen VR-Brillen unzufrieden. Er verbaute deren Teile sowie hochauflösende Handydisplays in seine eigenen Prototypen. Das sechste Modell mit dem Namen "Rift" schickte er dem Programmierer John Carmack¹¹, Schöpfer des Spiels "Doom". Dieser modifizierte die Rift und machte das Headset für das Spiel "Doom 3" kompatibel. Einer Demo des Spiels im Juni 2012 auf der E3 in Los Angeles folgte eine Kickstarter-Kampagne. Aufgrund des großen Erfolgs war VR auf einmal in aller Munde und viele weitere namenhafte Entwickler und Programmierer schlossen sich Palmer Luckey an und gründeten das Unternehmen Oculus VR.

Die Oculus Rift DK2¹² besteht ähnlich wie der Vorgänger das DK1-Modell aus einem hochauflösenden 5,7-Zoll OLED Display sowie einem Gyroskop. Diese Technologien sind in jedem Smartphone vorhanden und wurden auf das HMD adaptiert. Auf das Display werden via Splitscreen für das jeweils linke und rechte Auge Bilder abgespielt. Genau

¹¹John Carmack: Schöpfer des EGO-Shooter Genres, war bis zu seinem Wechsel 2013 zu "Oculus VR" bei "idSoftware" tätig

¹²DK: Development Kit

wie beim menschliche Auge, werden so Räume aus leicht unterschiedlichen Perspektiven dargestellt und es entsteht eine Dreidimensionalität. Um das Sichtfeld zu erhöhen werden die Bilder leicht verzerrt abgebildet, diese Verzerrung wird jedoch mit Hilfe von Linsen wieder aufgehoben. Um Kopfbewegungen und deren Ausrichtung zu erfassen, dient das Gyroskop und eine eigens entwickelte Infrarotkamera. Diese Kamera muss auf den Anwender gerichtet sein, um so die Positionsänderungen des Kopfes sowie die Tiefe tracken zu können. Ist das HMD zur Kamera jedoch um 180° gedreht, kann dieses nicht getrackt werden. Vorgängermodelle wie die DK1 hatten eine hohe Bewegungsunschärfe zu beklagen. Mit Hilfe der neuen Funktion "low persistence" konnte die Unschärfe extrem minimiert werden. Diese Technologie ist aus der Erkenntnis entstanden, dass das menschliche Auge und der Kopf nie zu 100% still stehen. Somit werden die Pixel, nachdem sie für einen Bruchteil einer Sekunde ein Bild dargestellt haben, auf Schwarz gestellt. Die Pixel werden erst dann wieder mit Inhalten bespielt, wenn die Bewegungserkennung der Brille eine Änderung registriert. Zusätzlich kann man mittels zweier USB 2.0 Anschlussbuchsen weitere Eingabegeräte an die VR-Brille anschließen. Zur Audiowiedergabe ist es möglich über eine 3,5mm Klinkenbuchse Audioabspielgeräte anzuschließen.



Abbildung 2.12: Oculus Rift DK2

2.3.2 Project Morpheus

Sony hat mit der Herausgabe der Next Generation Konsole Playstation 4 große Erfolge gefeiert. 2016 soll exklusiv für die Konsole eine VR-Brille mit dem Namen Project Morpheus (siehe Abbildung 2.13) erscheinen. Die Brille hat starke Ähnlichkeit mit der Oculus Rift Dev Kit 2. Pro Auge verfügt die bikulare HMD über ein 5,7 Zoll OLED Display mit einer Auflösung von 960 x 1080. Allerdings ist das Sichtfeld mit 90 Grad etwas geringer

als bei der Rift. Eine besondere Eigenschaft der Brille ist, dass diese sich automatisch auf die Dioptrien des Benutzers einstellt um gerade Brillenträgern möglichst viel Komfort zu bieten. Getrackt wird die Brille mit Hilfe der Kamera der Playstation 4. Zusätzlich werden die Position und die Stellung des Kopfes mit Sensoren zur Messung der Beschleunigung und Gyroskopen ermittelt. Die Brille soll mit den schon auf dem Markt sich befindenen "PlayStation Move Motion Controllern" kompatibel sein. Zusätzlich zu einer Klinkenbuchse verfügt die Sony Morpheus über ein eingebautes 3D Soundsystem. Dieses stellt sich auf die jeweilige Kopfposition sowie -neigung ein und passt sich jedem Blickwinkel an.



Abbildung 2.13: Sonys Project Morpheus

2.3.3 HTC Vive

Der Smartphone Hersteller "HTC" hat im Frühjahr 2015 verkündet an einer VR-Brille zu arbeiten. In Kooperation mit dem Softwareunternehmen "Valve" soll eine eigens für die Computerspieleplattform "Steam" eine Virtual Reality Brille mit dem Namen "HTC Vive" (siehe Abbildung 2.14) auf dem Markt kommen.

Das Tracken der Brille und die Bewegungserkennung soll mit Hilfe von 70 Sensoren erfolgen. Laut Hersteller ist es möglich, sich in einem 4,5 m² großen Raum zu bewegen, ohne das Motion Sickness¹³ auftritt. Grund soll die besondere Form der Positionserkennung sein. Die Brille arbeitet nicht wie üblich nur mit Gyroskopen und Magnetometern, sondern ebenfalls mit Laserscannern, die das Umfeld bzw. den Raum scannen. Desweiteren soll eine Bildwiederholfrequenz von 90 Hz und somit 90 Bilder pro Sekunde gewährleistet werden. Für die Bilder sorgen zwei Displays, womit die VR-Brille zu den bikularen HMDs zählt, mit einer Pixel-Auflösung von je 1.200 x 1.080. Es verfügt über kein integriertes Soundsystem, jedoch über eine Klinkenbuchse, an die man ohne weiteres Kopfhörer anschließen kann. Um das virtuelle Spielerlebnis zu erweitern, arbeiten die Entwickler der

¹³Übelkeit beim Tragen einer VR-Brille

HTC Vive an VR Game Controllern, mit denen man in der virtuellen Umgebung agieren kann.



Abbildung 2.14: HTC Vive

Name	Oculus Rift DK2	Playstation VR	HTC Vive
Hersteller	Oculus VR	Sony	HTC / Valve
Display	5,7-Zoll OLED	5,7-Zoll OLED	2 x OLED
Auflösung / Pixeldichte	1920 x 1080 / ca. 386 ppi	1920 x 1080 / ca. 386 ppi	2160 x 1200
Pixel pro Grad (diagonal)	ca. 14.45	ca. 14.45	ca. 14.68
Auflösung pro Auge	960 x 1080	960 x 1080	1200 x 1080
Framerate	75 Hz	120 Hz	90 Hz
Sichtfeld (diagonal)	ca. 100°	ca. 100°	ca. 110°
Gewicht	440 g (ohne Kabel)	unbek.	unbek.
IPD und Dioptrien Anpassung	2 Paar Linsen, verstellbares Display, IPD via Software anpassbar	unbek.	unbek.
Audio	Extern, 3D binaural	extern, 3D Sound	Extern
Releasedatum	2014 im Juli	(angekündigt) Q2 2016	(angekündigt) April 2016
Besonderheiten	low persistence	Für Playstation 4 optimiert. Move Controller Support	Lighthouse Tracking, VR-Controller, Chaperone Sicherheitssystem

Tabelle 2.1: VR-Brillen im Vergleich.

(Modifiziert nach <http://www.vrnerds.de/hardvr/96-2/vr-brillen-vergleich/>, Stand: 15.01.2016)

3 Versuchsaufbau

Das VR-System wurde wie bereits erwähnt aus den folgenden drei Technologien aufgebaut: dem Motion Capturing Anzug "Perception Neuron", der Game Engine "Unity 3D" sowie dem Head Mounted Display "Oculus Rift DK 2". Grundidee war es, eine Umgebung zu schaffen, die der Realität gleicht. Man bewegt sich also in einer echten Umgebung, die mittels eines 3D Modells nachgebaut und mit nicht vorhandenen virtuellen Objekten ergänzt wurde.

Der MoCap Anzug der Firma Noitom wurde mit 21 Neuronen bestückt, maximal wären 32 möglich gewesen. Um jedoch bei technischen Ausfällen der Sensoren reagieren zu können und Ersatz parat zu haben, fiel die Wahl auf das folgende Setup (siehe Abbildung 3.1).

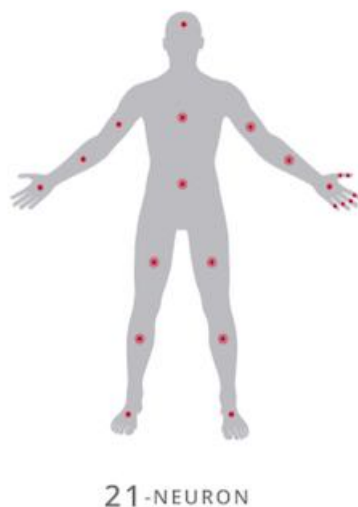


Abbildung 3.1: MoCap Setup mit 21 Neuronen

Der Datenstrom zwischen dem Hub des Anzugs und dem PC wurde über ein eigens eingerichtetes WLAN Netzwerk realisiert. Hierzu diente ein WLAN-Router TL-WR841N der Firma "TP-Link". Dieser hat eine Übertragungsbandbreite von 300 Mbit pro Sekunde welches bei einer Datenrate des HUBs von 27,8 Mbit pro Sekunde mehr als ausreichend ist. Die Übertragung der Daten erfolgt im 2,4 GHz Band über das verbindungslose Netz-

werkprotokoll UDP.

Zur Kalibrierung des Anzugs sowie der Bearbeitung der Bewegungsdaten dient die Software Axis Neuron. Hierbei handelt es sich um die Pro Version, die zwar noch nicht auf dem Markt ist, jedoch auf Nachfrage an den Hersteller als Beta Version zur Verfügung gestellt wurde. Da bei der Durchführung des Versuchs unter anderem das Auf- und Absteigen von Treppen untersucht werden sollte, war es notwendig die Option, den Boden als konstanten Bezug anzusehen, zu deaktivieren (siehe Kapitel 4).

Axis Neuron und Unity laufen auf dem selben Rechner. Um die Anzugsinformationen in Echtzeit an Unity weiterzugeben, bedarf es lediglich einer Anpassung der Broadcasting-Einstellung in Axis Neuron (siehe Abbildung 3.2).

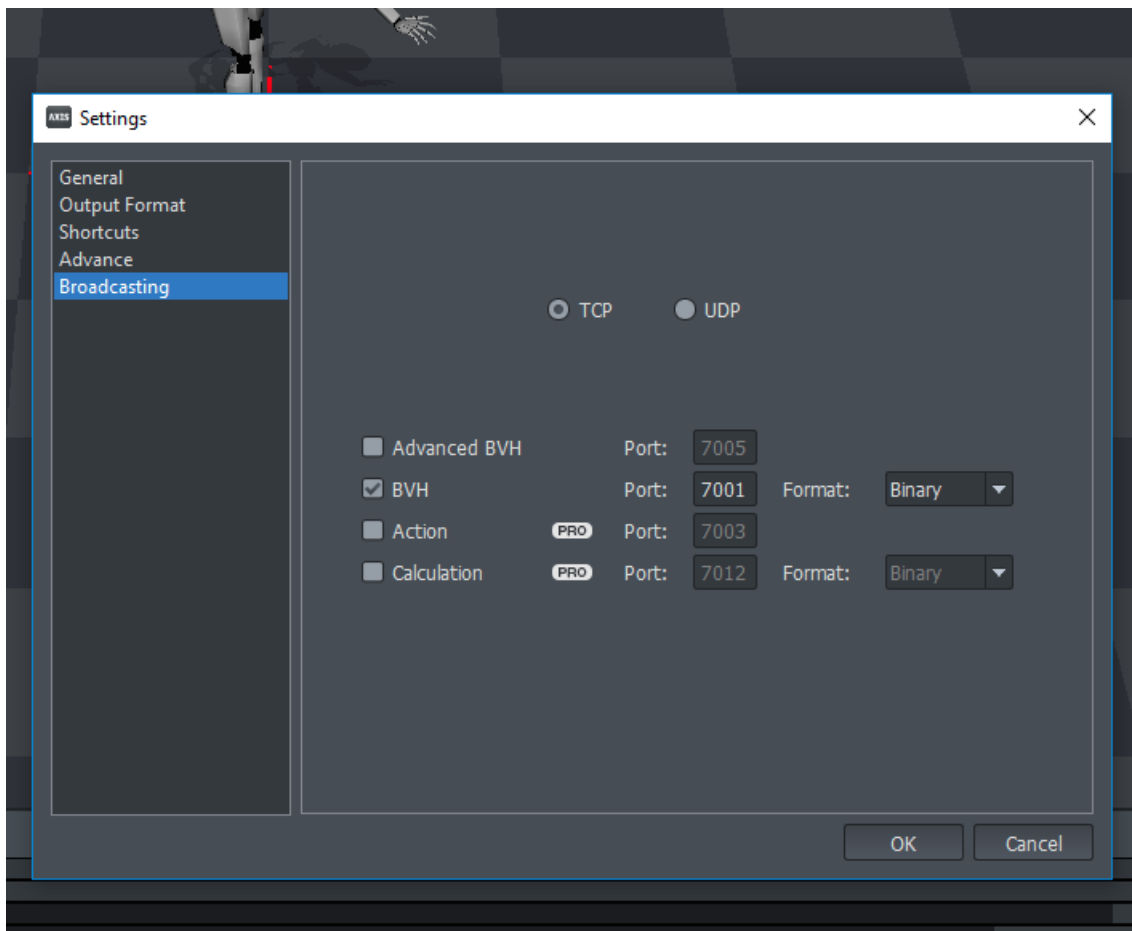


Abbildung 3.2: Axis Neuron Echtzeitstreaming Setup

Die Daten werden im BVH¹⁴ Format von Axis Neuron Pro an Unity 3D gesendet. Hierbei handelt es sich um ein Dateiformat, welches eigens für Motion Capturing Dateien von der Firma "Biovision" entwickelt wurde. Eine BVH Datei besteht aus zwei Bausteinen. Zum Einen aus der "Header Section", diese beinhaltet Informationen über die Skelett

¹⁴BVH: Biovision Hierarchical Data

Hierarchie, also aus wievielen beweglichen Elementen besteht das biomechanische Modell und in welcher Reihenfolge sind die Elemente angeordnet, sowie dessen Ausgangsposition. Zum Anderen liefert eine BVH Datei die Bewegungsinformation, welche als "Data Section" bezeichnet wird. Damit Unity 3D den von Axis Neuron Pro gesendet Datenstrom empfangen und verarbeiten kann, musste ein Unity SDK¹⁵, entwickelt von "Noitom", installiert werden. Für diesen Versuch wurde folgende SDK genutzt: "PerceptionNeuronUnityIntegration_0.2.5".

Es enthält folgende Paketinhalte:

- Neuron: beinhaltet die wichtigste Komponenten wie Prefabs, Scenes, Scripts und Texturen
- NeuronDebug: Systemdaten des Neuron Debuggers
- NeuronExamples: Beispiele zu möglichen Anwendungen in Unity
- Plugins: Datenbibliothek für die 32 und die 64 Bit Versionen

Bei der Entwicklung der virtuellen Welt war der erste Schritt die Erstellung der 3D Umgebung. Da gerade das Treppensteigen und Laufen auf einer flachen Ebene untersucht werden sollte, bot sich das Treppenhaus des Neubaus der HAW Hamburg, Finkenau 35 an. Aus verschiedenen CAD Dateien, auf denen der Grundriss der einzelnen Stockwerke zu sehen ist, wurde das Gebäude in dem Architekturprogramm "Vectorworks 2015" nachgebaut, allerdings nur das Treppenhaus und die angrenzenden Flure. Auf angrenzende Büros sowie diverse Labore und andere Räumlichkeiten wurde verzichtet. Somit sind in dem 3D Modell auch keine Türen der angrenzenden Räume zu sehen. Die Vergabe der Texturen und Materialien der Wände, des Bodens usw. erfolgte nicht in Unity, sondern schon der Einfachheit halber direkt bei der Konstruierung in Vectorworks. Da es sich bei der 3D Umgebung nur um das Treppenhaus handelt, wurden wie in Tabelle 3.1 zu sehen, lediglich vier verschiedene Texturen und die dazugehörigen Materialien verwendet. Dieses Modell wurde dann in Vectorworks als fbx.¹⁶ Datei exportiert. Dieses Dateiformat ist kompatibel mit Unity und man kann es ohne weiteres in die Game Engine importieren.

¹⁵SDK: Software Development Kit

¹⁶fbx: FilmBox Datei

Treppenhaus	Textur / Material
Boden (Abbildung 3.3)	Farbe: Pantone 19-3906 TPX-Dark Shadow Helligkeit: 100% Spiegelung: Plastik Textur: Naturstein-Granit (Stärke:100%)
Wände (Abbildung 3.4)	Farbe: Pantone 11-0602 TPX-SnowWhite Helligkeit: 100% Spiegelung: keine Textur: Naturstein-Granit (Stärke: 50%)
Stufenleiste (Abbildung 3.5)	Farbe: Pantone 14-4503 TPX-Metal Helligkeit: 100% Spiegelung: metallisch Tektur: keine
Fenster (Abbildung 3.6)	Farbe: keine Helligkeit: - Spiegelung: Glas / Flüssigkeit Textur: Glas / Flüssigkeit

Tabelle 3.1: Flächentextur des Treppenhausmodells

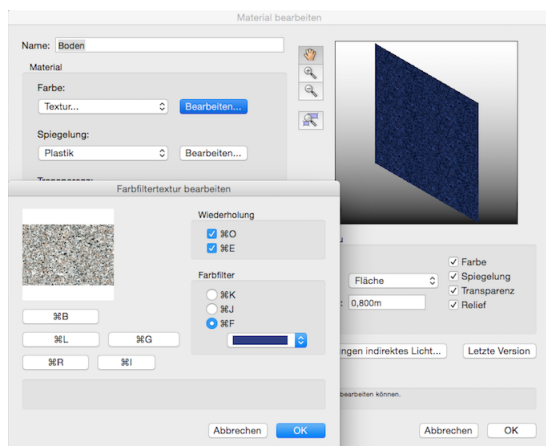


Abbildung 3.3: Bodentextur
(Vectorworks 2015)

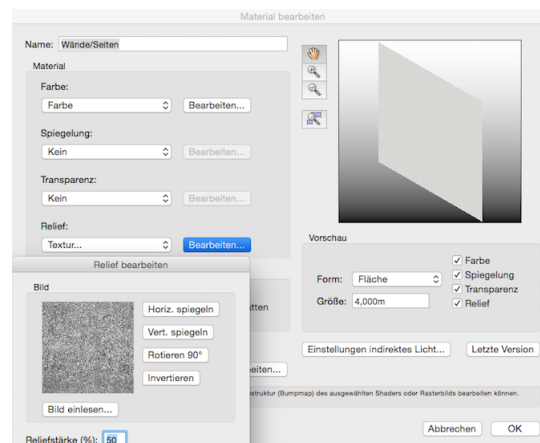


Abbildung 3.4: Wandtextur
(Vectorworks 2015)

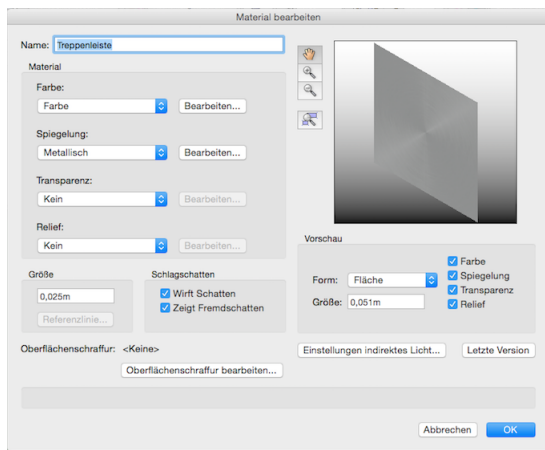


Abbildung 3.5: Stufenleistentextur
(Vectorworks 2015)

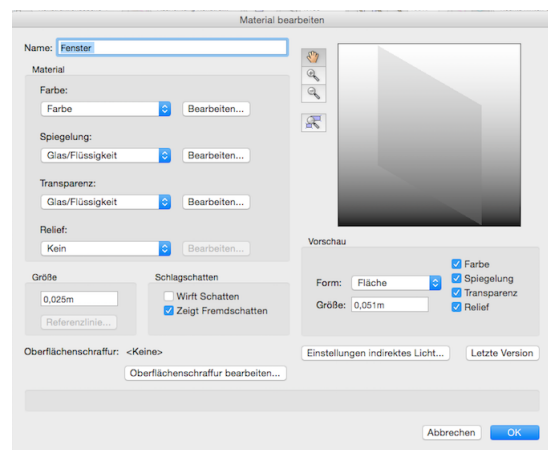


Abbildung 3.6: Fenstertextur
(Vectorworks 2015)

Nach dem Import der fbx. Datei in Unity folgte die Hinzunahme von virtuellen Objekten. Diese sind vorgefertigte 3D Modelle und wurden mit Hilfe des von Unity bereitgestellten "Asset Store" der virtuelle Welt hinzugefügt. Zu diesen Objekten gehören beispielsweise Palmen mit zugehörigem Sandboden (siehe Abbildung 3.7), oder ein virtuellen Ball. Hintergrund dieses Objektes war es zu prüfen, ob der Ball virtuell mit dem Fuß gespielt oder sogar aufgehoben werden konnte.

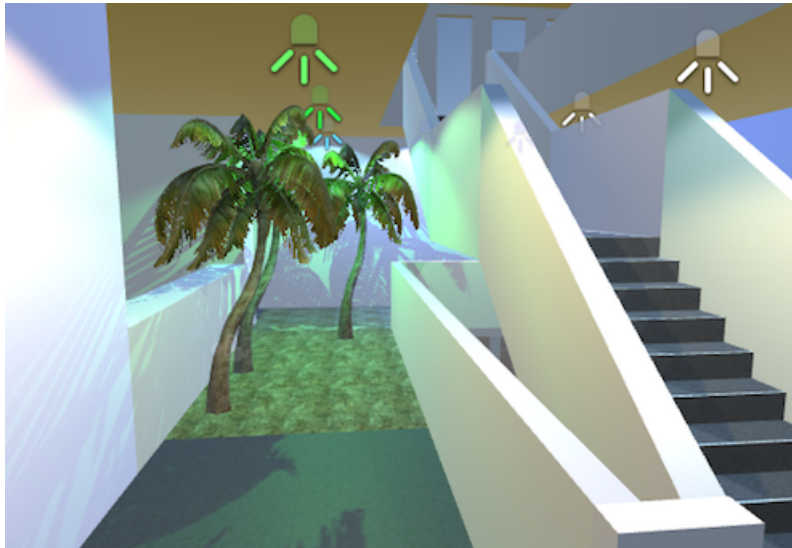


Abbildung 3.7: virtuelle Palmen 3D Treppenhausmodell

Die Oculus Rift DK2 wurde über ein HDMI Kabel, welches zur Übertragung des Video-signals dient und einem USB Kabel, welches für den Datentransfer und der Stromversorgung des HMDs zuständig ist, mit dem PC verbunden. Die Infrarot Kamera, die für das "Positional Tracking" der Oculus Rift zuständig ist, wurde natürlich nicht installiert, da

Positionsänderungen über den Motion Capturing Anzug registriert werden. Die Oculus Rift DK2 besitzt, unter anderem aufgrund ihrer deutlich besseren Auflösung und höheren Bildwiederholungsrate, eine relativ hohe Systemanforderung an den Computer, an den sie angeschlossen ist. In der Tabelle 3.2 stehen sich die Anforderungen der Oculus Rift und die vorhandene Leistungsparameter des im späteren Versuch genutzen Rechners im Vergleich gegenüber. Man erkennt deutlich, dass die Systemvoraussetzungen des genutzen Rechners deutlich ausreicht und die Leistungsanforderungen mehr als erfüllt.

	min Systemanforderungen Oculus Rift DK2	Leistungsparameter Tower PC
CPU	i5-4590	i7-4690
Arbeitsspeicher	8GB	32GB
Grafikarte	Nvidia GeForce GTX 600 / AMD Radeon HD 7000	4 x Nvidia GeForce GTX Titan
Schnittstelle	USB 2.0	USB 2.0 und USB 3.0
Betriebssystem	Windows 7 SP1	Windows 10

Tabelle 3.2: Vergleich Anforderungen Oculus Rift DK 2 mit Tower PC

4 Versuchsdurchführung

Um einen reibungslosen Testverlauf garantieren zu können, galt es die einzelnen Systeme zusammenzuführen und abhängig von den verschiedenen Versuchen aufeinander abzustimmen.

4.1 Inbetriebnahme

Schritt 1: Arbeitsplatz, Hardwareinstallation, Startposition

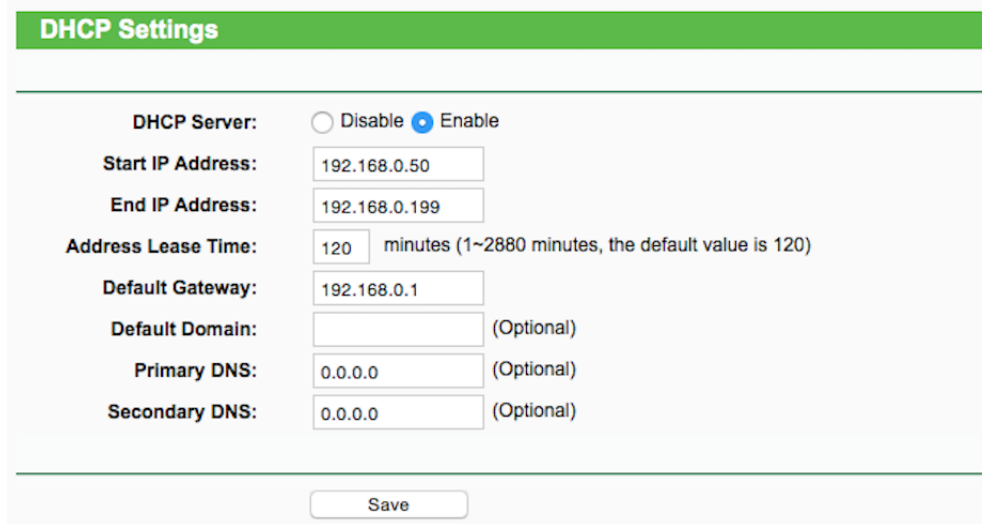
Die Oculus Rift DK 2 wurde über ein HDMI und ein USB Kabel mit dem PC verbunden. Somit musste der Arbeitsplatz dementsprechend direkt im Treppenhaus sein (siehe Abbildung 4.1). Hier galt es den PC möglichst am Rand zu platzieren, um zum Einen nicht direkt unter der Feuerschutzwand zu sein und zum Anderen den Gang für andere Personen begehbar zu halten. Da das HMD selbst nur über ca. 2m lange USB und HDMI Kabel verfügt, war keine großartige Mobilität gegeben. Somit wurden diese jeweils um 10m verlängert. Auch ein fixer Startpunkt auf dem Boden war wichtig. Dieser glich dem Startpunkt des virtuellen 3D Modells, damit es so nicht schon vor dem Start des Versuchs zu Positionsdifferenzen kam.



Abbildung 4.1: Arbeitsplatz im Treppenhaus des Neubaus der HAW Hamburg

Schritt 2: Anzug, HUB, Netzwerkeinstellungen, Axis Neuron Pro

Wie unter Kapitel 3 Versuchsaufbau zu lesen, wurde ein 21 Neuron Setup für den MoCap Anzug "Perception Neuron" gewählt. Für ein korrektes Positionstracking musste der Anzug mit Hilfe verschiedener Körperposen kalibriert werden. Um ein weiteres Anschlusskabel an den Rechner zu vermeiden, wurden die Positionsinformationen der Sensoren des Anzugs über eine Wifi-Verbindung realisiert. Der Router wurde dahingegen konfiguriert, dass er nur IP Adressen im Adressraum von bis vergibt (siehe Abbildung 4.2). Das Beziehen der IP Adresse des PCs erfolgte automatisch.



DHCP Server:	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Start IP Address:	<input type="text" value="192.168.0.50"/>
End IP Address:	<input type="text" value="192.168.0.199"/>
Address Lease Time:	<input type="text" value="120"/> minutes (1~2880 minutes, the default value is 120)
Default Gateway:	<input type="text" value="192.168.0.1"/>
Default Domain:	<input type="text"/> (Optional)
Primary DNS:	<input type="text" value="0.0.0.0"/> (Optional)
Secondary DNS:	<input type="text" value="0.0.0.0"/> (Optional)

Abbildung 4.2: Routerkonfiguration, IP Zuweisung mittels DHCP

In der Software Axis Neuron Pro wurde unter Netzwerkeinstellungen der Wifi Mode aktiviert und die IP Adresse des PCs übernommen. Um den Motion Capturing Anzug mit dem WLAN Netz zu konfigurieren, musste der HUB als erstes via USB Kabel mit dem PC verbunden werden, dann unter Wfi das richtige WLAN Netzt ausgewählt und das zugehörige Wireless Passwort eingegeben werden. Nachdem der HUB von dem Rechner getrennt und mit dem Akku verbunden wurde, hat er sich automatisch mit dem WLAN Netz verbunden. Die Verbindung zwischen dem Anzug und Axis Neuron Pro war somit gegeben.

Schritt 3: Unity, 3D Modell, Prefab, Kopftracking

Um die virtuelle 3D Umgebung zu erstellen wurden das Treppenhausmodell sowie die Palmen und der Ball als Prefabs mittels Drag&Drop in die Szene eingefügt. Die SDK zur Intergrierung des Perception Neuron Anzugs bietet das Prefab "Robot" welches ebenfalls durch Drag&Drop in die Umgebung gezogen werden konnte. Es beinhaltet das gleiche bio-

mechanischen Modell wie das in der Software Axis Neuron Pro. Dieses Modell wurde mit dem Skript "NeuronAnimatorInstance.cs", welches von der SDK mitgeliefert wurde, belegt. Es transformiert die Neuron Motion Daten auf das Skelett. Um dies zu ermöglichen, musste die lokale IP Adresse des PCs im Inspector angegeben werden, da Axis Neuron Pro und Unity 3D auf dem selben Rechner ausgeführt werden (siehe Abbildung 4.3). In diesem Fall war es die IP Adresse 127.0.0.1. Außerdem wurde die Portnummer, auf der die BVH Daten von Axis Neuron Pro gesendet werden, angeglichen. Dies ist unabdingbar, da bei unterschiedlichen Ports es zu keinem erfolgreichen Datenaustausch kommen kann bzw. Unity die Streamingdaten von Axis Neuron nicht empfängt.

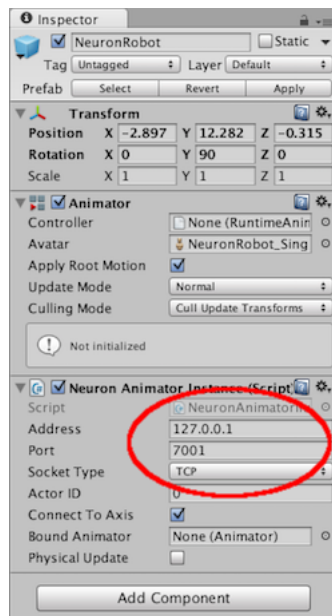


Abbildung 4.3: Robot-Einstellungen über den Inspector

Schritt 4: Anschluss Oculus Rift

Um die Virtual-Reality-Brille in Unity nutzen zu können bedurfte es nur einer kleinen Einstellungsänderung in den sog. "Player Settings", indem die Einstellung "Virtual Reality Supported" aktiviert wurde (siehe Abbildung 4.4). Mit dieser Änderung wurde der statische Blickwinkel der "Main Camera" durch eine variable Sicht, abhängig von der Positionsänderung der VR-Brille, ersetzt. Um eine realwirkende Perspektive zu erhalten, musste die Main Camera auf Kopfhöhe positioniert werden. Damit die "Main Camera" den Bewegungen des biomechnischen Modells folgt und nicht an den angegebenen Koordinaten stehen bleibt, musste diese in der Hierarchie als sog. "Child-Object"¹⁹ unter

¹⁹Child-Object: Positionskoordinaten beziehen sich auf das , "Parent-Object, dem höchstgestellten Objekt in der Hierarchie. Verändert sich die Position des "Parent-Object", so verändert sich auch das "Child-Object"

”Robot_Head” im Robotverzeichnis abgelegt werden (siehe Abbildung 4.4).

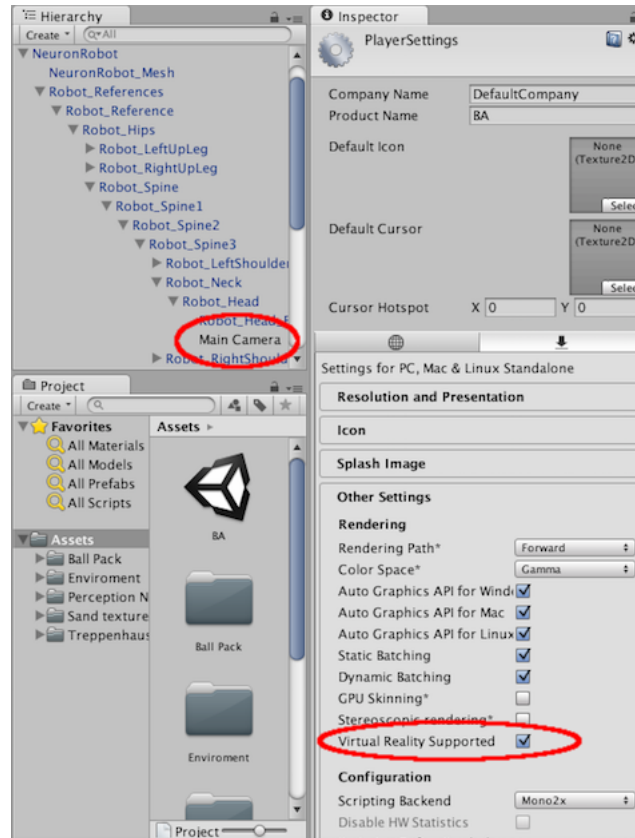


Abbildung 4.4: Playereinstellungen und Main Camera als ”Child-Object”

Um das auftretende Problem, dass sich das Bild bei einer 180 Grad Kopfdrehung um 360 Grad dreht liegt daran, dass sich die Bewegungsänderungen des Kopfsensors und der Oculus Rift addieren. Einerseits kann die Kopfrotation mittels eines Scripts halbiert werden. Andererseits kann im Script ”Neuron Animator Instance” des ”NeuronRobot” Modells die Wertübergabe vom Anzug auf das Neuron Modell in Zeile 246 gelöscht werden.

4.2 Versuch

Der Versuch selbst begann mit ersten Testläufen der Axis Neuron Pro Software (Kapitel 5.1). Hierbei wurde auf das Streaming in Unity 3D und somit auf das Tragen der Oculus Rift verzichtet. Lediglich die Bewegungen des Anzugs auf einer Ebene und beim Treppen steigen sollten aufgenommen und näher untersucht werden. Hierbei wurden verschiedene Einstellungsmöglichkeiten in der Software vorgenommen und mögliche Verbesserungen bzw. Verschlechterungen untersucht. Zu den Einstellungen zählen: ”Anti Mag” und ”Ground constrain”.

”Anti-Mag” dient, wie der Name es schon vermuten lässt, dazu, möglich auftretende

magnetische Felder und die daraus resultierenden Fehler bei der Positionserkennung der Sensoren herauszurechnen.

”Ground Constrain” kann aktiviert bzw. deaktiviert werden. Ist Ground Constrain aktiv und die Bewegungen finden auf einer Ebene statt, dient der Boden somit als direkter und fixer Bezugspunkt. Beim Leiter hochsteigen oder Treppen laufen, sollte Ground Constrain deaktiviert werden, da der Anwender auf verschiedenen hohen Ebenen interagiert.

Die aufgenommenen Bewegungen wurden dann ohne Nachbearbeitung in Unity gestreamt, um zu testen, wie stark der aufkommende Drift ist. Vor jedem Durchlauf wurde der Anzug mit den dementsprechenden Posen neu kalibriert um eine optimale Ausgangspointion zu haben.

Nachdem die Vorbereitungen abgeschlossen waren, wurde das komplette Virtual Reality Setup getestet. Anfangs wurde der Test auf einem Gang im Treppenhaus und somit auf einer Ebene ausgeführt. Einfache Bewegungen sollten hier untersucht werden, sowie die Interaktion mit einem virtuellen Ball. Außerdem wurde geprüft, ob der Anwender wirklich gerade läuft und nicht zu einer Seite wegdriftet.

Dann folgte die Untersuchung, ob in einer virtuellen Umgebung das Steigen realer Treppen möglich ist. Oberste Regel war es, die Treppen hoch zu laufen, da man im Falle eines Sturzes die Treppe hinauf fällt und nicht hinunter, was einen wesentlich größeren Schaden für die Testperson zur Folge hätte. Während dieser Test wurde ebenfalls das Aufkommen von ”Motion Sickness” bzw. Unwohlsein beim Anwender geprüft.

5 Ergebnisse

Die Konfiguration der einzelnen Systeme ist durch zahlreiche Dokumentationen nicht sehr schwer. Durch SDKs, die von den Herstellern zum Download bereit gestellt werden, war es theoretisch relativ einfach, die Programme aufeinander abzustimmen. Jedoch gab es zwei praktische Probleme, die sich nicht logisch erschlossen haben.

Zum Einen kam es häufiger zum Absturz von Unity 3D, sobald die Oculus Rift angeschlossen war. Dieser Fehler kann an der neuesten Unity Softwareversion 5.3.2f1 liegen, mit der in diesem Versuch gearbeitet wurde. Sobald nun die Unity Einstellung "Virtual Reality Support" aktiviert wurde, stürzte Unity ab. Dies geschah auch bei komplett neuen Projekten. Die Lösung dieses Problems war ein Neustart der Oculus Rift Software "Runtime".

Zum Anderen erfolgte die Verbindung von Axis Neuron Pro und Unity willkürlich. Das kann daran gelegen haben, dass es sich bei der Neuron Software um eine Beta Version gehandelt hat. Bei früheren Testläufen mit der Standard Axis Neuron Version ist dieser Fehler nicht aufgetreten. Die Ursache des Problems wurde nicht herausgefunden. Durch mehrmaliges Neustarten der Software oder des kompletten Rechners bzw. das An- und Ausschalten von verschiedenen Streaming Optionen in beiden Programmen, konnte eine Verbindung zwischen Axis Neuron Pro und Unity 3D hergestellt werden.

5.1 Axis Neuron Pro

Bei den ersten Tests ging es um die Aufnahme der Bewegungen des Motion Capturing Anzugs. Hierbei wurden lediglich die Bewegungen auf einer Ebene und beim Treppensteigen aufgenommen.

Die ersten Feststellung war, dass ohne Schuhe, die Sensoren am Fuß den metallischen Gegenständen im Boden direkt ausgesetzt waren. Die Folge war eine sofortige komplette "Fehlstellung" der Füße. Einen großen Unterschied machte es dann, als über den Sensoren Schuhe getragen wurden. So half die Gummisohle deutlich, mögliche elektromagnetische Felder des Bodens zeitweise zu unterdrücken. Nach einer gewissen Zeit war jedoch eine Drehung der virtuellen Füße unvermeidbar. Gerade beim Treppen steigen war ein solcher Fehler festzustellen (siehe Abbildung 5.1).



Abbildung 5.1: virtuelle Rotation des linken Fußes

Generell hielt die Kalibrierung des Motion Capturing Anzugs nicht lange vor. Aufgrund der vielen metallischen Gegenständen in der Umgebung wie dem Fahrstuhl oder dem Metallstreifen auf dem Boden der Brandschutztür, war der Anzug vielen elektromagnetischen Feldern ausgesetzt. Außerdem könnten Stromkabel, die in den Wänden und im Boden verliefen, die Sensoren in ihrer Positionierung stören.

Die Folge war, dass sich das biomechanische Modell verformte und die natürlichen Bewegungen des Trägers des MoCap Anzugs mit der Zeit nicht mehr zu erkennen waren. Später glichen die Bewegungen denen von Marionetten.

Warum die Trackingsensoren so empfindlich auf externe Störeinflüsse reagieren, liegt an dem Aufbau der Sensoren selbst. Die verwendeten Bauteile, wie Kondensatoren oder Spulen und die geringen Spannungen, sind äußerst anfällig für metallische Objekte und Elektrogeräte, die ihre elektromagnetischen Felder einstreuen.

Die Aufnahmen, die beim Laufen auf einer Ebene im Treppenhaus entstanden sind, weisen generell nur Unterschiede im Aussehen der Bewegungen auf. Des Weiteren wurde darauf geachtet, dass der Läufer zu seinem Startpunkt zurückkehrt, um womöglich auftretenden Drift zu erkennen.

Die Einstellungsoption "Ground Constrain" war bei allen Läufen aktiviert, da sich die Testperson auf einer gleichbleibenden Ebene bewegt hat und das Modell in "Axis Neuron Pro" sonst bei jedem Schritt in den Boden gerutscht wäre (siehe Abbildung 5.2).



Abbildung 5.2: Fuß versinkt in den Boden bei deaktivierter "ground constrain"

Sobald die Option Anti-Mag aktiviert war, war eine leichte Verbesserung bei der Wiederkehr in die Ausgangsposition zu erkennen.

Ob allerdings das Verhältnis der Bewegungen und der zurückgelegten Distanz mit der Realität übereinstimmt, konnte mit Hilfe von Unity geklärt werden. Die Aufzeichnungen wurden in das 3D Treppenhaus gestreamt. Wichtig war auch hier, dass die Startpositionen übereinstimmen, da es sonst beim Start schon zu Verfälschungen gekommen wäre. Allerdings wurde bereits nach den ersten Metern deutlich, dass die Relationen zwischen der Realität und der aufgenommenen Strecke nicht übereinstimmen (siehe Abbildung 5.3).

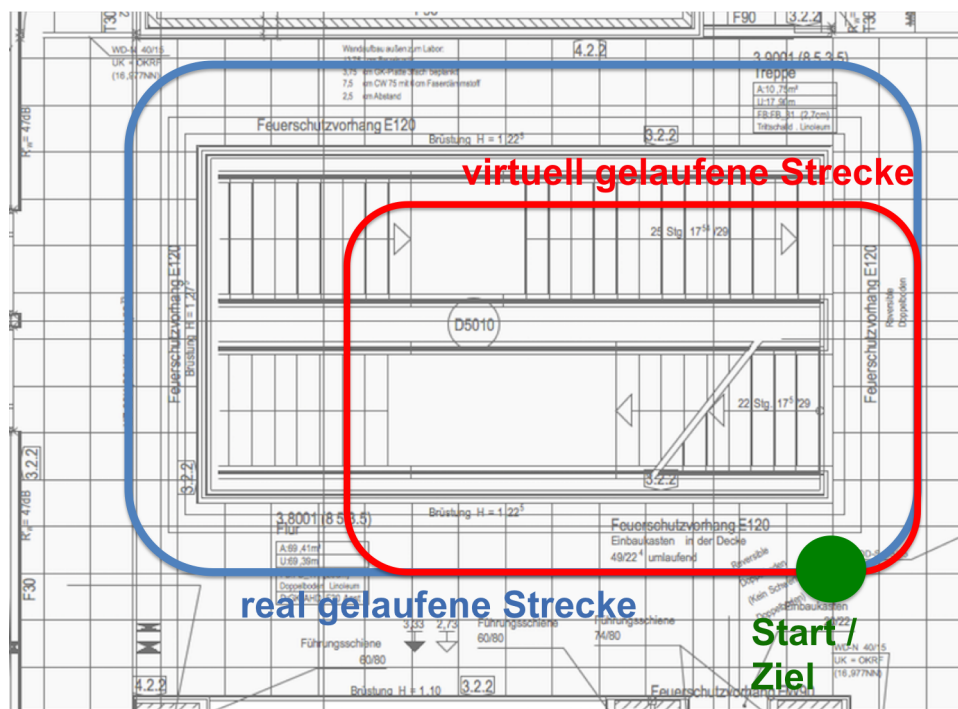


Abbildung 5.3: zurückgelegter virtueller Weg verglichen mit der Realität

Gerade die Sensoren an den Füßen müssen exakt arbeiten um beispielsweise die Schrittgröße richtig zu messen. Allerdings wurden diese dahingehend beeinflusst, dass die reale Fußposition nicht richtig weitergegeben wurde. So befand sich das virtuelle Modell in der 3D Welt, im Verhältnis zur Realität, an einem falschen Ort (siehe Abbildung 5.4). Das Modell ist jedoch relativ exakt zu dem Startpunkt zurückgekehrt. Das bedeutet, dass am Start- bzw. Endpunkt weniger bis gar kein Drift vorhanden war. In anderen Bereichen des Treppenhauses hingegen müssen sich stärkere elektromagnetische Felder befinden, welche die Sensoren dementsprechend negativ beeinflusst haben.

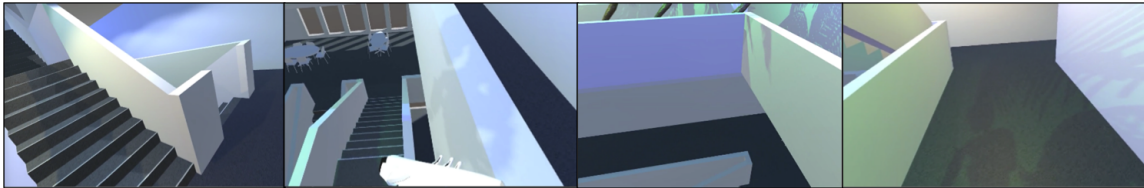


Abbildung 5.4: Axis Neuron Aufnahmen in Unity gestreamt

Damit beim Treppensteigen die Füße des Modells in Axis Neuron nicht immer wieder auf den Boden sinken, wurde die Option "Ground Constrain" deaktiviert. Allerdings war in den Aufnahmen schon zu sehen, dass die Füße trotz alledem bei jeder Stufe unterschiedlich stark nach unten sanken. Als die Aufnahmen ebenfalls in Unity gestreamt wurden, war zu erkennen, dass der MoCap-Anzug einem starken Drift ausgesetzt war. Um den elektromagnetischen Feldern und weiteren Störeinflüssen aus dem Weg zu gehen, wurde eine weitere Szene auf einer Treppe außerhalb des Gebäudes aufgenommen. Inklusiv der Option "Anti-Mag" ist eine deutliche Verbesserung zu sehen. Die Bewegungen des Modells laufen flüssiger und es sieht im Allgemeinen realistischer aus. Jedoch mit zunehmender Treppenhöhe, beginnen auch hier die Füße ein wenig nach unten zu driften. Außerdem verdreht sich mit der Zeit das rechte Bein. Grund dafür könnte wiederum ein magnetisches Feld sein. Da die Treppe vom Innenhof links entlang der Kantine verläuft, ist es möglich, dass Leuchten oder Lüfter der Kantine, die viel Strom verbrauchen, Grund für das Störfeld sind. Um einen wirklich realistischen Treppenauf- bzw. Treppenabstieg darzustellen, ist es unabdingbar, eine Nachbearbeitung der sog. "Contact points" durchzuführen. Mit dieser Nachbearbeitung legt man die Höhe der Bezugsebene, auf der sich der jeweilige Fuß gerade befindet, fest.

5.2 VR Setup auf einer Ebene

Bei den zweiten Testdurchläufen wurde das komplette Setup in Betrieb genommen. Die Bewegungen des vom Anwender getragenen Motion Capturing Anzugs wurden über Axis

Neuron Pro in Echtzeit in Unity gestreamt und die Inhalte aus Unity direkt auf die Oculus Rift DK 2, die wiederum der Anwender trägt, übertragen (siehe Abbildung 5.5).



Abbildung 5.5: komplettes VR-Setup

Die Bewegungen sahen anfangs sehr realistisch aus und das Gehen auf einer Ebene war für eine gewisse Zeit möglich. Nach ca. einer halben Minute jedoch war der Drift so stark, dass keine gute Orientierung im Raum mehr möglich war und der Anwender links bzw. rechts gegen Wände gelaufen ist, die laut den gezeigten Inhalten auf der Oculus Rift noch weiter entfernt sein müssten. Außerdem hat sich das biomechanische Skelett verdreht. So zeigten beispielsweise die Füße und Beine beim geradeaus Gehen in eine falsche Richtung. Darüber hinaus ist es ebenfalls vorgekommen, dass der virtuell zurückgelegte Weg viel kleiner war als der in der Realität. Bei diesem Punkt ist es tatsächlich zu einer leichten Form von Motion Sickness gekommen, da große Schritte getätigt wurden, sich das VR-Modell jedoch in der 3D Welt kaum fortbewegt hat. Es entstand ein kurzes Übelkeitsgefühl und die Beine haben gekribbelt. Dieser Zustand hielt aber nicht lange an, sodass nach einer kurzen Pause die Symptome nachließen und ohne die VR-Welt zu verlassen, weiter darin agiert werden konnte.

Die Stärke des Drifts und die Verformungen waren davon abhängig, wo sich der Anwender bewegt hat bzw. wo sich starke elektromagnetische Störfelder befinden, die die Inertialsensoren des Anzugs stören. Auf dem Boden befand sich ein ca. 10 cm breiter Metallstreifen der Brandschutzwände. Fanden die Bewegungen über diesem statt, kam es sofort zu Drehungen in den Beinen und Füßen. Die im vorherigen Absatz erklärte Problematik des virtuellen nicht Vorankommens war vor allem am Anfang des Gangs im Treppenhaus zu

spüren, da sich in unmittelbarer Nähe ein Fahrstuhl befindet und dieser ein großes Störfeld produziert (siehe Abbildung 5.6).

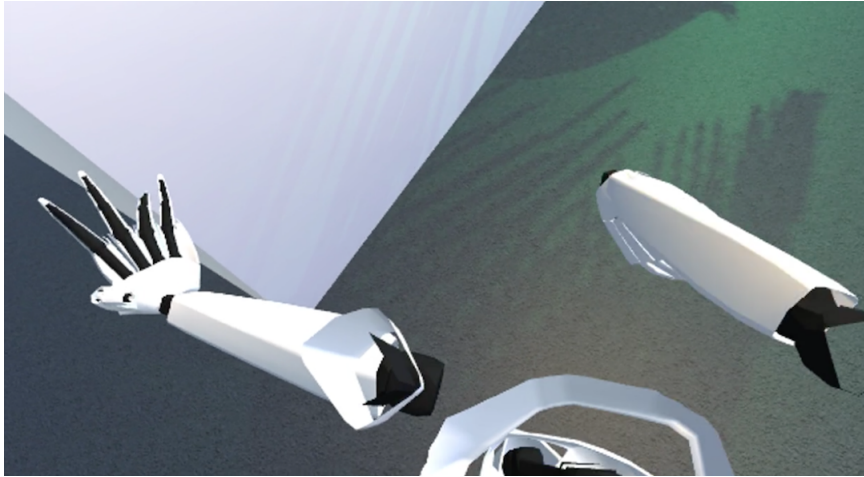


Abbildung 5.6: Verformung der Arme in Folge der elektromagnetischen Störung durch den sich in unmittelbarer Nähe befindenden Fahrstuhls

Der virtuelle Ball konnte ohne weiteres gespielt werden. Das Verhalten des Balles war davon abhängig, mit welchen physikalischen Eigenschaften dieser ausgestattet wurde. Ob der Ball hätte aufgehoben werden können, konnte nicht untersucht werden, da sich die Arme und Hände, je dichter diese dem Boden kamen, verdreht haben und nach außen zeigten. Die Folge war eine Neukalibrierung des MoCap Anzugs. Beim Spielen des Balles hat das Gefühl etwas zu Berühren gefehlt, genauso wie beim Berühren der Blätter der virtuellen Palmen (siehe Abbildung 5.7). Durch dieses fehlende Gefühl und der fehlerhaften Rotation des Skeletts wurde das immersive Gefühl gestört.



Abbildung 5.7: Bei Berührungen eines virtuellen Objektes fehlt das haptische Feedback

5.3 VR Setup auf der Treppe

Bei dem dritten Testdurchlauf wurde ebenfalls das Zusammenspiel des kompletten VR-Setups überprüft. Nun galt es das Treppensteigen in einer virtuellen Umgebung zu untersuchen.

Der Startpunkt war einen Meter vor der ersten Stufe. Anfangs stimmten die Ausrichtungen des biomechanischen Skeletts und die reale Position überein. Jedoch schon nach dem ersten Schritt Richtung Stufe störten wiederum elektromagnetische Felder die Sensorik, die Füße begannen sich zu verdrehen und der ganze Körper driftete nach rechts. Die Füße versanken in den Stufen und nach kurzer Zeit war der Drift so stark, dass die virtuelle Treppe verlassen wurde und das virtuelle Modell in der Luft schwebte (siehe Abbildung 5.8).



Abbildung 5.8: Vergleich VR mit Realität: Die Füße sind in den Treppenstufen versunken und der Körper driftet nach rechts

Es war nicht möglich freihändig die Stufen hinauf zu gehen, da es sonst schnell zu einem Sturz hätte kommen können. Motion Sickness trat allerdings nicht auf, da die Bewegungen der virtuellen Welt im Vergleich zu den Bewegungen in der Realität so surreal waren, dass der Kopf erkannt hat, dass das Gesehene nicht der Wirklichkeit entspricht. Aufgrund dessen, dass die Stufen nicht freihändig hinauf gestiegen werden konnten, hielt sich die Testperson am Geländer der Treppe fest. Somit hatte diese einen festen Halt, was ebenfalls einem möglichen Aufkommen von Übelkeit entgegenwirkte. Wie in Abbildung 5.9 deutlich zu erkennen, befindet sich der virtuelle Körper neben der Treppe und schwebt in der Luft obwohl die Testperson sich auf der Treppe Höhe der neunten Stufe befindet.



Abbildung 5.9: Aufgrund des starken Drifts nach rechts befindet sich die Testperson virtuell neben der Treppe

Da es nicht möglich war, längere Testläufe zu absolvieren, kam es nicht zu einer kompletten Immersion, und es entstand zu keinem Zeitpunkt das Gefühl in einer anderen Welt zu sein. Die aufkommenden Fehler waren zu drastisch, die Konzentration beschränkte sich eher darauf nicht zu fallen bzw. zu stolpern, als sich wirklich auf die 3D Welt einzulassen. Allerdings gilt es festzuhalten, dass ein Motion Capturing Anzug nicht für solche Anwendungen konzipiert wurde. Er dient ursprünglich der Aufzeichnung von Bewegungen, die in den meisten Fällen noch nachbearbeitet werden. Ein VR-System mit einem inertialbasierten MoCap-Anzug aufzubauen, kann nur in einer kontrollierten Umgebung durchgeführt werden. Somit können Fehlerquellen wie elektromagnetische Störfelder minimiert werden. Aber selbst dann wird es nach einer gewissen Zeit zu Trackingfehlern kommen, da diesem Trackingsystem ein Fixpunkt fehlt. Die Sensoren bekommen lediglich bei der Kalibrierung über verschiedene Körperposen einmal ihre Position mitgeteilt, danach nicht mehr. Die Positionsmessungen können nun jederzeit durch äußere Störfaktoren beeinflusst werden. Diese Fehler bleiben so lange erhalten, bis eine Neukalibrierung erfolgt.

6 Fazit

Ein VR-System mit einem Motion Capturing Anzug, Unity 3d und einer Oculus Rift aufzubauen, ist ohne weiteres möglich, aber nicht frei von Positionsfehlern. Diese können durch eine Kombination verschiedener Trackingsysteme beseitigt werden.

6.1 Zusammenfassende Bewertung

Die Software, welche die einzelnen Technologien miteinander verknüpft ist Unity. Die Entwickler stellen mit den SDKs wie "Oculus Utility" oder "Perception Neuron Unity Integration" die wichtigsten Inhalte für eine einfache und erfolgreiche Kalibrierung der jeweiligen Hardware in Unity bereit. Der Umgang mit Unity selbst war nach einer kurzen Einarbeitungsphase kein Problem. Ein großer Vorteil war, das Treppenhaus Modell in einem Architekturprogramm zu konstruieren und dann einfach per Drag&Drop in die 3D Engine zu laden. Einzig die recht eigenwillige Kommunikation zwischen Unity 3D und Axis Neuron Pro sorgte für Probleme, was jedoch daran gelegen haben könnte, dass es sich bei der Neuron Software um eine Beta-Version gehandelt hat. Auch die Konfigurierung der Oculus Rift in Unity erfolgte ohne Probleme wobei es hin und wieder zu Abstürzen von Unity gekommen ist, sobald die Rift angeschlossen war. Bei diesem Fehler könnte die verwendete Unity Version eine Rolle gespielt haben. Es wurde die zu diesem Zeitpunkt aktuellste Version 5.3.2f1 verwendet. Bei früheren Versionen war dieser Fehler nicht bekannt.

Hardwaretechnisch hat alles funktioniert. Nur war dort mit dem auf Inertialsensoren basierende Motion Capturing Anzug "Perception Neuron" der Firma Noitom die größte Fehlerquelle vertreten. Aufgrund des Aufbaus und der Funktionsweise der Sensoren zur Positionserkennung ist die korrekte Positionswiedergabe stark von der Umgebung abhängig. Metallische Gegenstände sowie elektrische Geräte und deren elektro-magnetische Felder stören die Inertialsensoren so stark, dass es zu massivem Drift kam. So befanden sich nicht nur die virtuellen Körperteile an den falschen Positionen, oder waren extrem verdreht, auch die beim Gehen zurückgelegte Strecke stimmte nicht mit der Realität überein. Auf einer Ebene war es möglich für eine gewisse Zeit in die AV einzutauchen und das Virtual Reality Erlebnis zu genießen. Beim Treppensteigen war dies unmöglich, da die Trackingfehler sehr stark waren und keine Immersion entstehen konnte.

Nicht zu vergessen ist jedoch, dass ein solcher Tracking Anzug nicht dafür hergestellt wurde, um durch eine virtuelle Welt zu gehen, sondern um Bewegungen aufzuzeichnen. Somit ist die Leistung des Anzugs enorm groß und in einem kontrollierten Umfeld könnte ein noch besseres Tracking, ohne Drift und Positionsfehler, erreicht werden.

6.2 Ausblick

Eine höhere Mobilität könnte erreicht werden, würde das HMD über Funk mit dem PC verbunden werden. Außerdem könnten so Stolpergefahren als auch zusätzliche elektromagnetische Störfelder vermieden werden.

Der große Vorteil des inertialen Trackingverfahrens ist die hohe Framerate und somit die hohe Bewegungserkennung. Allerdings ist das System sehr anfällig gegen äußere Störeinflüsse und besitzt keine festen Referenzpunkte. Würde dieses System durch solche ergänzt werden, beispielsweise durch ein optisches Infrarot-Trackingsystem mit Markern, könnten solche Punkte gesetzt werden. Durch die besondere Beschaffenheit der Marker, werden diese sehr genau von der Kamera erfasst. Sollte es zu einer Verdeckung kommen, das bedeutet, die Kamera kann den Marker optisch nicht mehr erfassen, könnten die Bewegungsänderungen über den inertialbasierten Trackinganzug erkannt und berechnet werden. Kleinste Veränderungen, wie beispielsweise das Bewegen der einzelnen Finger, würde ebenfalls über den Motion Capturing Anzug erkannt werden. Somit ließe sich eine optimale Bewegungserkennung auch in schwierigen Umgebungen realisieren.

Abbildungsverzeichnis

2.1	biomechanisches Softwaremodell https://www.gumstix.com/applications/motion-workshop/ , Stand: 15.01.2016	7
2.2	korrekte Sensorposition https://www.researchgate.net/profile/Per_Slycke/publication/239920367_Xsens_MVN_Full_6DOF_Human_Motion_Tracking_Using_Minature_Inertial_Sensors/links/0f31752f1f60c20b18000000.pdf , Stand: 15.02.2016	8
2.3	mechanischer MoCap-Anzug "Gypsy 7" der Firma "Meta Motion" http://www.metamotion.com/gypsy/Gypsy-6-torso.html , Stand: 15.01.2016	9
2.4	Perception Neuron Full-Body Anzug http://www.noitom.com/index.php/mocap/perception-neuron , Stand: 19.01.2016	10
2.5	Neuron Sensor http://www.noitom.com/index.php/mocap/perception-neuron , Stand: 19.01.2016	10
2.6	Neuron Hub http://www.noitom.com/index.php/mocap/perception-neuron , Stand: 28.01.2016	11
2.7	Axis Neuron https://i.ytimg.com/vi/LvkirChw4Nc/maxresdefault.jpg , Stand: 22.01.2016	12
2.8	menschliches Modell der Xsens Software "MVN Studio" https://www.xsens.com/products/mvn-biomech/ , Stand: 15.01.2016	13
2.9	Xsens MVN Anzug http://www.vrlab.ctw.utwente.nl/eq/Xsens.html , Stand: 15.01.2016	14

2.10	"Ted" aus dem gleichnamigen Film gespielt von Seth MacFarlane	
	http://www.ifc.com/2012/06/seth-mcfarlane-ted ,	
	Stand: 15.01.2016	15
2.11	Gears of War 1	
	http://www.ifc.com/2012/06/seth-mcfarlane-ted ,	
	Stand: 12.02.2006	17
2.12	Oculus Rift DK2	
	https://www1.oculus.com/order/ ,	
	Stand: 18.01.2016	21
2.13	Sonys Project Morpheus	
	http://www.vrnerds.de/sonys-project-morpheus/ ,	
	Stand: 15.01.2016	22
2.14	HTC Vive	
	http://www.vrnerds.de/htc-vive-das-steamvr-headset-von-htc/ ,	
	Stand: 15.01.2016	23
3.1	Motion Capturing Setup	
	Bildschirmfoto	
	Stand: 10.02.2016	25
3.2	Axis Neuron Setup	
	Bildschirmfoto	
	Stand: 12.02.2016	26
3.3	Bodentextur (Quelle: Vectorworks 2015)	
	Bildschirmfoto	
	Stand: 12.02.2016	28
3.4	Wandtextur (Quelle: Vectorworks 2015)	
	Bildschirmfoto	
	Stand: 12.02.2016	28
3.5	Stufenleistentextur (Quelle: Vectorworks 2015)	
	Bildschirmfoto	
	Stand: 12.02.2016	29
3.6	Fenstertextur (Quelle: Vectorworks 2015)	
	Bildschirmfoto	
	Stand: 12.02.2016	29
3.7	virtuelle Palmen 3D Treppenhausmodel	
	Bildschirmfoto	
	Stand: 22.02.2016	29

4.1	Arbeitsplatz im Treppenhaus des Neubaus der HAW Hamburg	
	Handyfoto	
	Stand: 24.02.2016	31
4.2	Routerkonfiguration, IP Zuweisung mittels DHCP	
	Bildschirmfoto	
	Stand: 22.02.2016	32
4.3	Robot-Einstellungen über den Inspector	
	Bildschirmfoto	
	Stand: 22.02.2016	33
4.4	Playereinstellungen und Main Camera als "Child-Object"	
	Bildschirmfoto	
	Stand: 22.02.2016	34
5.1	virtuelle Rotation des linken Fußes	
	Bildschirmfoto	
	Stand: 24.02.2016	37
5.2	Fuß versinkt in den Boden bei deaktivierter "ground constrain"	
	Bildschirmfoto	
	Stand: 26.02.2016	37
5.3	zurückgelegter virtueller Weg verglichen mit der Realität	
	Bildschirmfoto	
	Stand: 26.02.2016	38
5.4	Axis Neuron Aufnahmen in Unity gestreamt	
	Bildschirmfoto	
	Stand: 26.02.2016	39
5.5	komplettes VR-Setup	
	Bildschirmfoto	
	Stand: 26.02.2016	40
5.6	Verformung der Arme in Folge der elektromagnetischen Störung durch den sich in unmittelbarer Nähe befindenden Fahrstuhls	
	Bildschirmfoto	
	Stand: 26.02.2016	41
5.7	Bei Berührungen eines virtuellen Objektes fehlt das haptische Feedback	
	Bildschirmfoto	
	Stand: 26.02.2016	41
5.8	Vergleich VR mit Realität	
	Stand: 26.02.2016	42

5.9 Aufgrund des starken Drifts nach rechts befindet sich die Testperson virtuell
neben der Treppe
Stand: 26.02.2016 43

Tabellenverzeichnis

2.1	VR-Brillen im Vergleich	
	http://www.vrnerds.de/hardvr/96-2/vr-brillen-vergleich/ ,	
	Stand: 15.01.2016	24
3.1	Flächentextur des Treppenhausmodels	28
3.2	Vergleich Anforderungen Oculus Rift DK 2 mit Tower PC	30

Literaturverzeichnis

- [1] AMD: 16K Auflösung pro Auge für echte Immersion notwendig
<http://www.vrnerds.de/amd-16k-aufloesung-pro-auge-fuer-echte-immersion-notwendig/>,
2015
letzter Zugriff: 27. 02. 2016
- [2] Auralisation
<http://www.muellerbbm.de/bau/raumakustik/auralisation/>
letzter Zugriff: 07. 01. 2016
- [3] AXIS NEURON SOFTWARE
<https://neuronmocap.com/content/axis-neuron-software>
letzter Zugriff: 15. 02. 2016
- [4] Baumann, Tobias; Hao, Tang; He, Yuanhui; Shoda, Rick; "PERCEPTION NEURON
UNITY HANDBOOK"
[https://neuronmocap.com/sites/default/files/downloads/unity_](https://neuronmocap.com/sites/default/files/downloads/unity_integration.pdf)
[integration.pdf](https://neuronmocap.com/sites/default/files/downloads/unity_integration.pdf), 2015,
letzter Zugriff: 12. 02. 2016
- [5] Biovision BVH
<http://research.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html>
letzter Zugriff: 12. 02. 2016
- [6] Damgrave, R.G.J.; Lutters, D. "The Drift of the Xsens Moven Motion Capturing Suit
during Common Movements in a Working Environment"
<https://core.ac.uk/download/files/23/139022.pdf>, 2009,
letzter Zugriff: 15. 02. 2016
- [7] FMOD
<http://www.fmod.org/about-us/>
letzter Zugriff: 12. 02. 2016
- [8] GOING UP AND DOWN STAIRS.
<https://neuronmocap.com/forum/going-and-down-stairs>
letzter Zugriff: 06. 02. 2016

- [9] Head Mounted Displays & Datenbrillen
<http://www.vdc-fellbach.de/wissen/vr-hardware/head-mounted-displays>
letzter Zugriff: 17. 01. 2016
- [10] HTC Vive
<http://www.htcvive.com/us/>
letzter Zugriff: 15. 02. 2016
- [11] HUB WIFI SPECIFICATIONS
<https://neuronmocap.com/forum/hub-wifi-specifications>
letzter Zugriff: 10. 02. 2016
- [12] Kaur, Kal "Motion Capture Sensor Systems"
<http://www.azosensors.com/Article.aspx?ArticleID=43>
letzter Zugriff: 15. 02. 2016
- [13] Kolorz, Niklas "Schnell und Einfach: Game-Engines"
[http://www.giga.de/extra/schnell-und-einfach/videos/schnell-und-einfach-game-engines/,2015,](http://www.giga.de/extra/schnell-und-einfach/videos/schnell-und-einfach-game-engines/,2015)
letzter Zugriff: 28. 02. 2016
- [14] Krosta, Michael "Special: Playstation VR *FAKTEN SPIELE & POTENZIELLE HIGHLIGHTS*"
http://www.4players.de/4players.php/dispsbericht/PlayStation4/Special/36856/81593/0/PlayStation_VR.html, 2015,
letzter Zugriff: 20. 02. 2016
- [15] Luinge, Henk; Roetenberg, Daniel; Slycke, Per "Xsens MVN: Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors"
https://www.researchgate.net/profile/Per_Slycke/publication/239920367_Xsens_MVN_Full_6DOF_Human_Motion_Tracking_Using_Minature_Inertial_Sensors/links/0f31752f1f60c20b18000000.pdf, 2009,
letzter Zugriff: 20. 02. 2016
- [16] Popa, Justin "OCULUS RIFT: DK1 VS DK2"
<http://in2gpu.com/2014/08/10/oculus-rift-dk1-vs-dk2/>, 2014,
letzter Zugriff: 20. 02. 2016
- [17] PlayStation VR / Sony Morpheus
<http://www.vrbrillen.net/sony-morpheus/>
letzter Zugriff: 15. 02. 2016

- [18] Schmädig, Benjamin "Special: Virtual Reality *DIE GESCHICHTE DER VIRTUAL REALITY*"
http://www.4players.de/4players.php/dispatcher/Allgemein/Special/37321/81590/0/Virtual_Reality.html?refresh, 2015,
letzter Zugriff: 15. 02. 2016
- [19] Sonys Project Morpheus
<http://www.vrnerds.de/sonys-project-morpheus/>, 2014,
letzter Zugriff: 15. 02. 2016
- [20] Stelzmann, Robert "Partikelsysteme "
<https://www.inf.tu-dresden.de/content/institutes/smt/cg/teaching/seminars/ProseminarSS08/rstelzmann/Vortrag.pdf>, 2008,
letzter Zugriff: 07. 01. 2016
- [21] Unity *THE LEADING GLOBAL GAME INDUSTRY SOFTWARE*
<https://unity3d.com/public-relations>
letzter Zugriff: 15. 12. 2016
- [22] Unity, Learn with Unity *ROLL-A-BALL Tutorial*
<https://unity3d.com/learn>
letzter Zugriff: 15. 02. 2016
- [23] Unreal Engine
<https://www.unrealengine.com/what-is-unreal-engine-4>
letzter Zugriff: 15. 02. 2016
- [24] VR Brillen Vergleich
<http://www.vrnerds.de/hardvr/96-2/vr-brillen-vergleich/>
letzter Zugriff: 15. 01. 2016
- [25] Witt, Hendrik "WETTBEWERBSVORTEIL DURCH HEAD-MOUNTED DISPLAYS?"
<http://www.ubimax.de/index.php/de/news/publications/item/67-wettbewerbsvorteil-hmd>, 2014,
letzter Zugriff: 27. 02. 2016
- [26] Wöbbeking, Jan "Special: Virtual Reality *IMMERSION ODER ÜBELKEIT*"
http://www.4players.de/4players.php/dispatcher/Allgemein/Special/37321/81590/0/Virtual_Reality.html?refresh, 2015,
letzter Zugriff: 15. 02. 2016

- [27] Wöbbeking, Jan "Vorschau: HTC Vive *AUSFLUG AUF HOLOECK*"
http://www.4players.de/4players.php/dispbericht/PC-CDROM/Vorschau/36466/81549/1/HTC_Vive.html, 2015,
letzter Zugriff: 15. 02. 2016
- [28] Xsens Company
<https://www.xsens.com/company/>
letzter Zugriff: 15. 02. 2016
- [29] Xsens Motion Capture
<https://www.xsens.com/tags/motion-capture/>
letzter Zugriff: 15. 02. 2016
- [30] Xsens MVN
<https://www.xsens.com/products/xsens-mvn/>
letzter Zugriff: 15. 02. 2016