

Eine VR-Montageanwendung: Entwicklung und Evaluation

Masterarbeit

Alexander Zimmermann

Matrikelnummer: 69 28 69

Medien- und Kommunikationsinformatik

Hochschule Reutlingen

Alteburgstraße 150

72762 Reutlingen

Abgabetermin: 30. September 2015

Erstprüfer: Prof. Dr. Uwe Kloos

Zweitprüferin: Prof. Dr. Gabriela Tullius

Zusammenfassung

Durch die derzeitige Entwicklung auf dem Markt der VR-Technologien ist es möglich, dass Bewegungsaufnahmekameras und Headmounted-Displays auch für kleine und mittelständische Unternehmen, oder Endanwender verfügbar werden. In dieser Arbeit soll eine VR-Anwendung im Einsatz der Lehre von Technikern entwickelt, und auf ihre Akzeptanz bei Anwendern getestet werden. Dabei wird die Bewegungserkennung über Microsoft Kinect v2 umgesetzt, die Anwendung in Unity3d entwickelt, und die Ausgabe über die Oculus Rift DK2 bewerkstelligt. In Zusammenarbeit mit Anlagenbauer ‚Alfred Becker GmbH‘ und dem Landwirtschaftsmaschinenhersteller ‚Amazonenwerke H. Dreyer GmbH & CO. KG‘ wurden Anforderungen an das System gesammelt. Die Prüfung entgegen dieser Anforderungen, und die Nutzertests zur Analyse der Akzeptanz des entwickelten Systems, sind ebenfalls Teil dieser Masterarbeit. Die Auswertung ergab, dass die für diese Arbeit entwickelte Anwendung durch ihre Anpassbarkeit für die Verwendung bei Schulungen geeignet ist. Sowohl in Hinsicht auf die Erfüllung der Anforderungen, als auch auf die Akzeptanz bei Nutzern.

Abstract

The current trends and developments of VR technologies enable, that small and medium enterprises or end users can afford motion capturing devices and head mounted displays. In this work a VR application for use in education of technicians is developed and evaluated. To include movement detection in the project, the Microsoft Kinect v2 is used, in combination with the Oculus Rift DK2 for the output. The development environment is Unity3d. In cooperation with the plant engineers ‚Alfred Becker GmbH‘, and the agricultural machinery manufacturers ‚Amazonenwerke H. Dreyer GmbH & CO. KG‘ requirements were collected. The test against these requirements and the user testing to analyse the acceptance of the developed system are also part of this thesis. The evaluation showed, that the developed application meets the collected requirements, as well as the acceptance among users.

Inhaltsverzeichnis

Abbildungsverzeichnis	6
Tabellenverzeichnis	7
1 Einführung	8
2 Stand der Wissenschaft und Technik.....	10
2.1 CAD-Daten in Verwendung mit Spieleentwicklungsumgebungen....	11
2.2 Bildbasierende Bewegungsaufnahme	12
2.3 Motivationserhaltung in Übungsanwendungen und Rückkopplung über den Fortschritt	14
2.4 Microsoft KINECT in Übungs- und Testanwendungen	15
2.5 Montage in virtueller Welt	16
2.6 VR-Übungsanwendung.....	19
2.7 Interaktion über NUI mittels adaptierten Bewegungen	21
2.8 Oculus Rift, Kinect V2 und Unity3D	23
2.9 VR-Unterricht	24
2.10 Stand der Wissenschaft zusammengefasst	25
3 Ziele	27
3.1 Anforderungen an das System.....	27
3.1.1 Organisation	28
3.1.2 Generierung	28
3.1.3 Output	28
3.1.4 Inhalt.....	28
3.1.5 Umgebung	29
3.1.6 Interaktion	29
3.1.7 Nutzung	29
3.2 Lernmodule	29
4 Methoden und Material.....	31
4.1 Interaktions- und Anzeigegeräte	31
4.1.1 Microsoft Kinect v2.....	31
4.1.2 Oculus Rift VR	32

4.2	Entwicklungsumgebung	32
4.2.1	Unity 5.....	32
4.2.2	MonoDevelop.....	33
4.3	Usability-Tests.....	34
4.3.1	Induktive / deduktive Usability Tests	34
4.3.2	Lautes Denken	35
4.3.3	Fragebogen	36
4.3.4	Testpersonen	37
5	Vorbedingungen der Arbeit.....	38
5.1	Auswahl des Bewegungsaufnahmegeräts	38
5.2	Auswahl der Entwicklungsumgebung.....	40
5.2.1	Vorgang	40
5.2.2	Kriterien	41
5.2.3	Auswertung.....	42
6	Konzeption und Implementierung.....	45
6.1	Aufnahme der Anforderungen an das System	45
6.2	Vorgehen/Implementierung	48
6.2.1	Interaktion	48
6.2.2	CAD-Daten zu Objekten	50
6.2.3	Spielelogik	52
6.2.4	Ausgabe	52
6.3	Probleme	53
6.3.1	Cybersickness.....	53
6.3.2	Hindernisse	55
6.3.3	Trackingfragmente.....	56
6.3.4	Infrarotlicht des OVR	56
6.3.5	Trackinginkonsistenzen.....	56
7	Evaluation und Bewertung	58
7.1	Evaluation auf Basis der Anforderungen.....	58
7.2	Nutzertests	61
7.2.1	Auswertung der Fragebögen	62
7.2.2	Auswertung der Thinkaloud-Rückmeldungen.....	67
8	Diskussion	69

Literaturverzeichnis	70
Eidesstattliche Erklärung	73
Anhang	I
I. Glossar	I
II. Bewertung der Entwicklungsumgebungen	III
III. Anforderungen	IV
IV. Fragebogen	VI
V. Fragebogenauswertung des Nutzertests	IX
VI. Software	XI

Abbildungsverzeichnis

Abbildung 1: Bild aus CAD-Viewer [Bea2015]	11
Abbildung 2: Vergleich Razer Hydra zu bildbasiertem System [GrJa2014 S. 312]	13
Abbildung 3: JewelMine Spielausschnitt [LaKo*2012 S. 171]	14
Abbildung 4: VR-Aufbau nach Xia et al. [XiLo*2011 S. 384]	18
Abbildung 5: Microsoft Kinect v2 [Zim2015 S. 61]	31
Abbildung 6: Ausschnitt Unity3D	33
Abbildung 7: Leap Motion und OVR [Zim2015 S. 61]	38
Abbildung 8: Beispielarbeitsstation mit Leap Motion Tracking [Zim2015 S. 62]	40
Abbildung 9: Anforderungssammlung	45
Abbildung 10: Interviewleitfaden des Workshop-Interviews	47
Abbildung 11: CAD-Objekt im Unity-Editor	51
Abbildung 12: Die grünen Objekte sind fehlende Bauteile. Das gelbe Objekt ist der Platzhalter	52
Abbildung 13: Das Trackingsetup. Der schwarze Hintergrund soll dazu dienen, eventuelle Fragmente im Tracking zu reduzieren.	56
Abbildung 14: Vorbereitungsszene mit Quadern	62
Abbildung 15: Beispielfrage	62
Abbildung 16: Fragebogenauswertung Aufgabenangemessenheit	63
Abbildung 17: Fragebogenauswertung Selbstbeschreibungsfähigkeit	64
Abbildung 18: Fragebogenauswertung Erwartungskonformität	65
Abbildung 19: Fragebogenauswertung Lernförderlichkeit	66
Abbildung 20: Fragebogenauswertung Steuerbarkeit	66
Abbildung 21: Fragebogenauswertung Fehlertoleranz	67

Abbildung 22: Fragebogen Teil 1 (aus [Prü1997 S.4])	VI
Abbildung 23: Fragebogen Teil 2 (aus [Prü1997 S.5])	VII
Abbildung 24: Fragebogen Teil 3 (aus [Prü1997 S.6])	VIII
Abbildung 25: Auswertungsmatrix Teil 1	IX
Abbildung 26: Auswertungsmatrix Teil 2	X

Tabellenverzeichnis

Tabelle 1: Auswertung der VR-Übungsanwendung (siehe [MaVo2015 S. 12])	20
Tabelle 2: Benchmark Unreal Engine 4	42
Tabelle 3: Benchmark Unity 5	43
Tabelle 4: Symptome die nach Auswertung des Virtual Reality Symptom Questionnaires bei mindestens 20% der Probanden vorkam [DaNe*2014 S. 6]	53

1 Einführung

Maschinenhersteller, die ihre Mitarbeiter selbst ausbilden, nutzen für die Schulungen meist Frontalunterricht. Dabei werden PowerPoint-Folien, Bilder und Texte dazu verwendet, ein praktisches Thema zu lehren. Dieser Unterricht wird dadurch unterstützt, dass die Auszubildenden Übungen an extra dafür aufgearbeiteten Komponenten durchführen. Diese Komponenten, beispielsweise ein Motor, ist eine teure Investition, und sobald ein neuer Motor gebaut wird, muss auch das Übungsobjekt ausgetauscht werden. Zudem sind Aufgaben am praktischen Objekt aufwändig, weswegen sie meist in Gruppenarbeit bearbeitet werden müssen. Eine VR-Anwendung soll es ermöglichen, dass Auszubildende die Übungen alleine bearbeiten können. Ein großer Vorteil der virtuellen Umgebung ist, dass die Objekte, um die sich es in der Lerneinheit handelt, beliebig austauschbar sind. Das wachsende Angebot an VR-Hardware, angetrieben durch die Spiele- und Unterhaltungsindustrie, ermöglichen es, dass auch kleine und mittelständische Unternehmen für Schulungen auf VR-Technologien zurückgreifen können.

Im Rahmen des Projekts GLASSROOM¹ sollen VR- und AR-Systeme genutzt werden, um bestehendes Wissen zu verstärken oder neues Wissen zu bilden. Der VR-Teil zielt auf die oben angesprochene Problematik in der Ausbildung ab, das AR-System hingegen soll genutzt werden um Arbeiter bei der Arbeit zu unterstützen. Dabei arbeiten die Anlagenbauer ‚Alfred Becker GmbH‘ und Landwirtschaftsmaschinenhersteller ‚Amazonenwerke H. Dreyer GmbH & CO. KG‘ eng am Projekt mit. So soll sichergestellt werden, dass durch ständigen Austausch und Tests mit der richtigen Zielgruppe ein, in der Lehre anwendbares, Produkt entsteht. Durchgeführt wird das Projekt am Lehrstuhl für Informationsmanagement und Wirtschaftsinformatik (IMWI) der Universität Osnabrück und in Kooperation mit ‚IMC information multimedia communication AG‘, der Universität des Saarlandes und des ‚Fraunhofer-Institut für Arbeitswirtschaft und Organisation‘.

¹ www.imwi.uni-osnabrueck.de/projekte/glassroom.html

Im folgenden zweiten Kapitel wird ein Überblick über den Stand der Wissenschaft gegeben werden. Es sollen bereits bestehende Systeme aufgeführt und Grundlagen für diese Masterarbeit aufgezeigt werden. Das dritte Kapitel enthält die Anforderungen, welche an das hier entwickelte System gestellt werden, und welche Ziele mit dieser Masterarbeit erfüllt werden sollen. In Kapitel 4 sind die VR-Technologien sowie die verwendeten Werkzeuge aufgeführt. Zudem wird die Evaluationsmethode erläutert. Kapitel 5 beschreibt die Vorarbeit, die getätigt werden musste, um die Arbeit an dem praktischen Artefakt zu ermöglichen. Im sechsten Kapitel sind der Ablauf der Anforderungssammlung, sowie der Ablauf der Implementierung und der Probleme bei der Arbeit beschrieben. Kapitel 7 enthält die Evaluation des praktischen Produkts und eine kritische Betrachtung der Arbeit. Im achten und letzten Kapitel wird die Arbeit diskutiert und einen Ausblick auf weitere Arbeiten gegeben.

2 Stand der Wissenschaft und Technik

Um die Arbeit in Kontext zu stellen, sollen in diesem Kapitel relevante verwandte Arbeiten vorgestellt werden. Im ersten Unterkapitel wird auf ein Projekt von Nathan Beattie [Bea2015] eingegangen, welches er auf der Australian International Airshow 2015 (Avalon, Australien) im Februar 2015 präsentierte. Dort zeigte er, wie CAD-Modelle mittels Unity in VR zugänglich gemacht werden können. Die Arbeit in Kapitel 2.2 vergleicht ein Controller-basierendes und ein Bild-basierendes Interaktionsverfahren [GrJa2014]. Die Arbeit [LaKo*2012] von Belinda Lange et al. in Kapitel 2.3 stellt dar, wie Anwender einer VR-Trainingsapplikation zum Weiterüben motiviert werden. Kapitel 2.4 beinhaltet eine Arbeit, welche die Microsoft Kinect (in ihrer ersten Version) nutzt, um bestehende Übungs- und Testmethoden zu verbessern [Ki-Ad*2013]. Das fünfte Unterkapitel beschreibt einen Aufbau, der eine Montageübungsumgebung simuliert und den Nutzer mittels HMD und Trackingverfahren in die virtuelle Welt bringt [XiLo*2011]. Die Übungsanwendung in Kapitel 2.6 greift die gleiche Idee auf, verwendet hierfür aber Endnutzer-Hardware [MaVo2015]. Das siebte Unterkapitel beschäftigt sich mit dem Einsatz von natürlichen Gesten in VR-Anwendungen [HaBa2015]. Kapitel 2.8 beschreibt den gleichen Interaktionsansatz der auch in dieser Thesis verfolgt wurde, definiert dabei aber kein konkretes Anwendungsszenario [GrRo2014]. Das letzte Unterkapitel zielt auf den Einsatz von VR als Unterrichtsmittel ab [ShCh2014].

Sucht man in Google Scholar² nach „Virtual Reality“, so erhält man 2.200.000 Treffer. „Head-mounted-Display“ ergibt 1.010.000 Treffer. „Microsoft Kinect“ kommt nur noch auf 22.200 Treffer. Kombiniert man „Virtual Reality“ mit weiteren Suchbegriffen so nimmt die Zahl der Suchergebnisse rapide ab:

- „Virtual Reality Training“: 1.410.000 Treffer
- „Virtual Reality Training Head mounted display“: 67.000 Treffer

² <https://scholar.google.de/>

- “Virtual Reality Training Head mounted display Microsoft Kinect”: 1.920 Treffer
- “Virtual Reality Training Head mounted display Microsoft Kinect Assembly” kommt nur noch auf 825 Treffer.

Beschränkt man die Ergebnisse auf Treffer, die seit der Entwickler-Version der Microsoft Kinect v2 erschienen sind, so erhält Google man auf 312 Treffer, die teilweise ähnliche Inhalte haben wie diese Arbeit. Eine relevante Auswahl aus diesen Suchergebnissen ist im Folgenden aufgeführt.

2.1 CAD-Daten in Verwendung mit Spieleentwicklungsumgebungen

Das in [Bea2015] vorgestellte Experiment zeigt, wie mit einem VR-Aufbau eine interaktive Anwendung entstehen kann. Dieser Aufbau besteht aus einer Oculus Rift VR als Anzeigegerät, der Leap Motion als Interaktionsgerät und einer Anwendung, die mit Unity gebaut wurde. In dieser Anwendung wird ein CAD-Modell zur Anschauung und Interaktion bereitgestellt. Beattie nutzte hierbei die Leap Motion so, wie ursprünglich angedacht, d. h. als eine auf dem Tisch aufliegende Gestensteuerung.

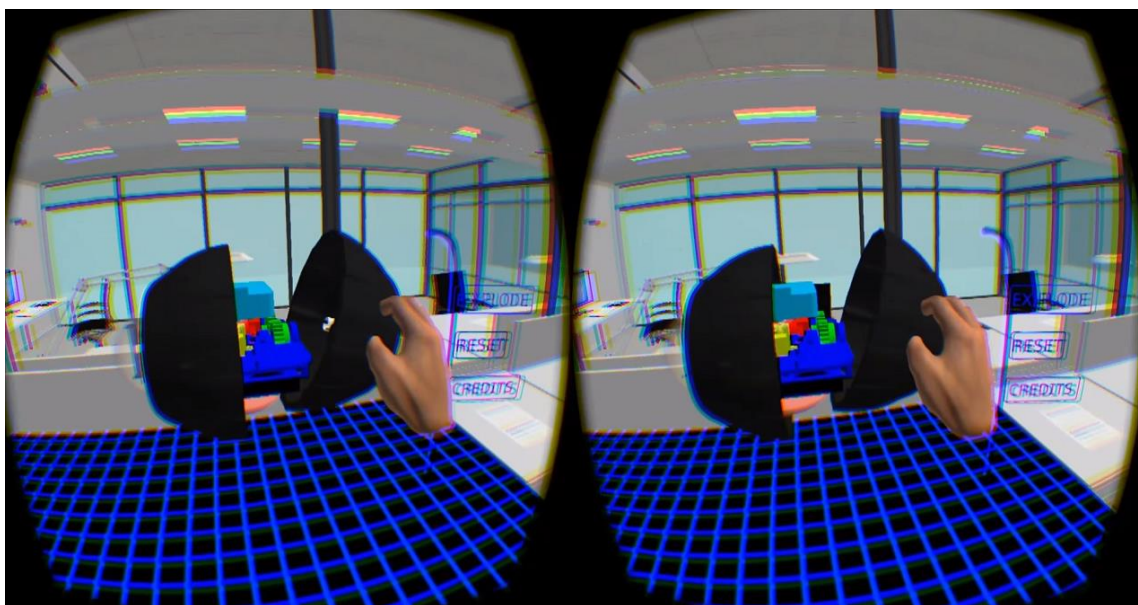


Abbildung 1: Bild aus CAD-Viewer [Bea2015]

Das Modell, welches in der Anwendung auseinander gebaut werden kann, ist etwa so groß wie ein Medizinball. Es schwebt über der virtuellen Tischfläche vor dem Nutzer und kann mit Greifbewegungen zerlegt werden. Durch das Positionstracking des OVR kann der Nutzer in und um das Modell herumschauen. Zusätzlich zum zerlegbaren Objekt gibt es noch Schaltflächen die sich per Berührung betätigen lassen (In Abbildung 1 rechts der Hand zu erkennen). So kann man das Objekt in alle Einzelteile zerspringen lassen, per Knopfdruck wieder zusammenbauen lassen, sowie sich Informationen über den Entwickler anzeigen lassen.

2.2 Bildbasierende Bewegungsaufnahme

Der Artikel von Grabowski und Jankowski in [GrJa2014] beschreibt einen Vergleich zwischen zwei verschiedenen VR-Systemen. Die Systeme sollen zur Schulung neuer Mitarbeiter einer Kohlemine genutzt werden. Motivation für eine VR-gestützte Schulung ist, dass 18% der Unfälle in der dort beschriebenen Kohlemine von neuen Mitarbeitern verursacht werden, obwohl diese Mitarbeiter nur 6% der Angestellten ausmachen. Das konkrete Szenario der Lernumgebung sind Sprengarbeiten, welche sich nur im Ansatz risikofrei üben lassen. Für die Studie wurden 20 Probanden herangezogen. Alle Probanden führten Übungen in beiden Systemen durch, und beantworteten einen Fragebogen zu dem System nach dem jeweiligen Durchlauf. Am Ende wurde dann noch ein weiterer Fragebogen ausgefüllt, welcher die beiden Systeme konkret gegenüberstellt. Zudem wurde noch abgefragt, wie nützlich die gezeigten Trainingsmethoden in Bezug auf die tatsächliche Arbeit ist.

Die beiden verglichenen Systeme waren Folgende:

- Ein hoch-immersiver Aufbau mit Datenhandschuh und bildbasierendem Bewegungsaufnahmeverfahren. Durch kabellose Datenübertragung wurde eine hohe Bewegungsfreiheit erzeugt, welche nur durch den physischen Raum gebrochen wird. Zudem erhielt die Hälfte der Probanden ein Sony HMZ-T1 HMD mit einem 45° Sichtbereich (kurz FOV). Die andere Hälfte benutzte ein OVR Developer Kit 1 mit 110° FOV.

- Ein niedrig-immersiver Aufbau bestehend aus einer Razer Hydra³, einem einfachen HMD (45° FOV) kombiniert mit einem AHRS-Gerät zur Kopfbewegungsmessung. Durch die kurzen Kabel der Hydra ergibt sich ein nur geringer Spielraum bei diesem VR-System.

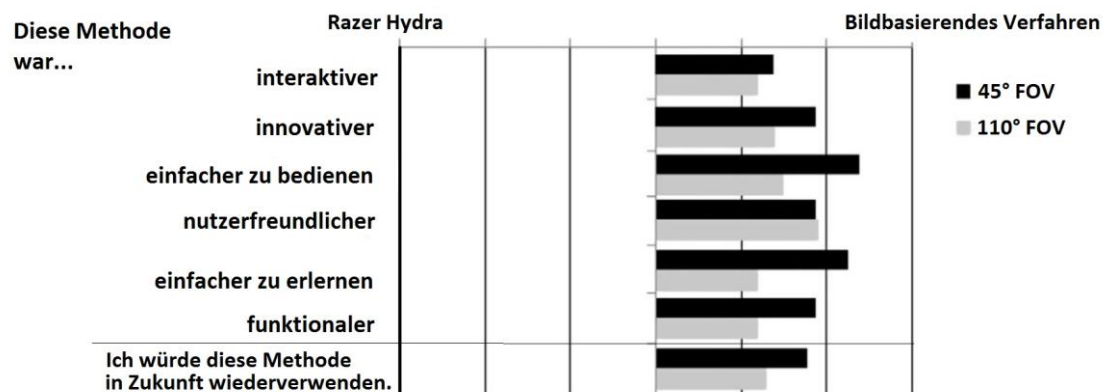


Abbildung 2: Vergleich Razer Hydra zu bildbasiertem System [GrJa2014 S. 312]

Das Szenario, welches durchlaufen wurde, war in beiden Aufbauten dasselbe. Die anschließende Befragung ergab, dass das bildbasierte System in allen abgefragten Bereichen vor dem Hydra-System bevorzugt wurde (siehe Abbildung 2). Zudem besteht ein Unterschied zwischen dem Aufbau mit dem 45°-FOV-HMD und dem 110°-FOV-HMD, zu Gunsten des Geräts mit niedrigerem Sichtwinkel. Die Autoren erwähnen, dass dies an der niedrigen Auflösung des OVR DK1 liegen kann. Darüber hinaus heben die Autoren einen Aspekt besonders hervor: „Especially useful are remarks from miners who take part in the experiment, because of their experience in the development of training programs for the miners.“ [GrJa2014 S. 312].

³ Razer Hydra: <http://www.razerzone.com/de-de/gaming-controllers/razer-hydra-portal-2-bundle>



Abbildung 3: JewelMine Spielausschnitt [LaKo*2012 S. 171]

2.3 Motivationserhaltung in Übungsanwendungen und Rückkopplung über den Fortschritt

Eine Anwendung zur körperlichen Ertüchtigung haben Belinda Lange et al. in ihrem Artikel in [LaKo*2012] vorgestellt. Durch Nutzung der Microsoft Kinect (in ihrer ersten Version), werden Patienten mit neurologischen Beeinträchtigungen erkannt und ihre Bewegung in die virtuelle Welt übertragen. Die Anwendung JewelMine' ist ein Spiel, welches den Patienten dazu bringen soll, sich zu strecken und zu bewegen. Dabei fährt eine Lore auf Schienen in einer Mine entlang und das Programm setzt Juwelen in die Umgebung ein (siehe Abbildung 3). Der Spieler sitzt in der fahrenden Lore und muss die Juwelen einsammeln. Das Programm erkennt die Bewegungsreichweite der Spieler und setzt die Juwelen innerhalb dieser Reichweite, oder aber etwas darüber, um den Bereich zu erweitern und den Spieler zu fordern. Das Spiel speichert den Punktestand und die Leistung ab, damit zu einem anderen Zeitpunkt an gleicher Stelle weitergemacht werden kann. Dies dient auch der Motivation. Die Bewegung der Spieler wird mit Hilfe einer Replay-Funktion abgespeichert und kann so im Nachhinein analysiert werden. Das dazu benötigte Analysehilfsmittel wurde ebenfalls in die Anwendung integriert.

2.4 Microsoft KINECT in Übungs- und Testanwendungen

In [KiAd*2013] wird ebenfalls gezeigt, wie medizinische Übungen und Tests mit Hilfe der Kinect umgesetzt werden können. Die Arbeit beschreibt, wie der zeitgesteuerte Aufstehen-und-Gehen-Test und der 10-Meter-Lauf-Test umgesetzt werden, sowie ein Messverfahren zur Ermittlung der Bewegungsfreiheit funktioniert. Die Zielgruppe der Tests sind Menschen mit Beeinträchtigungen der Bewegung, vornehmlich ältere Menschen.

Beim Aufstehen-und-Gehen-Test sitzt der Patient zu Beginn auf einem Stuhl. Von diesem Stuhl muss der Patient aufstehen, 3 Meter gehen, sich umdrehen und wieder zurück auf den Stuhl sitzen. Die Zeit, die für die einzelnen Phasen des Tests benötigt werden, wird gestoppt und das Ergebnis des Tests gibt eine Aussage darüber, wie selbstsicher der Patient bei Bewegungen ohne Hilfe ist. In der ursprünglichen Version des Tests misst ein Betreuer die einzelnen Zeiten der Phasen. In der Kinect-Umsetzung erkennt der Sensor anhand der Kopfhöhe bzw. Schritthöhe, wann der Patient steht. Um die Laufbewegung zu messen wurde eine Stange in 3 Meter Entfernung zum Stuhl aufgestellt, um welche der Patient herum laufen muss. Der Aufbau wurde aus verschiedenen Perspektiven von der Kinect erfasst um zu testen, aus welcher Perspektive sich die geringste Abweichung im Vergleich zu einer von Hand gemessenen Zeit ergibt. Die Zeitmessungen wurden für jeden Patienten mehrmals durchgeführt. Bei jedem Durchlauf wurden von der Kinect und von einem Betreuer die Zeiten erfasst.

Der zweite Test wurde ebenfalls auf diese Art verglichen. Beim 10-Meter-Lauf-Test wird die Zeit gestoppt, sowie die Schritte gezählt, die für die 10 Meter benötigt werden. Dabei laufen die Patienten die 10 Meter mal mit normaler und mal mit schnellstmöglicher Geschwindigkeit. Im Test ist der Start- und Zielpunkt genau definiert: „sobald ein Fuß die Start- bzw. Ziellinie berührt oder überschreitet“. Die Autoren schreiben, dass sich dies mit der Kinect nicht genau abbilden lässt. Stattdessen wurden Start- und Ziellinien gemalt, und wenn die Person mit ihrem Körpermittelpunkt über die Linie getreten ist, gilt der Start oder das Ziel als erreicht.

Um die Bewegungsfreiheit zu ermitteln wird der Winkel des zu betrachteten Körperteils zum übergeordneten Körperteil gemessen. Dabei werden die Gliedmaßen in

alle natürlichen Bewegungsrichtungen rotiert. Dieses Messverfahren wurde ebenfalls von einer Fachperson und der KINECT durchgeführt, um einen Vergleich zu schaffen.

Im Aufstehen-und-Gehen-Test ergab sich eine durchschnittliche Abweichung von 0,33 Sekunden, was einer prozentualen Abweichung von etwa 4% entspricht. Im 10-Meter-Lauf-Test waren es mit 0,15 Sekunden etwa 2%. Bei der Messung Bewegungsfreiheit ergab sich eine Abweichung von teilweise bis zu 67%. Schuld daran können nicht eindeutige Daten oder Verdeckungsprobleme sein.

Die Autoren benennen in ihrer Arbeit Probleme, die es für eine praktische Anwendung zu beachten gilt. So ist es zu vermeiden, dass sich bewegende Objekte im Sichtbereich der Kamera befinden. Beispielsweise können Vorhänge, die sich im Wind bewegen, als Personen erkannt werden. Sie schreiben weiter, dass eine zusätzliche Person im Trackingbereich akzeptierbar ist, solange sie sich nicht mit der Hauptperson vor der Kamera überschneidet. Die Arbeit zeigt, dass die Microsoft KINECT für Übungsanwendungen nutzbar ist, wenn auch nicht ohne Einschränkungen.

2.5 Montage in virtueller Welt

Das von Xia et al. entwickelte System in [XiLo*2011] zeigt, wie virtuelle Realität eingesetzt werden kann, um Montagearbeiten an komplexen Systemen trainierbar zu machen. Die Autoren begründen die Entwicklung damit, dass Schulungen in komplexen Fertigungsanlagen zu teuer sind und es zu lange dauert, den Umgang mit den Maschinen zu lernen. Grund dafür ist auch, dass es schwer ist, sich zweidimensionale Darstellungen von Maschinen in einem 3D-Raum vorzustellen (vgl. [XiLo*2011 S. 383]).

Für VR-Übungsanwendungen im Montageumfeld bezeichnen die Autoren folgende Punkte als wichtig:

1. Die Lernenden sollen so vollständig wie möglich in die Welt übertragen werden. Beispielsweise dienen dafür HMDs, sodass ein Mitten-drin-Gefühl entsteht.

2. Die Aktivität, die zur Montage vollführt wird, soll realgetreu sein. Die Autoren unterscheiden hier zwischen Arbeiten an einer fixen Position (beispielsweise an einem Laufband sitzend) und im freien Umfeld (sodass der Lernende um die Maschine herum laufen kann). Die verwendete Methode in der virtuellen Welt sollte auch der verwendeten Methode in der echten Welt entsprechen.
3. Damit ein Fertigungsprozess ganzheitlich und genau erlernt werden kann, ist haptisches Feedback nötig.

Das in der Arbeit gezeigte System besteht aus teilweise selbst entworfenen Komponenten. So stehen die Lernenden auf einer Plattform, die es ermöglicht, in der virtuellen Welt frei umher zu laufen ohne dabei die Position in der realen Welt zu verändern. Dies wird durch zwei Plattformen ermöglicht, die jeweils mit einem Servomotor verbunden sind. Die Plattformen bewegen sich unter den Füßen mit und dienen so als ‚beweglicher Boden‘. Die Position und Orientierung der Füße wird mittels Markern erkannt. Darüber hinaus besteht die Mensch-Maschine-Schnittstelle des Systems aus einem HMD, einer Phantom Omni als haptisches Gerät, dem CyberGlove als Datenhandschuh und einer sphärischen Leinwand um den Nutzer herum. Auf diese Leinwand, welche einen Durchmesser von 5,5m hat, wird per Rückprojektion die virtuelle Welt auf projiziert (siehe Abbildung 4).

Die Autoren erhalten die virtuellen Daten der Maschinen aus den CAD-Modellen der Anlagen. Diese werden über eine CAD-API aufgearbeitet und in die Formate OBJ oder SLP umgewandelt. Wichtig bei der Konvertierung ist, dass Polygondaten und der hierarchische Aufbau des Objekts mit seinen Teilobjekten unverändert bleiben. Die Polygondaten werden verwendet, um das Objekt in der Welt darzustellen und Kollisionserkennung möglich zu machen. Kollidiert das Objekt, welches von der Phantom Omni gesteuert wird, mit einem anderen Objekt, so wird eine haptische Rückkopplung an der Phantom ausgelöst. Um virtuelle Komponenten der Maschine zusammenzubauen, müssen Entfernung und Winkel zueinander in einem vordefinierten Bereich liegen. Ist beispielsweise eine Schraube nahe an dem entsprechenden Bohrloch und steht fast orthogonal zu der Stelle, so werden die Objekte zusammengefügt. Entsteht auf die Art eine Verknüpfung, so wird eine haptische Rück-

kopplung über die ‚Phantom‘ an den Nutzer ausgegeben um den Einsetz- bzw. Zusammensetzungsprozess so realistisch wie möglich abzubilden.

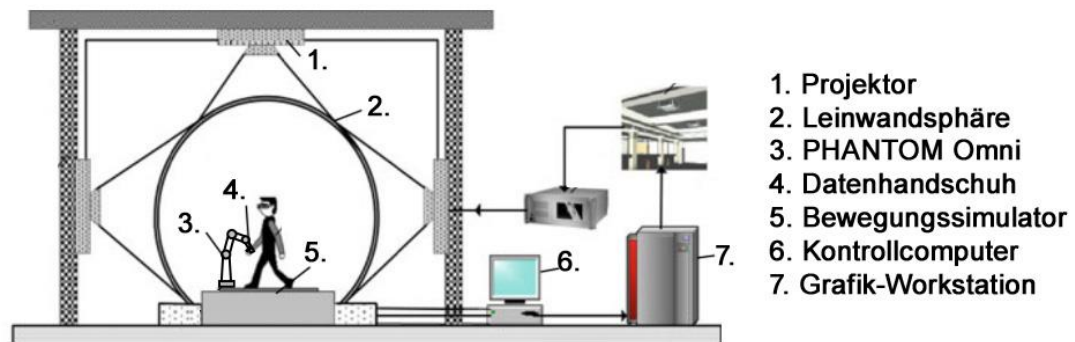


Abbildung 4: VR-Aufbau nach Xia et al. [XiLo*2011 S. 384]

Da die ‚Phantom Omni‘ keine Greifinteraktion möglich machen kann, wurde der Datenhandschuh ‚CyberGlove‘ dem System hinzugefügt. Es wurde versucht, den Handschuh an der Omni zu montieren um haptisches Feedback, sowie Greifinteraktionen an einer Hand zu vereinen. Diese Methode erwies sich als wenig praktikabel, da die Interaktion mit der virtuellen Umgebung zu sehr verändert wird. Eine Alternative ist laut den Autoren ein Handschuh, mit eingebautem Force-Feedback (vgl. [XiLo*2011 S. 395]).

Xia et al. sehen in der Übertragung von großen Fertigungsanlagen in die virtuelle Welt ein Problem. So seien die CAD-Daten zu komplex, um sie mit geringem Rechenaufwand darzustellen. Eine weitere Erschwernis ist, dass der verwendete Datenhandschuh Force-Feedback hat, die Phantom Omni mit haptischer Rückkopplung aber keine Greifinteraktionen zulässt. Zudem seien Handschuhe mit Force-Feedback teuer, welches dem Ansatz, ein kostengünstiges System zu entwerfen, widerspricht. Das in [XiLo*2011] gezeigte System vereint haptisches Feedback und eine Lauffläche, die nicht an Grenzen gebunden ist (beispielsweise das Ende des Trackingbereichs oder physische Grenzen, wie z.B. Wände). Die Autoren treffen keine Aussage darüber, ob die Erstellung der Lehr-/Lerninhalte mit geringem Aufwand möglich ist. Zudem gibt es auch keine Angaben über die Dauer der Einrichtung.

2.6 VR-Übungsanwendung

Die Autoren Matsas und Vosniakos zeigen in [MaVo2015] eine VR-Trainingsumgebung für die Fertigung von Luft- und Raumfahrtteilen. Im Fertigungsprozess, welcher mit VR trainiert werden soll, werden Teile aus Carbon-Fasern produziert. Dabei übergibt ein Roboter Faserstoffe an einen menschlichen Arbeiter. Dieser legt die Stoffe übereinander, bis das daraus entstandene Werkstück eine bestimmte Dicke erreicht hat. Diese Mensch-Maschine-Kollaboration erfordert ein Training, da Fehler bei der Arbeit mit einer Maschine mitunter fatale Folgen haben können. Beispielsweise gilt es, nicht in die Reichweite des Roboterarmes zu gelangen, um von ihm möglicherweise erfasst zu werden. Dennoch muss man nah genug zum Arm bleiben, um noch die von dem Roboter gereichten Teile nehmen zu können.

Matsas et al. verwendeten für die Erstellung des virtuellen Arbeitsplatzes die Unity3D-Engine. Als Anzeigegerät wurde das HMD ‚eMagin Z800 3DVisor‘ genutzt. Zur Interaktion mit der virtuellen Welt verwendeten die Autoren die Kinect in ihrer ersten Version, sowie Maus und Tastatur. Die Schnittstelle zwischen Kinect-Sensor und Unity3D heißt OpenNI. 3D-Objekte wurden mit Hilfe von 3DS Max und Rhinoceros erstellt. Der Avatar wurde mit Hilfe von ‚evolver‘ generiert.

In einem beispielhaft skizzierten Ablauf der Anwendung muss der Nutzer zuerst in einer Y-Pose kalibriert werden. Sobald die Kinect den User erkennt, werden die Bewegungsdaten auf den Avatar übertragen. Nun kann der Nutzer in der virtuellen Welt agieren. Um den Arbeitsprozess zu starten, gilt es, einen Knopf auf dem Arbeitsbereich zu drücken. Daraufhin startet der Roboter und reicht dem Arbeiter den ersten Stoff. Diesen nimmt der User und legt ihn vor sich auf die Arbeitsfläche, woraufhin der Roboter den nächsten Stofffetzen reicht. Dies wird solange wiederholt bis die richtige Dicke erreicht ist, bzw. bis eine vordefinierte Anzahl an Stoffen angebracht wurde. Gerät der Nutzer in den Arbeitsbereich des Roboters, so ertönt ein Warnsignal und die Oberflächen der Modelle beginnen rot zu pulsieren. Das Aufnehmen des Stofffetzens wird durch eine einfache Kollision des Handobjekts mit dem Stoffobjekt realisiert. Um den Stoff abzulegen, muss dieser die Arbeitsfläche berühren. Die Ausrichtung des Stoffes geschieht automatisch.

Zur Auswertung des Projekts wurden 30 Studenten eines Ingenieurstudiengangs herangezogen. Tabelle 1 zeigt die Ergebnisse.

Tabelle 1: Auswertung der VR-Übungsanwendung (siehe [MaVo2015 S. 12])

Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I feel like I was really moving in the scene	0	1	3	5	21
Wearing the HMD I feel present and involved in the virtual activity	2	1	5	15	7
I lost my concentration, even instantaneously, during the experiment	28	0	0	2	0
Wearing the HMD I feel immersed in the virtual environment	1	0	7	13	9
I feel like I was really moving an object with my hand, despite the fact that the object did not have physical mass	1	7	8	12	2
I did not easily perceive the red transparent surface that represented the robot workspace	29	0	0	0	1
Audiovisual stimuli (alarms) helped me being aware of a potentially hazardous workspace	0	1	1	8	20
I did not encounter any difficulties during the initial calibration process with the Kinect sensor	0	3	3	6	18
I feel like the avatar followed my body and head movements precisely	1	2	12	14	1
It was easy to take the patches from the robot's end-effector	0	0	9	13	8
It was easy to navigate/move in the virtual world (ease of movement, restraint)	0	3	12	12	3
I feel like my behavior in the virtual world didn't change, compared to my behavior in the real world	3	10	9	8	0
I feel more like I was participating in an amusing game	0	2	5	10	13

After the experiment I went through, I believe that H-R training tasks can be more attractive with the use of "serious games"	1	0	2	8	19
---	---	---	---	---	----

Die Auswertung zeigt, dass konzentriertes Lernen in VR-Anwendungen möglich ist, und dass eine virtuelle Umgebung als Schulungsmittel für die Schüler attraktiv ist. Die Erzeugung von Immersion und die Übertragung der Bewegung sind in dem Projekt bereits gut, obwohl die Autoren alte Technologien verwendet haben. Die Frage, ob die Bewegung von Kopf und Körper zwischen Avatar und echter Person übereinstimmen, konnte von den Studenten nur gemischt beantwortet werden.

2.7 Interaktion über NUI mittels adaptierten Bewegungen

In [HaBa2015] wird ein Kinect-basiertes NUI vorgestellt, bei der durch körperliche Bewegung ein Avatar gesteuert wird. In der Anwendung gilt es, den Avatar durch eine virtuelle Welt zu bewegen und Aufgaben zu erledigen. Dabei sieht der Nutzer die Welt aus den Augen des Avatars auf einem LCD-Bildschirm vor sich angezeigt. Die virtuelle Welt selbst ist in *Voxel-Game*⁴ erstellt. Charakteristisch für Voxel-Grafik ist, dass der kleinste Bestandteil einer Welt ein Voxel ist. Als Voxel wird ein „voluminierter Pixel“ bezeichnet. Typischerweise sind diese Voxel größer als Pixel, sodass sich eine bauklotzartige Erscheinung der Welt ergibt. Ein berühmtes Beispiel für ein voxel-basiertes Spiel ist *Minecraft*⁵. Für die Interaktion mit der Welt gibt es drei verschiedene „Modi“. Im Point-and-click-Modus wird mit dem Zeigefinger auf etwas gezeigt. Zeigt man einige Zeit auf etwas Bestimmtes, so wird dieses ausgewählt. Diese Interaktionsmethode ist bei der Menüführung und bei der Auswahl von Antworten auf Frageseiten gedacht. Im Movement-Task-Modus werden vorgegebene Bewegungen oder Figuren vom Nutzer nachgemacht, beispielsweise Kniebeugen oder Hampelmänner. Im Explore-Modus gibt es vier Bewegungsarten, welche in einer Transformation des Avatars resultieren. Läuft der Nutzer auf der Stelle, so

⁴ <http://www.voxelgame.com/>

⁵ <https://minecraft.net/>

läuft der Avatar nach vorne. Springt der Nutzer nach oben, so macht der Avatar einen Satz nach vorne, wobei so auch Höhen von bis zu einem Voxel überwunden werden können. Dreht der User sich mit den Schultern nach links oder rechts, so macht der Avatar eine Drehung nach links oder rechts. Ziel der Anwendung ist die körperliche Ertüchtigung von Kindern und Jugendlichen. Zudem soll es einem Lehrer möglich sein, Welten und Aufgaben selbst zu generieren. Dies geschieht unter anderem durch ein Modul, welches ‚Knowledge Bricks‘ genannt wird. Innerhalb dieses Werkzeuges lassen sich Aufgaben definieren und damit in der Welt verteilen.

In einer Testanwendung verteilten die Autoren Hachaj und Baraniewicz fünf verschiedene Aufgaben-Boxen. Erreicht der Nutzer eine dieser Boxen, so erhält er eine Aufgabe, die es zu erledigen gilt. Die Boxen werden in der Welt frei verteilt, und müssen vom User selbst gefunden werden. Die Testanwendung besteht aus einem Startpunkt, einem gewundenen Pfad, einem Haufen von Boxen und einem Häuschen mit engem Durchgang. Durch diese Welt muss sich der Nutzer hindurchmanövrieren, während die Zeit gemessen wird. Die fünf Boxen enthielten die Aufgaben: ‚laufe geradeaus‘, ‚laufe durch den geschlängelten Pfad‘, ‚Springe über den Boxenhaufen‘, ‚gehe durch den engen Durchgang‘, ‚Klatsche, mache Kniebeugen, mache Hampelmänner‘. Mit diesen Aufgaben kann gemessen werden, wie gut die Interaktion von der Kinect erkannt wird, und anhand der Zeit kann geschätzt werden, wie intuitiv die Interaktion ist.

Probanden der Evaluation waren 27 Personen, davon 13 Männer und 14 Frauen. 4 davon waren Vorschüler, 5 Grundschüler, 5 Jugendliche, sowie 13 Studierende. Keiner der Probanden hatte Erfahrung mit VR-Technologien. Bevor die Anwendung gestartet wurde, erklärte der Tester wie mit der Welt zu interagieren ist, zudem wurde vom Programm geprüft, ob die Interaktion verstanden wurde. Erst dann wurde der Level geladen. Die Auswertung ergab, dass alle Probanden die Anwendung in „annehmbarer Zeit absolvierten“ (vgl. [HaBa2015 S. 404]). Erwähnenswert ist auch, dass die Point-and-Select-Methode von allen Probanden ohne Probleme verstanden wurde. Die Autoren geben an, dass die verwendeten Interaktionsmodi für die getestete Altersgruppe erfolgreich verwendet werden kann (vgl. [HaBa2015 S. 405]).

Ähnlich wie im vorangegangenen Unterkapitel ist auch in der Arbeit von Hachaj und Baraniewicz zu bemerken, dass für die Interaktion mit der Welt teilweise veränderte Bewegungen verwendet werden. Dies dient dazu, den geringen Trackingbereich der Kinect zu kaschieren und die Anwendbarkeit der Technologie in dem Anwendungsfall so erst zu ermöglichen.

2.8 Oculus Rift, Kinect V2 und Unity3D

Die Autoren Greuter und Roberts verwenden für ihre Arbeit in [GrRo2014] die Oculus Rift DK1 sowie die Kinect V2, und entwickelten dafür eine Anwendung in Unity3D. Zudem machen sie es durch die Verknüpfung des OVR mit einem Tablet-PC möglich, von dem Computer, auf dem die Anwendung läuft, kabelgetrennt zu sein. In ihrem Aufbau gibt es die Trackingstation, die mit einem WiFi-Router und der Kinect ausgestattet ist. Auf dieser Tracking-Station läuft die Unity-Anwendung. Darüber hinaus gibt es den VR-Rucksack, in welchem sich der Tablet-PC, sowie eine Telefonbatterie (diese dient dazu, die OVR DK1 mit Strom zu versorgen, erst die DK2 machte es möglich, die Display-Brille über den USB-Anschluss mit Strom zu versorgen). In der Anwendung befindet sich der Nutzer in der Toskana-Demo, welche mit dem Oculus Rift SDK mitgeliefert wird. Diese Anwendung wurde durch eine Meldung erweitert, die erscheint, sobald man den Trackingbereich der Kinect v2 verlässt. Ebenfalls Teil des Aufbaus ist eine Projektion dessen, was der Nutzer sieht. Dies dient dazu, dass der Anleiter verfolgen kann, was der User sieht. Eine Interaktion mit der Welt per Greifbewegung ist möglich. Zudem lassen sich Dinge auswählen, indem man auf sie schaut. Dies geschieht durch einer Art unsichtbares Fadenkreuz im Mittelpunkt des Sichtbereichs. Greuter und Roberts identifizieren in ihrer Arbeit bereits Probleme die in derartigen VR-Aufbauten entstehen können. So seien vor allem der geringe Trackingbereich und Hindernisse, die sich in dem Bereich befinden, ein Problem. Zudem schreiben sie, dass die geringe Latenz der OVR DK1 zu Problemen wie „Bewegungskrankheiten“ führt (siehe [GrRo2014 S. 5]).

Die Autoren zeigen, wie mit den zu dem Zeitpunkt verfügbaren Technologien eine VR-Anwendung für Endanwender möglich gemacht werden kann. Sie beschreiben

einige Interaktionsmethoden wie die Problematik der fehlenden Maus umgangen werden kann, ohne dabei einen konkreten Anwendungszweck zu implementieren.

2.9 VR-Unterricht

Die letzte hier gezeigte Arbeit von Sharma und Chen zeigt, wie VR-Technologien und virtuelle Welten als Alternative zum physischen Unterricht eingesetzt werden können (siehe [ShCh2014]). Die Zielsetzung des vorgestellten Projekts ist, die Vorteile des klassischen Unterrichts von seinen Nachteilen zu trennen. So sind die Kommunikation zwischen Lehrern und Schülern, sowie die Darreichung von Bildern und Texten in einer realen Unterrichtssituation sehr einfach möglich. Nachteilig ist aber, dass für dafür ein Raum nötig ist, welcher Kosten verursacht und die Schüler und Lehrer zudem alle extra anreisen müssen. In der Anwendung können sich die Teilnehmenden in einem virtuellen Raum einfinden und nehmen entweder die Rolle des Schülers oder Lehrers ein. Der Lehrer hat darüber hinaus die Möglichkeit, die Rechte der Schüler zu erweitern und ihnen auf die Art und Weise das Wort zu erteilen oder zu entziehen. Ebenfalls kann der Lehrer seine Gestik mit einer Kinect in die virtuelle Welt übertragen. Die Anwendung wurde mit WorldViz Vizard erstellt, hierfür wurden die Objekte in 3DS Max teilweise vormodelliert. Zudem wurden Objekte verwendet, die von der VR-Engine zur Verfügung gestellt werden. Als Schnittstelle zwischen Kinect und Anwendung wurde FFAST verwendet. Um eine Schulungseinheit zu starten muss der Lehrer den Serverdienst starten und hat dann die Möglichkeit, in der Lobby der laufenden Anwendung verschiedene Räume zu erstellen. In diesen Räumen nimmt er dann die Position des Lehrers ein. Dargestellt wird die Anwendung über den PC-Monitor. Besonders heben die Autoren das Management-Werkzeug hervor, womit der Lehrer die Rechte der Schüler erweitern oder Einstellungen am Raum vornehmen kann. Der Einsatz der Kinect ist nur optional, hierfür muss erst das FFAST-Programm gestartet werden, damit die Übertragung der Sensordaten an die Anwendung funktioniert.

Der optionale Einsatz der Kinect ist eher kosmetisch als wirklich nützlich. Das Programm, welches die Autoren präsentieren, ähnelt stark einem 3D-Konferenzsystem als einer wirklichen VR-Anwendung.

2.10 Stand der Wissenschaft zusammengefasst

Die Arbeiten [Bea2015], [KiAd*2013] und [GrRo2014] nutzen teilweise dieselbe Technologie wie sie in dieser Masterarbeit verwendet wurde. Die von den Autoren benannten Probleme sind auch in dieser Arbeit zu beachten. Beattie und Greuter et al. verwendeten ihre Technologie ohne einen Kontext zu schaffen, Kitsunezaki et al. bettet ihren VR-Aufbau in ein Übungsszenario ein.

[GrJa2014] und [MaVo2015] testen ihre Systeme auf Nutzerakzeptanz. Dabei wird klar, dass bildbasierte Systeme wie in der Arbeit von Grabowski et al. vorgestellt, als intuitiver wahrgenommen werden als ein controller-basiertes Interface. Matsas et al. hingegen testet auf die Nutzbarkeit von einer bildbasierten Eingabemethode für Produktionsübungen.

Matsas et al. und die Arbeit von Hachaj et al. (siehe [HaBa2015]) nutzen dafür adaptierte Bewegungen. Auf die Art sollen Schwachstellen der Technologie umgangen werden. So werden in der Arbeit [MaVo2015] keine tatsächlichen Greifbewegungen verwendet, sondern es werden Objekte, sobald sie virtuell berührt werden, in die Hand oder auf das Werkstück platziert. In [HaBa2015] bewegen sich die Nutzer durch eine weitläufige Spielewelt, der echte Nutzer tritt aber auf der Stelle. So wird das Problem des geringen Trackingbereichs kaschiert.

In [XiLo*2011] wird versucht, den Nutzer ganzheitlich in eine virtuelle Welt zu bringen. Dies wird mit einer Laufplattform, einem HMD, Projektoren und einer Phantom Omni ermöglicht. [HaBa2015] und [XiLo*2011] beschränken die Bewegungen oder Interaktionsmöglichkeiten mit der virtuellen Welt nicht. So ist der Nutzer gezwungen entweder sich eine Vielzahl neuer Interaktionen anzueignen (das Laufen auf einer Plattform, der Umgang mit der Phantom Omni), oder aber bereits gelerntes Verhalten zu adaptieren (statt laufen in einem Raum, an einer Stelle laufen).

Die Arbeit von Lange et al. in [LaKo*2012] nutzt die Kinect, um Nutzer in einer virtuellen Welt zu trainieren. Dabei werden Abspielfunktion, Speicherstände und Bestenlisten genutzt, um die Nutzer zu motivieren. Dies ist vor allem in Ausblick auf die langfristige Verwendung des Produkts dieser Masterarbeit von Bedeutung.

[ShCh2014] nutzt die Kinect ebenfalls in einem Lehr-/Lernkontext. Dabei ist die Bewegungsaufnahmekamera aber nur als optionale Funktion eines Konferenzsystems in der Arbeit untergebracht. Auch diese Arbeit verwendet die Technologie, ähnlich wie in [HaBa2015] eher gezwungenermaßen.

3 Ziele

Um ein VR-basiertes Schulungssystem für Ausbildungsbetriebe zu realisieren gilt es, in enger Zusammenarbeit mit den Anwendern das System zu konzipieren. Die bisherige Schulung, ein an Frontalunterricht angelehnter Vortrag mittels PowerPoint, soll von einer VR-Anwendung ersetzt werden, die von jedem zu schulenden Mitarbeiter durchlaufen werden muss. Ziel der VR-gestützten Schulung ist es, die Effektivität des Unterrichts zu erhöhen. Dies soll erreicht werden, indem ein konkretes didaktisches Konzept in einer virtuellen Welt umgesetzt wird. Eine einfache und intuitive Bedienbarkeit der Anwendung, sowie eine realistische Darreichung von Interaktion und Umgebung, soll eine hohe Akzeptanz des Programms ermöglichen. Auf diese Weise soll sichergestellt werden, dass das Programm auch tatsächlich als Alternative eingesetzt wird.

Aus einem Workshop (genauer in Kapitel 6.1 beschrieben) gehen Anforderungen an das System hervor. Kapitel 3.1 beschreibt diese Anforderungen. Das zweite Unterkapitel behandelt die Lernmodule, die teilweise während des Thesis-Prozesses umgesetzt wurden.

3.1 Anforderungen an das System

Insgesamt wurden 85 direkte oder indirekte Anforderungen an das System gesammelt. 32 Anforderungen kamen dabei von Mitarbeitern der Firma Klima Becker, 53 Anforderungen von den Mitarbeitern der Amazonenwerke. Nach Abzug der Doppelnennungen, sowie der VR-irrelevanten Nennungen ergaben sich 62 Anforderungen, die in sieben Gruppen zuordenbar sind. Die Gruppen sind:

- Organisation
- Generierung
- Output
- Inhalt
- Umgebung
- Interaktion

- Nutzung

Eine Auflistung der Anforderungen gibt es im Anhang unter der Überschrift „III. Anforderungen“.

3.1.1 Organisation

Die Organisation betreffend ist es den Experten wichtig, dass es möglich ist, das System zu pflegen und zu warten. Erscheinen neue Maschinen oder eine neue Serie einer bereits vorhandenen Maschine, so sollen die neuen Daten dem bestehenden System hinzugefügt werden können. Da die Firmen auch Firmensitze außerhalb von Deutschland haben, entstand die Anforderung, dass das System auch mehrsprachig nutzbar ist. Anforderungen an die Ergonomie des Systems sind, dass es mobil einsetzbar ist, sowie einen schnellen Zugriff ermöglicht und damit eine geringe Einrichtungszeit hat. Weitere wichtige Punkte die die Organisation betreffen, sind die Kosten des Produkts, und dass Standardabläufe verwendet werden.

3.1.2 Generierung

Teil der Generierung ist die Einbindung der bereits vorhandenen Schulungsdaten. Bilder sowie Filme sollen über die Engine erreichbar und in der VR-Anwendung eingebettet sein. Eine weitere wichtige Anforderung an die Interaktion, ist die Verwendung der CAD-Daten der echten Maschinen. Die Übungseinheiten sollen, nach Meinung der Experten, einem Musterablauf entsprechen. Zudem sollen die Aufgaben störungsbasiert sein, also einem Reparaturverfahren aus einem reellen Szenario entsprechen.

3.1.3 Output

Unter dem Punkt ‚Output‘ wurde gebündelt, dass der Nutzer erfährt, wie gut er abgeschnitten hat, und dass es möglich ist, Papieranleitungen zu erzeugen.

3.1.4 Inhalt

Der Punkt ‚Inhalt‘ weist vorwiegend Informationen auf, die mit in der Anwendung angezeigt werden sollen. Beispielsweise sind es Hinweise über die verwendeten

Werkzeuge, Fehlerhinweise, Sicherheitshinweise, Stücklisten, Gewichte der Objekte, Teilenummern der Objekte, bauteilbezogene Parameter, etc. Darüber hinaus sind Anzeigen über die Objektaktivität gewünscht. Eine weitere Anforderung der Experten war es, dass man sich selbst im virtuellen Raum sieht, wenn man an sich selbst herunterschaut.

3.1.5 Umgebung

Folgende Anforderungen wurden an die Umgebung gestellt: Realitätsnähe, vorhandene Lichtsimulation, Anzeige von kompletten Anlagen (nicht nur einzelner Teile davon), die Einbettung von technischen Zeichnungen, und die Simulation von Anlagengeräuschen. In dieser Gruppe untergeordnet sind auch die Abschottung von der Außenwelt und Sprachneutralität.

3.1.6 Interaktion

Unter dieser Gruppe sind Anforderungen aufgeführt, die Arten der Interaktion innerhalb der Anwendung widerspiegeln. So wünschen sich die Experten, dass der Einbau von Teilen geübt werden kann, aber auch die Zerlegung eines Objekts. Die Hervorhebung und Beweglichkeit von Teilen ist unter dieser Gruppe mitdefiniert. Es sei auch wichtig, dass die Interaktionen beschränkt werden, um sich auf das Wesentliche zu konzentrieren. Eine intuitive Bedienung wurde ebenfalls als Anforderung an das System genannt.

3.1.7 Nutzung

Bereiche in denen die VR-Anwendung genutzt werden soll, sind der Innendienst, in Schulungen, bei der Montage und in der Produktion. Auch Bedienungsschulungen von neuen Maschinen sollen so ermöglicht werden, sowie die Fehlersuche und die Konstruktion soll das System unterstützen.

3.2 Lernmodule

Als ein Lernmodul wird in dieser Arbeit die Form einer Aufgabe definiert, die ein Schüler zu absolvieren hat. Diese Aufgabe ist in einem oder mehreren C#-Skripten verpackt und liefert Schnittstellen, über die die benötigten Materialien (Objekte

bzw. Teilobjekte, die es zusammen zu bauen gilt) dem Script übergeben werden können. In solchen Lernmodulen lassen sich vordefinierte Routinen ablegen, um sie für verschiedene Aufgaben zu nutzen. Beispielsweise bietet ein Script „MachineAssembly.cs“ mehrere Schnittstellen für Unity-GameObjects. Um eine neue Aufgabe zu gestalten, nimmt ein Lehrer dieses Script und fügt die zum Zusammenbau benötigten Teilobjekte hinzu. Die restliche Arbeit, wie die Ausbreitung der Objekte auf einer Werkbank oder die Hervorhebung welches Objekt als nächstes eingesetzt werden muss, übernimmt das Script. So beschränken sich die Arbeitsschritte zur Erstellung neuer Lerninhalte auf:

- (opt.) Konvertieren neuer CAD-Daten über den Autodesk Converter.
- (opt.) Import der daraus resultierenden .fbx-Datei.
- Auswahl der Aufgabe (z.B. MachineAssembly.cs).
- Hinzufügen der Aufgabe zur Spielwelt. (Leeres GameObject der Welt hinzufügen, diesem GameObject eine Script-Komponente geben, die das Aufgabenskript hält.)
- Verlinkung der Objektdaten in das Script des Lernmoduls. (Drag and Drop der einzelnen Objekte in die vorgesehenen Felder innerhalb der Script-Komponente des Aufgaben-Objektes.)
- Hinzufügen der vordefinierten Objektgruppen für das Kinect-Tracking und das Oculus-Rift-Tracking (in Unity ‚Prefabs‘ genannt).

Ziel ist es, die Gestaltung der Lernszenarien auf eben diese Schritte zu reduzieren.

4 Methoden und Material

4.1 Interaktions- und Anzeigegeräte

4.1.1 Microsoft Kinect v2

Die für Konsumenten ausgelegte Bewegungskamera von Microsoft ist als Zusatzausrüstung für die Microsoft Xbox One erhältlich. Das Gerät vereint eine Farbkamera, einen Tiefensensor, sowie ein Mikrofon. Der ursprünglich angedachte Einsatzzweck ist der, der Bewegungssteuerung für Spiele, welche für die Xbox entwickelt wurden. Durch die geringen Kosten von etwa 150 Euro (Stand 2015) sind mit der Kamera sogenannte „low-cost“-Entwicklungen möglich. Die Farbkamera löst mit 1920x1080 Bildpunkten auf und tut dies mit einer Wiederholrate von 30Hz. Der Tiefensensor tastet mit 512x424 Bildpunkten ab und spannt einen Sichtbereich von 70° in der horizontalen und 60° in der vertikalen auf. Der Abtastbereich beginnt 0,5m vor der Kamera und endet in 4,5m Entfernung von der selbigen. Das Gerät wird per USB3.0 Anschluss mit dem PC verbunden [MsKin2015].



Abbildung 5: Microsoft Kinect v2 [Zim2015 S. 61]

Zur Ermittlung der Abstände zwischen den Objekten und der Kamera nutzt der Tiefensensor die sogenannte „Time-of-flight“-Technik. Bei dieser Sensormethode wird von der Kamera ausgehend Licht ausgestrahlt und gemessen, wie lange das Licht

braucht, um wieder zur Kamera zurückreflektiert zu werden. Daraus ergibt sich für jeden Bildpunkt eine andere Zeit bzw. einen anderen Abstand [Mei2013].

4.1.2 Oculus Rift VR

Das von der Firma Oculus entwickelte Anzeigegerät ‚Oculus VR‘ (kurz OVR) nutzt eine Auflösung von 960x1080 Bildpunkten pro Auge. Die Bildwiederholrate beträgt maximal 75Hz und das Sichtfeld beträgt 100°. Zum Vergleich: das menschliche Auge hat ein Sichtfeld von 176°-180°. Strom bezieht es über die USB2.0 Verbindung vom Computer. Eine weitere Bezeichnung für ein solches Gerät ist ‚head-mounted-display‘, kurz HMD. Eine Haupteigenschaft des HMDs ist, dass die Neigung und Rotation von sich selbst in Relation zum Raum an den Computer übertragen werden kann. Somit können Blickrichtungsänderungen in die virtuelle Welt übernommen werden und ermöglicht so, dass der Nutzer sich umschauen kann. Im Lieferumfang des OVR ist eine Positionsbestimmungskamera enthalten, welche dazu dient, die Position des HMDs im Raum zuerkennen [OcVR2015].

4.2 Entwicklungsumgebung

4.2.1 Unity 5

Die Spieleentwicklungsumgebung des Herstellers Unity Technologies ist inzwischen in der Version 5.1.2 verfügbar. Die Engine ermöglicht es, 2D- und 3D-Anwendungen zu implementieren und diese für iOS, Mac, Android, Windows, Linux/Steam OS, Webplayer etc. zu bauen. Teil der Unity IDE ist auch die Unity API, welche eine Vielzahl an Scripting-Schnittstellen bietet. Ein großer Vorteil der vorimplementierten Funktionen ist eine höhere Performanz. Beispielsweise sind folgende Codezeilen äquivalent im Ergebnis.

- `Mathf.Sqrt((firstPosition - secondPosition).sqrMagnitude)`
- `Vector3.Distance(firstPosition - secondPosition)`

Die Methode ‚Distance()‘ aus der Unity-API ist aber im Vergleich zur rechenintensiven `Math.Sqrt()`-Methode bedeutend schneller.

Ein weiterer Teil der Grafikengine ist der Asset-Store. Dort lassen sich vorgefertigte Objekte, beispielsweise Avatare, kostenlos herunterladen oder auch kostenpflichtig erwerben. Prominente Beispiele für Spiele, die mit Unity3d entstanden sind, sind „Lara Croft: Relic Hunt“, „Poly Bridge“ und „Star Trek Timelines“. Mit in der Grafikengine integriert ist der Abspielmodus. Dort kann die entwickelte Anwendung getestet werden (siehe Abbildung 6).

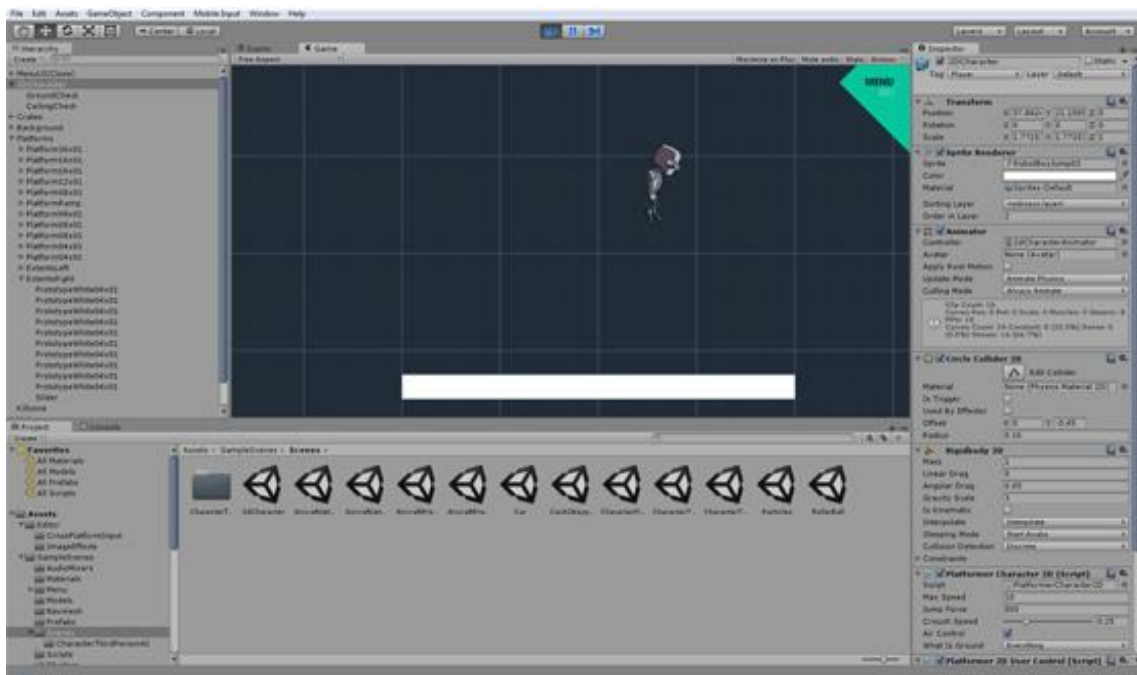


Abbildung 6: Ausschnitt Unity3D

4.2.2 MonoDevelop

MonoDevelop ist eine Entwicklungsumgebung für die Programmiersprachen C#, JavaScript, C/C++, Visual Basic .NET, F# und Vala. Das Programm wird mit Unity mitinstalliert und dient als IDE für die Scripte der Spieleanwendung. Ein Vorteil in einer IDE zu programmieren ist die Verfügbarkeit eines Debuggers. Zudem verfügt MonoDevelop über die typische Auto-Vervollständigung, die es ermöglicht, effektiver zu programmieren, im Vergleich zu einem einfachen Editor.

4.3 Usability-Tests

In empirischen Nutzertests mit potenziellen Endanwendern wird ein bestehendes, oder ein sich in der Entwicklung befindendes System geprüft. Dabei werden realistische Aufgaben durchgeführt und die Beobachtungen von beisitzenden Experten, Äußerungen der Nutzer über das Programm und zum Teil auch Messungen (bspw. der Dauer oder Herzfrequenz des Nutzers) als Hinweis auf Probleme genutzt (vgl. [SaBr2011 S. 163]).

4.3.1 Induktive / deduktive Usability Tests

Nach Rauterberg et al. [RaSi*1994] lassen sich Usability Tests in induktive und deduktive Tests einteilen.

Als induktive Testmethode wird ein Test dann bezeichnet, wenn er ein einzelnes System auf Schwachstellen überprüft. Diese Test-Art zielt darauf ab, ein bestehendes System zu verbessern und Rückkopplung über seine Qualität zu liefern.

Bei deduktiven Tests werden verschiedene Systeme miteinander verglichen und aus allen Alternativen das leistungsfähigste bestimmt. Aus dieser Testmethode können ebenfalls Verbesserungsvorschläge für ein einzelnes System hervorgehen. Diese Art der Rückmeldung liegt bei deduktiven Tests aber nicht im Vordergrund. Auch unterscheiden sich die Methoden in ihren Rahmenbedingungen. So schlagen die Autoren vor, dass deduktive Tests in Laboren stattfinden um standardisierte Testbedingungen zu schaffen. Induktive Tests finden hingegen am Arbeitsplatz des Endnutzers statt, um reale Bedingungen abzubilden (vgl. [SaBr2011 S. 163-164]).

Für die Evaluation des Thesis-Artefakts wird ausschließlich eine induktive Usability-Testmethode verwendet. Dabei ist zu beachten, dass die getestete Interaktion möglichst vollständig implementiert ist und der Testnutzer nicht von irrelevanten Systemverhalten oder -eigenschaften abgelenkt wird. Beispielsweise soll sich der Nutzer nicht mit der Suche nach einer Funktionalität beschäftigen, die noch nicht implementiert ist. Zudem stellen die Autoren Sarodnick und Brau fest, dass Testaufgaben realistischen Arbeitsszenarien entsprechen müssen, unter anderem damit fundierte Aussagen über die getestete Anwendung getroffen werden können. Übertragen auf diese Arbeit bedeutet dies, dass die vom Testnutzer durchgeführte Aufgabe

tatsächlich der des Endnutzers entspricht (vgl. [SaBr2011 S. 163-164]). Sichergestellt wird das dadurch, dass tatsächliche CAD-Daten einer vom Endnutzer genutzten Maschine verwendet werden, und dass die Art der Aufgabe (nach Absprache mit den Firmen) dem Zusammenbauen und Auseinandernehmen der Maschine entspricht.

Nach Rosson und Carroll [RoCa2002] ist es bei induktiven Tests wichtig, dass nur bestimmte Funktionalitäten und nicht das gesamte System getestet wird. Soll das System in Form eines Prototyps nur ausschnittsweise dargestellt werden, um es Personen zu erleichtern, Kritik anzubringen. Rosson et al. schreiben, dass fertige Systeme seltener in Frage gestellt werden als nur Systemteile. Dies steht in keinem Widerspruch mit der oben genannten Forderung, dass ein Prototyp in seiner Funktionalität vollständig sein muss. Der getestete Prototyp muss also die zu testende Problematik kapseln, um den Fokus auf das Wesentliche zu beschränken (vgl. [SaBr2011 S. 165-166]).

4.3.2 Lautes Denken

Die für diese Ausarbeitung verwendete Methode zur Erfassung von Handlungsabläufen oder Beweggründen der Nutzer, ist die ‚think aloud‘ bzw. ‚Lautes Denken‘-Methode. Die in [Lew1982] definierte Art der Erfassung beinhaltet, dass der Testnutzer beim Durchlaufen der Anwendung seine Gedankengänge laut ausdrückt. Nachteile des ‚Lautes Denken‘-Tests sind, dass es schwer ist, für ungeübte Nutzer ihre kognitiven Vorgänge in Worte auszudrücken. Dies führt auch dazu, dass sie für das Durchlaufen der Anwendung länger brauchen. Vor allem bei performanzorientierten Tests ist dies ungeeignet. Der Hauptvorteil ist, dass es durch diese Methode sehr einfach ist, Beweggründe der Nutzer zu identifizieren, da sie diese direkt mitteilen. Dabei muss sichergestellt sein, dass dieser seine Gedanken auch kommuniziert. Begünstigen lässt sich dies, indem ihm beim Testdurchlauf Fragen gestellt werden, um ihn zum Reden zu animieren.

„Was denken sie gerade?“

„Was denken Sie, was diese Meldung zu bedeuten hat?“

„Was denken Sie, was passiert ist?“

„Welche Reaktion haben Sie erwartet?“

Dabei notiert der Moderator Anmerkungen und Gedanken des Nutzers. Tonaufnahmen können dafür ebenfalls hilfreich sein (vgl. [SaBr2011 S. 170-171]).

Ein weiteres Mittel, um die Nutzer an die Evaluationsmethode herzuführen, kann eine kleine Aufgabe vorab sein, bei der der ‚Lautes Denken‘-Test geübt wird, aber dessen Ergebnis nicht relevant ist.

4.3.3 Fragebogen

Im Zuge der Evaluation des hier entwickelten Systems, soll mit Hilfe eines Fragebogens zusätzlich quantitative Information über die Meinung der Nutzer gesammelt werden. Generell können sich Fragebögen durch quantitativen (z. B. bei ‚Multiple Choices‘-Antwortmöglichkeiten oder Ratingskalen) oder aber qualitative Antworten (z.B. Freitexte) unterscheiden. Auch Mischfragen sind möglich (Antwortmöglichkeiten wie „Wenn ja, warum? / Wenn nein, warum nicht?“) (vgl. [SaBr2011 S. 181-182]). Da es sich bei der „Lautes denken“-Methode bereits um qualitative Rückmeldung handelt, beschränkt sich der Fragebogen der Nutzertest in dieser Arbeit auf quantitative Daten. Eine Voraussetzung für die Durchführung eines Fragebogens ist es, sich vorher mit der Thematik ausreichend auseinandergesetzt zu haben. Im Anschluss an den Usability-Test ist diese Voraussetzung gegeben.

Da die Entwicklung eines eigenen Fragebogens voraussetzt, dass der Fragebogen selbst auf Objektivität, Reliabilität und Validität geprüft wurde, empfiehlt es sich, auf bewährte und normierte Fragebogen zurück zu greifen (vgl. [SaBr2011 S. 182]). Der hier verwendete normierte Fragebogen ist ISONORM 9241/110-S [Prü1997]. Dieser ist dazu gedacht, die Software auf Grundlage der internationalen Ergonomie-Norm DIN EN ISO 9241-110 auf ihre Benutzbarkeit zu prüfen. Dabei beziehen sich die Fragen aus dem Fragebogen auf Themen wie die Erlernbarkeit der Software oder die Eindeutigkeit und Form der dargereichten Information. Auch ein wichtiger Kernpunkt ist die Effektivität der Anwendung.

4.3.4 Testpersonen

Nach [Nie1994] und [Vir1992] decken bereits 5-6 Personen 80% der Benutzbarkeitsprobleme auf. Da der Prozentsatz variieren kann, ist abzuwägen, ob die Zahl der Personen nicht angepasst werden soll. So ist bei sicherheitskritischen Systemen eine höhere Anzahl von Personen entscheidend, um bis zu 100% der Probleme aufzudecken. Zudem kann sich die Zahl der Personen auch durch das Budget ändern, oder dadurch, dass zu einem späteren Zeitpunkt weitere, tiefergehende Tests angedacht sind. Auch eine geringere Zahl von Subjekten ist bspw. beim Test von Consumer-Produkten möglich (vgl. [SaBr2011 S. 169]). Da es sich bei dem Projekt GLASSROOM um ein weiterlaufendes Projekt handelt, indem noch weitere Tests gemacht werden, ist eine Anzahl von 6 Personen oder mehr erstrebenswert.

Bei der Wahl der Testpersonen ist zu beachten, dass sie entweder repräsentative für die Zielgruppe sind und keine Vorkenntnis über das getestete System haben, oder aber Experten sind, die die Anwendung selbst zwar nicht kennen, aber mit ähnlichen Bedingungen bereits gearbeitet haben. Vorteil bei der Verwendung von Experten als Testpersonen ist, dass Ideen und Verbesserungen einfließen können. Tests mit Endanwendern sind vor allem dann wichtig, wenn die Interaktionsmethode fertig ist und ein konkretes Anwendungsszenario getestet werden soll.

5 Vorbedingungen der Arbeit

Dieses Kapitel bündelt Vorarbeiten zur Implementierung. Unter anderem ist es notwendig, eine begründete Entscheidung für eine Entwicklungsumgebung zu treffen, sowie die geeignete VR-Technologie zu wählen.

5.1 Auswahl des Bewegungsaufnahmegeräts

Zu Beginn der Entwicklung standen zwei Geräte für die Aufnahme der Bewegungen zur Auswahl. Zum einen die Microsoft Kinect v2, eine Kamera mit Tiefensensor mit der Microsoft für seine Xbox One die Steuerung von Spielen per Körperbewegung möglich macht. Und zum anderen die Leap Motion, welche ursprünglich eine Gestensteuerung ist, die auf einer Tischoberfläche aufliegt. Durch einen vom Hersteller bereitgestellten Aufsatz kann die Leap Motion aber auch an die OVR angebracht werden, und ist so für Motion Capturing einsatzfähig.



Abbildung 7: Leap Motion und OVR [Zim2015 S. 61]

Im Konferenzbeitrag in [Zim2015] hat der Autor den Unterschied der beiden Trackinggeräte durch einen virtuellen Versuchsaufbau ermittelt. Es wurde eine virtuelle Welt gebaut, die Arbeitsstationen enthält, deren Durchführung der Arbeit eines Monteurs entspricht. Um die Arbeitsstationen abzuarbeiten werden folgende Interaktionen benötigt: das Greifen von Gegenständen, das Tragen des Gegenstandes an

eine andere Position, das Einsetzen von Objekten in dafür vorgesehene Öffnungen, sowie das Betätigen von Knöpfen. Umgesetzt wurde dies in der Entwicklungsumgebung Unity in seiner vierten Version.

Als Resultat aus der Untersuchung ergab sich, dass die Leap Motion für den Anwendungszweck nicht geeignet sei. Zimmermann schreibt, dass die Gestensteuerung durch die Montage am HMD, die Bewegung der Hände nur noch von oben abgreifen kann. Gesten die von der Brille wegdeuten, werden durch die eigene Handfläche verdeckt und somit nicht erkannt. Auch ergebe sich durch den geringen Trackingbereich, dass Bewegungen mit weit ausgestreckten Armen nicht mehr durchgeführt werden können, ohne einen Verlust der Bewegungsinformationen hervorzurufen. Die Leap Motion habe aber auch große Vorteile. So sei es möglich, das Gerät, sowie das HMD an einen Laptop anzuschließen, welcher sich dann in einem Tornister verstauen ließe. Mit diesem Aufbau wäre man nicht an einen Trackingbereich gebunden, und könnte sich in der echten Welt frei bewegen. In der Praxis sei dies aber durch die hohe Anforderung an die Performanz der virtuellen Welt nicht möglich. Zimmermann beschreibt weiter, dass empfohlen werde, eine Bildwiederholrate von 75 Bildern pro Sekunde oder mehr zu erreichen, um Übelkeit bei der Benutzung von HMDs vorzubeugen (vergleiche hierfür [OcFAQ2015]).

Des Weiteren sei das Tracking der Leap Motion aber sehr genau. So werden zwar nur die Hände aufgenommen, diese werden aber bis auf die einzelnen Fingerglieder segmentiert und die Daten jedes einzelnen Segments vom Computer verarbeitet. Die Tatsache, dass nur die Hände aufgenommen werden, bedeutet auch, dass kein Positionstracking mit der Leap Motion alleine möglich sei. Auch die Kombination von Leap Motion und Positionstracking durch die OVR funktioniere nicht zusammen. Ist die Positionsbestimmung der OVR aktiv, strahlt das HMD mit Infrarotlampen um von der Kamera gesehen zu werden. Diese Strahlung wird aber von der Leap Motion als Interferenz erkannt, die es automatisch auszugleichen versucht. Das Resultat ist, dass die Qualität des Trackings sich noch weiter verschlechtert. Zudem sei durch diesen Aufbau der Vorteil zunichte, dass man trackingbereichsungebunden ist.

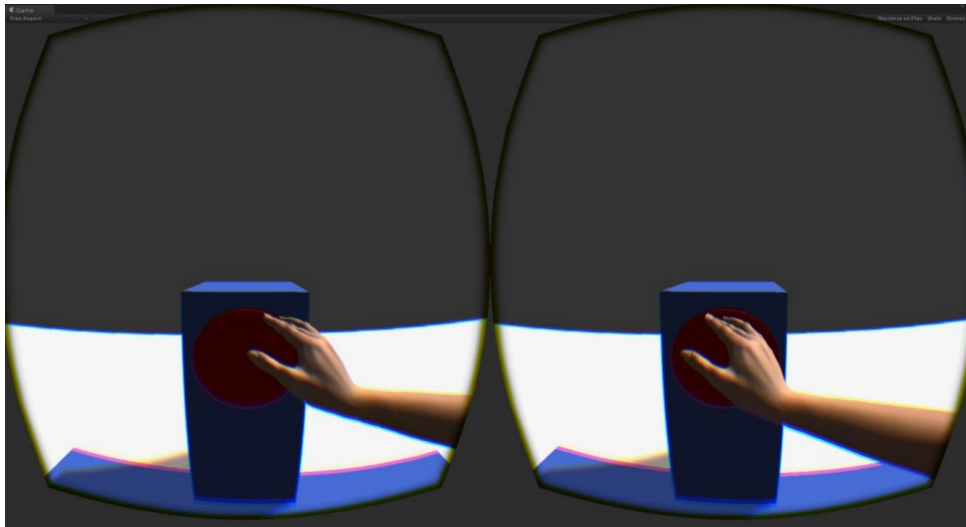


Abbildung 8: Beispielarbeitsstation mit Leap Motion Tracking [Zim2015 S. 62]

Auf der anderen Seite habe die Microsoft Kinect v2, habe mit weniger Verdeckungsproblemen zu kämpfen. Bewegungen, die hinter dem Körper und von der Kamera weg passieren werden zwar nicht erkannt, können aber vermieden werden. Durch den eingeschränkten Trackingbereich der Bewegungskamera ist das Herumlaufen um große virtuelle Objekte nicht möglich, aber durch Platzierung der Person in der virtuellen Welt lässt sich um dieses Problem herum arbeiten. Auch durch die Robustheit des Trackings sei die Kamera für die Aufgaben geeignet.

5.2 Auswahl des Entwicklungsumgebung

Zur Entwicklung des Systems kamen viele Spieleentwicklungsumgebungen in Frage. Für die engere Auswahl wurde die Unity Engine in der fünften Iteration, sowie die Unreal Engine 4 betrachtet. Grund hierfür war die Verfügbarkeit von Plug-Ins, die das Kinect v2 Tracking möglich machten. Aus den Zielen und Anforderungen des Lernsystems ließen sich Kriterien ableiten, die es von der Engine zu erfüllen galt. Im Anhang unter „II. Bewertung der Entwicklungsumgebungen“ ist die Nutzwertanalyse, die zur Auswahl der Engine erstellt wurde.

5.2.1 Vorgang

In einer Nutzwertanalyse wird ein Ziel definiert (Die Auswahl einer Engine), welches mit ihr zu erreichen gilt. Sind die Alternativen (Unreal Engine 4 und Unity 5) gefunden, werden Kriterien an die Alternativen definiert. Im darauffolgenden Schritt

werden die Kriterien nach ihrer Wichtigkeit bewertet und die Alternativen in den einzelnen Kriterien betrachtet. Je nachdem, wie gut die Alternative in einem Kritikpunkt abschneidet, erhält sie viel, wenig, oder keine Punkte (5 Punkte entspricht einem ‚sehr gut‘, 1 Punkt nur einem ‚mangelhaft‘. 0 Punkte wurden vergeben, wenn das Kriterium nicht erfüllt wurde). Diese Punkte werden mit der Gewichtung multipliziert und die Summe aller gewichteten Punktzahlen eines Systems ist ihre Abschlusswertung. Das System mit der höheren Abschlusswertung gilt es dann zu bevorzugen.

5.2.2 Kriterien

Unter der Kategorie ‚Rahmenbedingung‘ gebündelt, sind die Kosten des Systems für den Entwickler und Anwender, sowie die Hard- und Softwareanforderungen. Bei der Kategorie ‚Entwicklung‘ wird zwischen der Entwicklung die Zimmermann in dieser Masterarbeit machte und der Entwicklung die von den Anwendern später benötigt wird, unterschieden, um eigene Lernszenarios zu implementieren und neue Inhalte einzuspeisen. Die Kriterien der technologischen Entwicklung bestehen aus der Anbindung der Kinect, sowie der OVR, und der verfügbaren Dokumentation über die Entwicklungsumgebung. Die Entwicklung der Lernmodule setzt voraus, dass die Engine einfach erlernbar bzw. intuitiv bedienbar ist. Um neue Inhalte verfügbar zu machen, müssen diese aus CAD-Dateien importiert werden. Zudem darf die Objekthierarchie der Datei nicht verändert werden. Unter dem Kritikpunkt ‚Gestaltung von Lernmodulen‘ ist definiert, in welcher Form die jeweilige Engine ein Lernmodul abspeichern kann, um es später wieder verfügbar zu machen. Das Definieren von Schnittstellen soll es möglich machen, dass der Anwender nicht tiefgehendes Verständnis für die Skripte oder Logik des Programms oder der Engine benötigt. Er sollte nur an der richtigen Stelle das richtige Objekt einsetzen können, um mit der Engine zu arbeiten.

Unter der Kategorie ‚Laufzeit‘ sind Kriterien an das Endergebnis gebündelt. Hierunter fällt die Performanz des Programms, das Aussehen, sowie, ob es automatisch Daten über den Nutzer sammeln kann.

5.2.3 Auswertung

Unter genauer Betrachtung ergeben sich die Hardware- und Softwareanforderungen nicht aus den Engines, sondern aus den verwendeten Technologien. Das Kinect-SDK benötigt mindestens Windows 8. Die Oculus Rift verlangt, dass eine möglichst hohe Framerate (~75Hz) erreicht wird. Die Anforderungen sind deswegen für beide Engines dieselben. Bei den Kosten ist die Unreal Engine zu bevorzugen, da sie für Bildung und Ausbildung komplett frei ist. Die Unity Engine beschränkt einige Eigenschaften auf die kostenpflichtige Lizenz. Da diese aber nicht für die Entwicklung nötig sind, ist dies marginal.

Die OVR ist bei der Unreal Engine 4, sowie bei Unity 5 nativ verfügbar gemacht worden. Die Anbindung der Kinect ist bei beiden durch ein Plug-In möglich. Für die Unreal Engine ist dieses Plug-In aber einfacher in der Handhabung. Die Dokumentation ist bei beiden Engines auf den Webseiten der Hersteller verfügbar. Zudem ist die Anzahl der Anleitungen und Foren über Problemstellungen in der Spieleentwicklung bei beiden ebenfalls sehr hoch.

Tabelle 2: Benchmark Unreal Engine 4

Unreal Engine 4	
Format:	
FBX 2010	38,9Hz
FBX 2013.3	40,0Hz

In der Einfachheit bzw. Erlernbarkeit der Engine schnitten beide Systeme nicht gut ab. Die Unreal Engine nutzt zur Script-Erstellung das Engine-eigene ‚Blueprints‘-System. Damit ist es möglich, Funktionsknoten zusammen zu stecken und damit ein komplexes Verhalten zu erstellen. So können Spiele mit Interaktion gefüllt werden, ohne eine Zeile Code schreiben zu müssen. Allerdings benötigt dieses Verfahren ebenfalls Einarbeitung. Unity ist dafür weniger kompliziert aufgebaut. Hat man die

benötigten Bestandteile des Levels in seine Assets geladen, so ist es möglich, das ganze Level nur durch ‚zusammenstecken‘ der Komponenten zu bauen. Da es Teil der Anforderung ist, dass die Anwender keinen eigenen Code schreiben sollen, ist die Verwendung von Skripten statt ‚Blueprints‘ bei der Unity Engine zu vernachlässigen.

Tabelle 3: Benchmark Unity 5

Unity 5	
Format:	
FBX 2010	60,0Hz
FBX 2013.3	60,0Hz

Um CAD-Daten zu importieren ist eine vorherige Konvertierung in das .fbx-Format nötig. Dies geschieht bei beiden Engines auf gleiche Weise über den Converter von Autodesk. Die Importzeit der FBX-Datei ist bei den Entwicklungsumgebungen aber unterschiedlich. Die Unity Engine verändert die Objektdaten nicht, sondern erstellt eine weitere Datei, die die Objektdaten referenziert. Die Unreal Engine konvertiert das Objekt jedoch in ein Engine-eigenes Format. Während der Import eines Objekts mit 4 Millionen Vertices bei der Unity Engine 20 Minuten in Anspruch nahm, so benötigte die Unreal Engine 3 Stunden und mehr. Die Konvertierung in das eigene Format ist auch Schuld daran, dass die Knotenhierarchie des Objekts zerstört wird. Die Teilobjekte unterscheiden sich in der Unreal Engine nur noch an den verschiedenen Submaterials. Die Teilobjekte lassen sich in der Engine nicht mehr unabhängig voneinander ansprechen bzw. mit Logik belegen. Dies führt dazu, dass die CAD-Daten in der Unreal Engine nicht verwendbar sind. Um dies zu umgehen, ist eine vorherige Zerlegung des CAD-Objekts in seine Teilobjekte nötig. Dies geschieht in einem Modellierungsprogramm wie Blender oder 3DS Max und ist nur unter stark erhöhtem Zeitaufwand möglich.

Die Lernmodule lassen sich in der Unreal Engine in Blueprints ablegen, Unity verwendet dafür Skripte (in JavaScript oder C#). Während Blueprints die komplette Logik eines Objektes halten können, werden in den Skripten oft auch auf andere Skripte verwiesen und erstrecken sich somit nicht nur über eine Datei. Dafür ist es in der Unity Engine möglich, Schnittstellen zu definieren. So müssen vorgefertigte Lernmodule nicht zu einem späteren Zeitpunkt wieder geöffnet und gelesen werden. Es genügt, das richtige Objekt in der richtigen Schnittstelle zu verlinken.

Um das Lernprogramm auf seine Performanz zu testen, wurde ein CAD-Testobjekt in eine leere Welt geladen und geprüft, wie viele Bilder pro Sekunde bei der jeweiligen Engine erzeugt werden, wenn die Bilder auf die OVR gerendert werden. Im direkten Vergleich auf demselben Rechner, mit demselben Testobjekt, war die Unity Engine um rund 30% schneller gegenüber der Unreal Engine (siehe Tabelle 2 und Tabelle 3). Dies ist dadurch begründet, dass die Unreal Engine bereits Grafikshader nativ betreibt, während sie bei Unity erst hinzugefügt werden müssen. Der Realitätsgrad der Produkte beider Engines ist befriedigend. Aufgrund der Shader der Unreal Engine wirkt diese Lösung jedoch etwas realistischer. Um Daten über den Nutzer zu sammeln, beispielsweise darüber, wie schnell er war, oder wie viele Punkte er verdient hat, ist die Implementierung eines Skriptes nötig. Beide Engines liefern APIs um dies zu tun, vorgefertigte Funktionen gibt es nicht.

6 Konzeption und Implementierung

Dieses Kapitel enthält die praktischen Entwicklungen, die im Verlauf dieser Masterarbeit gemacht wurden. Im ersten Unterkapitel wird beschrieben, wie es zu den Ergebnissen des Kapitels 3.1 gekommen ist. In dem Unterkapitel 6.2 wird auf die Programmierarbeit eingegangen. Das letzte Unterkapitel „Probleme“ geht auf Erschwernisse bei der Entwicklung ein, und enthält Lösungen für selbige.

6.1 Aufnahme der Anforderungen an das System

Die Anforderungen an das System wurden in zwei Workshops aufgenommen. Teilnehmer und Gastgeber des ersten Workshops war der Heizungsbauer , Alfred Becker GmbH'. Im zweiten, gleich aufgebauten Workshop wurde die Firma ,AMAZONEN-Werke H. Dreyer GmbH & Co. KG ' befragt. Die befragte Gruppe beider Firmen bestand aus Mitarbeitern aller Abteilungen. So wurden Außendienstmitarbeiter, Monteure, Ausbilder und Büromitarbeiter nach Anforderungen interviewt.

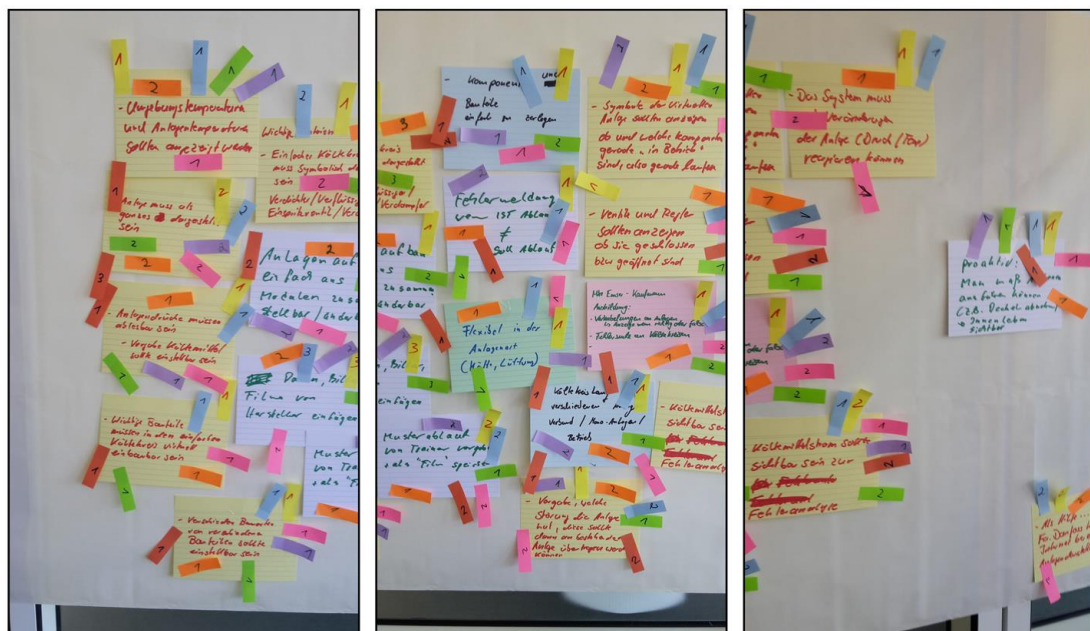


Abbildung 9: Anforderungssammlung

Die Workshops bestanden aus mehreren Teilen. Zuerst wurde das generelle Projektziel erläutert und die Personen so auf den Kontext eingestimmt. Waren die Teilnehmer auf das Thema fokussiert, wurden in einer Anforderungsaufnahmerunde die einzelnen Anforderungen aller Teilnehmer gesammelt. Dabei hatte jeder einen Stift und einen Stapel Zettel zur Verfügung um auf jedes Blatt eine separate Anforderung zu schreiben. Nachdem die Firmenmitarbeiter keine Anforderungen mehr stellten und eine Pause eingelegt wurde, galt es, die Anforderungen zu bewerten. Hierfür wurden die Zettel an einer Wand angebracht. Jeder der Teilnehmer musste nun jede der Anforderungen in seiner Wichtigkeit einstufen. Dabei war es wichtig, subjektiv auf die eigene Arbeit bezogen zu bewerten und dabei Punkte von 1 bis 3 zu vergeben (wobei 3 eine hohe Wichtigkeit bedeutet und 1 eine niedrige Wichtigkeit). Aus diesem Teil des Workshops erging bereits eine Vielzahl an Anforderungen, bewertet nach ihrer Wichtigkeit für die jeweilige Firma (siehe Abbildung 9).

Im darauffolgenden Teil wurden dann bereits entwickelte Prototypen gezeigt. Dabei durfte jeder Teilnehmer die Demos selbst ausprobieren. Vorgestellt wurden Entwicklungen die mit Leap Motion und dem OVR gemacht wurden, Vorläufer der praktischen Arbeit dieser Thesis und auch einfache Demos bei denen nur 3D-Modelle mit einem Spielecontroller umflogen wurden. Auf diese Weise sollten die Teilnehmer näher auf den VR-Kontext eingestellt werden, um im darauffolgenden Interview nach qualitativen Anforderungen abgefragt zu werden. Da nur begrenzt viele Teilnehmer gleichzeitig einen Prototypen ausprobieren konnten, mussten die übrigen Teilnehmer einen Fragebogen über ihre Erfahrung mit VR und ihre Rolle in der Firma ausfüllen.

Danach wurde jeder der Firmenmitarbeiter interviewt. Die Fragen des Interviewleitfadens (zu sehen in Abbildung 10) zielten vor allem darauf ab, Rückmeldung zu den Prototypen zu erhalten und konkrete Verbesserungsvorschläge zu bekommen. Die Resonanz zu den Thesis-Prototypen war bereits durchaus positiv. Zum einen konnten die Teilnehmer sehen, wie CAD-Daten in einer virtuellen Umgebung ‚begehrbar‘ gemacht wurden. Zum anderen wurde die Technologie dort bereits vorgestellt und ausprobiert. Der größte Teil der Teilnehmer konnte sich durchaus vorstellen, dass diese Anwendung in der Schulung von Nutzen oder gar von Vorteil wäre.

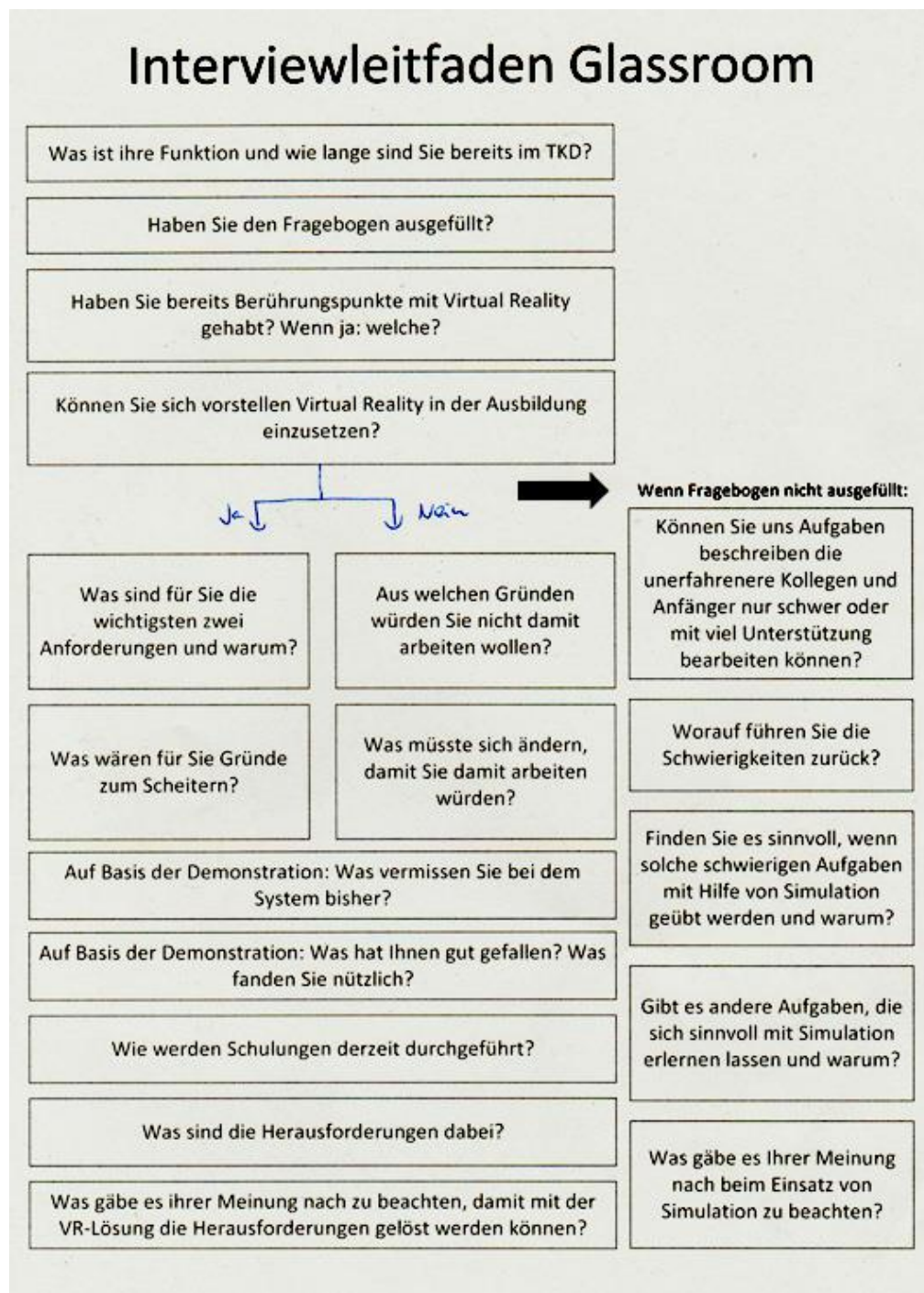


Abbildung 10: Interviewleitfaden des Workshop-Interviews

Die Auswertung der Interviews, sowie der Anforderungsplakate (Abbildung 9) führte zu den Ergebnissen von Kapitel 3.1. Diese dienten auch in der Bewertung (Kapitel 7) als Bewertungskriterien.

Wie bereits weiter oben beschrieben (Kapitel 4.3.3), muss ein Fragebogen auf seine Gütekriterien evaluiert werden. Nur so kann sichergestellt werden, ob die Ergebnisse objektiv, zuverlässig und zutreffend sind. Der Zweck dieses Fragebogens (Abbildung 10) ist es aber, Rückmeldung über das Projekt zu geben und Verbesserungsmöglichkeiten zu identifizieren. Die Verbesserung und Entwicklung des Produkts wird in iterativen Schritten wiederholt, bis die Verbesserungsvorschläge marginal sind (oder das Budget des Projekts erschöpft ist). Ab diesem Zeitpunkt kann das Projekt mit bereits evaluierten Fragebögen auf seine Nutzbarkeit geprüft werden, um die Tauglichkeit zu beweisen.

6.2 Vorgehen/Implementierung

6.2.1 Interaktion

Nachdem die Werkzeuge für die Entwicklung festgelegt waren, wurde nach einem bestehenden Plug-In für Unity gesucht, welches die Kinect v2 an die Engine anbindet. Das „Kinect v2 with MS-SDK“⁶ von RF Solutions bietet eine Schnittstelle für alle Sensordaten der Kinect. Dieses Plug-In liefert zu den Rohdaten auch teilweise gefilterte Daten, unter anderem um Zittern oder Schwierigkeiten bei der Erkennung der Gelenkorientierung auszugleichen.

Über den Kinect-Sensor ist es möglich, Farbbilddaten, Tiefenbilddaten, Infrarotbilder, Körperdaten, Zustandsdaten der Hände und sogar Mikrophon-Eingaben zu erhalten. Für die Entwicklung am Projekt wurden vor allem die Körperdaten, sowie die Zustandsdaten der Hände benötigt. Die Daten werden in Unity über den Kinect Manager erreicht, bzw. ausgegeben. Dieser Manager implementiert das Singleton-Pattern und dient als Zugangspunkt zur Kinect.

Um auf die Daten zuzugreifen wurde ein Interface implementiert, welches sich die Daten vom Manager abholt. Das Programm wurde so entwickelt, dass es um weite-

⁶ <https://www.assetstore.unity3d.com/en/#!/content/18708>

re Interfaces erweiterbar ist, um später weitere Interaktionsarten zuzulassen. Hierfür wurde ein Strategy-Pattern verwendet.

Die Körperdaten der getrackten Person werden auf einen Avatar übertragen, allerdings nimmt die Kinect keine genauen Fingerdaten auf. Die virtuelle Hand erhält nur einen Positionsvektor, Finger bleiben dabei statisch und auch eine eigene Orientierung (abgebildet als Quaternion) erhält sie nicht. Die Ausrichtung der Hand ergibt sich dabei aus dem Quaternion des übergeordneten Knochens, also des Unterarms. Um also Interaktionen mit der Hand möglich zu machen, erkennt die Kinect den Zustand der jeweiligen Hand. Diese Zustände sind Grip, Point, Release und None, wenn kein Zustand erkannt wird. Diese Zustände werden vom Plug-In zurückgegeben, wenn die Hand geschlossen ist, wenn mit dem Zeigefinger gezeigt wird, oder wenn sie geöffnet ist.

Durch die Positionsdaten und dem Zustand der Hand lassen sich nun Greifinteraktionen dann realisieren. Des Weiteren braucht man eine Erkennungs-Funktion die herausfindet, welche Objekte an der Position der Hand liegen. Bei einem Zustandswechsel von Release zu Grip wird dann das nächste Objekt an die Hand gebunden. Die Objekterkennung macht Unity über die Collider-Komponente möglich. So benötigt das zu greifende Objekt einen Collider. Dieser Collider kann darüber Rückmeldung geben ob eine Position sich innerhalb des Collider-Objekts befindet.

```
if(greifbaresObjekt.GetComponent<Collider>().  
    bounds.Contains(greif_position)){  
    greifeZu();  
}
```

Prüft, ob eine Position „greif_position“ innerhalb der Grenze eines Objekts „greifbaresObjekt“ liegt

Die Zustandswechsel sind innerhalb der Hand-Logik realisiert. Die Methoden onRelease() sowie onGrab() werden aufgerufen, wenn der Zustand von Grab zu Release oder umgekehrt, wechselt. Zusätzlich gibt es noch eine Methode onContinuedGrab(), welche immer dann aufgerufen wird, wenn in einem Frame und dem vorherigen Frame jeweils der Zustand auf Grab stand.

Die Aufgabe der Methode `onGrab()` ist es, das zu greifende Objekt zu finden und dieses Objekt zu benachrichtigen, dass es gegriffen wird. Die Methode `onRelease()` gibt dieses Objekt wieder frei und setzt auch den Zustand des Objektes wieder zurück. Der tatsächliche Griff, verbunden mit der Bewegung des Objektes, wird in `onContinuedGrab()` implementiert. Hier wird die Rigidbody-Komponente des Objekts über ihren `velocity`-Wert bewegt. Eine andere Herangehensweise wäre gewesen, die Positionsdaten des Objektes zu verändern. Dies führt aber zu sprunghaften Bewegungen des Objekts. Die Lösung über die Rigidbody-Komponente ist an die Arbeit von Beattie in [Bea2015] angelehnt, der diese Art und Weise der Greifinteraktion mit der Leap-Motion bereits löste.

```
Vector3 velocity = (greif_position -  
    greifbaresObjekt.transform.position) / Time.deltaTime;  
greifbaresObjekt.GetComponent<Rigidbody>().velocity = velocity;
```

Setzt die Geschwindigkeit eines Objektes „`greifbaresObjekt`“ auf den Vektor „`greif_position – greifbaresObjekt.position`“ in Relation zur Zeit

6.2.2 CAD-Daten zu Objekten

Um die Daten zu konvertieren wurde ein Converter von Autodesk verwendet (FBX Converter⁷). Ausgangsformat war dabei `.jt` und Zielformat `.fbx`. Das Objektformat `.fbx` ist in der Spieleentwicklung gängig und enthält neben den Informationen der Meshes auch die Knotendaten wie sie in der originalen Datei enthalten waren. Das Format lässt sich in Unity direkt weiterverwenden.

Eine typische Eigenschaft von CAD-Daten ist es, für jedes Teilobjekt einen eigenen Knoten zu haben. Da Maschinen aus vielen Teilobjekten bestehen, haben CAD-Daten im Umkehrschluss auch viele Knoten. Dies ist in der Spieleentwicklung eher untypisch, da versucht wird, an Polygonen und Knoten zu sparen um eine höhere Performanz zu erzeugen.

⁷ <http://usa.autodesk.com/adsk/servlet/pc/item?siteID=123112&id=22694909>

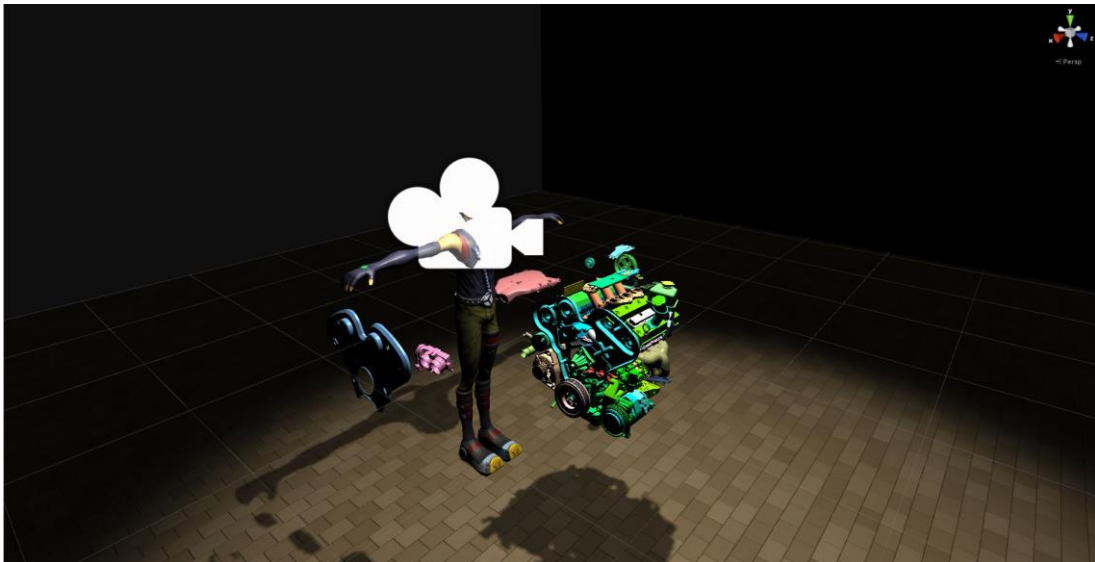


Abbildung 11: CAD-Objekt im Unity-Editor

Dies führt unter anderem zu dem Problem, dass es sehr aufwändig ist, jedes der Teilobjekte mit Logik zu versehen. Die Skripte, welche auf die Teilobjekte zugreifen wollen, müssen zuerst wissen, dass es die Teilobjekte überhaupt gibt. Um die Pflege des Programms möglichst einfach zu gestalten, wurde ein Script erstellt, welches die Teilobjekte automatisch in einer Collection ablegt. Dies wurde durch die Methode `BroadcastMessage()` realisiert, welche durch die Unity Engine-API mitgeliefert wird. Mit ihr ist es möglich, dass ein Eltern-Objekt eine Nachricht an alle ihre Kind-Objekte weiterleitet.

```
elternObjekt.BroadcastMessage ("registriereKindobjekt");
```

Methodenaufruf der Kinder, innerhalb des Elternobjekts

Daraufhin rufen alle Kind-Objekte die übergebene Methode auf und registrieren sich so eigenständig bei ihrem Eltern-Objekt:

```
public void registriereKindobjekt (){  
    GameObject.Find("ElternObjekt").  
        GetComponent<AssemblyScript>().registerChild(gameObject);  
}
```

Aufgerufene Registrierungsmethode innerhalb der Kinder

6.2.3 Spielelogik

Um ein einfaches Zusammenbau- und Auseinanderbau-Szenario möglich zu machen, wurde ein Script implementiert, welches eine Reihenfolge definiert, in welcher die Objekte auseinander und zusammengebaut werden müssen. Die Reihenfolge ergibt sich dabei aus der Hierarchie der Objektknoten. Die bewegbaren Spielobjekte sind dabei als Stack in dem Script referenziert.

Objekte, die nicht an der Reihe sind, sind dabei gefroren. Sie lassen sich also nicht bewegen. Ein Objekt, welches bereits richtig eingesetzt wurde, wird ebenfalls in den gefrorenen Zustand versetzt. Ist ein Objekt an der richtigen Position, erhält der Spieler einen Punkt. Insgesamt sind im Spiel also so viele Punkte zu holen, wie es Objekte zum Zusammenbauen gibt.

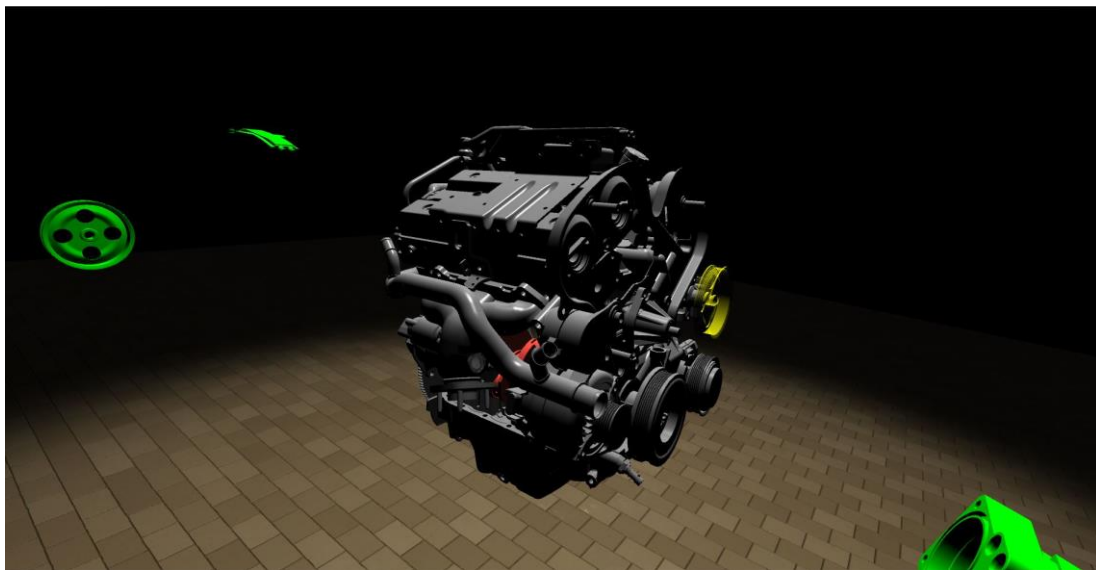


Abbildung 12: Die grünen Objekte sind fehlende Bauteile. Das gelbe Objekt ist der Platzhalter.

6.2.4 Ausgabe

Die Ausgabe erfolgte zu Beginn der Entwicklung noch über die von Oculus VR angebotenen Prefab-Objekte. Ab Unity Version 5.0 wird die OVR-Brille jedoch nativ von der Engine unterstützt. So ist es möglich, EXE-Anwendungen so zu bilden, dass das Bild direkt an das OVR übertragen wird.

6.3 Probleme

6.3.1 Cybersickness

Unter dem Begriff Cybersickness sammeln sich die Symptome Übelkeit, Orientierungslosigkeit und Schwindel, die man in einer virtuellen Umgebung empfinden kann. Die Ursachen dieser Symptome hängen von verschiedenen Faktoren ab, beispielsweise von der virtuellen Umgebung selbst, den verwendeten Technologien, sowie die Aufgaben, die in der virtuellen Welt erledigt werden müssen. In der Vergangenheit wurde ein Zusammenhang der Simulatorkrankheit, der Bewegungskrankheit und der Cybersickness erkannt. Zudem ähneln sich die Symptome auch stark mit dem „Space adaption syndrom“, welches Astronauten empfinden, wenn sie in der Schwerelosigkeit wandeln (vgl. [DaNe*2014 S. 2ff]).

Tabelle 4: Symptome die nach Auswertung des Virtual Reality Symptom Questionnaires bei mindestens 20% der Probanden vorkam [DaNe*2014 S. 6]

Nicht-okulare Symptome	Okulare Symptome
Müdigkeit	Augenmüdigkeit
Schläfrigkeit	Augenbeschwerden
Unwohlsein	Unwohlsein beim Sehen
Kopfschmerzen	Fokussierungsschwierigkeiten
Konzentrationsschwierigkeiten	Verschwommene Sicht
Benommenheit	Wunde/juckende Augen
Langeweile	

In einer Studie von Ames et al. [AmWo*2005] wurden Probanden nach Nutzung einer VR-Anwendung zu den empfundenen Symptomen von Cybersickness befragt (Tabelle 4).

Das Empfinden dieser Symptome führt unter anderem dazu, dass die Übungen in der virtuellen Welt ineffektiv werden, oder aber man dazu tendiert, das VR-System nicht zu verwenden [DaNe*2014 S. 4ff].

Die Autoren über die Ursachen: „The actual cause of cybersickness is not known and the underlying physiological responses uncertain.“ [DaNe*2014 S. 5]. Theorien für die Ursachen sind:

- Vergiftungstheorie: Der Körper denkt, er sei vergiftet. Ein Überlebensmechanismus tritt in Kraft, der den Körper dazu veranlasst, das restliche Gift aus dem Körper durch Erbrechen auszuscheiden.
- Haltungsinstabilitätstheorie: Der menschliche Körper hat den Drang, eine stabile Haltung einzunehmen. Dafür hat er bekannte Strategien um die stabile Haltung zu erreichen. In einer virtuellen Welt wird dem Körper eine instabile Haltung suggeriert, obwohl er eigentlich bereits eine stabile Position eingenommen hat. Um das auszugleichen fehlt dem Mensch die passende Strategie und dies führt wiederum zu Übelkeit.
- Sensorischer-Konflikt-Theorie: Die populärste Methode betrachtet den Gleichgewichts- und Sehsinn. Diese beiden Sinne arbeiten im Körper zusammen, um den Menschen stabil zu halten. In einer virtuellen Welt kann es aber zu einem Konflikt zwischen dem System des Sehsinns und dem des Gleichgewichtssinns kommen (beispielsweise bei einer virtuellen Achterbahnfahrt).

Keine der drei genannten Theorien erklärt aber, warum nur manche Personen von Cybersickness betroffen sind, und andere hingegen nicht. Zudem lässt keine Theorie eine Schätzung zu, wie schwerwiegend der Effekt von Cybersickness in einer virtuellen Welt sein wird [DaNe*2014 S. 5].

Die Stärke der Cybersickness hängt von verschiedenen Faktoren ab. Diese sind in individuelle Faktoren wie: Alter, Geschlecht, Krankheiten und Körperhaltung des Nutzers, technologische Faktoren (Verzögerungszeit, Flicker, Qualität der Kalibration und Ergonomie), sowie Faktoren der Aufgabe in der virtuellen Welt (Kontrolle,

die der Nutzer über sein Verhalten in der Welt hat, Dauer der Interaktion oder des VR-Erlebnisses) [DaNe*2014 S. 5-6].

Cybersickness lässt sich subjektiv, sowie objektiv messen. Bei einer subjektiven Messung wird der Nutzer befragt, und die Antwort auf einer Punkteskala vergeben. Ein Beispiel für so ein Verfahren ist das ‚Pensacola Motion Sickness Questionnaire‘. Dies ermöglicht es, den Grad der Cybersickness zu indexieren (‚Pensacola Diagnostic Index‘ nach [GrWo*1968]). Um das Empfinden von Cybersickness objektiv zu ermöglichen, werden Messgeräte benötigt, um die Herzfrequenz, die EEG-Rate, sowie die Frequenz des Blinzeln und die Magenverstimmung zu messen. In (vgl. [DaNe*2014 S. 6-7]).

Im Hinblick auf diese Arbeit ist es daher vor allem wichtig, auf das Auftreten der Symptome zu achten. Durch realitätsnahe Abbildung der echten Bewegung auf die virtuelle Welt soll die Cybersickness reduziert werden. Auf Kamerafahrten und abrupte Bildwechsel oder Einblendungen muss verzichtet werden, um die Benutzbarkeit und die Häufigkeit der Nutzung des Systems zu begünstigen. Zudem sollten Aufgaben kurz genug sein, um Pausen in den Verlauf einer Lerneinheit einbauen zu können.

6.3.2 Hindernisse

Aus der Arbeit [KiAd*2013] (beschrieben in Kapitel 2.4) geht hervor, dass Hindernisse im Trackingbereich zu einem Problem werden können. Um das Verletzungsrisiko des Nutzers zu minimieren, ist ein Bereich zu definieren (und diesen auch bspw. mit Klebeband abzugrenzen). In diesen Bereich darf zur Laufzeit des Programms niemand eindringen und auch kein Gegenstand stehen. Während der Arbeit an dem praktischen Projekt dieser Arbeit hat sich gezeigt, dass Nutzer abschätzen können, welche Gegenstände in der näheren Umgebung sind, anhand dessen, wo sie sich vor Aufsetzen des HMDs befanden. Ist ein Objekt für das Tracking nötig, da sich ein Nutzer bspw. setzen muss, so ist darauf zu achten, dass der Stuhl nicht seine Position ändert, nachdem der Nutzer die Brille aufgezogen hat.

6.3.3 Trackingfragmente

Der Treiber der Kinect v2 erkennt Menschen automatisch und speichert die Position der Joints in einem Körperobjekt ab. Oft werden aber Gegenstände von der Software als Körper erkannt und ebenfalls als Datenobjekt abgelegt. Besonders oft galt dies für Bürostühle, die nach Anwendungsstart in den Trackingbereich geschoben wurden.



Abbildung 13: Das Trackingsetup. Der schwarze Hintergrund soll dazu dienen, eventuelle Fragmente im Tracking zu reduzieren.

6.3.4 Infrarotlicht des OVR

Die Oculus verfügt über ein eigenes Positionsbestimmungssystem. Die dafür benötigte Kamera erkennt das OVR anhand seiner Infrarotlichter, die bei Aktivierung aufleuchten. Diese Lichter überstrahlen die ausgesandten Infrarotimpulse der Kinect. Dies führt dazu, dass der Avatar zu zittern scheint, oder Zustandsdaten der Hände nicht optimal erkannt werden. Diese Option der Positionsbestimmung lässt sich aber abschalten.

6.3.5 Trackinginkonsistenzen

Werden die Rohdaten direkt auf den Avatar übertragen, so passiert es oft, dass in einzelnen Frames die Zustands- oder Transformationsdaten springen. Um dies zu

reduzieren wurde eine Filterung als Schnittstelle implementiert, die statt die direkten Daten, den Mittelwert der Daten in einem Zeitabschnitt weiterreicht. Der Zeitabschnitt ist dabei wählbar. Dabei muss die Anzahl der Frames angegeben werden, die für die Filterung betrachtet wird. Diese Zahl muss mindestens größer gleich 3 sein.

7 Evaluation und Bewertung

In diesem Kapitel werden die Anforderungen auf Erfüllung bzw. Erfüllbarkeit geprüft, sowie die Ergebnisse der Nutzertests diskutiert.

7.1 Evaluation auf Basis der Anforderungen

Wie in Kapitel 3.1 beschrieben, lassen sich die Anforderungen in die Kategorien Organisation, Generierung, Output, Inhalt, Umgebung, Interaktion und Nutzung einteilen. Der Teil „Organisation“ beinhaltet Anforderungen, welche die Anwendung betreffen, beispielsweise die Pflegbarkeit, Mehrsprachigkeit und Bezahlbarkeit. Um dem gerecht zu werden, wurden für die Lernanwendung Consumer-Produkte wie die Kinect und die Oculus Rift verwendet, sowie Unity als Entwicklungsumgebung herangezogen. So ist die Anwendung „bezahlbar“, bzw. im Preis skalierbar, da je nach Budget eine größere Anzahl von Kinect-Kameras und Oculus-Rift-Displays angeschafft werden können. Die Pflege des Produkts ist dadurch möglich, dass die 3D-Objekte innerhalb der Lernmodule austauschbar sind. So ist sichergestellt, dass ein neues Produkt in bereits bestehende Lernmodule eingesetzt werden kann. Die Mehrsprachigkeit ist in den Folgeschritten nach dieser Masterarbeit zu gewährleisten.

Ein Kernpunkt der Masterarbeit war es, CAD-Daten in der Entwicklungsumgebung verfügbar zu machen. Unter anderem deswegen fiel die Wahl des Entwicklungswerkzeugs auf die Unity Engine. In dieser Engine ist es, wie in Kapitel 5.2 beschrieben, möglich, die CAD-Daten zu importieren, und dabei die Struktur der Daten beizubehalten (eine vorherige Konvertierung in das FBX-Format vorausgesetzt). Dies ist vor allem deshalb wichtig, weil die CAD-Daten in den anwendenden Firmen bereits vorhanden sind. So müssen 3D-Modelle nicht erst modelliert werden, um eine Lernanwendung daraus zu bauen, sondern es können bereits bestehende Daten verwendet werden.

Durch Erstellung von Lernmodulen kann sichergestellt werden, dass Aufgabentypen vordefiniert gebündelt werden. Als Beispiel wurde eine Anwendung geschrieben, die das Lernmodul eines Montagevorgangs abbildet. Der Anwender kann dabei ein 3D-Modell auseinander- bzw. zusammensetzen. Die Reihenfolge, in der die Teile

zusammengesetzt werden müssen, lassen sich vom Anwendungsersteller selbst festlegen.

In Texturen lassen sich Bilder und auch Filme abbilden um diese dem Anwender verfügbar zu machen. Auf diese Weise können innerhalb der Lernumgebung Lernmaterialien, sowie Maschineninformationen dargereicht werden.

Durch die Verwendung der Kinect und die Übertragung der Bewegung auf einen Avatar ist es möglich, eine virtuelle Repräsentation von sich selbst zu sehen. Dies dient zum einen der Immersion, zum anderen aber zur Orientierung innerhalb der Umgebung. Die Bewegungskamera als Schnittstelle zur Anwendung gilt als intuitiv, so können Greifbewegungen, wie man sie in der echten Welt macht, auch in der virtuellen Welt verwendet werden. Auf diese Weise lassen sich 3D-Objekte mit der Hand bewegen.

Dank des OVR können sich Nutzer in der Lernumgebung umsehen, als würden sie sich selbst in dieser Welt befinden. Eine realistische Einschätzung von Größen und Entfernungen ist auf diese Art ermöglicht, vorausgesetzt die Objektdaten und der Avatar sind im gleichen Maßstab zueinander. Außerdem ist der Anwender mit der Display-Brille von der echten Welt um sich herum abgeschottet. Dies soll begünstigen, dass sich der Lernende auf die Aufgabe konzentriert.

Um eine hohe Motivation bei der Bewältigung der Aufgaben zu erzeugen, kann ein Punktesystem eingeführt werden. Beispielhaft wurde das in der Anwendung umgesetzt, indem es für jeden richtigen Montageschritt einen Punkt gibt. Eine komplexere Wertung würde auch bei der Umsetzung eines didaktischen Konzepts Sinn machen. So können Dinge die besonders wichtig sind einen großen Punktebonus geben und unwichtigere, optionale nur einen kleinen. Eine geführte Highscore-Liste kann die Nutzer dazu anleiten, die Anwendung ein weiteres Mal durchzuspielen, um sich vielleicht zu verbessern und sich so von seinen Kollegen abzusetzen.

Eine Anforderung des Inhalts war es, dass Anlagen bzw. Maschinen flexibel dargestellt werden. Innerhalb der Engine sind Teilobjekte an- bzw. abschaltbar. So können auch nur Teile einer großen Maschine in einer Lernanwendung behandelt werden.

Durch die Entwicklung von Lernmodulen (Kapitel 3.2) soll sichergestellt werden, dass Aufgaben einem festen Schema erfolgen. Nur die Objekte, welche der Aufgabe zu Grunde liegen, sollen änderbar sein. Auch dies war eine Anforderung, die im vorneherein als wichtig erachtet wurde.

Von den genannten Anforderungen gibt es auch einige, die nicht umgesetzt wurden, oder die sich nicht umsetzen lassen. So wurde zum Beispiel genannt, dass das System eine Schnittstelle zu anderen Systemen haben soll. Durch Plug-Ins lassen sich Schnittstellen zu anderen Systemen zwar hinzufügen, es kommt aber darauf an, um welches System es sich handelt, und ob eine Verbindung Sinn ergibt. Beispielsweise wäre eine Ausgabe der Highscore-Liste in einem Tabellen-Format zur Weiterverarbeitung in MS Excel denkbar.

Ein weiterer genannter Punkt war, dass sich Durchläufe der Nutzer in der virtuellen Welt aufnehmen lassen, um sie zu einem späteren Zeitpunkt wieder anzuschauen. Umsetzbar wäre dies durch eine zusätzliche Kamera. Diese kann über ein Movie-Capture-Plug-In (oder einem zusätzlichen Programm) eine Filmdatei rendern lassen. Allerdings werden dann drei Bilder pro Zeitframe berechnet (linkes Auge, rechtes Auge und das Bild für die Aufnahme). Dies verringert die Bildwiederholrate auf dem OVR und ist in Bezug auf Cybersickness nicht wünschenswert. In Hinsicht auf die weitere Arbeit ist also zu prüfen, mit was für Zielsystemen gearbeitet wird.

Die Anforderungen hinsichtlich der Nutzung (Kapitel 3.1.7) sind weitestgehend nicht erfüllt. Die Anwendung ist als Lernumgebung für Schulungen konzipiert. Ein Einsatz in der Montage oder der Produktion ist daher nicht vorgesehen.

Die Firma Oculus hat angekündigt, dass die Verkaufsversion des OVR auch Kopfhörer beinhaltet. Simulierte Anlagengeräusche wurden im Prototyp dennoch nicht beachtet. Hierfür müsste man die entsprechenden Geräusche aufnehmen und in der Anwendung unterbringen. Allerdings wird der Lernende dadurch vom Ausbilder getrennt. In weiteren Arbeitsschritten wäre es wichtig zu erfahren, ob dies in Anbetracht der Didaktik erwünscht ist. Umsetzbar wäre auch diese Anforderung.

Viele der Anforderungen beziehen sich auch auf die Anzeige von arbeitsrelevanten Daten. Möglich wäre die Darreichung durch Texturen oder UI-Einblendungen. Der

Prototyp dieser Masterarbeit soll jedoch nicht durch ein hohes Aufgebot an Informationen ablenken. Im Mittelpunkt stehen die Interaktion und die Umsetzung eines Lernmoduls. In einer fertigen Anwendung wäre es möglich, alle möglichen Informationen an den Nutzer auszugeben. Auch hier wäre zu prüfen, ob es in Anbetracht der Didaktik sinnvoll ist, dem Nutzer diese Informationen zugänglich zu machen.

7.2 Nutzertests

Der Test der VR-Montageanwendung wurde in vier Teile unterteilt. Um auswertbare Daten zu erhalten, wurde der Test der Anwendung in der Thinkaloud-Methode durchgeführt. Die Aufzeichnungen der Nutzerantworten dienten dabei als qualitative Antworten. Um die Nutzererfahrung auch als quantifizierte Daten zu erhalten, wurde zusätzlich noch ein Fragebogen (ISONORM 9241/10) verwendet. Als Testpersonen wurden Probanden eingeladen, die bei ihrer Arbeit Erfahrung mit virtueller Realität oder virtuellen Umgebungen sammeln konnten. Ziel bei der Wahl von Experten als Testanwender war es, die Anwendung möglichst kritisch zu betrachten, aber auch Verbesserungsvorschläge und Ideen zu erhalten.

Der Nutzertest war in verschiedene Phasen unterteilt. Während der Vorbereitung durfte der Proband ein Papier-Labyrinth bearbeiten, und dabei laut kommentieren. Dies diente dazu, dass die ungewohnte Situation, seine Gedanken laut auszusprechen, eingeübt wurde. Damit sollten Hemmungen bei der Kommentierung der eigentlichen Anwendung beseitigt werden. Als Zweites galt es eine Vorbereitungsanwendung zu durchlaufen. Dabei begab sich der Anwender in die virtuelle Realität und konnte sich mit der Interaktion via Kinect vertraut machen. Hier wurde bereits die Thinkaloud-Methode angewandt. Dabei war es wichtig, dass die Interaktionsmöglichkeit nicht sofort erklärt wurde, sondern der Nutzer selbst darauf kommt, wie zu greifen ist und wofür die Hervorhebung der Objekte gedacht ist. Im dritten Teil des Tests durchlief der Nutzer die Montageanwendung. Dabei wurde ein teilweise zerlegter Motor gezeigt, dessen fehlende Teile um den Nutzer herum platziert waren. Das als nächstes einzufügende Bauteil wurde dabei durch ein gelbes Leuchten angezeigt. Der Anwender musste also erkennen, welches Teil als nächstes in der Baureihe war, und dieses dann selbstständig einsetzen. Zusätzlich galt es vom Nut-

zer ein hervorgehobenes Bauteil im Motor zu finden und sich so hinzuknien, dass es für ihn bestmöglich zu sehen war. Zum Schluss beantwortete der Nutzer den Fragebogen und konnte noch weitere Rückmeldung zur Anwendung geben.

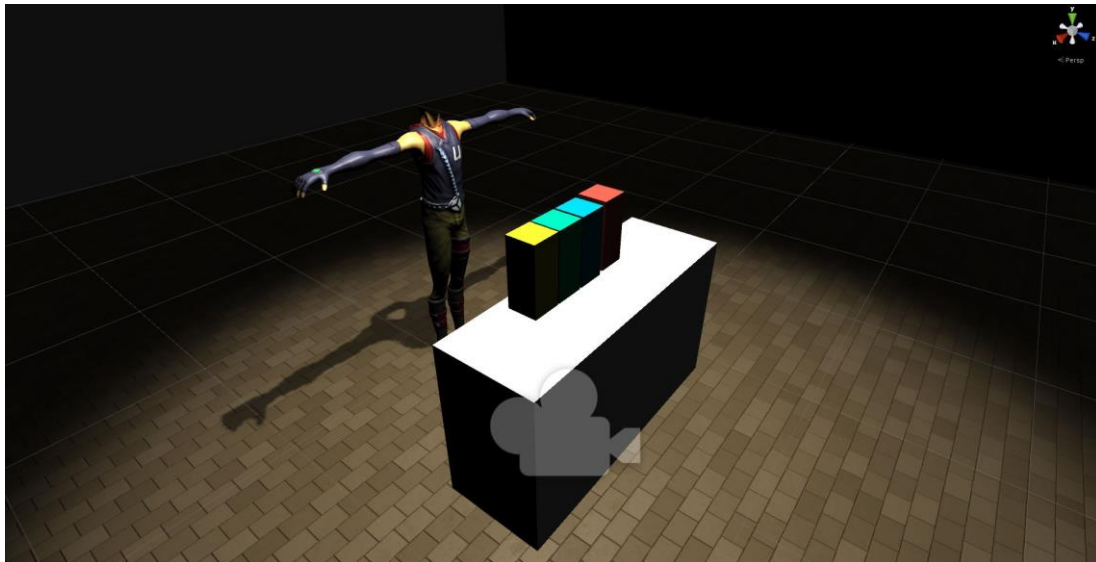


Abbildung 14: Vorbereitungsszene mit Quadern

7.2.1 Auswertung der Fragebögen

Der Fragebogen ISONORM 9241/110-S (siehe [Prü1997] und im Anhang unter „IV. Fragebogen“) beurteilt eine Software nach ihrer Ergonomie. Der Autor Prof. Prümper zog dafür die Norm DIN EN ISO 9241-110 heran. Die Fragen sind einzuteilen in: Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Erwartungskonformität, Lernförderlichkeit, Steuerbarkeit und Fehlertoleranz einer Anwendung. Jede Frage enthält eine Aussage, welche zu bewerten ist, je nachdem, ob sie zutrifft, oder nicht zutrifft (siehe Abbildung 15).

Beispiel 1

Die Software...	---	--	-	-/+	+	++	+++	Die Software...
ist schlecht.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ist gut.

Abbildung 15: Beispielfrage

Diese Einschätzung ist im Auswertungsschritt einer Zahl zuzuordnen.

- --- (drei Minus) entspricht einer Wertung von 1.
- +++ (drei plus) entspricht einer Wertung von 7.

Die acht Probanden mussten zu Beginn ihre VR-Vorerfahrung nach selbiger Methode einschätzen. Die durchschnittliche VR-Erfahrung der Probanden lag bei einem Wert von 6,0.

Da es sich bei der getesteten Anwendung nicht um eine fertige Software, sondern genaugenommen um eine Interaktionsmethode handelt, sind die letzte Sektion des Fragebogens, sowie Frage sk2 nicht möglich zu beantworten bzw. übertragbar. In folgenden Abbildungen werden die Mittelwerte der Antworten mit ihren Standardabweichungen angezeigt. Im Anhang unter „V. Fragebogenauswertung des Nutzertests“ sind die tabellarischen Auswertungen der einzelnen Fragen aufgeführt.

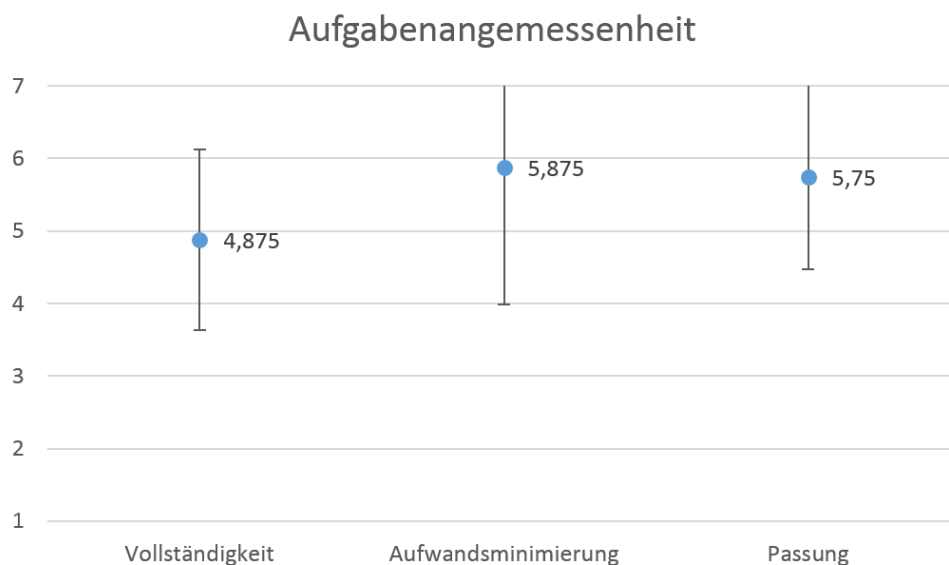


Abbildung 16: Fragebogenauswertung Aufgabenangemessenheit

Der gesamte Fragenbereich „Aufgabenangemessenheit“ erhielt einen Mittelwert von 5,5. Wie in Abbildung 16 zu sehen ist, sind die Nutzer besonders damit zufrieden, dass keine überflüssigen Eingaben notwendig sind.

Der Bereich „Selbstbeschreibungsfähigkeit“ erhielt einen Durchschnittswert von 4,7. Wobei die Frage nach der Unterstützungsmöglichkeit, ob auf Verlangen situationsspezifische Erklärungen erhalten werden können, schlechter abschnitt als die beiden anderen Fragen der Kategorie. Dies ist darauf zurückzuführen, dass es in dem einfachen Prototyp keine schriftliche Hilfestellung gibt. Diese fehlte, um den Proband nicht von der Interaktionsmethode abzulenken, die es hauptsächlich zu bewerten galt.

Die Erwartungskonformität, durchschnittlich mit 5,9 gewertet, umfasst Aspekte der einheitlichen Gestaltung, Rückmeldung darüber, welchen Zustand das Programm derzeit hat und die einheitliche Interaktionsmethode. Vor allem Letzteres schnitt mit einem Mittelwert von 6,75 und einer Standardabweichung von 0,46, besonders gut ab.

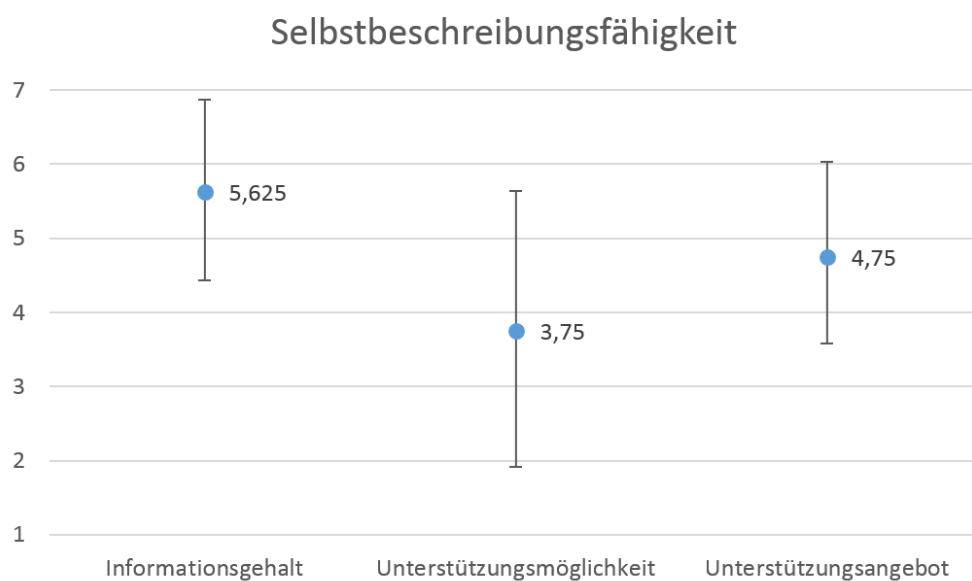


Abbildung 17: Fragebogenauswertung Selbstbeschreibungsfähigkeit

Der Mittelwert des Abschnitts „Lernförderlichkeit“ ist mit 6,2 der am besten abschneidende. Durch die Verwendung eines NUI, in diesem Fall also die Kinect, kommt die gute Erlernbarkeit zustande. Die etwas schlechter bewertete Erschließbarkeit rührt daher, dass die Interaktionen mit Objekten nicht wie in der echten

Welt ablaufen. Beispielsweise werden große Objekte nicht mit zwei Händen genommen, sondern gleich wie kleine Objekte mit nur einer Hand gegriffen.

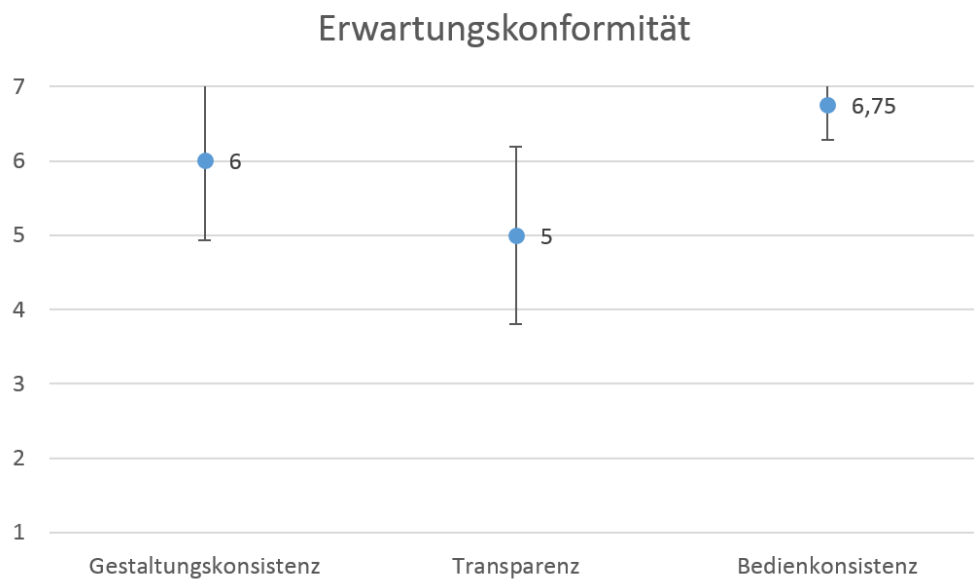


Abbildung 18: Fragebogenauswertung Erwartungskonformität

In der Steuerbarkeit schnitt die Anwendung wesentlich schlechter ab. Mit einem Mittelwert von 4,3 bei nur zwei Fragen ist dies der schlechteste Abschnitt. Besonders oft wurde bemängelt, dass die Anwendung unnötige Unterbrechung der Arbeit erzwingt. Dies wird hervorgerufen durch die Erkennung des Trackings, und durch die Art und Weise der Umsetzung von Objektbewegung. Die hohe Standardabweichung von 2,26 unterstreicht den Zwiespalt bei den Nutzern über diesen Aspekt.

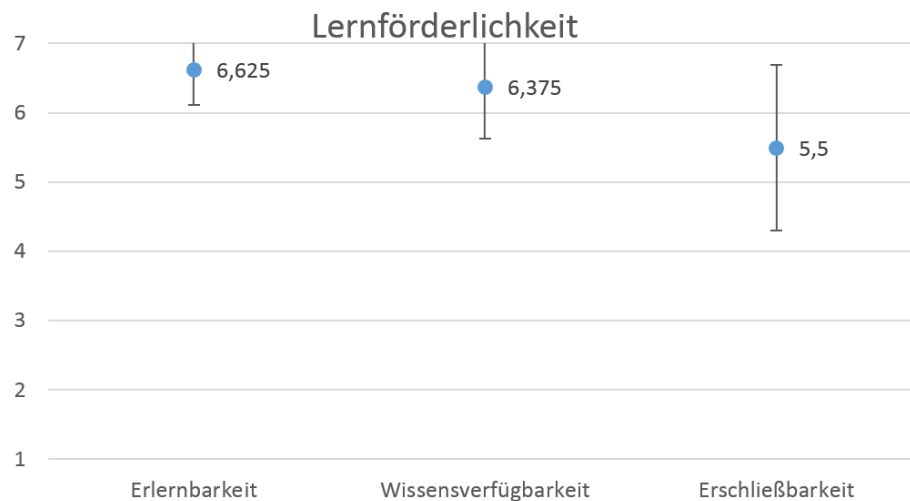


Abbildung 19: Fragebogenauswertung Lernförderlichkeit

Die Probleme bei der Steuerbarkeit führten bei den Tests auch zu einer schlechten Bewertung der gesamten Fehlertoleranz. Mit einem Mittelwert von 4,5 ist dieser Bereich nur unwesentlich besser bewertet als der vorangegangene. Vor allem wurde bemängelt, dass nicht ausreichend Hinweise über die Fehlerbehebung angeboten wurden.

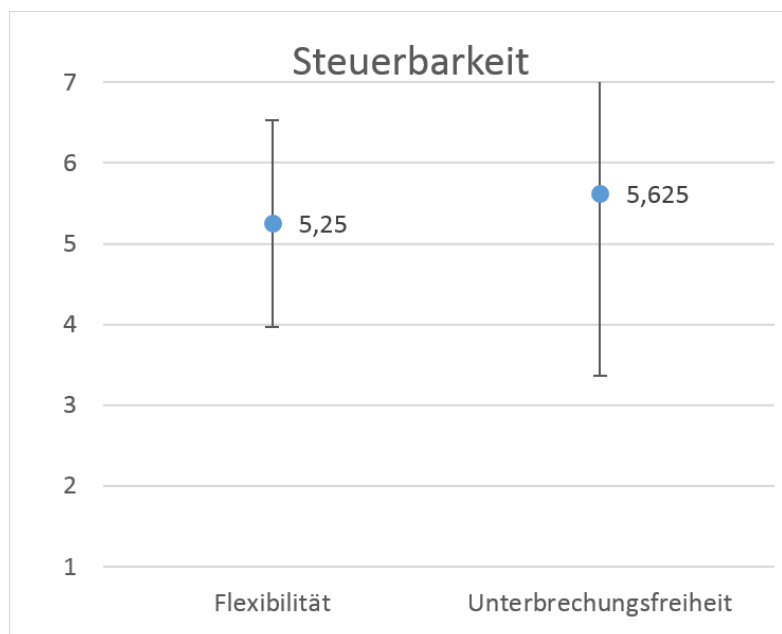


Abbildung 20: Fragebogenauswertung Steuerbarkeit

Insgesamt waren den Nutzern zu wenige Hinweise gegeben, was die aktuelle Aufgabe ist, oder worin der Fehler ihrer Falscheingabe liegt. Für weitere VR-Anwendungen muss untersucht werden, wie sich Hinweise über ein HMD darstellen lassen. Zudem ist es wichtig das Tracking und die Interaktion mit der Welt weiter zu verbessern. So müssen die Collider um die Hände vergrößert werden, sodass Objekte besser gegriffen werden können, aber auch die Objektbewegung grundsätzlich verändert wird.

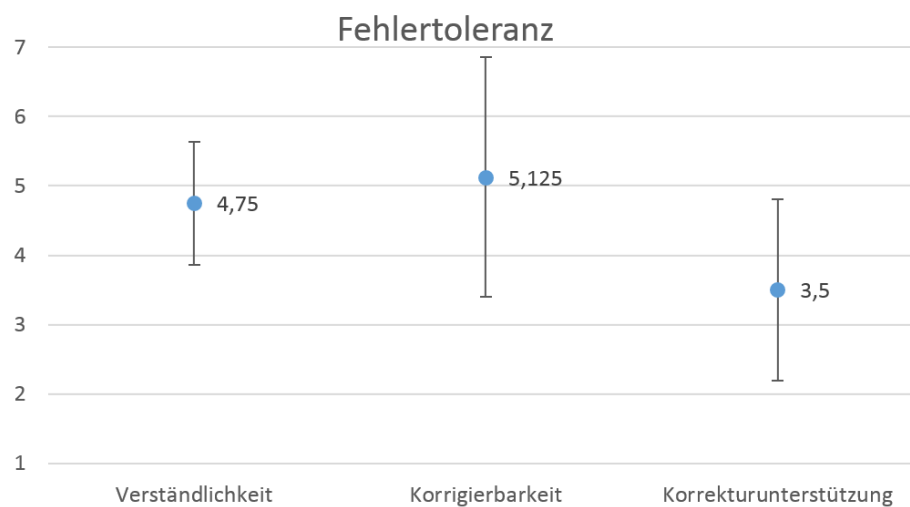


Abbildung 21: Fragebogenauswertung Fehlertoleranz

7.2.2 Auswertung der Thinkaloud-Rückmeldungen

Die Art und Weise wie gegriffen werden muss, wurde von den meisten direkt verstanden. Nur zwei Personen versuchten zunächst mit zwei Händen zu greifen. Auch die Hervorhebung, welche auftritt, wenn ein Objekt greifbar wird, wurde von fünf der acht Probanden direkt richtig erkannt. Das Manövrieren im Level und das Betrachten von Objekten wurde von allen Probanden positiv aufgenommen. Vor allem das Hinknien und genaue Betrachten der hervorgehobenen Kurbelwelle des Motors überzeugte durch die Darstellung mittels HMD.

Allerdings wirkte sich die Dämpfung der Bewegung zu stark aus. Diese wird zwar benötigt, um Fehler im Tracking auszugleichen, doch die Stärke führte bei drei Pro-

banden sogar zu Schwindel. Ebenfalls machte sich bei zwei Probanden eine niedrigere Bildfrequenz bemerkbar.

Die Aufnahme von Objekten funktionierte bei vielen Probanden nur schlecht. So war es für manche Teilnehmer schwer, Objekte zu erreichen, da die Collider der Hände kleiner waren als von den meisten angenommen. Dies führte bei vielen Probanden zu einem Frustrationsgefühl. Auch rotierten die Objekte oft schlagartig um ihren verschobenen Ursprungspunkt. Dies sah für die Probanden aus, als würde das Objekt aus ihrer Hand *springen*.

Die Anwendung selbst wurde sehr positiv gewertet. So wurde die gute Qualität der Grafik bemerkt sowie, dass das Leveldesign generell sehr schlicht und aufgabenorientiert war. Die eingeblendete Hilfe, die kennzeichnet, welches Objekt an welcher Stelle eingesetzt werden muss, erkannten fünf der Probanden sofort. Auch die Hilfe beim Einsetzen der Objekte bewerteten die Nutzer gut. Die Interaktion in der Anwendung führte sogar bei dem einen oder anderen Nutzer zu einem erhöhten Spaßfaktor. Ein Nutzer bemerkte sogar, dass er sich haptisches Feedback einbildete, da der Immersionsgrad derartig hoch gewesen sei.

Generell wurde erwähnt, dass der Aufbau von HMD und der markerloser Bewegungsaufnahmekamera besser sei, als eine Powerwall oder Cave. Mehr als die Hälfte der Anwender sahen die Technologien als einschränkungslos anwendbar in einem VR-Lernprogramm. Nur zwei Personen fanden, dass dies nur unter Einschränkungen möglich sei.

8 Diskussion

Die Auswertung der Nutzertests und die Betrachtung der gesammelten Anforderungen zeigen, dass die Entwicklung einer derartigen VR-Lernanwendung in die richtige Richtung geht. Die Einrichtungszeit des OVR-Kinect-Aufbaus und der geringe Preis sprechen für eine unkomplizierte Anschaffung und Einbringung auch in mittelständische Unternehmen. Die Probanden konnten sich vorstellen, dass ein derartiger Aufbau in der Schulung tatsächlichen Einsatz findet. Vorher sind aber noch Probleme zu beseitigen. So führen zum Beispiel die starke Dämpfung und die teilweise niedrige Framerate zu gelegentlicher Übelkeit. Auch Probleme bei der Objektbewegung oder die sprunghafte Transformation der Teile sind Probleme, die auf dem Weg zu einer fertigen Anwendung zu beseitigen sind. Die Kernfrage der Arbeit, ob ein derartiger Aufbau zur Akzeptanz beim Nutzern führt, kann aber mit ‚Ja‘ beantwortet werden.

Auch passt eine Kombination von CAD-Daten und Unity gut in die Arbeitssituationen von Maschinenbauunternehmen. Durch die Konvertierung der Daten mittels dem Autodesk Converter in das .fbx-Format, ist ein unkomplizierter Import der Daten möglich. Eine Vor- oder Nachbereitung der Daten bleibt somit aus.

Durch die Gestaltungsfreiheiten die durch die Verwendung einer Spieleentwicklungsumgebung gegeben sind, lassen sich leicht Informationen über die Geräte einbinden, die Anwendung mehrsprachig gestalten, sowie die aktuellen Maschinendaten einpflegen. Die Möglichkeit innerhalb der Engine auf die Unity-API zuzugreifen, erleichtert es immens eine spannende Darreichungsform von Lehr-/Lerninhalten zu schaffen.

Um die Anwendung und den Aufbau bereit für Feldversuche bei den Maschinenbauunternehmen zu machen, wäre es in folgenden Arbeiten wichtig, Trackinginkonsistenzen zu bereinigen und eine mit HMDs verwendbare GUI einzubinden. Dies soll in weiterer Arbeit im Projekt GLASSROOM auch geschehen.

Literaturverzeichnis

- [AmWo*2005] S. L. Ames, J. S. Wolffsohn, N. A. McBrien: "The development of a symptom questionnaire for assessing virtual reality viewing using a head-mounted display", 2005. Optometry & Vision Science, 82
- [Bea2015] N. Beattie: „CAD Experiment: Disassemble a Spherical Robot in VR“, 2015. <http://blog.leapmotion.com/cad-experiment-disassemble-spherical-robot-vr/> (Zugriff am 26 Mai 2015)
- [DaNe*2014] S. Davis, K. Nesbitt, E. Nalivaiko: "A Systematic Review of Cyber-sickness", 2014. IE2014. ACM, Newcastle, Australia
- [GrJa2014] A. Grabowski, J. Jankowski: „Virtual Reality-based pilot training for underground coal miners“, 2014. Safety Science 72, S. 310-314
- [GrWo*1968] A. Graybiel, C. D. Wood, E. F. Miller, D. B. Cramer: Diagnostic criteria for grading the severity of acute motion sickness, 1968. Aerospace Medicine Research Labs, 39
- [GrRo2014] D. S. Greuter, P. D. J. Roberts: „SpaceWalk: Movement and Interaction in Virtual Space with Commodity Hardware“, 2014. ACM, Newcastle, Australia
- [HaBa2015] T. Hachaj, D. Baraniewicz: „Knowledge Bricks - Educational immersive reality environment“, 2015. International Journal of Information Management, Elsevier Ltd
- [KiAd*2013] N. Kitsunezaki, E. Adachi, T. Masuda und J.-i. Mizusawa: „KINECT Applications for The Physical Rehabilitation“, 2013. IEEE
- [LaKo*2012] B. Lange, S. Koenig, E. McConnell, C.-Y. Chang, R. Juan, E. Suma, M. Bolas, A. Rizzo: „Interactive Game-Based Rehabilitation Using the Microsoft Kinect“, 2012. IEEE Virtual Reality 2012, Orange County, CA, USA

- [Lew1982] C. Lewis: "Using the 'thinking aloud' method in cognitive interface design", 1982. Research Reports RC 9265 (40713), IBM. Yorktown Heights, NY
- [MaVo2015] E. Matsas, G.-C. Vosniakos: „Design of a virtual reality training system for human-robot collaboration in manufacturing tasks“, 2015. Springer-Verlag, Frankreich
- [Mei2013] J. Meisner: „Collaboration, expertise produce enhanced sensing in Xbox One“, 2013. <http://blogs.microsoft.com/blog/2013/10/02/collaboration-expertise-produce-enhanced-sensing-in-xbox-one/> (Zugriff am 12 Mai 2015)
- [MsKin2015] Microsoft-Datenblatt: „Microsoft Kinect v2 Datenblatt“, 2015. <http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx> (Zugriff am 12 Mai 2015)
- [Nie1994] J. Nielsen: "Usability Engineering", 1993. AP Professional Ltd. London, UK
- [OcFAQ2015] Oculus-FAQ: „Oculus Rift Development Kit 2 FAQ“, 2015. <https://support.oculus.com/hc/en-us/articles/201835987-Oculus-Rift-Development-Kit-2-FAQ> (Zugriff am 12 Mai 2015)
- [OcVR2015] Oculus-Datenblatt: „Oculus Rift VR Datenblatt“, 2015. <https://www.oculus.com/dk2/> (Zugriff am 12 Mai 2015)
- [Prü1997] J. Prümper: "Der Benutzungsfragebogen ISONORM 9241/10: Ergebnisse Zur Reliabilität und Validität", 1997. Software-Ergonomie '97, Stuttgart
- [RaSi*1994] M. Rauterberg, P. Sinas, O. Strohm, E. Ulich, D. Waeber: "Benutzerorientierte Softwareentwicklung. Konzepte, Methoden und Vorgehen zur Benutzerbeteiligung", 1994. Teubner Verlag, Stuttgart

-
- [RoCa2002] M. B. Rosson, J. M. Carroll: "Usability Engineering. Scenario-based Development of Human-Computer Interaction", 2002. Morgan Kaufmann. San Francisco, CA.
- [SaBr2011] F. Sarodnick, H. Brau: "Methoden der Usability Evaluation - Wissenschaftliche Grundlagen und praktische Anwendung", 2011. Verlag Hans Huber, Bern, Schweiz
- [ShCh2014] S. Sharma, W. Chen: „Multi-user VR Classroom with 3D interaction and real-time motion detection“, 2014. International Conference on Computational Science and Computational Intelligence, IEEE
- [Vir1992] R. A. Virzi: "Refining the test phase of usability evaluation: how many subjects is enough?", 1992. Human Factors 34. S. 457-468
- [XiLo*2011] P. Xia, A. M. Lopes, M. T. Restivo, Y. Yao: „A new type haptics-based virtual environment system for assembly training of complex products“, 2011. Springer-Verlag, London
- [Zim2015] A. Zimmermann: „Mit Virtual Reality zum E-Learning - eine prototypische Anwendung“, 2015. Informatics Inside '15 - Interaction Design, S. 58-67

Eidesstattliche Erklärung

Ich versichere ehrenwörtlich:

- Ich habe diese Thesis selbstständig verfasst,
- alle benutzten Quellen und Hilfsmittel – dazu zählen auch sinngemäß übernommene Inhalte, leicht veränderte Inhalte sowie übersetzte Inhalte – in Quellenverzeichnissen, Fußnoten oder direkt bei Zitaten angegeben,
- alle wörtlichen und sinngemäßen Zitate von Textstücken, Tabellen, Grafiken, Fotos, Quellcode usw. aus fremden Quellen als solche gekennzeichnet und mit seitengenauen Quellenverweisen versehen, und
- alle nicht als Zitate gekennzeichneten Inhalte selbst erstellt.
- Ich kenne und achte den Leitfaden für gute wissenschaftliche Praxis „Wissenschaftliches Arbeiten – Leitfaden des Studiengangs Medien- und Kommunikationsinformatik, Fakultät Informatik, Hochschule Reutlingen“.
- Die von mir eingereichten Dokumente und Artefakte wurden noch nicht in dieser oder ähnlicher Form einer anderen Kommission zur Prüfung vorgelegt.

Mir ist bekannt, dass unmarkierte und unbelegte wörtliche und bildliche Zitate und Paraphrasen Plagiate sind und nicht als handwerkliche Fehler, sondern als eine Form vorsätzlicher Täuschung der Prüfer gelten, indem der Plagiator fremde Gedanken als eigene Gedanken vortäuscht, um sich eine bessere Leistungsbewertung zu erschleichen.

Mir ist bekannt, dass Plagiarismus die Standards guter wissenschaftlicher Praxis, die Regeln des Studiengangs Medien- und Kommunikationsinformatik, die Studien- und Prüfungsordnung der Hochschule Reutlingen (§ 10 Täuschung und Ordnungsverstoß) und das Landeshochschulgesetz von Baden-Württemberg (§ 3 Wissenschaftliche Redlichkeit Abs. 5, § 62 Exmatrikulation Abs. 3) missachtet und seine studienrechtlichen Folgen vom Nichtbestehen bis zur Exmatrikulation reichen.

Mir ist auch bekannt, dass Plagiate sogar das Urheberrechtsgesetz (§ 51 Zitate, § 63 Quellenangabe, § 106 Unerlaubte Verwertung urheberrechtlich geschützter Werke) verletzen und zivil- und strafrechtliche Folgen nach sich ziehen können.

Reutlingen, den 26. September 2015

Alexander Zimmermann

Anhang

I. Glossar

WORT	BEDEUTUNG
AHRS	Kurz für 'Attitude Heading Reference System'. Ein System welches die Drehung, Beschleunigung und Lage auf der X-, Y- und Z-Achse des Sensorgeräts messen kann.
API	Kurz für 'Application programming interface'. Eine API ist eine Sammlung von Schnittstellen die dem Verwender Funktionen zur Verfügung stellt, ohne dass er diese selbst implementieren muss.
Blueprints	Im Kontext der Unreal Engine ist das eine Art und Weise wie Funktionen erstellt werden können. Diese visuelle Scriptsprache ermöglicht es, Eigenschafts- und Funktionsknoten miteinander zu koppeln um ein Verhalten zu erstellen.
CAD-Modell	Ein digitales 3D-Modell welches bei der Konstruktion von Werkstücken hilft. CAD steht für ,computer-aided design'.
Cybersickness	Krankheitsgefühl welches man bei der Verwendung von VR-Technologie empfinden kann.
FOV	Kurz für 'Field-of-view'. Ein gradueller Wert der angibt, wie groß der horizontale Winkel des Sichtbereichs ist.
HMD	Kurz für 'Head-mounted-display'. Ein Anzeigegerät welches wie eine Brille bzw. wie ein Helm auf dem Kopf getragen wird.
Low-cost-	Niedrig-preisige Entwicklung, meist für Konsumenten und

development	nicht für Firmen. Es gibt keine genaue Definition, ab wann etwas niedrig-preisig ist und wann es ein ‚high-cost-development‘ ist.
NUI	Kurz für ‚Natural user interface‘, eine Art der Interaktion mit Programmen, bei dem natürliche Gesten verwendet werden. Die Idee ist, dass man kein neues Verhalten lernen muss, sondern bereits gelernte Interaktion intuitiv einsetzen kann.
OVR	Kurzform für „Oculus Virtual Reality“, das von Oculus entwickelte Anzeigegerät.
(Grafik-) Shader	Shader ermöglichen es, Render-Effekte zu erzeugen. Man unterscheidet zwischen softwareseitigen und hardwareseitigen Shadern.
Thinkaloud	Eine Testmethode, bei der eine Software-Anwendung durchlaufen wird, und der Nutzer laut kommentiert was er tut und was er denkt.

II. Bewertung der Entwicklungsumgebungen

Kriterien:	Gewichtung:	Engine:			
		Unity 4		Unreal Engine 5	
		Wert	gewichteter Wert	Wert	gewichteter Wert
Rahmenbedingungen					
Hardwareanforderung	3	0	0	0	0
Softwareanforderung	3	0	0	0	0
Kosten	4	4	16	5	20
Entwicklung					
Technologie					
Anbindung OVR	6	5	30	5	30
Anbindung Kinect V2	6	3	18	4	24
Dokumentation	5	5	25	5	25
Lernmodule					
Einfachheit/Erlernbarkeit	7	3	21	2	14
Dauer des CAD-Datenimports	4	5	20	1	4
Verwendbarkeit der importierten CAD-Daten	9	5	45	0	0
Gestaltung von Lernmodulen	7	4	28	5	35
Definieren von Schnittstellen	5	5	25	1	5
Laufzeit					
Performanz des Ergebnisses	7	4	28	2	14
Aussehen/Realismus	5	3	15	4	20
Sammeln von Daten über Nutzer	3	2	6	2	6
Summe:			277		197

Gewichtung von 1-10, hohe Gewichtung bedeutet eine hohe Wichtigkeit.

Bewertungsmaßstab von 5 (sehr gut) bis 1 (mangelhaft). Eine Wertung von 0 bedeutet, dass das Kriterium nicht erfüllt wurde.

III. Anforderungen

Generierung: Geführtes Training nach Musterablauf E-Learning Inhalte einbinden (Schnittstellen zu anderen Systemen) Modulare Objektgenerierung Normen einhalten Szenario nach geführtem Training aufnehmen Störungsbasiertes Training CAD-Daten einbinden Filme einbinden Bilder einbinden	Organisation: Regelmäßige Pflege Internationaler Zugriff Bezahlbarkeit Verwendung für Standardabläufe Standards setzen Wartungsfreundlichkeit Mobiler Zugriff Schneller Zugriff Drahtlose Ausführung Sprachneutralität Handlichkeit
Interaktion: Änderung im System müssen angezeigt werden Einbau von Teilen Einblenden von Teilen Arbeitsschritte validiert/Interaktion bewusst eingeschränkt Bewegung von Teilen Simulation einer Infrarot-Wärmebild-Kamera Zerlegung des Objekts Intuitiv	Nutzung: <i>(Nutzung im Innendienst)</i> Nutzung in Schulungen <i>(Nutzung in Montage)</i> <i>(Nutzung in Produktion)</i> Nutzung zur Bedienungsschulung <i>(Nutzung in Konstruktion)</i> <i>(Nutzung zur Fehlersuche)</i> <i>(Nutzung durch Monteure vor Ort)</i>
Umgebung: Lichtsimulation Abstandsmessung Realitätsnah Motivierend Abschottung von der Umwelt Simulierte Anlagengeräusche	Output: Erzeugen von Papieranleitungen Erzeugen von Schnitten

Inhalt: Metadaten zu Objektteilen Fehlerhinweise Bewegungsdaten zu Teilobjekten Selbstdarstellung Interface für Maschinenparameter Flexibilität in der Anlagendarstellung Umgebungssimulation (simulierte) Temperaturen sollen angezeigt werden Sicherheitshinweise Werkzeughinweise Bauteilbezogene Parameter anzeigen Stückliste des Objektes anzeigen Modellbaum des Objektes anzeigen Teilenummer des Objektes anzeigen Material des Objekts anzeigen Gewicht des Objekts anzeigen Spezifikation des Objekts anzeigen Zustandsanzeige von regelbaren Objekten Anzeige der Objektaktivität Symbolische Abstraktion als zusätzliche Anzeige der Bauteile Ablauf anzeigen Anzeige von kompletten Anlagen Technische Zeichnungen einbetten	
--	--

(eingeklammerte Anforderungen gelten für das AR-System, welches nicht Thema der Thesis ist.)

IV. Fragebogen

	Die Software ...	--- -- - -/+ + ++ +++	Die Software ...
aa1	bietet nicht alle Funktionen, um die anfallenden Aufgaben effizient zu bewältigen.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	bietet alle Funktionen, um die anfallenden Aufgaben effizient zu bewältigen.
aa2	erfordert überflüssige Eingaben.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	erfordert keine überflüssigen Eingaben.
aa3	ist schlecht auf die Anforderungen der Arbeit zugeschnitten.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	ist gut auf die Anforderungen der Arbeit zugeschnitten.
sb1	liefert in unzureichendem Maße Informationen darüber, welche Eingaben zulässig oder nötig sind.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	liefert in zureichendem Maße Informationen darüber, welche Eingaben zulässig oder nötig sind.
sb2	bietet auf Verlangen keine situationsspezifischen Erklärungen, die konkret weiterhelfen.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	bietet auf Verlangen situationsspezifische Erklärungen, die konkret weiterhelfen.
sb3	bietet von sich aus keine situationsspezifischen Erklärungen, die konkret weiterhelfen.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	bietet von sich aus situationsspezifische Erklärungen, die konkret weiterhelfen.
ek1	erschwert die Orientierung durch eine uneinheitliche Gestaltung.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	erleichtert die Orientierung durch eine einheitliche Gestaltung.

Abbildung 22: Fragebogen Teil 1 (aus [Prü1997 S.4])

	Die Software ...	--- -- - -/+ + ++ +++	Die Software ...
ek2	informiert in unzureichendem Maße über das, was es gerade macht.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	informiert in ausreichendem Maße über das, was es gerade macht.
ek3	lässt sich nicht durchgehend nach einem einheitlichen Prinzip bedienen.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	lässt sich durchgehend nach einem einheitlichen Prinzip bedienen.
lf1	erfordert viel Zeit zum Erlernen.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	erfordert wenig Zeit zum Erlernen.
lf2	erfordert, dass man sich viele Details merken muss.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	erfordert nicht, dass man sich viele Details merken muss.
lf3	ist schlecht ohne fremde Hilfe oder Handbuch erlernbar.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	ist gut ohne fremde Hilfe oder Handbuch erlernbar.
sk1	erzwingt eine unnötig starre Einhaltung von Bearbeitungsschritten.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	erzwingt keine unnötig starre Einhaltung von Bearbeitungsschritten.
sk2	ermöglicht keinen leichten Wechsel zwischen einzelnen Menüs oder Masken.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	ermöglicht einen leichten Wechsel zwischen einzelnen Menüs oder Masken.

Abbildung 23: Fragebogen Teil 2 (aus [Prü1997 S.5])

	Die Software ...	--- -- - -/+ + ++ +++	Die Software ...
sk3	erzwingt unnötige Unterbrechungen der Arbeit.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	erzwingt keine unnötigen Unterbrechungen der Arbeit.
ft1	liefert schlecht verständliche Fehlermeldungen.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	liefert gut verständliche Fehlermeldungen.
ft2	erfordert bei Fehlern im Großen und Ganzen einen hohen Korrekturaufwand.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	erfordert bei Fehlern im Großen und Ganzen einen geringen Korrekturaufwand.
ft3	gibt keine konkreten Hinweise zur Fehlerbehebung.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	gibt konkrete Hinweise zur Fehlerbehebung.
lk1	lässt sich schwer erweitern, wenn für mich neue Aufgaben entstehen.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	lässt sich leicht erweitern, wenn für mich neue Aufgaben entstehen.
lk2	lässt sich schlecht an meine persönliche, individuelle Art der Arbeits erledigung anpassen.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	lässt sich gut an meine persönliche, individuelle Art der Arbeits erledigung anpassen.
lk3	lässt sich - im Rahmen ihres Leistungsumfangs - von mir schlecht für unterschiedliche Aufgaben passend einrichten.	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	lässt sich – im Rahmen ihres Leistungsumfangs - von mir gut für unterschiedliche Aufgaben passend einrichten.

Abbildung 24: Fragebogen Teil 3 (aus [Prü1997 S.6])

V. Fragebogenauswertung des Nutzertests

Auswertungsmatrix ISONORM 9241-110-S

Software:		Einschätzung der eigenen Erfahrung:									
Anzahl der Fragebögen:	8	6	6	7	5	5	7	6	6	6	6

Faktor	Frage		Wert aus dem Fragebogen (Wert liegt zwischen 1 (---) und 7 (++++))							
	Kürzel	Aspekt der Frage	1	2	3	4	5	6	7	8
Aufgabenangemessenheit	aa1	Vollständigkeit	5	6	5	5	6	3	3	6
	aa2	Aufwandsminimierung	7	7	4	6	7	7	2	7
	aa3	Passung	6	7	7	6	6	5	3	6
Selbstbeschreibungsfähigkeit	sb1	Informationsgehalt	6	6	6	6	7	3	6	5
	sb2	Unterstützungsmöglichkeit	1	6	5	6	4	3	2	3
	sb3	Unterstützungsangebot	4	4	5	5	7	5	5	3
Erwartungskonformität	ek1	Gestaltungskonsistenz	6	7	5	4	6	6	7	7
	ek2	Transparenz	5	6	7	5	4	5	3	5
	ek3	Bedienkonsistenz	7	6	7	7	7	6	7	7
Lernförderlichkeit	lf1	Erlernbarkeit	7	6	7	7	7	6	7	6
	lf2	Wissensverfügbarkeit	7	5	7	6	7	6	7	6
	lf3	Erschließbarkeit	6	7	5	5	4	6	7	4
Steuerbarkeit	sk1	Flexibilität	4	5	6	6	7	6	3	5
	sk2	Wechselmöglichkeit				1	4	4		6
	sk3	Unterbrechungsfreiheit	7	6	7	7	7	2	2	7
Fehlertoleranz	ft1	Verständlichkeit	4	6	6	4	5	4	5	4
	ft2	Korrigierbarkeit	6	4	7	6	7	2	5	4
	ft3	Korrekturunterstützung	3	4	6	4	2	2	3	4
Individualisierbarkeit	ik1	Erweiterbarkeit								4
	ik2	Personalisierbarkeit								6
	ik3	Aufgabenflexibilität			6					4
Punktsomme Fragebogen			91	98	108	96	104	81	77	109

Abbildung 25: Auswertungsmatrix Teil 1

Man beachte, dass der Teil ‚Individualisierbarkeit‘ sowie die Frage nach der Wechselmöglichkeit aufgrund der mangelnden Übertragbarkeit der Fragen nicht beantwortet wurden.

Auswertungsmatrix ISONORM 9241-110-S

Software:		ZimmermannMasterthesis	
Anzahl der Fragebögen:		8	

Faktor	Frage		Summe Einzelfragen	Summe Einzel- fragen / Anzahl Fragebögen	Summe Faktor	Mittelwert Faktor	ISONORM- Wert	Mittel- wert	Varianz	Standard- abweich- ung
	Kürzel	Aspekt der Frage								
Aufgabenangemessenheit	aa1	Vollständigkeit	39	4,9	16,5	5,5	95,5	4,875	1,55	1,25
	aa2	Aufwandsminimierung	47	5,9				5,875	3,55	1,89
	aa3	Passung	46	5,8				5,75	1,64	1,28
Selbstbeschreibungsfähigkeit	sb1	Informationsgehalt	45	5,6	14,1	4,7		5,625	1,41	1,19
	sb2	Unterstützungsmöglichkeit	30	3,8				3,75	3,36	1,83
	sb3	Unterstützungsangebot	38	4,8				4,75	1,36	1,16
Erwartungskonformität	ek1	Gestaltungskonsistenz	48	6,0	17,8	5,9		6	1,14	1,07
	ek2	Transparenz	40	5,0				5	1,43	1,20
	ek3	Bedienkonsistenz	54	6,8				6,75	0,21	0,46
Lernförderlichkeit	lf1	Erlernbarkeit	53	6,6	18,5	6,2		6,625	0,27	0,52
	lf2	Wissensverfügbarkeit	51	6,4				6,375	0,55	0,74
	lf3	Erschließbarkeit	44	5,5				5,5	1,43	1,20
Steuerbarkeit	sk1	Flexibilität	42	5,3	12,8	4,3		5,25	1,64	1,28
	sk2	Wechselmöglichkeit	15	1,9				3,75	4,25	2,06
	sk3	Unterbrechungsfreiheit	45	5,6				5,625	5,13	2,26
Fehlertoleranz	ft1	Verständlichkeit	38	4,8	13,4	4,5		4,75	0,79	0,89
	ft2	Korrigierbarkeit	41	5,1				5,125	2,98	1,73
	ft3	Korrekturunterstützung	28	3,5				3,5	1,71	1,31
Individualisierbarkeit	ik1	Erweiterbarkeit	4	0,5	2,5	0,8		4		
	ik2	Personalisierbarkeit	6	0,8				6		
	ik3	Aufgabenflexibilität	10	1,3				5	2,00	1,41

Abbildung 26: Auswertungsmatrix Teil 2

VI. Software

Auf der CD enthalten:

- Unity 5.1.2 Projekt mit verwendeten Assets
- Demoszene für Motorzusammenbau
- Nutzertest-Vorbereitungsszene