

Fakultät für Technik

Studiengang Wirtschaftsingenieurwesen

Global Process Management

**Virtual Reality: Produktanalyse und Konzipierung einer
Laborübung zum Thema Head-Mounted Displays für
eine Bildungseinrichtung**

Bachelor-Thesis

Erstprüfer Prof. Dr. Rer. Pol. Raphael Volz

Zweitprüfer Prof. Alfred Schätter

vorgelegt von Nicklas Trier

Matrikelnummer 307137

Abgabetermin 21.06.2016

Eidesstattliche Erklärung:

Ich versichere, die beiliegende Thesis selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben, sowie alle wörtlichen und sinngemäß übernommenen Stellen in der Arbeit gekennzeichnet zu haben.

Ort, Datum

Unterschrift

Danksagung

Für die erbrachte Hilfe in Rat und Tat bedanke ich mich bei meinem Erstprüfer Prof. Dr. Rer. Pol. Raphael Volz. Mein besonderer Dank gilt meinem Betreuer Sascha Friedrich für seine umfassende und stets hilfreiche fachliche Unterstützung. Weiterhin gebührt mein Dank allen befragten Personen, die mit ihrer persönlichen Meinung einen wertvollen Teil zur Arbeit beigetragen haben.

Inhaltsverzeichnis

1	Einführung	1
2	Grundlagen.....	3
2.1	Virtual Reality	5
2.2	Augmented Reality	8
2.3	Head-Mounted Displays	11
2.3.1	Direktsicht HMD	12
2.3.2	Video HMD	12
2.3.3	See-Through HMD	12
2.3.4	Interaktives HMD	13
2.4	Head-Up Displays.....	14
2.5	Zusammenfassung	15
3	Anwendungsgebiete von HMDs	16
3.1	PLM-Zyklus der Unternehmen.....	17
3.2	Privates Umfeld	19
3.3	Zusammenfassung	19
4	Oculus Rift Development Kit 2.....	20
4.1	Zusammenfassung	22
5	Das Google Cardboard Konzept	23
5.1	Zusammenfassung	23
6	Genutzte Google Cardboard Geräte	24
6.1	Dark Shader von ZAAK ZURICH BERLIN – Cardboard Kits Europe.....	24
6.2	Zeiss VR One.....	26
6.3	Zusammenfassung	28
7	HMD Development	29

7.1	Native HMD Development.....	30
7.1.1	Rift von Oculus.....	30
7.1.2	Google Cardboard Apps	31
7.2	HMD Web Development.....	32
7.2.1	WebGL	33
7.2.2	WebVR	33
7.2.3	Three.js	34
7.2.4	A-Frame von Mozilla und GLAM.....	35
7.3	Zusammenfassung	36
8	Produktanalyse zur Anschaffung eines HMDs für Lehrzwecke in einer Bildungseinrichtung	37
8.1	Festlegung und Definition der Kriterien für die Anschaffungsempfehlung 38	
8.2	Formulierung des Entscheidungsproblems.....	42
8.3	Vorstellung der Verfahren	43
8.3.1	Nutzwertanalyse	43
8.3.2	Analytical Hierarchy Process	43
8.4	Gewichtung der Entscheidungskriterien.....	45
8.4.1	Gewichtung der Entscheidungskriterien mit der Nutzwertanalyse ...	45
8.4.2	Ergebnisse aus der Paarvergleichsmethode mit der Nutzwertanalyse...	47
8.4.3	Gewichtung der Entscheidungskriterien mit dem AHP.....	53
8.4.4	Ergebnisse aus der Paarvergleichsmethode mit dem AHP	56
8.4.5	Vergleich: Gewichtung der Kriterien mit der Nutzwertanalyse gegen Gewichtung der Kriterien mit dem AHP	69
8.5	Bewertung der Entscheidungsalternativen	72
8.5.1	Gewichtete Bewertung der Entscheidungsalternativen mit der Nutzwertanalyse	76
8.5.2	Gewichtete Bewertung der Entscheidungsalternativen mit dem AHP..	78
8.6	Fazit der Anschaffungsempfehlung.....	80

8.7 Zusammenfassung	80
9 Konzipierung einer Laborübung zur HMD Einführung und Entwicklung	81
9.1 Motivation für die Laborübung	82
9.2 Vorstellung der genutzten Technologien.....	83
9.3 Programmierung der Webanwendung	86
9.3.1 Erste Station.....	86
9.3.2 Zweite Station.....	92
9.3.3 Dritte Station.....	94
9.3.4 Vierte Station	98
9.3.5 Fünfte Station.....	99
9.3.6 Andere Inhalte der Laborübung.....	100
9.4 Zusammenfassung	102
10 Evaluation der Laborübung	104
11 Ausblick	105
12 Fazit.....	107
13 Literaturverzeichnis	108
14 Anhang.....	113

Abkürzungen

AHP	<i>Analytical Hierarchy Process</i>
AMOLED	Active Matrix Organic Light Emitting Diode
API	<i>Application Programming Interface</i>
AR	<i>Augmented Reality</i>
CPU	<i>Central Processing Unit</i>
CSS	<i>Cascading Style Sheets</i>
DK	<i>Development Kit</i>
DOM	<i>Document Object Model</i>
FOV	<i>Field of View</i>
GPS	Global Positioning System
GPU	<i>Graphics Processing Unit</i>
HDMI	High Definition Multimedia Interface
HMD	<i>Head mounted Display, Head-Mounted Display</i>
HTML	<i>Hypertext Markup Language</i>
MR	<i>Mixed Reality</i>
NASA	National Aeronautics and Space Administration
PC	<i>Personal Computer</i>
PLM	<i>Product Lifecycle Management</i>
QR	<i>Quick Response</i>
URL	<i>Uniform Resource Locator</i>
USB	<i>Universal Serial Bus</i>
VR	<i>Virtual Reality</i>

Abstract

Virtual Reality and Augmented Reality are arguable the next big thing in computer technology. To enable as many people as possible to use this technologies Head-Mounted Displays of different types are currently released. This bachelor thesis explains what Virtual Reality, Augmented Reality and Head-Mounted Displays are and describes the different aspects of engineering that are used to run them. In this paper you can find in depth details on the Rift from Oculus and on Head-Mounted Displays that make use of the Google Cardboard concept. Furthermore, areas of application for Head-Mounted Displays are discussed, for private users and within the Product-Lifecycle-Management of businesses.

This bachelor thesis also takes a closer look on the development of content for Head-Mounted Displays. It discusses native Head-Mounted Display development and Head-Mounted Display web development. The focus is on Head-Mounted Display web development. The paper contains a comparison between the Rift from Oculus, the Zeiss VR One and the Dark Shader from ZAAK ZURICH BERLIN – Cardboard Kits Europe. An Analytical Hierarchy Process and a utility analysis are performed to find out which Head-Mounted Display suits best the needs of an education institution. Later on the introduced web technologies, for example WebVR API and A-Frame, are used to create a laboratory exercise about Head-Mounted Displays and Virtual Reality for an education institution.

1 Einführung

Bereits im Jahre 1982 verwendete der Autor Damien Broderick den Begriff *Virtual Reality* (im Folgenden VR) in seinem Science-Fiction Roman *The Judas Mandala*. In den folgenden Jahren diente das Konzept der virtuellen Realität als Träger für verschiedene Filme. So handelt beispielsweise der Film *Tron* aus dem Jahre 1982 von einer virtuellen Welt welche sich innerhalb eines Computers befindet. 1999 erschien der erste *Matrix* Film, welcher von einer virtuellen Welt mit dem Namen *Matrix* handelt, in jener die gesamte Menschheit lebt ohne sich dessen bewusst zu sein.

Technisch gesehen begann die Entwicklung von VR in den 60er Jahren mit dem Amerikaner Ivan Sutherland. Er beschrieb die Verbindung zwischen Rechnern, dem Design, der Konstruktion, der Navigation und dem Erleben von virtueller Welten. Zusätzlich erfand er einen Datenhelm, welcher es dem Nutzer erlaubt eine simulierte, simple 3D-Umgebung zu betrachten. Er nannte diesen Datenhelm „Head-Mounted Display“ [1] S. 19. Mitte der achtziger Jahre entwickelte die NASA eine multi-sensorische Workstation für die Simulation von Weltraumstationen. Jaron Lanier war 1987 der erste Wissenschaftler der den Begriff „Virtual Reality“ gebrauchte. Er nutzte diesen Begriff im Zusammenhang mit der Entwicklung einer Fortführung des Head-Mounted Displays von Sutherland. Das „EyePhone“ getaufte Head-Mounted Display bot eine Auflösung von 442 x 238 Pixeln, in einer späteren Version 720 x 238 Pixeln [1] S. 19–20. Allerdings wurde das EyePhone von der damaligen verfügbaren Technologie stark eingeschränkt. So konnte es nur fünf bis sechs Bilder pro Sekunde darstellen und das komplette System, bestehend aus Computern und Head-Mounted Display, kostete mehr als 250.000 \$ [2]. Folglich war das EyePhone für den Massenmarkt gänzlich ungeeignet.

Aktuell drängen verschiedene Hersteller mit unterschiedlichen Head-Mounted Displays auf den Massenmarkt. Die Head-Mounted Displays befinden sich dabei in einer Preisspanne von 10 € bis 1500 €, auch abhängig davon ob der Käufer bereits ein leistungsstarkes Smartphone bzw. einen leistungsstarken PC besitzt.

Ziel dieser Arbeit ist die Bewertung der unterschiedlichen Head-Mounted Displays in Hinsicht auf technische, wirtschaftliche und anderweitige Anforderungen und die Erstellung einer Laborübung welche das Head-Mounted Display verwendet. Die Bewertung zielt dabei auf eine Anschaffungsempfehlung ab. Inhalt der Laborübung sind die Funktionsweise, Einsatzgebiete und Bedienung eines Head-Mounted Displays. Zusätzlich umfasst die Laborübung die Programmierung einer Webanwendung welche das Head-Mounted Display als Ein- und Ausgabegerät verwendet.

Diese Bachelorthesis baut sich wie folgt auf: Zu Beginn dieser Arbeit werden die notwendigen Grundlagen erläutert, wie Definitionen und Erklärungen zu Virtual Reality oder Head-Mounted Display. Im Anschluss sollen die Anwendungsgebiete von Head-Mounted Displays betrachtet werden, sowohl im PLM-Zyklus der Unternehmen als auch im privaten Umfeld. Anschließend werden die genutzten Head-Mounted Displays und deren Funktionsweise vorgestellt. Darauf folgt die Entwicklung von Inhalten für Head-Mounted Displays. In diesem Zusammenhang wird der Fokus auf unterschiedliche Entwicklungskonzepte und Technologien gerichtet. Im weiteren Verlauf der Arbeit wird eine Produktanalyse zur Anschaffungsempfehlung der Head-Mounted Displays durchgeführt und eine Laborübung zur Verwendung der Head-Mounted Displays konzipiert. Den Abschluss macht eine Evaluation der Laborübung und ein daran angeschlossenes Fazit über die Arbeit.

2 Grundlagen

In diesem Kapitel werden alle nötigen Fachbegriffe eingeführt und erläutert. Weiterhin wird dargelegt wie Virtual Reality als auch Augmented Reality erzeugt wird und worin sich die beiden unterscheiden. Abschließend werden die Begriffe Head-Mounted Display und Head-Up Display definiert und der grundlegende Aufbau der Systeme dargestellt.

Um zuvor die verschiedenen Arten der Virtualität in Abhängigkeit des Realitätsanteils besser einordnen zu können bietet es sich an, die Mixed Reality (im Folgenden MR) – Taxonomie nach Paul Milgram et al. zu betrachten [3] S. 246. „Demnach handelt es sich bei MR um ein Kontinuum, welches sich zwischen der Realität und der Virtualität (virtuellen Realität) erstreckt, wobei der Anteil der Realität kontinuierlich abnimmt, während sich der der Virtualität entsprechend erhöht. Soweit der Anteil der Virtualität hier überwiegt, ohne dass die Umgebung dabei ausschließlich virtuell ist (virtuelle Realität), so spricht man von Augmentierter Virtualität (engl. *Augmented Virtuality*). Ist hingegen der Anteil der Realität größer, so handelt es sich um AR.“ [3] S. 246.

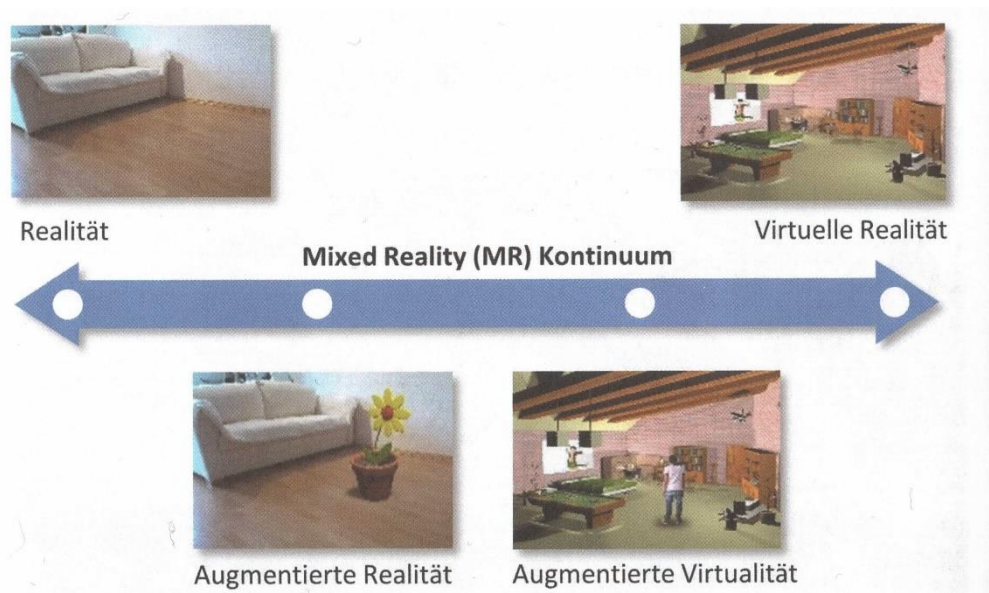


Abbildung 1: Mixed Reality Taxonomie. nach Milgram et al. (1995) [3] S. 246

Abbildung 1 zeigt die oben beschriebene Mixed Reality Taxonomie nach Milgram. Auf der linken Seite des Pfeils befindet sich die Realität, auf der rechten Seite die

virtuelle Realität. Dazwischen findet man die Mischformen aus beiden. Wobei einmal die Realität als Basis dient, welche mit virtuellen Inhalten erweitert wird und einmal die Virtuelle Realität als Basis dient, welche in diesem Fall mit realen Inhalten angereichert wird.

2.1 Virtual Reality

Was ist Virtual Reality? Wie bei vielen anderen Fachbegriffen gibt es auch für Virtual Reality keine einheitliche Definition, sondern vielmehr eine Auswahl an Definitionen die sich mehr oder weniger ähnlich sind. So definiert beispielsweise der Brockhaus virtuelle Realität als „eine mittels Computer simulierte Wirklichkeit oder künstliche Welt, in die Personen mithilfe technischer Geräte sowie umfangreicher Software versetzt und interaktiv eingebunden werden.“ [4] S. 5–6. Eine weitere Definition von Rademacher, beschreibt Virtual Reality als “[...] a simulation in which computer graphics is used to create a realistic looking world. [...] a high end user computer interface that involves real-time simulation and interactions through multiple sensorial channels. These sensorial modalities are visual, auditory, tactile, smell, and taste.” [5] S. 19.

Virtual Reality lässt sich wohl am besten Charakterisieren, wenn man die Abgrenzung zur „traditionellen“ Computergraphik betrachtet, als dem Wissenschaftsgebiet, aus welchem die Virtual Reality hervorgegangen ist [1] S. 13. Siehe Abbildung 2.

3D-Computergraphik	Virtuelle Realität
Rein visuelle Präsentation	Multimodale Präsentation: visuell, akustisch, haptisch
Präsentation nicht notwendigerweise zeitkritisch	Echtzeitdarstellung
Betrachterunabhängige Präsentation (exozentrische Perspektive)	Betrachterabhängige Präsentation (egozentrische Perspektive)
Statische Szene oder vorberechnete Animation	Echtzeitinteraktion und -simulation
2D-Interaktion (Maus, Tastatur)	3D-Interaktion (Körperbewegung, Hand-, Kopf- u. Körpergestik) + Spracheingabe
Nicht-immersive Präsentation	Immersive Präsentation

Abbildung 2: Merkmale von VR im Vergleich zu konventioneller Computergraphik [1] S. 14

Virtual Reality hat das Ziel den Nutzer zu überzeugen, dass er sich in einer anderen Welt aufhält als er sich tatsächlich physisch befindet. Erreicht wird dieses Ziel, indem das menschliche Gehirn, im speziellen der visuelle Cortex und Teile des Gehirns welche für die Wahrnehmung von Bewegung zuständig sind, getäuscht wird [6] S. 2. Diese Täuschung der Wahrnehmungssinne wird erreicht, indem bestimmte Technologien genutzt werden.

Eine entscheidende Rolle spielt dabei das sogenannte Stereosehen. Stereosehen hat das Ziel einen dreidimensionalen Eindruck zu bewirken, obwohl man nur eine zweidimensionale Displayfläche verwendet [7] S. 38. Dies bedeutet, dass mit Hilfe von Stereosehen und eines Head-Mounted Displays eine dreidimensionale Virtualität erzeugt werden kann. Technisch wird Stereosehen wie folgt realisiert: Ausgangspunkt ist ein Nutzer welcher die zweidimensionale Displayfläche betrachtet, siehe Abbildung 3.

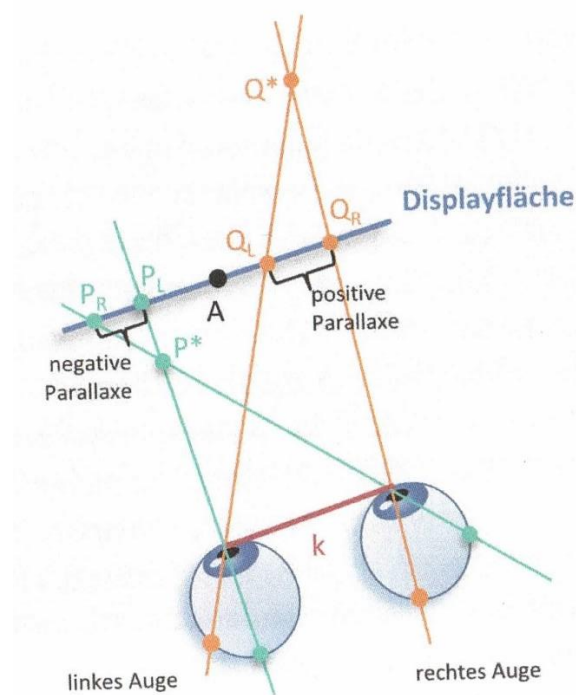


Abbildung 3: Stereosehen [7] S. 37

Der Nutzer betrachtet das Display indem er den Punkt A auf der Displayfläche fixiert. Auf der Displayfläche werden nun die Punkte P_R und P_L dargestellt. Dabei wird durch technische Vorkehrungen dafür gesorgt, dass Licht vom Punkt P_R nur in das rechte Auge des Nutzers gelangt und das Licht aus Punkt P_L nur in das linke Auge des Nutzers trifft. Der Abstand zwischen den Punkten wird als Parallaxe bezeichnet. Auf die Situation, dass Licht aus zwei verschiedenen Punkten jeweils in nur ein Auge trifft reagiert der visuelle Cortex des Nutzers indem es sich die Lichtreize aus beiden Punkten als der Lichtreiz aus einem Punkt P^* erklärt. Sprich der visuelle Cortex erkennt nicht die Lichtreize aus den beiden einzelnen Punkten, sondern rechnet die Lichtreize dem Punkt P^* zu. Dadurch, dass der visuelle Cortex die

beiden einzelnen Punkte fusioniert, ist es gelungen einen Punkt zu erzeugen, welcher sich, für die Wahrnehmung des Nutzers, vor der Displayfläche befindet. Vertauscht man die Reihenfolge der Punkte, so ist es möglich Punkte hinter der Displayfläche darzustellen. In Abbildung 3 ist das durch die Punkte Q_L , Q_R und Q^* dargestellt [7] S. 38–39. "In der VR kann man also unter Ausnutzung der Eigenheiten der menschlichen Wahrnehmung ein Stereodisplay realisieren, das nicht nur einen zweidimensionalen, sondern auch einen plastischen dreidimensionalen Bildindruck erzeugt, indem durch geeignete Wahl der Parallaxe Objekte vor oder hinter dem Bildschirm erscheinen." [7] S. 39. Somit können durch die Darstellung mehrerer Punkte beliebige Formen erzeugt werden, welche wiederum Grundlage einer dreidimensionalen Virtualität sind.

Wie zuvor erwähnt hat VR das Ziel dem Nutzer glaubhaft zu machen, dass er sich in einer anderen Welt befindet. Die Illusion der virtuellen Welt soll möglichst überzeugend sein. In diesem Zusammenhang spricht man von Präsenz. Präsenz beschreibt das subjektive Gefühl, dass man sich selbst in der virtuellen Umgebung befindet und dass diese Umgebung real für den Nutzer ist. Reize aus dem realen Umfeld des Nutzers werden dabei ausgeblendet [7] S. 46. Im Zusammenhang mit Präsenz spielt die Immersion eine entscheidende Rolle bei virtuellen Umgebungen. Immersion beschreibt den Grad des Eintauchens in eine virtuelle Umgebung. Folglich wird Präsenz dann erreicht, wenn der Nutzer einer möglichst immersiven virtuellen Umgebung ausgesetzt ist. Man könnte auch sagen, dass virtuelle Umgebungen möglichst immersiv gestaltet werden, damit der Nutzer die virtuelle Umgebung als präsent wahrnimmt. Präsenz wird erreicht, wenn der Nutzer sich bewusst in der VR befindet und sich wie in der realen Welt verhält. So reagiert der Nutzer bei einer in der VR auftretenden scheinbaren Gefahrensituation entsprechend körperlich, wie etwa mit erhöhten Puls oder erweiterten Pupillen [7] S. 46.

2.2 Augmented Reality

Dieses Kapitel führt den Begriff Augmented Reality ein und stellt die fünf wesentlichen Komponenten eines Augmented Reality Systems dar.

"Unter Augmentierter Realität (engl. Augmented Reality, deutsch teilweise auch als erweiterte oder angereicherte Realität bezeichnet) versteht man allgemein die Anreicherung der Realität durch künstliche virtuelle Inhalte. Dabei kommt es zu einer Verschmelzung der Realität mit der Virtualität.“ Abbildung 4 „[...] zeigt ein Beispiel eine reale Szene und deren Anreicherung um ein virtuelles Objekt.

Entscheidend hierbei ist, dass diese Erweiterung nicht statisch und einmalig wie in der unteren Abbildung, sondern kontinuierlich und angepasst an den aktuellen Standpunkt des jeweiligen Betrachters passiert.“ [3] S. 241–242.



Abbildung 4: Verschmelzung einer realen Umgebung (links) mit einem virtuellen Objekt (rechts) zur Augmentierten Realität (Mitte) [3] S. 242

In der Wissenschaft ist weitestgehend die Definition nach Azuma aus dem Jahre 1997 etabliert (Azuma 1997). „Danach versteht man unter AR eine Kombination (Überlagerung) von Realität und Virtualität, die interaktiv ist, bei welcher die Darstellung in Echtzeit erfolgt und in der 3D-Objekte (geometrisch) registriert sind.“ [3] S. 245. Vereinfacht lässt sich Augmentierte Virtualität in fünf Teilkomponenten aufteilen:

1. Videoaufnahme
2. Tracking
3. Registrierung
4. Darstellung
5. Ausgabe

Videoaufnahme

Mit Hilfe einer Kamera wird ein Videobild der Umgebung des Betrachters erstellt [3] S. 242.

Tracking

Tracking bedeutet die Berechnung der Position, der Lage und der Orientierung. Im Zusammenhang mit AR bedeutet es den Blickpunkt des Betrachters möglichst genau zu erfassen. Somit liefert das Tracking eine Transformation aus dem Nutzer- bzw. Kamerakoordinatensystem in das Koordinatensystem der virtuellen Umgebung [3] S. 242–243.

Registrierung

„Unter geometrischer Registrierung versteht man die Verankerung oder das korrekte Einpassen der künstlichen virtuellen Inhalte in die Realität. Dies bedeutet, dass auf Basis der Positions- und Lageschätzung des Trackings, das Koordinatensystem der einzelnen virtuellen Inhalte und der beobachteten Realität so in Beziehung gesetzt werden, dass virtuelle Inhalte in der Realität fest verortet (registriert) erscheinen. Dies führt dazu, dass ein sich in der virtuellen Welt nicht bewegendes künstliches Objekt, auch in der Realität einen scheinbar festen Platz hat, unabhängig vom sich verändernden Standpunkt des Betrachters (beziehungsweise der Kamera).“ [3] S. 243.

Darstellung

„Basierend auf der sich durch die geometrische Registrierung ergebenden Transformation und der jeweiligen Kameraperspektive werden die virtuellen Inhalte wiedergegeben (Rendering). Dabei wird das aufgenommene Videobild durch die virtuellen Inhalte perspektivisch korrekt überlagert, wodurch die eigentliche Augmentierung erfolgt. Für eine möglichst nahtlose Überlagerung müssen hierbei ggf. zusätzlich die Auflösung und Schärfe des virtuellen Bildes angepasst werden.“ [3] S. 243.

Ausgabe

Abschließend wird das augmentierte Videobild über ein Display ausgegeben, an welchem die Kamera angeschlossen ist. [3] S. 243.

2.3 Head-Mounted Displays

Dieser Abschnitt definiert den Begriff Head-Mounted Display und stellt den grundlegenden Aufbau dar. Anschließend werden unterschiedliche Arten von HMDs näher betrachtet.

„Head-Mounted Displays (HMDs) könnte man ins Deutsche als „am Kopf angebrachte Anzeigen“ übersetzen. Das bedeutet, dass man mobile Visualisierungs- und Interaktionssysteme in Form eines Helms oder einer Datenbrille am Kopf trägt.“
Abbildung 5 „[...] zeigt die prinzipiellen Bestandteile eines HMDs. Ein miniaturisiertes Display erzeugt ein Bild, welches dem Nutzer vergrößert über eine Optik dargestellt wird. [...] Das Display sowie die im HMD integrierten Sensoren werden über eine HMD-Elektronik angesteuert. Eine Schnittstelle verbindet das HMD mit einem Rechner, um Videodaten übertragen zu können. Eine Haltevorrichtung in Form einer Brille oder eines Helms trägt alle im HMD integrierten Funktionselemente.“ [8] S. 142.

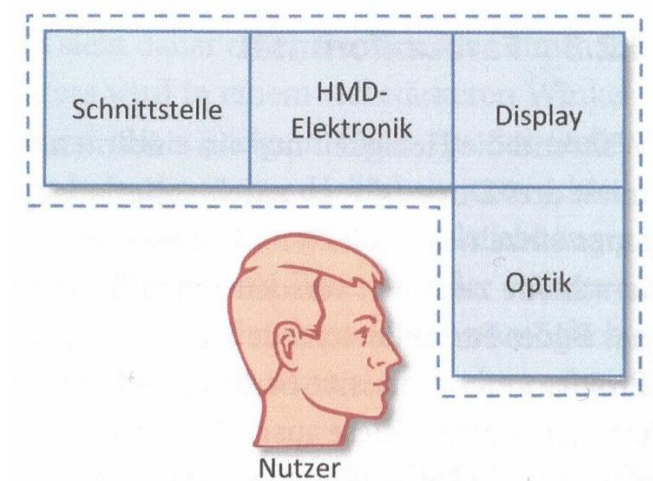


Abbildung 5: Prinzipielle Bestandteile eines HMDs [8] S. 142

Man unterscheidet zwischen vier Arten von HMDs:

1. Direktsicht HMD
2. Video HMD
3. See-Through HMD
4. Interaktives HMD

2.3.1 Direktsicht HMD

Direktsicht bedeutet, dass der Nutzer über eine Linse auf das integrierte Display des HMD blickt. Direktsicht HMDs grenzen den Nutzer auf Grund ihrer Bauweise komplett von der Umgebung ab. Der Nutzer kann die reale Umgebung nicht mehr betrachten. Ein Direktsicht HMD kann nur virtuelle Bilder ausgeben. Folglich kann mit diesem HMD nur virtuelle Realität betrachtet werden und keine augmentierte Realität [8] S. 147–148.

2.3.2 Video HMD

Beim Video HMD handelt es sich um eine Erweiterung des Direktsicht HMD. Die Erweiterung besteht aus einer Frontkamera, welche an das HMD montiert wird. Die Kamera dient zur Aufnahme von Bildern der realen Umgebung des Nutzers, da der Nutzer ja auf Grund der Bauweise des Direktsicht HMD die reale Umgebung nicht sehen kann. Die Frontkamera ist mit der HMD-Elektronik verbunden und überträgt die aufgenommen Bilder an das angeschlossene Computersystem. Mit Hilfe von Software werden reale Objekte mit virtuellen Objekten verschmolzen und ein virtuelles Bild an den Nutzer ausgegeben [8] S. 149–150.

2.3.3 See-Through HMD

Der offensichtliche Unterschied zwischen See-Through HMD und Direktsicht HMD als auch Video HMD ist, dass der Nutzer nicht bauartbedingt von seiner realen Umgebung abgeschirmt ist. Wie bei den beiden obigen HMDs wird auch beim See-Through HMD von einem Display in virtuelles Bild erzeugt und von einer Linse vergrößert. Somit kann der Nutzer gleichzeitig Inhalte seiner realen Umgebung und virtuelle Inhalte wahrnehmen. Der Unterschied zum Video HMD besteht

darin, dass das Bild der realen Umgebung des Nutzers nicht von einer Kamera aufgenommen wird, sondern direkt vom Nutzer gesehen wird. Das HMD überlagert dann das vom Nutzer Gesehene mit zusätzlichen virtuellen Informationen. Dafür muss ein Bildverarbeitungssystem in Echtzeit reale Objekte im Blickfeld des Nutzers erfassen, erkennen, rendern, und die entsprechenden virtuellen Informationen darstellen [8] S. 150–151.“

2.3.4 Interaktives HMD

Interaktiv bedeutet, dass der Nutzer mit den vom HMD angezeigten Informationen interagieren kann. Das HMD besteht aus einer der obigen Architekturen für die Visualisierung und einem Eingabegerät. Mit Hilfe des Eingabegerätes kann der Nutzer mit der virtuell angezeigten Information interagieren [8] S. 151. Mögliche Eingabegeräte sind beispielsweise Motion-Tracker, Finger-Tracker, GPS-Empfänger oder Eye-Tracking Systeme [8] S. 153 oder auch eine Computertastatur [8] S. 151

Diese Arbeit beschäftigt sich im späteren Verlauf mit interaktiven HMDs, da diese das am meisten immersive Erlebnis für den Nutzer bieten.

2.4 Head-Up Displays

Dieser Abschnitt beinhaltet eine kurze Einführung zum Thema Head-Up Displays. Es wird die Funktionsweise, einzelnen Bestandteile und die Vorteile erläutert.

Die grundlegende Idee hinter einem Head-Up Display ist die Überlagerung der realen Umgebungen mit Informationen, welche mit Hilfe eines virtuellen Bildes bereitgestellt werden. Die wesentlichen Bauteile aus denen ein Head-Up Display besteht sind die Bilderzeugungseinheit, das Spiegelsystem und der Combiner [9] S. 3. Folglich handelt es sich bei einem Head-Up Display um ein vereinfachtes, stationäres Augmented Reality System. Mit Hilfe des Spiegelsystems wird das von der Bilderzeugungseinheit erzeugte virtuelle Bild auf den Combiner abgebildet. Heutzutage handelt es sich beim Combiner in der Regel um die Frontscheibe eines Fahrzeuges auf welcher kleinflächig symbolische Informationen zur gefahrenen Geschwindigkeit oder Navigationshinweise angezeigt werden. Abbildung 6 zeigt die Funktionsweise eines Head-Up Displays im Fahrzeug.

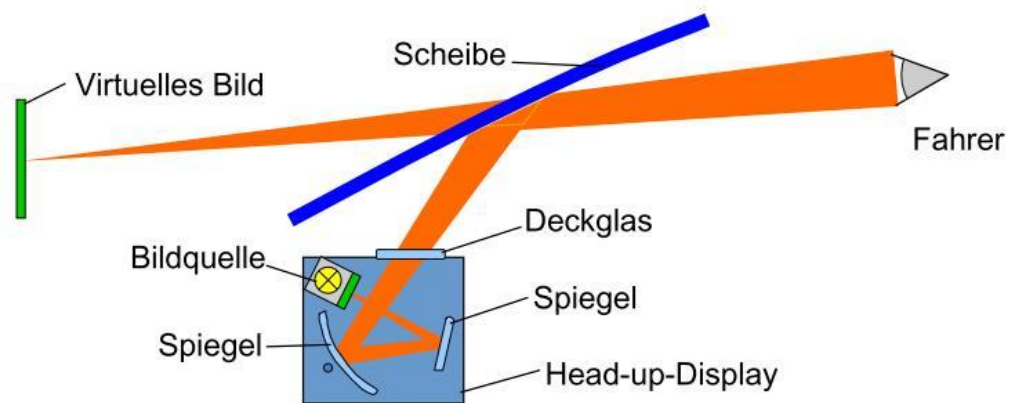


Abbildung 6: Funktionsweise eines Head-Up Display [9] S. 7

Die Vorteile eines HUD im Fahrzeug sind ein kleiner Blinkwinkel, da der Fahrer seine Blinkwinkel nicht mehr vergrößern muss, um die tiefliegenden Armaturen abzulesen. Ein HUD bietet außerdem den Vorteil, einer geringeren Fokussierungszeit, da die Informationen in der Frontscheibe in einer größeren Entfernung erscheinen als die herkömmlicher Innenrauminstrumenten [10] S. 26–27.

2.5 Zusammenfassung

In diesem Kapitel wurde erklärt was unter Virtual Reality, Augmented Reality, Head-Mounted Display und Head-Up Display zu verstehen ist. Des Weiteren wurden die wesentlichen Technologien vorgestellt, die benötigt werden, um Virtual Reality und Augmented Reality zu erzeugen. Abschließend wurde die Bestandteile und die Funktionsweise von Head-Mounted Displays als auch Head-up Displays erklärt.

3 Anwendungsgebiete von HMDs

In diesem Kapitel werden Einsatzfelder für HMDs betrachtet. Unterschieden wird dabei zwischen privaten und betrieblichen Anwendungen.

3.1 PLM-Zyklus der Unternehmen

Im Unternehmen können die Einsatzgebiete von HMDs, mittels des PLM-Zyklus veranschaulicht werden. Darunter versteht man:

„Im gesamten Produktlebenszyklus (engl.: product lifecycle), das heißt von der Idee über Entwicklung und Konstruktion, Produktion sowie Vertrieb und Service bis zur Außerbetriebnahme eines Produktes, entstehen große Mengen an verschiedensten Daten, Dokumenten und Informationen“ [11] S. 9. „Das Product Lifecycle Management ist ein integrierendes Konzept zur IT-gestützten Organisation und Verwaltung aller Informationen über Produkte und deren Entstehungsprozesse über den gesamten Produktlebenszyklus hinweg, sodass die richtige Information zum richtigen Zeitpunkt in der richtigen Form an der richtigen Stelle zur Verfügung steht“ [11] S. 10.

In der Entwicklung können beispielsweise Entwürfe realitätsnäher visualisiert und Prototypen kostengünstiger erstellt werden, als dies mit traditionellen Verfahren der Fall ist. Weiterhin können Fertigungsstraßen vor Produktionsbeginn in einer virtuellen Welt dargestellt werden. Planungsfehler können in virtuellen Anlagen oder in einer virtuellen Fabrik wesentlich einfacher und kostengünstiger behoben werden als dies in der realen Produktion möglich ist [1] S. 9.

HMDs bieten weiterhin Vorteile beim Training von Personal. Mit Hilfe von Simulationen können Mitarbeiter auf ihr Arbeitsumfeld vorbereitet werden. Dieses Training kann zeitlich so stattfinden, das zum Zeitpunkt der Inbetriebnahme bereits gut ausgebildetes Personal verfügbar ist. Zusätzlich können mit Hilfe von HMDs Extremsituationen, wie etwa ein Flugzeugabsturz, trainiert werden ohne Mitarbeiter tatsächlich zu gefährden. Oftmals bieten solche Simulationen auch wirtschaftliche Vorteile, da Lehrmittel wie etwa Schiffe, Flugzeuge oder LKWs nicht real vorhanden sein müssen, sondern simuliert werden können [1] S. 9–10.

Im Vertrieb bietet es sich an HMDs beispielsweise auf Messen einzusetzen, um Kunden Produkte besser vorstellen zu können [1] S. 10. Audi bietet potenzielle Käufer eine Virtual Reality Präsentationen seiner Autos. Dadurch können die zukünftigen Kunden ihre Wunschkonfiguration schon vor dem Kauf anschauen und

testen. Außerdem muss der Verkäufer nicht zwangsläufig ein Auto in der Wunschkonfiguration des Kunden vor Ort haben. Mit Hilfe des HMDs kann der Kunde verschiedene Konfigurationen besser vergleichen, als mit einem Verkaufsprospekt [12].

Die hier vorgestellten Anwendungsgebiete sind nur ein kleiner Ausschnitt aller möglichen Anwendungsgebiete von HMDs.

3.2 Privates Umfeld

Aktuell befinden sich auf dem Markt für private Nutzer folgende Head-Mounted Displays:

1. Oculus Rift
2. HTC Vive
3. PlayStation VR
4. HMDs die das Google Cardboard Prinzip nutzen

Anwendungsgebiete für diese Head-Mounted Displays sind hauptsächlich das Spielen von Videospielen in Verbindung mit PC, Konsole und Smartphone. Neben dem Spielen von Videospielen dienen die HMDs auch zum Konsumieren von speziellen Videos, Bildern und anderen Inhalten, die über das Internet bereitgestellt werden. Diese Inhalte müssen allerdings entsprechend aufbereitet werden, um einen Mehrwert bei der Konsumierung über HMDs zu erzeugen.

In den nächsten Kapiteln werden einige der oben genannten HMDs genauer vorgestellt und analysiert.

3.3 Zusammenfassung

Dieses Kapitel startet mit einer Definition von Product Lifecycle Management Zyklus und nennt einige Beispiele für die Anwendung von HMDs in diesem. Weiterhin wird auf Anwendungsmöglichkeiten von HMDs für den privaten Nutzer eingegangen, sowie aktuell verfügbare HMDs für private Anwender aufgelistet.

4 Oculus Rift Development Kit 2

Zunächst wird das Modelle der Firma *Oculus* näher betrachtet. Das Head-Mounted Display *Rift Development Kit 2* besteht aus zwei Teilen. Erstens dem Head-Mounted Displays selbst, welches vom Nutzer als eine Brille getragen wird, zu sehen in Abbildung 7.



Abbildung 7: Oculus Rift DK 2 [13] S. 1

Zweitens aus der Infrarot Kamera welche für das Positionstracking zuständig ist, siehe Abbildung 8.



Abbildung 8: Infrarotkamera des Rift DK 2 [14]

Herzstück der Datenbrille des Head-Mounted Displays ist das Display eines *Galaxy Note 3* Smartphones von *Samsung*. Bei dem Display handelt es sich um ein so genanntes Super AMOLED Display [14] mit einer Auflösung von 1920 mal 1080 Pixeln [6] S. 13. Das Display verfügt über eine Bildwiederholungsrate von 75 Hz [14]. Das HMD bietet ein Sichtfeld von 100 Grad und wiegt 453 g. Vor dem Display befinden sich zwei Linsen. Um den Tragekomfort möglichst hoch zu halten, ist die Gesichtsmaske der Rift, welche als Auflagefläche für das Gesicht des Nutzers dient, mit Schaumstoff gepolstert. Um einen sicheren Halt der Brille zu ermöglichen, wird diese mit Hilfe eines an drei Stellen fixierten und einstellbaren Gummibandes am Kopf des Nutzers gehalten.

Neben dem Display befinden sich innerhalb des Rift ein Beschleunigungssensor, ein Gyroskop und ein Magnetometer welche mit einer Wiederholungsrate von 1000 Hz Messungen vornehmen [14]. Diese drei Sensoren haben innerhalb des HMDs eine vergleichbare Funktion wie in einem Smartphone. Der Beschleunigungssensor berechnet für jede der drei Achsen des HMD die momentane Beschleunigung in m/s^2 . Das Koordinatensystem ist relativ zum Gerät [15], was bedeutet, dass der Nullpunkt des Koordinatensystems im Mittelpunkt des HMD liegt. Das Gyroskop misst Änderungen der Lageorientierung und Änderungen der Winkelgeschwindigkeit [16]. Das Magnetometer „misst das magnetische Feld in Micro-Tesla und somit die Ausrichtung des Geräts in Bezug auf das magnetische Feld der Erde. Es hat eine Komponente, die zum magnetischen Nordpol zeigt, was als Kompass verwendet werden kann.“ [15]. Der Verbund aus drei Sensoren ist notwendig, da das Magnetometer eine schlechte Genauigkeit für schnelle Bewegungen hat, aber sehr präzise misst je länger die Messdauer ist. Das Gyroskop und der Beschleunigungssensor wiederum reagieren sehr schnell und genau auf Änderungen [16]. Somit kann nur durch Kombination der Daten aus allen drei Sensoren eine vollständige Lagebestimmung des HMD realisiert werden.

Unterhalb des halb durchlässigen Kunststoffgehäuses befinden sich 40 Infrarot LEDs, welche über die gesamte Oberfläche der Rift verteilt sind [14]. Sie dienen als Marker für das Positionstracking mit der Infrarot Kamera. Positionstracking bedeutet in diesem Zusammenhang, dass die Position des Kopfes im dreidimensionalen Raum relativ zu der Kamera abgefragt wird. Somit kann der Nutzer durch Kopfbewegungen innerhalb der virtuellen Realität interagieren. Lehnt sich der Nutzer

beispielweise nach vorne um einen Gegenstand innerhalb der virtuellen Realität genauer zu betrachten, wird diese Bewegung vom System erkannt und der dargestellte Gegenstand größer dargestellt [14].

Die Kamera verfügt über ein Weitwinkelobjektiv [14]. Über der Linse befindet sich ein Filter welcher sichtbares Licht filtert und nur infrarotes Licht durchlässt [14]. Dadurch wird von der Kamera nur das Licht der infraroten LEDs des HMD aufgenommen. Diese Art von Tracking wird als Outside-In-Verfahren bezeichnet, da die Kamera von außerhalb des Interaktionsbereichs die Szene aufnimmt [17] S. 107.

Um das System aus Datenbrille und Kamera betreiben zu können ist ein PC notwendig. Das HMD wird über ein HDMI und ein USB Kabel mit dem PC verbunden. Die Kamera wird über ein sogenanntes Sync Kabel und ein weiteres USB Kabel mit dem PC verbunden [6] S. 13.

4.1 Zusammenfassung

Dieses Kapitel stellt das HMD Rift der Firma Oculus vor. Es werden die technischen Komponenten und deren Funktionsweise erläutert. Das System besteht aus der Datenbrille selbst und einer Tracking Kamera und wird mit Hilfe eines PCs betrieben.

5 Das Google Cardboard Konzept

Beim Google Cardboard Konzept handelt es sich im Kern um eine Erweiterung für das Smartphone. Dabei erweitert das Cardboard Konzept das Smartphone so, dass ein Head-Mounted Display entsteht. Hardwareseitig wird das Smartphone erweitert, indem es das Smartphone in eine bestimmte Halterung eingebettet, welche über Linsen und eine Art Gesichtsmaske für den Träger verfügt. Softwareseitig sind spezielle Apps nötig um das Stereosehen und das Tracking der Kopfbewegungen zu gewährleisten. Der größte Unterschied zur Rift von Oculus besteht darin, dass Cardboard Halterungen komplett ohne eigenständige Elektronik auskommen und stattdessen die Ressourcen verwenden die vom Smartphone bereitgestellt werden. So verfügen die Halterungen über keine Sensoren, sondern es werden ausschließlich die Sensoren des Smartphones genutzt.

Im Mai 2014 stellt Google auf der I/O Keynote das Cardboard Konzept erstmals vor und vergab an alle Teilnehmer ein Google Cardboard Halterung. Google stellt diese Halterung nicht selbst her, sondern stellt alle notwendigen Daten zu Verfügung, so dass jeder sein eigenes Cardboard bauen und auch verkaufen kann [6] S. 20. Google stellt dabei eine App zur Verfügung, welche das Smartphone des Nutzers mit Hilfe eines QR-Codes an die Parameter, wie etwa Abstand der Linsen, der Cardboard Halterung anpasst.

Um mit Hilfe des Google Cardboard Konzepts virtuelle Realität zu konsumieren ist nur ein Smartphone mit einem entsprechen großem Bildschirm notwendig. Dieses wird im Querformat in der Halterung fixiert.

5.1 Zusammenfassung

Dieses Kapitel behandelt das Google Cardboard Konzept. Es wird das Funktionsprinzip dieses Konzeptes vorgestellt. Anschließend werden die Unterschiede zwischen dem Google Cardboard Konzept und der *Rift* von *Oculus* erläutert.

6 Genutzte Google Cardboard Geräte

Wie oben beschrieben handelt es sich beim Google Cardboard Konzept um ein Open Source Projekt, somit gibt es zahlreiche Hersteller die eine Halterung nach dem Google Cardboard Konzept zur Verfügung stellen. In diesem Kapitel werden die Halterungen die für diese Arbeit verwendet wurden beschrieben.

6.1 Dark Shader von ZAAK ZURICH BERLIN – Cardboard Kits Europe

Die Cardboard Halterung von *ZAAK ZURICH BERLIN – Cardboard Kits Europe* namens *Dark Shader* besteht aus Pappkarton. Der Karton hat eine Stärke von 1,5 mm und ist mit Cellophan beschichtet um Schaden durch Feuchtigkeit zu vermeiden. Neben der Pappkarton Konstruktion besitzt sie zwei Linsen und an der linken Seite einen Magnetschalter, siehe Abbildung 9.



Abbildung 9: ZAAK Dark Shader - Cardboard Kits Europe [18] S. 1

Die Lasche an der Vorderseite der Halterung kann Smartphones mit einer Displaygröße von bis zu 5,5 Zoll aufnehmen. Die Lasche wird mit Hilfe der beiden Klettverschlüsse an der Oberseite der Halterung fixiert und somit das Smartphone sicher in der Halterung gehalten. Die Aussparung an der Vorderseite der Lasche ist notwendig, damit die Kamera des Smartphones nicht vom Pappkarton abgedeckt wird. Dadurch kann mit dem *Dark Shader* auch augmentierte Realität konsumiert werden. Im Inneren des *Dark Shader* befinden sich zwei Linsen. Die Linsen haben einen Durchmesser von 25 mm und bestehen aus BK7 Borosilikatglas. Dieses Glas zeichnet sich durch hohe Reinheit als auch hohe Widerstandfähigkeit aus [18] S. 1–2.

Der Magnetschalter dient als Bedienelement. So ist es für den Nutzer möglich bestimmte Ereignisse innerhalb der virtuellen oder augmentierten Realität mit dem Schalter auszulösen. Der Schalter wird bei der Bedienung von seiner Ruhelage aus nach unten gezogen, diese Bewegung nimmt das Magnetometer des Smartphones wahr und löst ein Ereignis innerhalb der Software aus [19]. Die Halterung kostet zehn Euro und wiegt 68 Gramm.

6.2 Zeiss VR One

Die *VR One* von *Zeiss* ist ebenfalls eine Halterung für Smartphones, welche nach dem Google Cardboard Konzept funktioniert. Im Gegensatz zur Halterung von *ZAAK ZÜRICH BERLIN – Cardboard Kits Europe* besteht sie allerdings aus Kunststoff und verfügt über ein Band, welches die Halterung am Gesicht des Nutzers fixiert, ähnlich wie bei der *Rift* von *Oculus*, siehe Abbildung 10.



Abbildung 10: VR One von Zeiss [20]

Die folgenden technischen Spezifikationen wurden der Produktseite der VR One entnommen [20]. Die Halterung unterstützt Smartphones mit einer Bildschirmdiagonale von 4,7 Zoll bis 5,2 Zoll. Für jedes Smartphone ist eine extra Schale notwendig, welche als Aufnahme für das Smartphone dient, siehe Abbildung 11. Anschließend wird die Schale mit dem Smartphone in die Halterung geschoben.

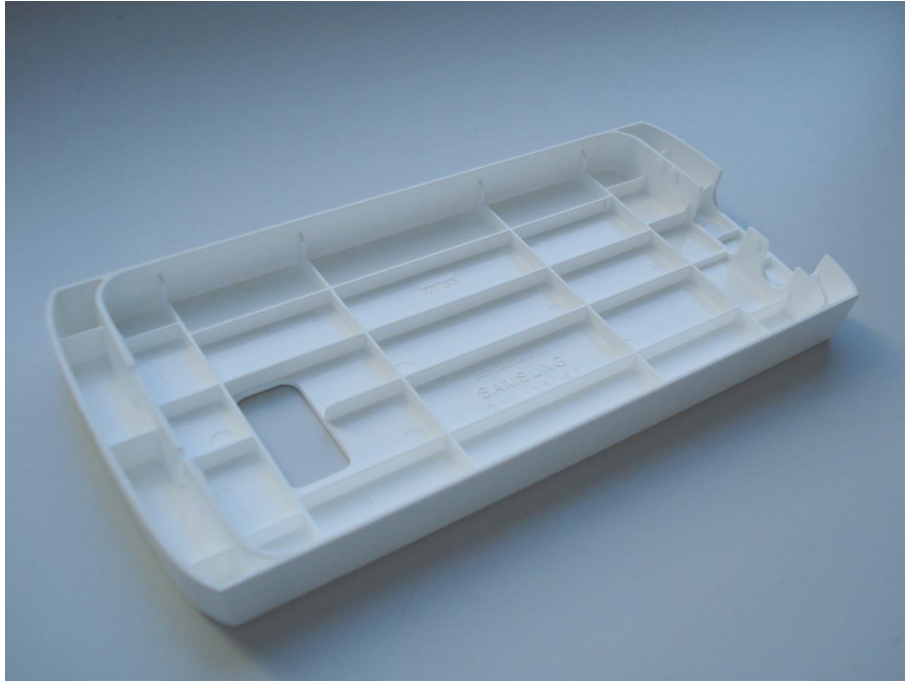


Abbildung 11: Schale zur Aufnahme des Smartphones [21]

Wie Abbildung 11 zu entnehmen ist, verfügt die Schale über eine Aussparung an ihrer Rückseite, diese Aussparung befindet sich an derselben Position wie die Kamera des jeweiligen Smartphones. Zusätzlich besteht die Vorderseite der *VR One* aus durchsichtigem Kunststoff, wodurch das konsumieren von augmentierter Realität ermöglicht wird. Die Schalen können gekauft oder aber selbständig, mit Hilfe eines 3D-Druckers, gedruckt werden. Die nötigen Daten für das selbständige Drucken werden dabei von Zeiss zur Verfügung gestellt. Allerdings hält sich aktuell die Anzahl der unterstützten Smartphones noch in Grenzen.

Die Halterung von Zeiss bietet ein Sichtfeld von 100 Grad und unterstützt einen Augenabstand des Nutzers von 53 mm bis 77 mm. Verbaut werden zwei Zeiss Präzision Linsen mit einem Durchmesser von 40 mm. Die Gesichtsmaske ist mit Schaumstoff gepolstert und das Gehäuse verfügt über Luftschlitze um ein Beschlagen der Linsen zu vermeiden. Das verwendete Band kann mit Hilfe von Klettverschlüssen verstellt werden.

Im Gegensatz zur Halterung *Dark Shader* verfügt die *VR One* über keinen Magnetschalter. Die *VR One* von Zeiss kostet zum Zeitpunkt der Erstellung dieser Arbeit 100 €.

6.3 Zusammenfassung

In diesem Kapitel wurden die *Google Cardboard* Halterungen vorgestellt und auf ihre technischen Merkmale eingegangen. Es wurden die Halterungen VR One von Zeiss und Dark Shader von ZAAK ZURICH BERLIN – Cardboard Kits Europe verwendet.

7 HMD Development

Dieses Kapitel behandelt die Entwicklung von Inhalten für HMDs. Genauer gesagt die Entwicklung von Virtual Reality Inhalten. Dabei wird unterschieden zwischen so genannten Native HMD Development und HMD Web Development.

Native HMD Development heißt, dass die Inhalte für eine spezielle Plattform entwickelt werden, mit Hilfe der verfügbaren Software Development Kits von dieser Plattform. Diese Inhalte sind dann auch nur auf dieser Plattform nutzbar [22] S. 18. So sind etwa Anwendungen die für die Rift von Oculus entwickelt wurden nicht ohne weitere Anpassungen auf Smartphones für Google Cardboard nutzbar.

HMD Web Development bedeutet, dass Inhalte nicht plattformspezifisch entwickelt werden, sondern darauf ausgelegt sind, in Webbrowsern zu laufen. Dabei werden im Wesentlichen Technologien wie HTML, CSS und JavaScript genutzt. Somit können Anwendungen die webbasierend sind auf jedem Gerät verwendet werden, das einen Browser besitzt, der diese Technologien unterstützt. Die gleiche Anwendung kann so, ohne Anpassungen, sowohl auf Smartphones als auch PCs genutzt werden [22] S. 19.

7.1 Native HMD Development

Dieses Kapitel behandelt das Native HMD Development für die Rift von Oculus und für Google Cardboard Apps.

7.1.1 Rift von Oculus

Um für die Rift von Oculus Inhalte zu entwickeln gibt es die Möglichkeit direkt auf eine 3D API zu programmieren, wie beispielsweise OpenGL oder DirectX. Allerdings ist dies sehr aufwändig, da diese Art von Programmierung ein großes Wissen über 3D Programmierung voraussetzt [6] S. 27.

Deswegen bietet es sich an, einen Spiele-Engine für die Programmierung zu verwenden. Dadurch wird die Programmierung wesentlich einfacher, da der Engine viele Standardfunktionen bereits enthält und diese nicht aufwändig programmiert werden müssen, was eine großer Zeitvorteil ist [6] S. 47.

Mit folgenden Spiele-Engines können Inhalte für die Rift von Oculus erstellt werden:

- Unity
- Unreal Engine
- Crytek CryENGINE
- Autodesk Stingray
- MaxPlay
- Amazon Lumberyard

Dabei ist es für die Entwicklung notwendig, die Oculus Runtime, das Software Development Kit und die Betriebssoftware von Oculus für die jeweilige Spiele-Engine zu installieren [6] S. 38. Diese Programme bzw. Daten können von der Webseite von Oculus heruntergeladen werden.

7.1.2 Google Cardboard Apps

Inhalte für Google Cardboard Apps zu programmieren ist vergleichbar damit, Inhalte für die Rift von Oculus zu erstellen. Hier kann ebenfalls mit der Kombination aus der Spiele-Engine Unity und dem Cardboard Software Development Kit eine virtuelle Welt erzeugt werden.

Außerdem kann, wie bei der Rift eine Anwendung ausschließlich mit dem Software Development Kit von Google Cardboard programmiert werden. Hierbei werden die Technologien Java und OpenGL verwendet. Allerdings bietet sich wiederum, auf Grund der im Spiele-Engine enthaltenen Standardfunktionen, eine Verwendung eben dieses an [6] S. 86.

7.2 HMD Web Development

Wie bereits erwähnt, ist es das Ziel von HMD Web Development Anwendungen zu entwickeln die browserbasierend sind. Browserbasierende Anwendungen haben folgende Vorteile [6] S. 63–64:

- Sofortiger Zugang: Es ist kein Herunterladen oder Installieren der Anwendung nötig, stattdessen muss nur eine URL eingetippt oder ein Link genutzt werden, um die Anwendung zu starten.
- Einfache Integration von anderen Webinhalten: Aufgrund der Tatsache, dass nahezu jeder Web Service APIs zur Verfügung stellt, können Daten aus Quellen wie YouTube, sozialen Netzwerken oder Wikipedia einfach in die Anwendung integriert werden.
- Plattformunabhängigkeit: HTML kann auf jedem PC, Smartphone oder Tablet ausgeführt werden.
- Schnelles und kostengünstiges Entwickeln von Inhalten: HTML und JavaScript sind die einfachsten, plattformunabhängigen Entwicklungstechnologien um Anwendungen zu erstellen. Außerdem sind die meisten Entwicklungswerkzeuge für HTML und JavaScript Anwendungen kostenfrei und Open Source.
- Einfache Bereitstellung: Anwendungen können serverseitig einfach aktualisiert bzw. überarbeitet werden, ohne dass der Nutzer eine neue Version der Anwendungen herunterladen bzw. installieren muss.

Damit Virtual Reality Inhalte im Browser dargestellt werden können, muss der Browser in der Lage sein, 3D Inhalte zu rendern und mit den angeschlossenen HMDs zu kommunizieren. Dies wird mit Hilfe von WebGL und WebVR erreicht, was im Folgenden näher betrachtet wird.

7.2.1 WebGL

WebGL steht für Web Graphics Library und wird von der Khronos Group bereitgestellt. Die Khronos Group definiert WebGL wie folgt:

„WebGL is a cross-platform, royalty-free web standard for a low-level 3D graphics API based on OpenGL ES 2.0, exposed through the HTML5 Canvas element as Document Object Model interfaces. Developers familiar with OpenGL ES 2.0 will recognize WebGL as a Shader-based API using GLSL, with constructs that are semantically similar to those of the underlying OpenGL ES 2.0 API. It stays very close to the OpenGL ES 2.0 specification, with some concessions made for what developers expect out of memory-managed languages such as JavaScript. WebGL brings plugin-free 3D to the web, implemented right into the browser. Major browser vendors Apple (Safari), Google (Chrome), Mozilla (Firefox), and Opera (Opera) are members of the WebGL Working Group.” [23]

Dies bedeutet, dass WebGL eine plattformunabhängige, lizenzfreie 3D Grafik API auf Basis von OpenGL ist. Mit Hilfe von WebGL werden 3D Inhalte ohne weitere Plug-Ins direkt in den Browser integriert. WebGL ist in allen gängigen Browsern, wie Mozilla von Firefox oder Chrome von Google, vorhanden.

WebGL wiederum dient als Basis für WebVR, was im nächsten Kapitel behandelt wird.

7.2.2 WebVR

WebVR steht für Web Virtual Reality. Das Ziel von WebVR ist es, den Browser und damit WebGL so zu erweitern, dass Virtual Reality Inhalte und HMDs unterstützt werden [24]. WebVR wurde im Jahr 2014 von Vlad Vukićević und Josh Carpenter entwickelt. Vukićević arbeitet als Ingenieur für Mozilla während Carpenter als VR Wissenschaftler für Google arbeitet. Das ursprüngliche Ziel war die Integration von Browsern, so dass sie mit der Rift von Oculus genutzt werden können. Mittlerweile werden alle gängigen HMDs unterstützt [6] S. 65.

WebVR erweitert WebGL um folgende Schlüsseleigenschaften [6] S. 66:

- HMDs erkennen und verbinden: WebVR erkennt die HMDs unterschiedlicher Hersteller und verbindet den Webbrowser mit dem am PC angeschlossenen HMD.

VR Fullscreen Modus: Der Browser wird so erweitert, dass Inhalte im Fullscreen Modus angezeigt werden können und zwei Bilder gleichzeitig gerendert werden.

Head Tracking

Mit Hilfe einer JavaScript API werden Kopfbewegungen überwacht, so dass die Position und die Orientierung des Kopfes bekannt sind und damit die Kamera der Szene angepasst werden kann, bevor ein neues Bild in der Anwendung gerendert wird.

Somit kann mit Hilfe von WebGL und WebVR eine browserbasierende Virtual Reality Anwendung betrieben werden, welche HMDs als Ein- und Ausgabegerät nutzt.

7.2.3 Three.js

Bei Three.js handelt es sich um eine Bibliothek, welche auf WebGL aufsetzt. Sie verringert den Programmieraufwand, im Gegensatz zur reinen WebGL Programmierung wesentlich. So würde es beispielsweise mehrere hundert Zeilen Code brauchen, um einen einfachen Würfel mit reinem WebGL zu erstellen. Denselben Würfel mit Hilfe von Three.js zu erstellen, würde nur einen Bruchteil an Zeilen mit Code brauchen [25].

Außerdem bietet Three.js ein intuitives Set von Objekten, die in jeder üblichen 3D Grafik Anwendung zu finden sind. Zusätzlich ist Three.js ein OpenSource Projekt welches über Github zugänglich ist und ständig von einem Team von Entwicklern verbessert wird [6] S. 71.

7.2.4 A-Frame von Mozilla und GLAM

Diese Kapitel behandelt Auszeichnungssprachen. Auszeichnungssprachen sind maschinenlesbare Sprachen für die Gliederung von Daten und Texten [26]. Hypertext Markup Language (HTML) ist beispielsweise eine weit verbreitete Auszeichnungssprache. Im Folgenden werden nun Auszeichnungssprachen vorgestellt, die zur Entwicklung von Virtual Reality Inhalten verwendet werden.

Bei A-Frame handelt es sich um einen OpenSource Framework, welcher vom MozVR Team von Mozilla entwickelt wurde. Das Team von MozVR entwickelte dieses Framework, weil sie erkannten, dass es nur wenige Entwickler gibt, die über die Kenntnisse verfügen, Virtual Reality Inhalte direkt über WebGL zu entwickeln. Mit Hilfe von A-Frame möchte es Mozilla Web Designern, Web Entwicklern und 3D Künstlern ermöglichen, Virtual Reality Inhalte zu entwickeln, ohne dass sie sich Wissen über WebGL aneignen müssen. Vielmehr reicht ihr bereits vorhandenes Wissen über HTML und JavaScript aus. A-Frame läuft auf Three.js, was heißt, dass es die Funktionen von Three.js übernimmt, diese aber unter selbständigen Tags „versteckt“ und somit den vom Entwickler zu schreibenden Code verringert. Außerdem nutzt A-Frame ein sogenanntes „entity-component-system“. Diese Art von System ist ein Standard aus der Spiele Industrie. Mit Hilfe dieses Systems werden die Inhalte leicht erweiterbar, kombinierbar und sind sehr flexibel. Auf der Webseite von A-Frame beschreibt das MozVR Team A-Frame im Kern als:

“Entity-component-system-based three.js framework with a DOM interface”

Inhalte, die mit A-Frame entwickelt wurden sind browserbasierend und funktionieren somit auf PCs, Android und iOS Smartphones als auch auf der Rift von Oculus [27].

GLAM steht für Graphic Library And Markup und wurde von Toni Paraisi erfunden. Es handelt sich dabei um eine deklarative Sprache. GLAM nutzt WebGL in Verbindung mit Three.js und erweitert es durch einfach zu nutzende Tags und Style Eigenschaften. Um mit Hilfe von GLAM 3D Objekte zu erstellen, werden einzelne Elemente definiert, wie etwa ein Würfel, und anschließend die Attributes des Würfels festgelegt. Zusätzlich können so genannte Event Listeners installiert werden,

um etwa den Würfel auf die Mausbewegung eines Nutzers reagieren zu lassen [6] S. 82.

7.3 Zusammenfassung

Dieses Kapitel beschäftigt sich mit den Möglichkeiten der Entwicklung von Inhalten für HMDs. Es werden Techniken zum nativem Entwickeln und Webentwickeln vorgestellt. Beim nativen Entwickeln werden Software und Tools verwendet, die vom Hersteller des HMD bereitgestellt werden. Beim Web Entwickeln werden verschiedene Browsertechnologien verwendet, die aufeinander aufbauen. Abbildung 12 zeigt die Browsertechnologien.

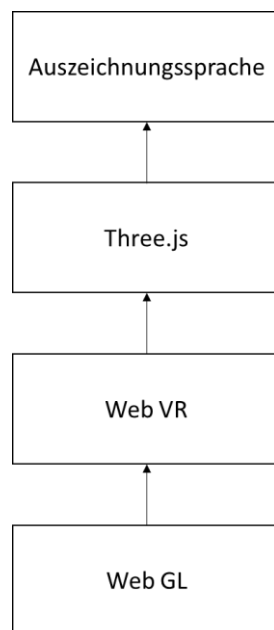


Abbildung 12: Browsertechnologien für HMD Web Development

Web GL dient als Basis, während der Entwickler mit der Auszeichnungssprache interagiert.

8 Produktanalyse zur Anschaffung eines HMDs für Lehrzwecke in einer Bildungseinrichtung

Dieses Kapitel behandelt den Vergleich zwischen der Rift von Oculus, dem Dark Shader von ZAAK ZURICH BERLIN – Cardboard Kits Europe und der VR One von Zeiss. Dabei soll die Tauglichkeit der verschiedene HMDs als Lehrgegenstand innerhalb einer Bildungseinrichtung überprüft werden. Nachfolgend werden zuerst die Vergleichskriterien festgelegt und definiert.

8.1 Festlegung und Definition der Kriterien für die Anschaffungsempfehlung

In diesem Abschnitt werden die Vergleichskriterien für die Nutzwertanalyse und den AHP festgelegt und erläutert. Manche Kriterien verfügen über Unterkriterien, Hauptkriterien werden durch Unterstreichungen von ihren jeweiligen Unterkriterien unterschieden.

Kosten

Dieses Kriterium bewertet die Kosten des jeweiligen HMDs. Dabei stehen die Anschaffungskosten in Vordergrund. Zusätzlich sollen aber auch evtl. auftretende laufende Kosten bewertet werden.

Robustheit

Mit Robustheit wird die Langlebigkeit des jeweiligen HMDs bewertet. Dabei wird beurteilt wie gut das HMD äußere Einflüsse, die bei der Nutzung auftreten, übersteht. Die Robustheit des HMDs ist nicht zuletzt auf die Eigenschaften der verwendeten Materialien zurückzuführen.

Tragekomfort

Der Tragekomfort beschreibt den Komfort den das HMD aufweist, wenn es vom Nutzer verwendet wird. Es wird durch folgende Unterkriterien beschrieben:

Gewicht

Gewicht meint das Gesamtgewicht des HMDs, welches auf dem Nutzer bei der Verwendung wirkt.

Polsterung

Die Polsterung beschreibt die Qualität der Polsterung die am HMD angebracht ist und als Kontaktfläche zwischen Nutzer und HMD dient.

Qualität der Immersion

Dieses Kriterium beschreibt, wie stark die Immersion ist, die das HMD erzeugen kann. Ist das HMD dazu in der Lage dem Nutzer eine virtuelle Welt als präsent vorzutäuschen? Dieses Hauptkriterium ist von folgenden Unterkriterien abhängig:

Displayauflösung

Die Displayauflösung gibt an, aus wie vielen Bildpunkten die Inhalte auf dem Display des HMD bestehen. Dabei gilt, je höher die Auflösung desto mehr Platz steht für die Darstellung zu Verfügung und desto schärfer können Inhalte für den Betrachter dargestellt werden.

Qualität der grafischen Darstellung

Qualität der grafischen Darstellung meint, ob die mit dem HMD verbundene Hardware leistungsstark genug ist, um virtuelle Welten glaubwürdig aussehen zu lassen, oder aber, ob die Grafikleistung nur virtuelle Welten darstellen kann, die auch leicht als solche erkennbar sind.

Präzision des Trackings

Bei der Präzision des Trackings spielt die Latenz eine entscheidende Rolle. „Die Zeitspanne, die ein System für die Reaktion auf eine Eingabe benötigt, bezeichnet man dabei als Latenz (engl. Latency). Je größer die Latenz des Systems, desto größer ist der zeitliche Abstand der wahrnehmbaren Konsequenz einer Handlung und desto dissoziierter ist die Wirkung auf den Nutzer.“ [28] S. 196.

Bei HMDs mit Head-Tracking stellt die Berechnung der dynamischen Perspektive des Nutzers eine besondere Anforderung an die Latenz eines HMD dar. Sobald der Nutzer nur noch virtuelle Inhalte wahrnimmt, wirkt sich eine hohe Latenz besonders stark auf die Qualität des Erlebnisses für den Nutzer aus. Schwindelgefühle sind die Folge, welche zunehmen, je höher die Latenz des Systems ist [28] S. 197. „Meehan et al. (2003) konnten z. B. eine deutlich höhere Zahl an Personen mit Schwindelgefühlen feststellen, wenn sie die Latenz eines HMDs von 50 ms auf 90 ms erhöhten. Eine Latenz von unter 50 ms wird für HMDs empfohlen (Brooks 1999; Ellis et al. 1997).“ [28] S. 197.

Frames per Second

Mit dem Kriterium Frames per Second, also Bilder pro Sekunde, wird gezeigt wie hoch die Bildwiederholungsrate und damit die Reaktionszeit des verwendeten Displays ist. Ein weiter wichtiger Faktor ist auch die Schwankung der Wiederholungsrate. Es ist wenig hilfreich, wenn bei einer Wiederholungsrate von 100 Hz (also 100 Bilder pro Sekunde) 99 der Bilder innerhalb der ersten 5 ms dargestellt werden und

das letzte Bild erst nach weiteren 995 ms angezeigt wird. Die kritische Wiederholungsrate bei einer Einzelbildabfolge liegt bei 50 Hz, allerdings ist ein Bild erst ab einer Wiederholungsrate von 100 Hz absolut flimmerfrei [28] S. 198.

Sichtfeld (Field of View)

Unter FOV versteht man den horizontalen und vertikalen Winkel ausgehend vom Auge des Nutzers, von dem die virtuellen Informationen wahrgenommen werden, siehe Abbildung 13. Je größer das FOV ist, desto stärker ist die Illusion des Nutzers, in eine virtuelle Welt versetzt zu werden, da diese umso weiträumiger angezeigt wird. [8] S. 143.

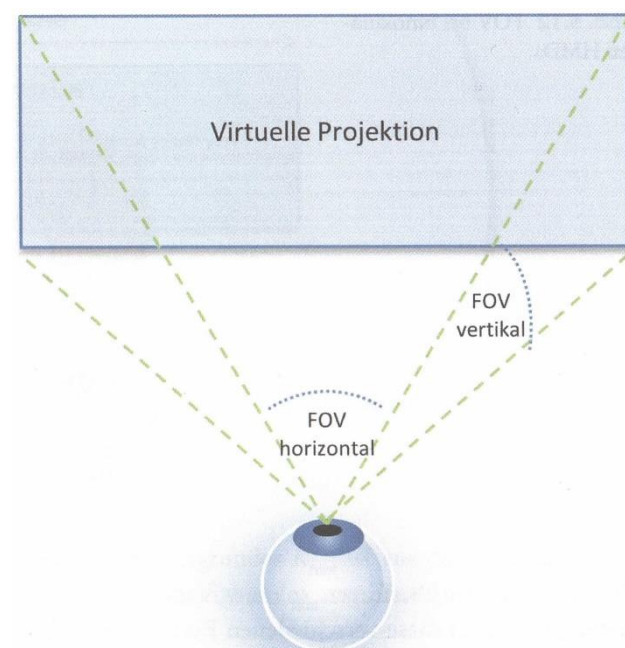


Abbildung 13: Prinzipskizze des FOV ausgehend vom Nutzerauge [8] S. 143

Benutzerfreundlichkeit

Benutzerfreundlichkeit bewertet, wie einfach es ist ohne Vorkenntnisse das HMD zu bedienen. Dieses Hauptkriterium teilt sich wiederum in folgende Unterkriterien auf:

Inbetriebnahme

Mit dem Kriterium Inbetriebnahme wird ausgedrückt, wie leichtgängig die notwendigen Vorkehrungen zu erledigen sind, um das HMD erstmals zu betreiben.

Bedienbarkeit

Bei der Bedienbarkeit wird ermittelt, wie intuitiv die Verwendung des HMDs ist, oder aber ob für den Nutzer eine gewisse Einarbeitungszeit notwendig ist.

Dokumentation

Die Dokumentation gibt an, welche Informationen der Hersteller über das HMD zu Verfügung stellt, beispielsweise Betriebsanleitungen, Handbücher, Wartungspläne.

Mobilität des Systems

Mobilität des Systems beschreibt, wie frei sich der Nutzer während er das HMD gebraucht, bewegen kann. Ist seine Bewegung komplett uneingeschränkt oder ist es nötig, dass er an einem PC sitzt? Wird seine Bewegung durch Kabel eingeschränkt?

Open Source Software

Open Source Software bewertet, ob die Software die notwendig ist, um das HMD zu betreiben, frei zugänglich ist.

Open Hardware

Das Kriterium Open Hardware zeigt auf, ob technische Zeichnungen zu der verwendeten Hardware öffentlich verfügbar sind. Damit können in Schadensfällen viele Reparaturen vom Nutzer selbst durchgeführt werden.

Verfügbare Software

Dieses Kriterium misst, wie viele Anwendungen für das jeweilige HMD zur Verfügung stehen. Dadurch wird auch dargestellt, wie beliebt das jeweilige HMD bei Entwicklern bzw. Nutzern ist.

Entwicklung für diese Plattform

Dieses Kriterium bewertet, wie leicht die Entwicklung von Inhalten für die jeweilige Plattform des HMD ist.

Anhand dieser Kriterien gilt es nun, eine klare Formulierung des Entscheidungsproblems zu treffen.

8.2 Formulierung des Entscheidungsproblems

In diesem Abschnitt wird das Entscheidungsproblem ausformuliert. Es lautet wie folgt:

„Welche der drei HMDs eignet sich am besten als Lehrgegenstand einer Laborübung zum Thema IoT-HMD?“

Dabei ist auf die höhere Gewichtung der Kriterien Kosten und Robustheit zu achten. Anschließend werden die einzelnen Kriterien gewichtet.

8.3 Vorstellung der Verfahren

Dieser Abschnitt stellt die Verfahren Nutzwertanalyse und Analytical Hierarchy Process vor, die für die Entscheidungsfindung genutzt werden.

8.3.1 Nutzwertanalyse

Bei der vorliegenden Entscheidung zwischen einem der drei HMDs, handelt es sich um eine komplexe Aufgabe. Die Fragestellung verfügt über eine hohe Anzahl an Entscheidungskriterien. Weiterhin ist es nicht von vorne herein möglich, eine eindeutige Rangfolge der Entscheidungskriterien festzulegen. Außerdem soll die Entscheidungsfindung nachvollziehbar sein und dokumentiert werden. Zusätzlich soll die Entscheidungsfindung von mehreren Personen durchgeführt werden.

Bei Fragestellungen mit dieser Charakteristik bietet es sich an, auf das Verfahren der Nutzerwertanalyse zurückzugreifen [29] S. 2–3. "Nutzwertanalysen dienen dazu, komplexe Entscheidungen zu treffen. Das Mittel, das dazu eingesetzt wird, ist die „Fragmentierung“: Das Gesamtproblem, das es zu entscheiden gilt, wird in Teilprobleme zerlegt und diese, wenn erforderlich, wiederum in Teilprobleme." [29] S. 1. Die Zerlegung in Teilprobleme wird im nächsten Abschnitt dadurch realisiert, dass verschiedenen Eigenschaften der HMDs durch unterschiedliche Hauptkriterien und deren Unterkriterien dargestellt werden.

8.3.2 Analytical Hierarchy Process

Der Analytical Hierarchy Process (AHP) ist ein Verfahren zur Entscheidungsfindung, welches von Thomas Saaty entwickelt wurde. Das Ziel ist dabei ebenfalls, die beste Lösung aus einer Anzahl von Alternativen zu finden. Es kommt bei der Gewichtung der Kriterien auch die Paarvergleichsmethode zum Einsatz. Im Gegen-

satz zur Nutzwertanalyse prüft der AHP die Inkonsistenz und er erlaubt eine Inkonsistenz bzw. Intransitivität der Entscheidung bis zu einem gewissen Grad. Anschließend stellt er Mittel zur Verfügung, um diese zu verbessern [30] S. 1.

8.4 Gewichtung der Entscheidungskriterien

In diesem Abschnitt wird der Gewichtung der Entscheidungskriterien der beiden Verfahren miteinander verglichen, sowie auf Stärken bzw. Schwächen eingegangen. Zuerst wird die Nutzwertanalyse näher betrachtet und anschließend der AHP.

8.4.1 Gewichtung der Entscheidungskriterien mit der Nutzwertanalyse

Die Gewichtung der Entscheidungskriterien wird von mehreren Personen durchgeführt, um eine möglichst repräsentative Gewichtung zu erhalten. Es wird die Paarvergleichsmethode verwendet. Zuerst werden alle zu gewichtenden Kriterien in einer Kreuztabelle eingetragen, siehe Abbildung 14.

Kriterium	A	B	C	D	E
A					
B					
C					
D					
E					

Abbildung 14: Kreuztabelle für Paarvergleichsmethode [29] S. 14

Anschließend werden alle Personen gebeten, für sich zu entscheiden, ob im direkten Vergleich Kriterium A wichtiger ist als Kriterium B. Wenn Kriterium A wichtiger ist, wird der Buchstabe A ins frei Feld der Kreuztabelle eingetragen, sollte B wichtiger sein so wird B eingetragen. Dieser Schritt wird mit allen Kriterien durchgeführt. Die Gegenfrage ist nicht notwendig: Wenn bewertet wurde, dass A wichtiger als B ist, muss B unwichtiger als A sein. Der Vergleich eines Kriteriums mit sich selbst ist sinnlos und darum werden diese Felder – hier dunkel hinterlegt – nicht ausgefüllt. Aus diesen Feldern ergibt sich eine Diagonale. Auf Grund der Tatsache das die Gegenfrage nicht notwendig ist, muss von Teilnehmern nur eine Hälfte der Kreuztabelle ausgefüllt werden, entweder oberhalb oder unterhalb der

Diagonale, die jeweils andere Seite der Diagonale ergibt sich, wegen den Gegenfrage, von selbst, siehe Abbildung 15.

Kategorie	A	B	C	D	E	Σ
A		B	A	A	A	3
B	B		C	B	B	3
C	A	C		C	E	2
D	A	B	C		D	1
E	A	B	E	D		1

Abbildung 15: Ausgefüllte Kreuztabelle für die Paarvergleichsmethode [29] S. 33

Ein Unentschieden zwischen zwei Kriterien lässt die Nutzerwertanalyse nicht zu, der Teilnehmer muss immer ein Kriterium dem anderen gegenüber bevorzugen. Anschließend wird an der rechten Seite der Kreuztabelle die Anzahl der Nennungen des jeweiligen Kriteriums in seiner Zeile aufsummiert, siehe Abbildung 15. Wichtig ist, dass das *Axiom der Transitivität* vom Verfahrensteilnehmer berücksichtigt wird. Transitivität bedeutet, dass, wenn das Kriterium A dem Kriterium B präferiert (vorgezogen) wird und gleichzeitig das Kriterium B dem Kriterium C präferiert wird, folglich das Kriterium A auch gegenüber dem Kriterium C präferiert werden muss [31] S. 16–17. Die entsprechende Formel lautet:

$$A > B \wedge B > C \Rightarrow A > C \quad (1)$$

Die durch die Summierung zu Stande gekommen Zahlen entsprechen der Gewichtung der Kriterien. So wird im obigen Beispiel das Kriterium A dreifach gewichtet, während Kriterium C zweifach gewichtet und Kriterium E nur einfach gewichtet wird. [29] S. 14.

Für den vorliegenden Vergleich zwischen den drei HMDs sind die Kreuztabellen zur Gewichtung der Entscheidungskriterien im Anhang zu finden. Die Anhänge eins bis vier beinhalten dabei die Vorlagen zur Durchführung der Paarvergleichsmethode. Bei Anhang 1 handelt es sich um die Gewichtung aller Hauptkriterien. Anhang 2 dient als Vorlage zur Gewichtung der Unterkriterien des Tragekomforts, während Anhang 3 zur Gewichtung der Unterkriterien der Qualität der Immersion

genutzt wird. Zuletzt befindet sich in Anhang 4 die Gewichtung der Unterkriterien zur Benutzerfreundlichkeit. Die Anhänge fünf bis acht enthalten exemplarisch das ausgefüllte Äquivalent zu den Vorlagen aus den Anhängen eins bis vier.

8.4.2 Ergebnisse aus der Paarvergleichsmethode mit der Nutzwertanalyse

Insgesamt wurde die Paarvergleichsmethode zwischen den Kriterien von fünfzehn Personen durchgeführt. Tabelle 1 zeigt die Ergebnisse jeder Person aus der Paarvergleichsmethode. Im oberen Teil der Tabelle ist die Gewichtung der Unterkriterien Tragekomfort, Qualität der Immersion und Benutzerfreundlichkeit zu sehen. Anschließend wird die Gewichtung der Hauptkriterien gezeigt. Am Ende jeder Spalte wurden die einzelnen Werte aus der Befragung der unterschiedlichen Personen aufsummiert und anschließend die entsprechende Prozentzahl gebildet. In der Summe werden innerhalb der Unterkriterien, als auch innerhalb der Hauptkriterien, 100 % erreicht.

	Kriterium	Person 1	Person 2	Person 3	Person 4	Person 5	Person 6	Person 7	Person 8	Person 9	Person 10	Person 11	Person 12	Person 13	Person 14	Person 15	Summe	Prozent	
Tragekomfort	Gewicht	0	0	1	1	0	1	1	1	1	1	0	1	1	0	1	10	66,67%	
	Polsterung	1	1	0	0	1	0	0	0	0	0	1	0	0	1	0	5	33,33%	
																	15	100,00%	
Qualität der Immersion	Displayauflösung	0	2	4	1	2	1	0	2	0	1	2	1	3	1	1	21	14,09%	
	Qualität der grafischen Texturen	2	0	1	2	2	2	1	1	3	2	1	0	3	2	0	22	14,77%	
	Präzision des Trackings	4	4	3	3	3	4	2	4	2	4	4	4	3	4	4	52	34,90%	
	FPS	1	3	2	1	3	0	4	2	1	3	0	3	0	0	3	26	17,45%	
	Sichtfeld	3	1	0	3	0	3	3	0	4	0	3	3	2	1	3	28	18,79%	
																	Summe	149	100,00%
Benutzerfreundlichkeit	Inbetriebnahme	1	0	1	1	2	1	0	2	0	1	1	1	1	0	0	12	27,27%	
	Bedienbarkeit	2	2	2	1	1	2	2	1	2	2	2	2	2	2	2	27	61,36%	
	Dokumentation	0	0	0	1	0	0	1	0	1	0	0	0	0	1	1	5	11,36%	
																	Summe	44	100,00%
Hauptkriterien	Kosten	8	7	5	2	9	4	6	2	1	1	4	3	4	2	4	62	9,42%	
	Robustheit	4	4	2	4	8	5	7	1	3	2	7	4	3	1	2	57	8,66%	
	Tragekomfort	8	3	2	7	5	6	9	4	2	0	7	6	9	2	0	70	10,64%	
	Qualität der Immersion	4	9	8	5	7	8	0	5	4	5	8	7	6	4	9	89	13,53%	
	Benutzfreundlichkeit	8	3	8	4	1	8	8	6	9	5	4	8	6	5	3	86	13,07%	
	Mobilität des Systems	4	1	4	5	2	8	0	3	5	4	4	5	1	4	5	55	8,36%	
	Open Source Software	5	1	6	1	1	2	0	7	7	7	2	1	6	7	7	60	9,12%	
	Open Hardware	0	6	0	0	5	2	0	0	6	7	1	0	2	3	8	40	6,08%	
	Verfügbare Software	3	5	5	8	3	2	2	8	8	4	8	9	6	9	1	81	12,31%	
	Entwicklung für diese Plattform	1	2	5	9	4	0	2	9	0	9	0	2	1	8	6	58	8,81%	
																Summe	658	100,00%	

Tabelle 1: Ergebnisse aus der Paarvergleichsmethode mit der Nutzwertanalyse

Um die Daten aus Tabelle 1 verständlicher zu gestalten, bietet es sich an, die Verteilung der Punktzahl für das jeweilige Kriterium mit Hilfe eines Boxplot Diagramms zu visualisieren. Mit Hilfe des Boxplot können das Minimum, das Maximum, der Median, das erste Quartil, das dritte Quartil sowie Ausreißer nach oben als auch nach unten dargestellt werden. Außerdem erkennt der Leser anhand des Boxplot in welchem Bereich sich die einzelnen Gewichtungen der unterschiedlichen Personen verteilen. So wird beispielsweise für das Kriterium Kosten ein Boxplot erzeugt, welcher die Werte aus den Befragungen aller fünfzehn Personen enthält. Figure 16 zeigt diesen Boxplot in blauer Farbe. Betrachtet man Figure 16 genauer, ohne im Detail auf einen bestimmten Boxplot einzugehen, stellt man fest, dass jeder Boxplot eine große Streuung der einzelnen Werte aufzeigt. Das Minimum liegt bei jedem Boxplot mindestens bei eins, während das Maximum mindestens acht beträgt. Aus diesem Zusammenhang ergibt sich auch der Umstand, dass keiner der Boxplot über Ausreißer verfügt. Der Interquartilsabstand, welcher dem Abstand zwischen dem ersten und dritten Quartil entspricht, ist entsprechend groß. Als Ausreißer werden Punkte bezeichnet, welche sich mehr als das eineinhalbfache des Interquartilsabstandes unter dem ersten Quartil bzw. über dem dritten Quartil befinden. Als weitere Ursache, für eine Verteilung der Datenpunkte wie in Figure 16, ist die Art der Entscheidungsfindung, zwischen zwei Kriterien innerhalb des Paarvergleichs, für die Nutzwertanalyse zu nennen. Wie bereits in 8.4.1 erwähnt, wird immer die Zahl des Kriteriums eingetragen, welches beim Vergleich zwischen zwei Kriterien präferiert wird. Somit handelt es sich um eine binäre Entscheidung und es ergeben sich nur ganzzahlige Werte, siehe Tabelle 1. Aus diesem Grund macht es auch keinen Sinn ein Boxplot Diagramm für die Unterkriterien zu erstellen, da sie eine zu geringe Anzahl an unterschiedlichen Zahlenwerten enthalten. Kreuztabellen, die nach der oben beschriebenen Methode ausgefüllt wurden, können maximal Zahlenwerte von null bis $n - 1$ enthalten, wobei n gleich der Dimension der Kreuztabelle ist. Somit kann beispielsweise die Tabelle der Unterkriterien

des Tragekomforts maximal die Zahlenwerte null und eins enthalten, siehe Tabelle 1.

Im Gegensatz zur Nutzwertanalyse bietet der Analytical Hierarchy Process eine wesentliche bessere Bewertung zweier Kriterien zueinander, was zu einer geringeren Streuung führt und damit die oben genannten Probleme nicht auftreten. Dieser Zusammenhang kann im späteren Verlauf der Arbeit betrachtet werden.

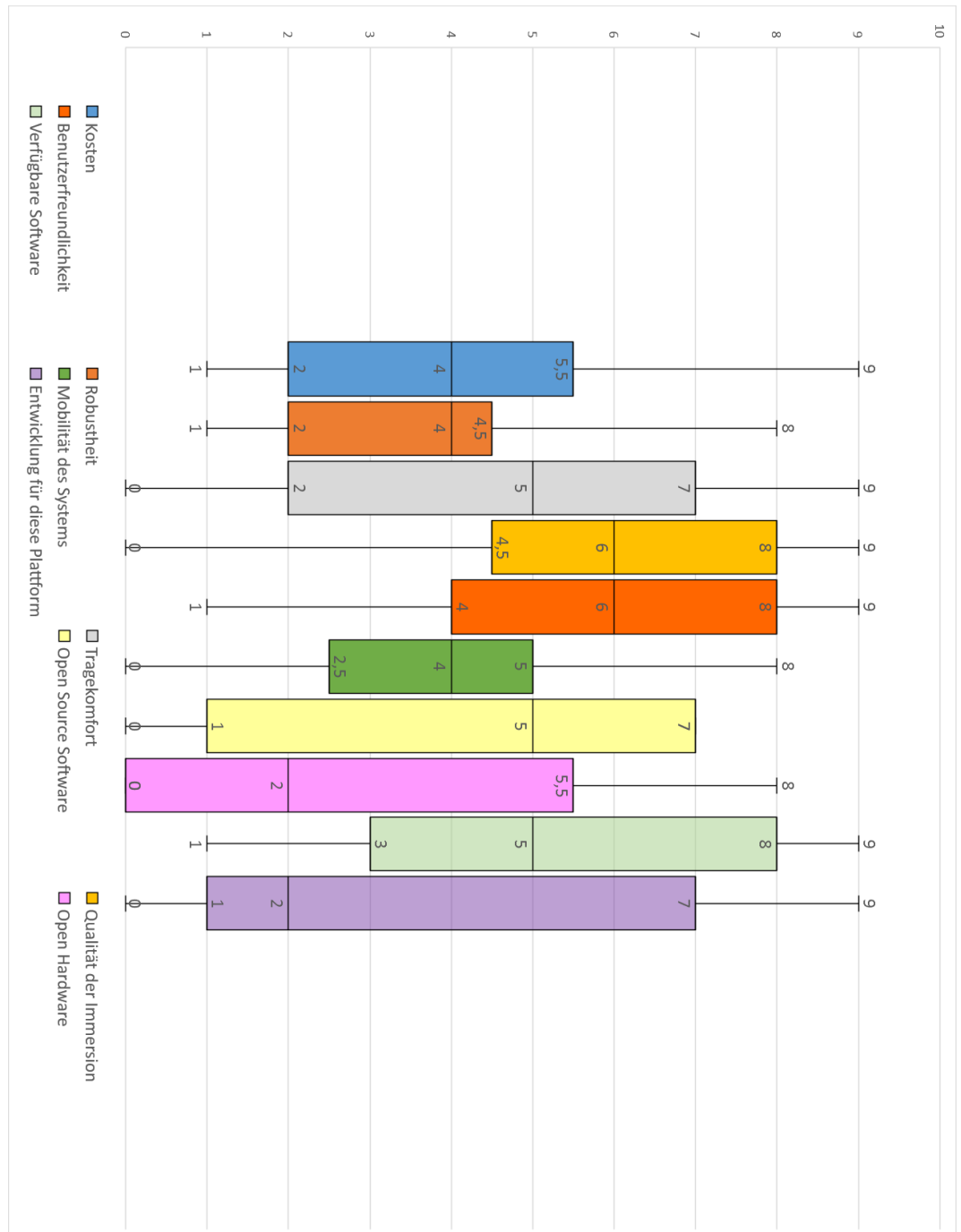


Figure 16: Boxplotdiagramm Hauptkriterien Nutzwertanalyse

Die tatsächliche Gewichtung der einzelnen Kriterien wird durch die Summierung der Zeile des jeweiligen Kriteriums berechnet, welche in der vorletzten Spalte aus

Tabelle 1 zu sehen sind. Anschließend werden diese Zahlen noch in Prozent dargestellt, um die Verarbeitung zu erleichtern. Tabelle 2 zeigt diese Ergebnisse nochmals in kompakter Form.

	Kriterium	Summe	Prozent
Tragekomfort	Gewicht	10	66,67%
	Polsterung	5	33,33%
		15	100,00%
Qualität der Immersion	Displayauflösung	21	14,09%
	Qualität der grafischen Texturen	22	14,77%
	Präzession des Trackings	52	34,90%
	FPS	26	17,45%
	Sichtfeld	28	18,79%
		149	100,00%
Benutzerfreundlichkeit	Inbetriebnahme	12	27,27%
	Bedienbarkeit	27	61,36%
	Dokumentation	5	11,36%
		44	100,00%
Hauptkriterien	Kosten	62	9,42%
	Robustheit	57	8,66%
	Tragekomfort	70	10,64%
	Qualität der Immersion	89	13,53%
	Benutzerfreundlichkeit	86	13,07%
	Mobilität des Systems	55	8,36%
	Open Source Software	60	9,12%
	Open Hardware	40	6,08%
	Verfügbare Software	81	12,31%
	Entwicklung für diese Plattform	58	8,81%
		658	100,00%

Tabelle 2: Prozentuale Gewichtung der Kriterien

So wird beispielsweise das Kriterium Benutzerfreundlichkeit, bei den Hauptkriterien, mit 13,07 % gewichtet. Mit Hilfe der erarbeiteten Gewichtung gilt es nun im nächsten Schritt der Nutzwertanalyse die verschiedene Entscheidungsalternativen zu bewerten.

Zunächst wird aber im nächsten Abschnitt die Gewichtung der Entscheidungskriterien mit dem AHP dargestellt.

8.4.3 Gewichtung der Entscheidungskriterien mit dem AHP

Als weiteren Vorteil gegenüber der Nutzwertanalyse, nutzt der AHP bei der Paarvergleichsmethode eine präzisere Skala. Somit wird mit Hilfe des AHP Kriterium A nicht nur wichtiger als Kriterium B gewertet, sondern der AHP gibt auch Aufschluss darüber wie viel wichtiger Kriterium A gegenüber Kriterium B ist. Die Skala, die diesen Zusammenhang zum Ausdruck bringt, sieht wie folgt aus [31] S. 8:

Mögliche Werte für a_{ij}	Bedeutung der möglichen Werte für a_{ij}
1	gleiche Bedeutung der beiden Elemente
3	etwas höhere Bedeutung eines Elements
5	deutlich höhere Bedeutung eines Elements
7	viel höhere Bedeutung eines Elements
9	sehr viel höhere Bedeutung eines Elements
2, 4, 6, 8	Zwischenwerte
$\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7}, \frac{1}{8}, \frac{1}{9}$	Wenn a_{ij} einen der Werte von 1 bis 9 annimmt, ist a_{ji} der Reziprokwert.

Abbildung 17: Skala Paarvergleichsmethode AHP [31] S. 9

Die Kennzeichnung a_{ij} in Abbildung 17 steht für das Feld innerhalb der Kreuztabelle, welches durch die Zeile i und der Spalte j bestimmt ist. Im Gegensatz zur Nutzwertanalyse erlaubt es der AHP Kriterien als gleich wichtig zu beurteilen. Ist dies der Fall, bekommen beide Felder den Wert eins.

Item Number	Item Number	1	2	3	4
	Item Description	Gewicht	Polsterung	Mit Brille nutzbar	subjektives Empfinden
1	Gewicht	1,00	5,00000	7,00000	5,00000
2	Polsterung	0,20	1,00	9,00000	1,00000
3	Mit Brille nutzbar	0,14	0,11	1,00	0,20000
4	subjektives Empfinden	0,20	1,00	5,00	1,00
5	Sum	1,54	7,11	22,00	7,20

Abbildung 18: Kreuztabelle AHP

Der Wert für a_{12} in Abbildung 18 ist somit 5,0 und damit wäre das Kriterium Gewicht fünfmal wichtiger als das Kriterium Polsterung, was in Bezug auf Abbildung 17 bedeutet, dass das Gewicht eine deutlich höhere Bedeutung wie die Polsterung hat.

Aus diesem Zusammenhang ergibt sich auch, die Konsistenz Bedingung für den AHP. Diese Bedingung lautet [31] S. 13:

$$a_{ik} * a_{kj} = a_{ji} \quad (2)$$

Wendet man diese Bedingung auf Abbildung 18 an, so lautet sie beispielsweise:

$$a_{12} * a_{24} = a_{14}$$

Setzt man nun die Werte aus den jeweiligen Feldern ein ergibt sich:

$$5 * 1 = 5$$

Somit ist die Bedingung der Konsistenz in diesem Fall erfüllt. Allerdings lässt sich, bei der Verwendung der Paarvergleichsmethode mit der Skala aus Abbildung 17, Inkonsistenz kaum vermeiden [31] S. 17. Deshalb hat Saaty eine Kennzahl entwickelt, die besagt, ob die vorhandene Inkonsistenz akzeptabel ist oder aber die Paarvergleichsmethode überarbeitet werden muss. Diese Kennzahl nennt sich *Consistency Ratio*. Die Formel zur Berechnung der Consistency Ratio ist [31] S. 15:

$$C.R. = \frac{C.I.}{R.I.} \quad (3)$$

R.I. bezeichnet in der obigen Formel den so genannten *Random Index*. Dieser ist ein aus zufällig ermittelten, reziproken Matrizen gebildeter durchschnittlicher Konsistenzindex. Die in Tabelle 3 dargestellten Werte geben den Wert des Random Index in Abhängigkeit der Dimensionen n der Kreuztabelle wieder [31] S. 15.

Anzahl der Dimensionen n	Random Index
1	0
2	0
3	0,58
4	0,90
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45
10	1,49

Tabelle 3: Werte des Random Index

So beträgt beispielsweise bei der Kreuztabelle aus Abbildung 18 der Wert des Random Index 0,90, da die Anzahl der Dimensionen gleich vier ist.

$C. I.$ aus obiger Formel bezeichnet den so genannten *Consistency Index*. Er wird mit Hilfe des größten Eigenwertes λ_{\max} und der Dimension n der Kreuztabelle berechnet. Die Formel lautet [31] S. 15:

$$C. I. = \frac{\lambda_{\max} - n}{n - 1} \quad (4)$$

Der größte Eigenwert für die Kreuztabelle aus Abbildung 18 beträgt 4,371. Damit ergibt sich für den Consistency Index:

$$C. I. = \frac{4,371 - 4}{4 - 1} = 0,124$$

Nun kann mit Hilfe des Random Index die Consistency Ratio berechnet werden:

$$C. R. = \frac{0,124}{0,90} = 0,138$$

Thomas Saaty empfiehlt für den Fall, dass die Consistency Ratio größer oder gleich 0,1 ist eine Überarbeitung der Kreuztabelle, was für den vorliegenden Fall somit erforderlich ist [31] S. 16. Die erhöhte Consistency Ratio lässt darauf schließen, dass die Entscheidungen in der Kreuztabelle nicht ausreichend konsistent sind bzw. das *Axiom der Transitivität* verletzt wird [31] S. 16–17.

Die Anhänge neun bis zwölf zeigen die verwendeten Vorlagen für die Paarvergleichsmethode des AHP, während die Anhänge dreizehn bis sechzehn exemplarisch die ausgefüllten Äquivalente enthalten. Bei der Vorlage und der ausgefüllten Vorlage des Tragekomforts zeigt die Konsistenzprüfung den Wert „#DIV/0!“ an. Dies ist dem Umstand geschuldet, dass nur zwei Kriterien innerhalb dieser Tabelle verglichen werden und somit keine Inkonsistenz auftreten kann, folglich funktioniert die Rechnung auch nicht.

Die in dieser Arbeit genutzten Excel Dateien für den AHP wurden von Thomas Pyzdek entwickelt und sind unter folgendem Link verfügbar: <http://sixsigmatraining.com/six-sigma-tools/ahp-spreadsheet.html>. Die Dateien wurden anschließend auf das hier vorliegende Entscheidungsproblem angepasst bzw. erweitert.

8.4.4 Ergebnisse aus der Paarvergleichsmethode mit dem AHP

Die Vorlagen für den AHP haben ebenfalls fünfzehn Personen ausgefüllt. Allerdings haben es gerade mal ein Drittel der Personen geschafft, die geforderte Consistency Ratio von 0,1 in allen Tabellen einzuhalten.

Als Grund für die Nichteinhaltung konnte oftmals die Unwissenheit über die Konsistenzbedingung und das Axiom der Transitivität genannt werden. Anschließend wurden die Personen auf diese Bedingung aufmerksam gemacht und Hilfestellung gegeben, um die Tabellen dementsprechend auszufüllen. Dabei stellte sich heraus, dass die befragten Personen auf Grund der geforderten Konsistenz nicht mehr in der Lage waren, ihre eigene Meinung beim Ausfüllen der Vorlagen zu vertreten, da diese nicht die Konsistenzbedingung erfüllen würde. Somit wird von der Konsistenzbedingung die persönliche Meinung verfälscht. Aus diesem Grund wurde die Entscheidung getroffen, die Konsistenzbedingung für die Befragung innerhalb dieser Arbeit auszusetzen. Der Grund dafür ist, dass die persönliche Meinung bei einer Befragung unterschiedlicher Personen als wichtiger zu sehen ist, als die hundertprozentige Konsistenz der Entscheidungen der einzelnen Personen. Eine vollständige Konsistenz hat sich im Rahmen dieser Arbeit und der verfügbaren Zeit als nicht praktikabel herausgestellt.

Tabelle 4 zeigt die Ergebnisse der Befragungen für den AHP. Die letzte Spalte zeigt den Median für das jeweilige Kriterium.

	Kriterium	Person 1	Person 2	Person 3	Person 4	Person 5	Person 6	Person 7	Person 8	Person 9	Person 10	Person 11	Person 12	Person 13	Person 14	Person 15	Median
Tragekomfort	Gewicht	75,00%	75,00%	16,67%	50,00%	87,50%	16,67%	88,89%	33,33%	75,00%	80,00%	75,00%	66,67%	80,00%	14,29%	85,71%	75,00%
	Polsterung	25,00%	25,00%	83,33%	50,00%	12,50%	83,33%	11,11%	66,67%	25,00%	20,00%	25,00%	33,33%	20,00%	85,71%	14,29%	25,00%
Qualität der Immersion	Displayauflösung	43,03%	6,27%	8,70%	11,75%	7,61%	4,84%	7,73%	12,32%	10,83%	8,71%	39,07%	4,83%	23,47%	21,66%	11,17%	10,83%
	Qualität der grafischen Texturen	3,83%	3,51%	32,94%	14,03%	4,84%	12,52%	13,78%	9,85%	19,54%	8,71%	8,39%	20,09%	14,66%	19,98%	3,81%	12,52%
	Präzession des Trackings	25,67%	52,61%	22,58%	21,33%	11,88%	37,86%	50,09%	42,25%	30,09%	40,11%	21,31%	25,74%	51,02%	14,43%	47,11%	30,09%
	FPS	17,92%	25,44%	8,59%	13,65%	52,09%	21,88%	25,51%	25,69%	9,47%	35,69%	19,98%	14,21%	7,32%	5,73%	16,10%	17,92%
	Sichtfeld	9,56%	12,15%	27,19%	39,24%	23,57%	22,89%	2,89%	9,90%	30,09%	6,78%	11,25%	35,13%	3,53%	38,19%	21,83%	21,83%
Benutzerfreundlichkeit	Inbetriebnahme	57,36%	22,12%	21,14%	32,45%	8,97%	18,04%	23,54%	12,43%	28,42%	10,38%	21,14%	14,29%	20,74%	25,00%	9,15%	21,14%
	Bedienbarkeit	28,64%	72,73%	68,64%	23,52%	30,33%	74,82%	70,70%	75,09%	61,94%	23,11%	65,51%	57,14%	73,78%	50,00%	70,71%	65,51%
	Dokumentation	13,99%	5,15%	10,22%	44,03%	60,70%	7,14%	5,76%	12,47%	9,64%	66,51%	13,35%	28,57%	5,48%	25,00%	20,14%	13,35%
Hauptkriterien	Kosten	19,44%	5,50%	18,50%	6,67%	14,15%	8,43%	1,29%	16,64%	9,86%	16,10%	12,95%	6,62%	9,76%	1,74%	12,73%	9,86%
	Robustheit	10,16%	6,32%	16,61%	13,18%	13,04%	13,11%	5,26%	6,79%	9,80%	16,41%	9,23%	9,28%	3,36%	2,72%	3,13%	9,28%
	Tragekomfort	13,20%	12,25%	18,76%	10,24%	22,56%	21,36%	2,45%	3,59%	10,06%	5,03%	5,20%	4,93%	15,17%	4,39%	2,80%	10,06%
	Qualität der Immersion	6,02%	11,94%	8,73%	12,67%	10,56%	27,00%	32,31%	28,41%	20,11%	8,44%	15,83%	10,82%	6,46%	9,00%	11,11%	11,11%
	Benutzerfreundlichkeit	18,46%	23,63%	16,35%	11,13%	29,02%	10,88%	5,96%	20,75%	19,71%	2,53%	16,23%	13,73%	30,71%	17,94%	8,56%	16,35%
	Mobilität des Systems	19,15%	7,56%	8,57%	9,32%	6,69%	3,39%	7,58%	6,46%	23,44%	2,13%	8,94%	5,54%	2,79%	7,03%	20,15%	7,56%
	Open Source Software	7,87%	3,71%	13,06%	12,31%	3,10%	1,96%	12,88%	14,05%	4,44%	1,31%	13,54%	14,83%	15,68%	13,30%	16,55%	12,88%
	Open Hardware	5,74%	1,12%	1,80%	7,32%	2,08%	1,96%	16,55%	6,67%	5,11%	4,31%	3,03%	9,79%	5,10%	8,39%	21,23%	5,11%
	Verfügbare Software	4,66%	36,73%	6,43%	13,44%	3,82%	21,33%	6,48%	4,35%	5,05%	4,33%	13,78%	29,64%	16,49%	26,66%	1,86%	6,48%
	Entwicklung für diese Plattform	6,42%	2,34%	2,30%	14,85%	6,11%	1,69%	20,34%	3,41%	3,53%	50,53%	12,38%	5,95%	5,59%	19,93%	13,00%	6,11%

Tabelle 4: Ergebnisse aus der Paarvergleichsmethode mit dem AHP

Im Folgenden sollen nun die Daten aus Tabelle 4 mit Hilfe von Boxplotdiagrammen visualisiert werden. Dabei können im Gegensatz zur Nutzwertanalyse auch für alle Unterkriterien aussagekräftige Boxplot erstellt werden, da es sich hier nicht um eine binäre Entscheidung handelt, sondern die in Abschnitt 8.4.3 vorgestellte Skala zur Bewertung der Kriterien zueinander verwendet wird. Dadurch entstehen beispielsweise in der Tabelle für das Unterkriterium Tragekomfort wesentlich mehr unterschiedliche Datenpunkte als das bei der Nutzwertanalyse der Fall ist, vergleiche Tabelle 1 zu Tabelle 4.

Figure 19 zeigt das Boxplot Diagramm zu den Unterkriterien des Tragekomforts. Dabei sind die Elemente des Boxplot, beispielsweise Median oder Quartil, beschriftet. Figure 20 zeigt das gleiche Diagramm um die Lage der einzelnen Datenpunkte innerhalb des Boxplot erweitert, aber auf Grund der Übersicht ohne Beschriftung von Datenpunkten. Auffällig ist, dass keines der beiden Kriterien über Ausreißer verfügt, was sicherlich an der großen Spannweite der Datenpunkte liegt. Tatsächlich liegt bei beiden Kriterien das Minimum und das Maximum ähnlich weit auseinander. Beim Kriterium Gewicht sind es 88,89 % minus 14,29 %, was einer Spannweite von 74,6 Prozentpunkten entspricht. Die Spannweite berechnet sich stets aus der Differenz zwischen größten Datenpunkt minus kleinsten Datenpunkt einer Beobachtung. Für das Kriterium Polsterung ergibt sich ein Bereich von 74,61 Prozentpunkten. Die Größe der Spannweite kann in diesem Zusammenhang als Grad für die Übereinstimmung der befragten Personen gesehen werden. Eine große Spannweite zeigt, dass ein Kriterium, je nach Person sehr unterschiedlich gewichtet wurde, während eine geringe Streuung darauf hindeutet, dass ein Kriterium von der Mehrheit der Personen als ähnlich wichtig empfunden wird.

Betrachtet man die Verteilung der Datenpunkte in Figure 20 fällt auf, dass bei beiden Kriterien die Punkte relativ gleichmäßig über den Boxplot verteilt sind.

Der Median unterscheidet sich bei den beiden Kriterien deutlich, das Gewicht weist einen Median von 75 % auf während die Polsterung im Median nur mit 25 % bewertet wurde.

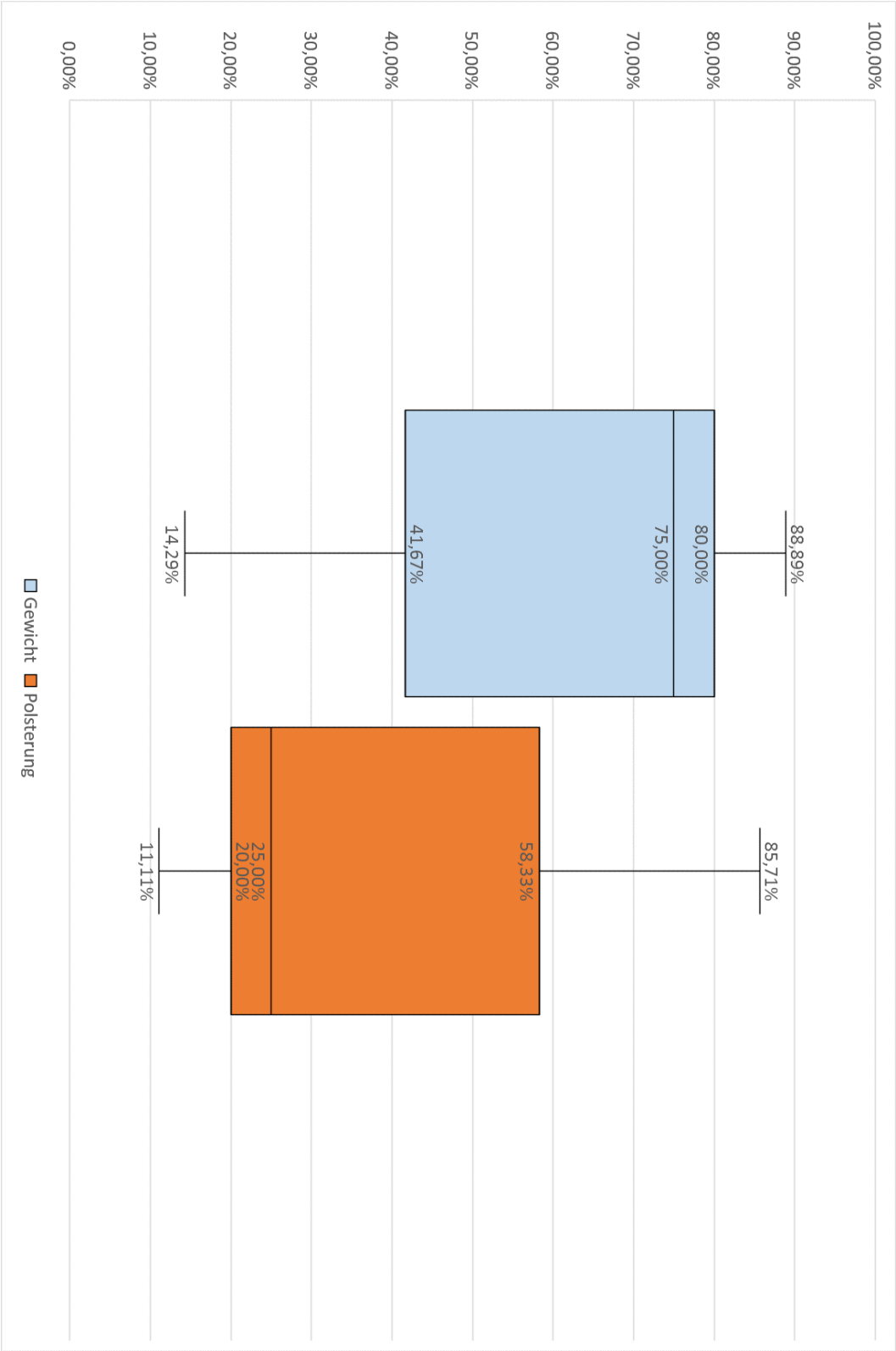


Figure 19: Boxplot zu den Unterkriterien des Tragekomforts

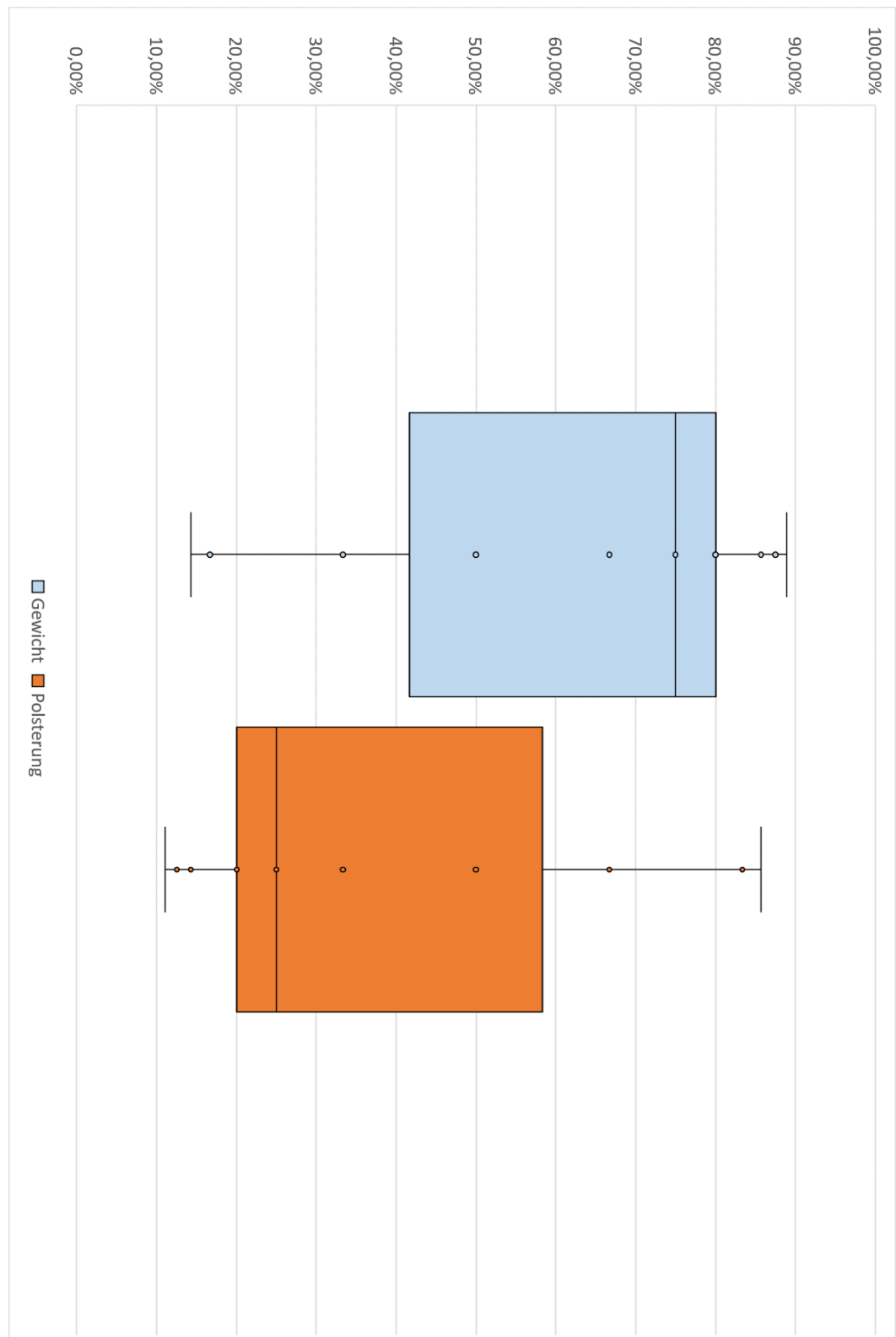


Figure 20: Boxplot mit Datenpunkten zu den Unterkriterien des Tragekomforts

Das Boxplot Diagramm zu den Unterkriterien der Qualität der Immersion wird in Figure 21 dargestellt, Figure 22 zeigt das Äquivalent mit Datenpunkten.

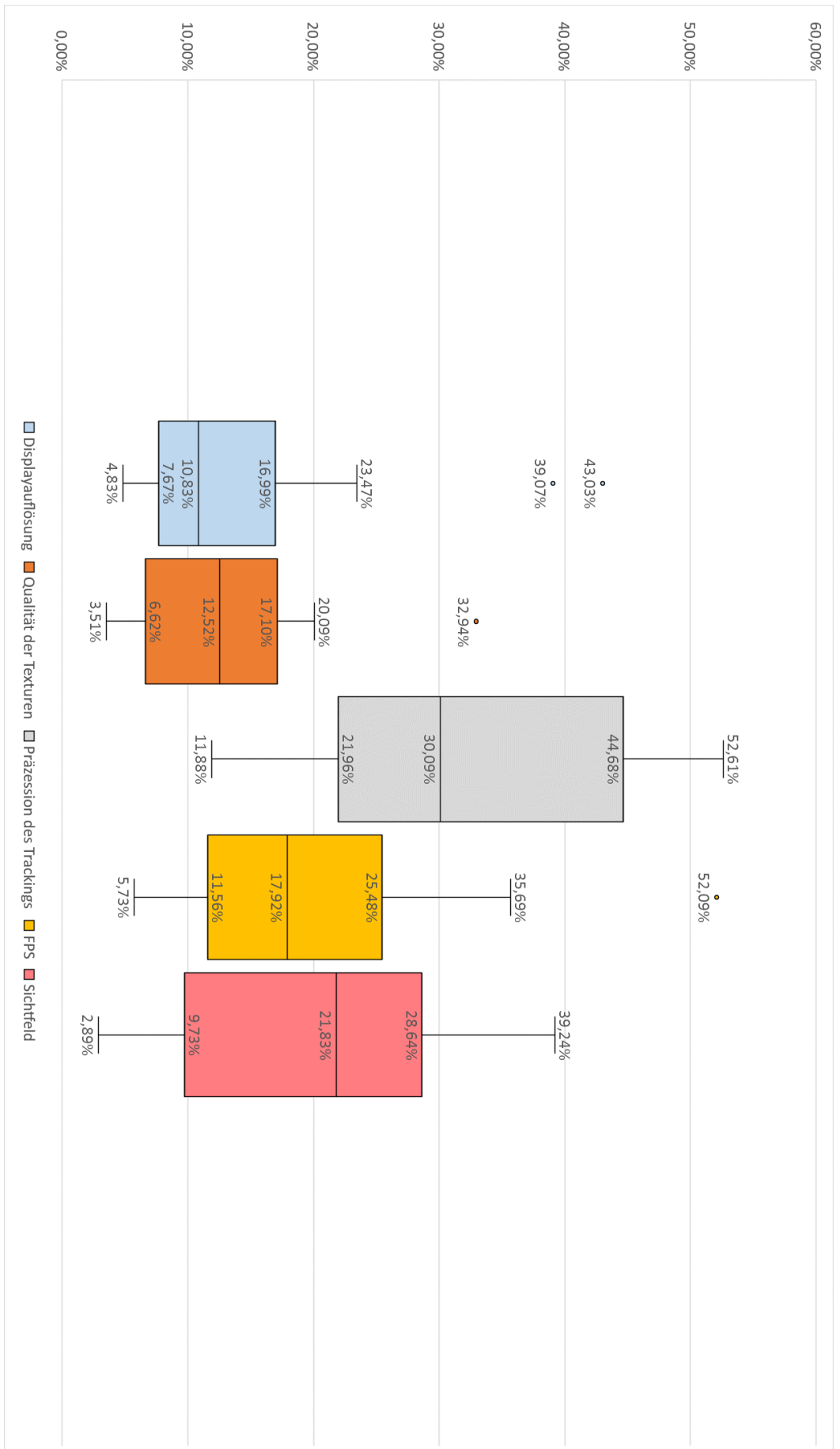


Figure 21: Boxplot der Unterkriterien der Qualität der Immersion

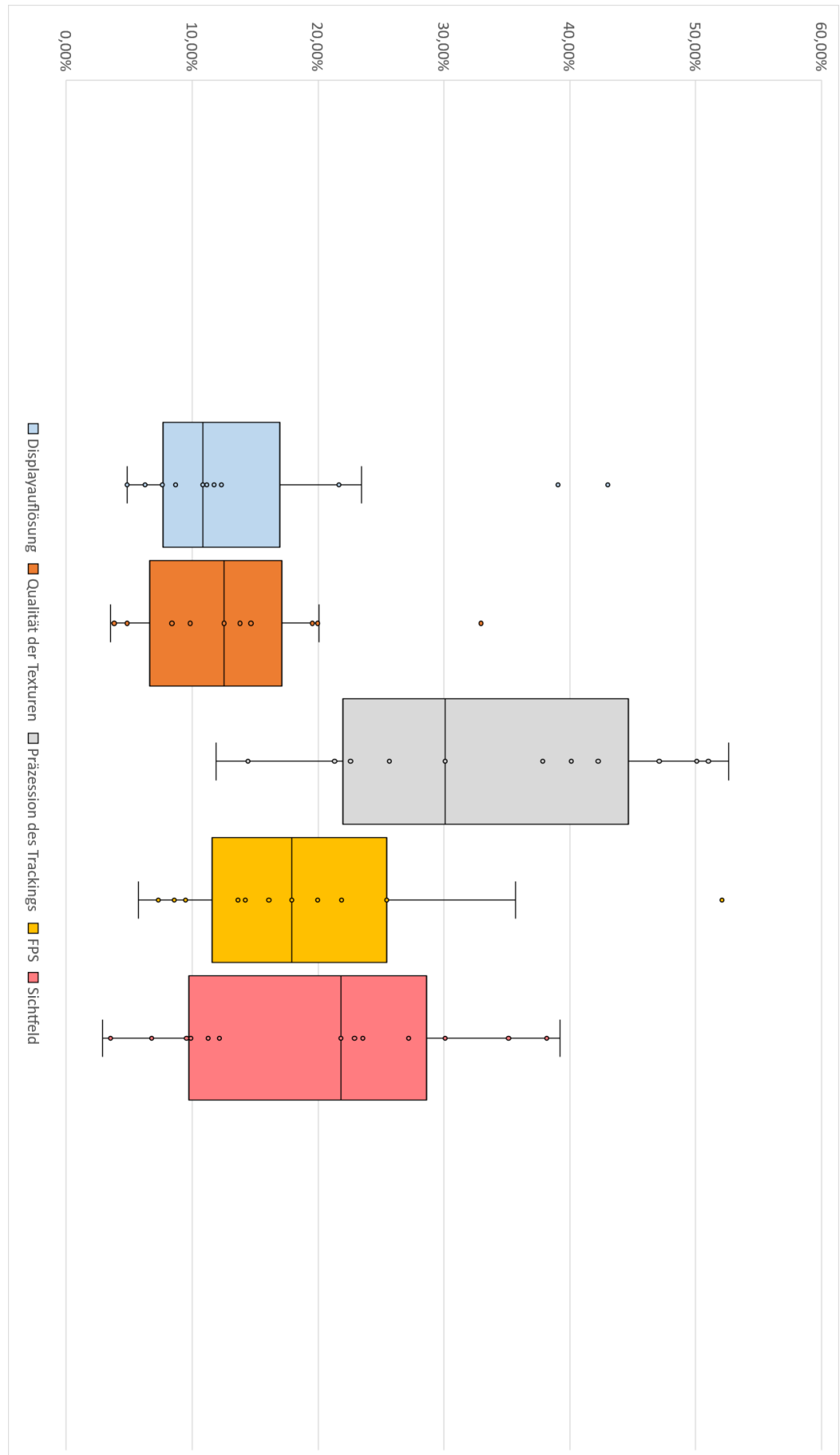


Figure 22: Boxplot mit Datenpunkten der Unterkriterien der Qualität der Immersion

Bei den Unterkriterien der Qualität der Immersion fällt auf, dass sie im Gegensatz zu den Unterkriterien des Tragekomforts, über eine wesentlich geringere Spannweite verfügen. Das Kriterium Präzision des Trackings weist eine relativ große Spannweite auf, sie beläuft sich auf 40,73 Prozentpunkte, während das Kriterium Qualität der grafischen Texturen gerade mal eine Spannweite von 29,43 Prozentpunkte besitzt. Folglich waren sich die befragten Personen insgesamt bei der Beurteilung stärker einig, im Gegensatz zu den Kriterien des Tragekomforts.

Das Kriterium Qualität der grafischen Texturen besitzt einen Ausreißer bei 32,94 %, das Gleiche gilt für das Kriterium Displayauflösung und FPS. Displayauflösung verfügt über zwei Ausreißer bei 43,03 % und 39,07 %, FPS hat einen Ausreißer bei 52,09 %. Auf Grund der Ausreißer verfügen die beiden Kriterien über eine große Spannweite, für die Displayauflösung beträgt sie 38,20 % und für FPS ist sie am Größten mit 46,37 %. Die Spannweite des Kriteriums Sichtfeld befindet sich mit 36,35 % im Mittelfeld zu den Spannweiten der anderen Kriterien.

Betrachtet man die Verteilung der Datenpunkte, weisen die Kriterien Displayauflösung und Sichtfeld Besonderheiten auf. Beim Kriterium Displayauflösung findet eine Konzentration der Punkte im unteren Bereich des Boxplot bis knapp über den Median statt. Dadurch lässt sich darauf schließen, dass die Mehrheit der Testpersonen eine Gewichtung von ca. 10 % als richtig erachten, während nur ein geringerer Anteil eine höhere Gewichtung als richtig erachtet, was wiederum auf die Ausreißer zurückgeführt werden kann. Beim Kriterium Sichtfeld findet eine Konzentration der Punkte sowohl im Bereich vom Median bis hin zum Maximum als auch im Bereich des 1. Quantil bis hin zum Minimum statt. Folglich teilen sie die befragten Personen bei diesem Kriterium in zwei Lager auf, ein Lager sieht eine Gewichtung beim bzw. oberhalb des Median von 21,83 % während dem anderen Lager eine Gewichtung rund um das 1. Quantil bei 9,37 % als passend erachtet.

Betrachtet man die einzelnen Mediane der Kriterien, so ergeben sich drei Gruppen. Gruppe eins beinhaltet die Kriterien Displayauflösung und Qualität der Texturen mit einem Median bei ca. 11 %. Gruppe zwei besteht aus den Kriterien FPS und Displayauflösung mit einem Median von ca. 20 %. Das Kriterium Präzision des Trackings gehört mit einem Median von ca. 30 % zu Gruppe drei.

Figure 23 und Figure 24 zeigen die Boxplot Diagramme zu den Unterkriterien der Benutzerfreundlichkeit.

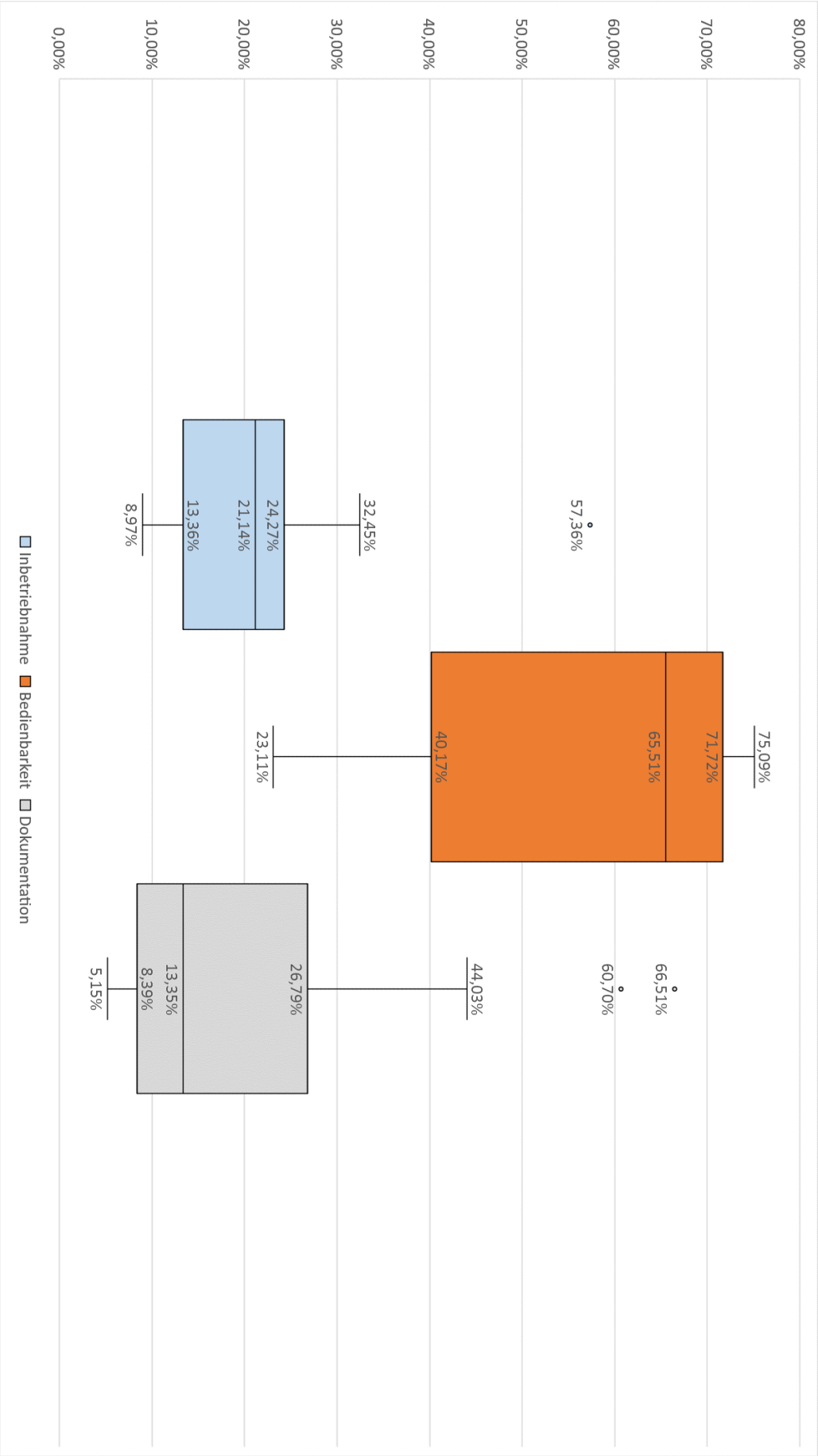


Figure 23: Boxplot der Unterkriterien der Benutzerfreundlichkeit

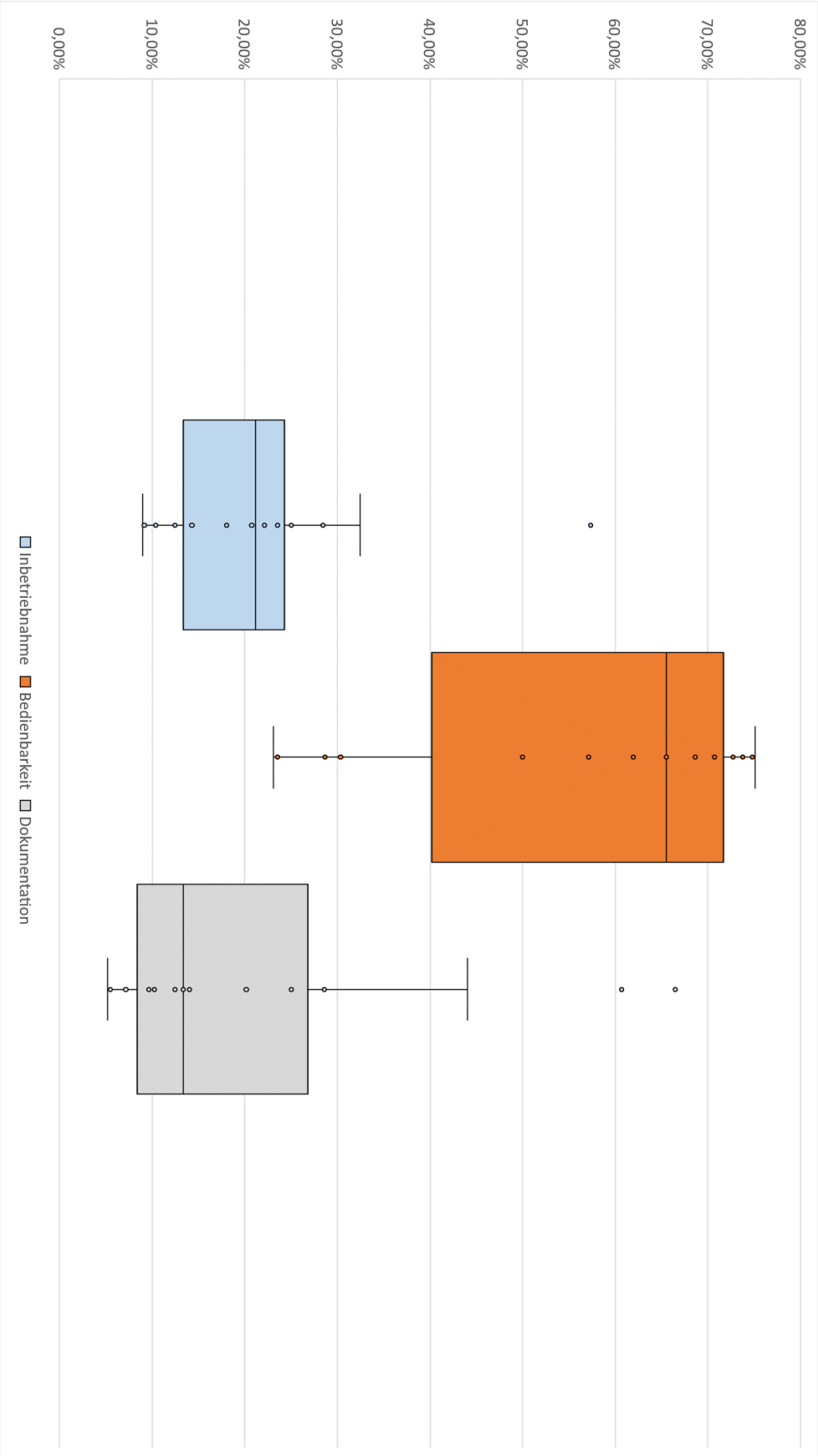


Figure 24: Boxplot mit Datenpunkten der Unterkriterien der Benutzerfreundlichkeit

Das Kriterium Inbetriebnahme verfügt, im direkten Vergleich mit den anderen Kriterien dieser Tabelle, über eine geringe Spannweite von 48,40 % Prozentpunkten. Außerdem besitzt es einen Ausreißer bei 57,36 %, welcher zu Größe der Spannweite beiträgt. Hier hat eine Person das Kriterium mehr als doppelt so wichtig beurteilt als es der Rest der Personen im Median von 21,14 % getan hat.

Beim Kriterium Bedienbarkeit zeigt sich eine große Spannweite von 51,98 %. Daraus folgt, dass es hier keine Ausreißer gibt. Des Weiteren wird die Bedienbarkeit im Median mit 65,61 % als das mit Abstand wichtigste Unterkriterium der Benutzerfreundlichkeit gesehen. Die Datenpunkte konzentrieren sich innerhalb der oberen zwei Drittel des Boxplot und nur eine geringe Anzahl der Befragten gewichtet das Kriterium mit weniger als 50 %.

Der Boxplot zum Kriterium Dokumentation hat die größte Spannweite von 61,36 % und zeigt eine Ausreißergruppe oberhalb von 60 %, was im Zusammenhang mit der Konzentration der Datenpunkte im Bereich vom Minimum bis zum Median in Zusammenhang steht. Die Ausreißergruppe hat das Kriterium mehr als dreimal so wichtig bewertet, als es der Rest der Personen im Median getan hat. Der Median ist mit 13,35 % der niedrigste innerhalb der Unterkriterien der Benutzerfreundlichkeit. Das lässt darauf schließen, dass eine umfangreiche Dokumentation von der Mehrheit der befragten Personen als nicht sehr wichtig angesehen wird. Vermutlich nähern sich diese Personen mit einer Art „trial and error“ Strategie neuen Produkten an, anstatt im Voraus Dokumentation zu lesen.

Figure 25 und Figure 26 zeigt die Boxplot Diagramme aus dem AHP zu den Hauptkriterien.

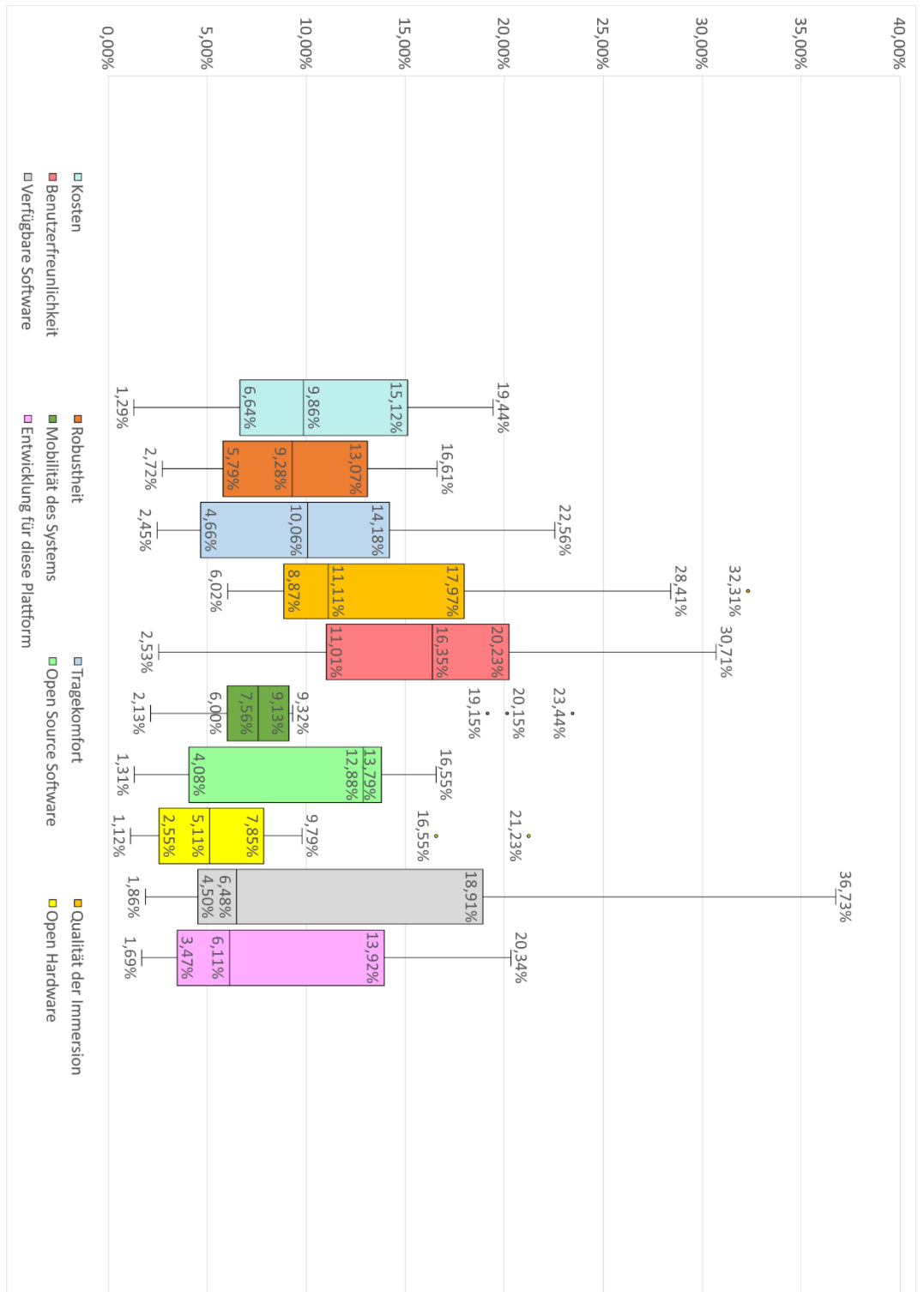


Figure 25: Boxplot der Hauptkriterien mit dem AHP

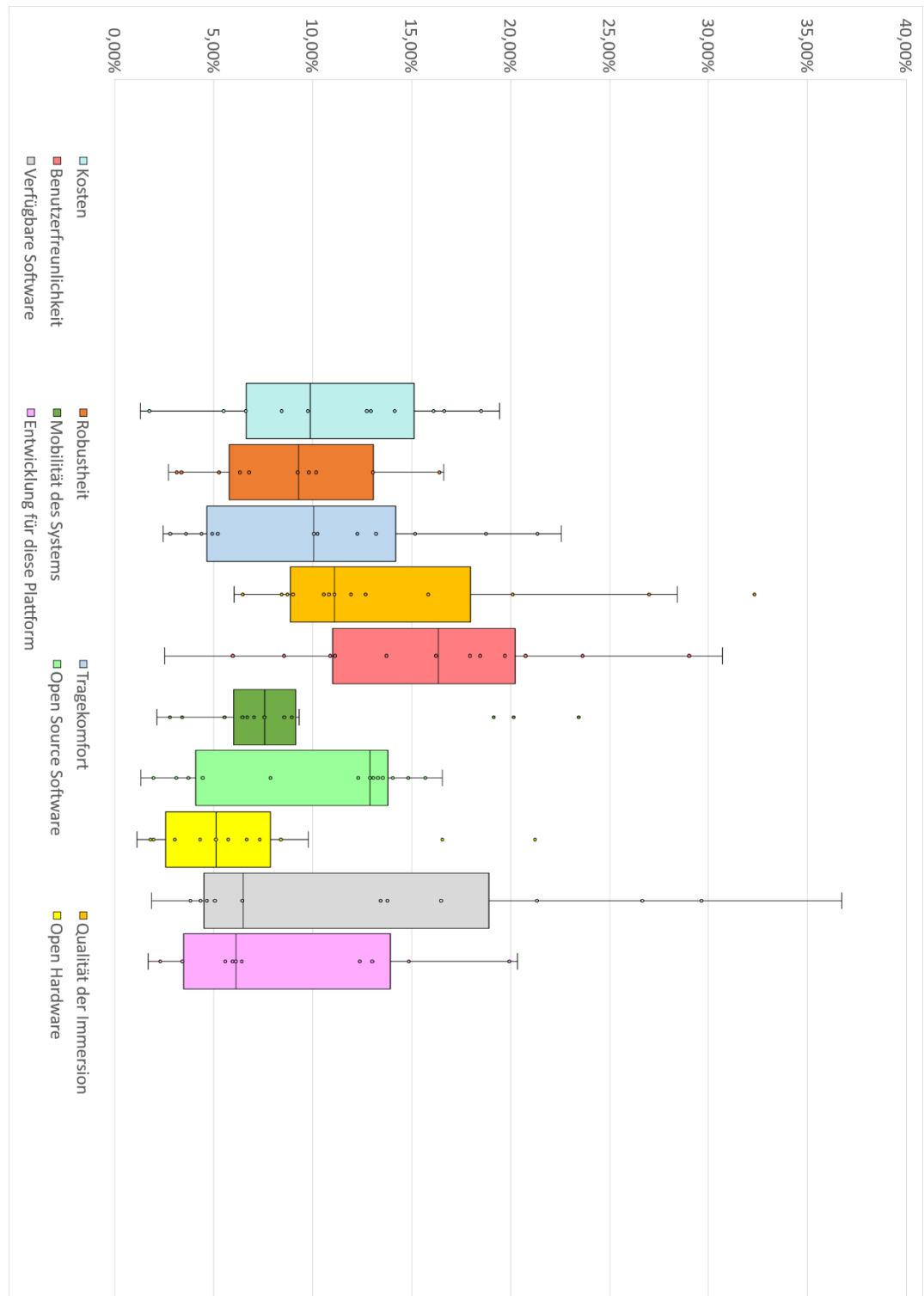


Figure 26: Boxplot mit Datenpunkten der Hauptkriterien mit dem AHP

Betrachtet man die Spannweite der Kriterien in diesem Diagramm, fällt auf, dass sie im Mittel kleiner sind als die Spannweiten aus den anderen Diagrammen. Der Mittelwert der Spannweiten für die Hauptkriterien beträgt 21,68 %, für die Unterkriterien der Benutzerfreundlichkeit sind es 53,91 %, für die Unterkriterien der Qualität der Immersion 38,22 % und für die Unterkriterien des Tragekomforts

74,60 %. Folglich herrschte bei der Gewichtung der Hauptkriterien eine deutlich höhere Einigkeit, als das bei der Gewichtung der Unterkriterien der Fall ist.

Bei den Spannweiten der einzelnen Hauptkriterien fällt das Kriterium verfügbare Software mit einer Spannweite von 34,87 % ins Auge. Auch das Kriterium Benutzerfreundlichkeit hat eine große Spannweite mit 28,18 %. Zur Gruppe der Kriterien mit kleiner Spannweite gehören das Kriterium Robustheit mit 13,89 % und das Kriterium Open Source Software mit 15,23 %. Alle anderen Kriterien verfügen über eine mittlere Spannweite von ca. 20 %.

Betrachtet man den Median der einzelnen Kriterien ergeben sich drei Gruppen. Gruppe eins enthält die Kriterien mit dem niedrigsten Median. Diese sind Open Hardware mit 5,11 %, Entwicklung für diese Plattform mit 6,11 %, verfügbare Software mit 6,48 % und Mobilität des Systems mit 7,56 %. Gruppe zwei enthält Kriterien mit mittleren Median. Dazu gehören, Kosten mit 9,86 %, Robustheit mit 9,28 %, Tragekomfort mit 10,06 % und Qualität der Immersion mit 11,11 %. Die Gruppe der Kriterien mit großem Median besteht aus dem Kriterium Open Source Software mit 12,88 % und Benutzerfreundlichkeit mit 16,35 %.

Im nächsten Schritt des AHP, der die möglichen Entscheidungsalternativen bewertet, wird der Median als Zahlenwert verwendet, der die Gewichtung festlegt. Gegenüber dem Durchschnittswert bietet der Median den Vorteil, dass er robust gegen Ausreißer ist. Somit wird eine Verfälschung der Gewichtung auf Grund der zahlreichen Ausreißer verhindert.

8.4.5 Vergleich: Gewichtung der Kriterien mit der Nutzwertanalyse gegen Gewichtung der Kriterien mit dem AHP

Dieser Abschnitt vergleicht die Ergebnisse aus der Gewichtung der Entscheidungskriterien der Verfahren Nutzwertanalyse und Analytical Hierarchy Process. Figure 27 zeigt zu jedem Kriterium jeweils den Wert des Median aus dem AHP und den aufsummierten, prozentualen Wert aus der Nutzwertanalyse. Die Werte der Nutzwertanalyse werden in blau dargestellt, die Werte aus dem AHP in orange.

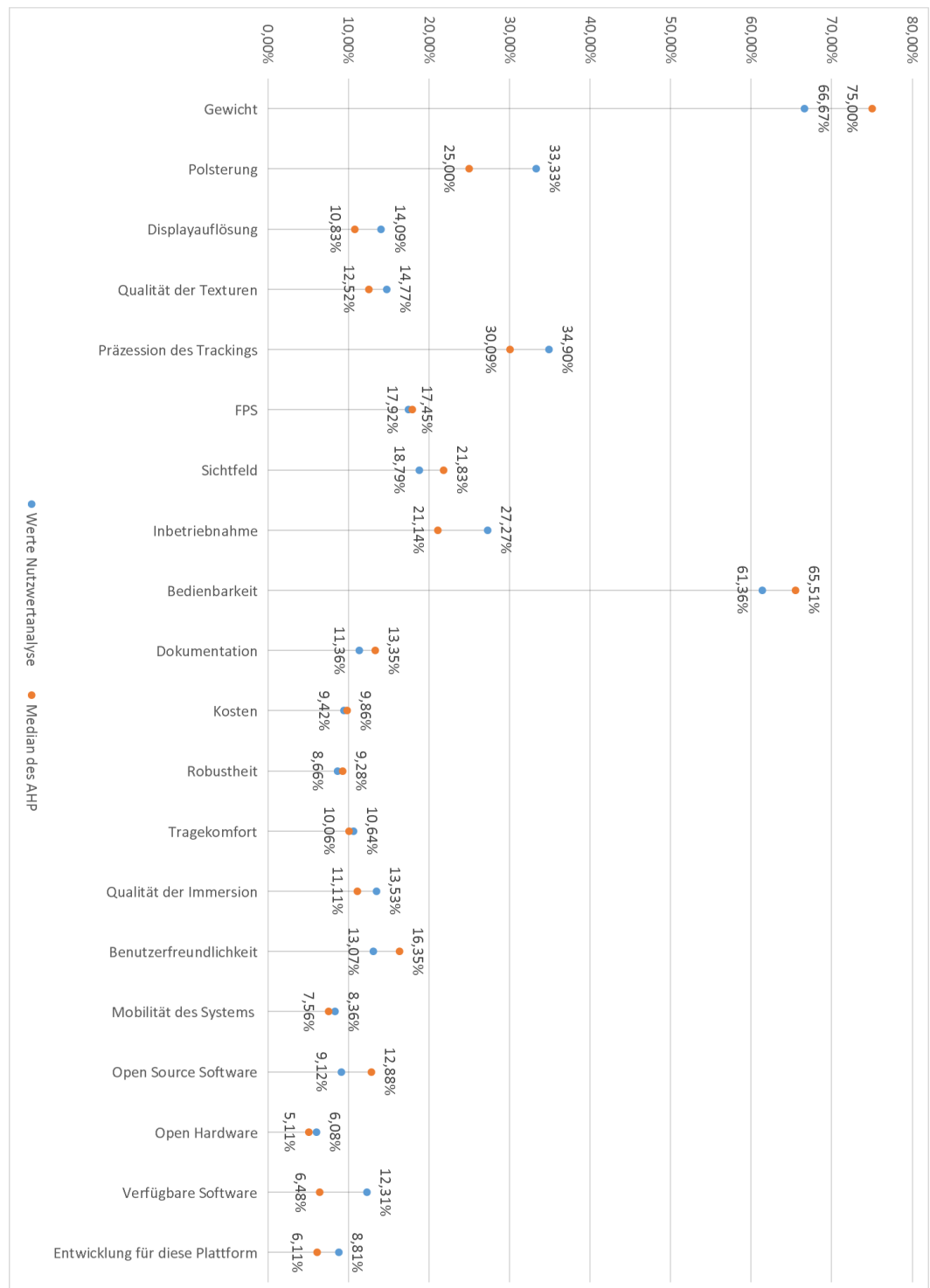


Figure 27: Vergleich Gewichtung Nutzwertanalyse zu AHP

Beim Kriterium Gewicht beträgt der Wert aus der Nutzwertanalyse 66,67 % und der Wert aus dem AHP 75,00 %. Dadurch weicht der Wert aus dem AHP um 12,50 % vom Wert der Nutzwertanalyse ab. Für das Kriterium Polsterung ergibt sich eine Abweichung von 33,33 % der Nutzwertanalyse gegenüber dem AHP. Das Kriterium der Displayauflösung besitzt eine Abweichung von 30,18 %. Bei den Kriterien Qualität der Texturen und Präzision des Trackings beträgt diese Abweichung

17,90 % bzw. 16,00 %. Hingegen ergibt sich beim Kriterium FPS keine nennenswerte Abweichung der Verfahren. Das gleiche gilt für die Kriterien Kosten, Robustheit, Tragekomfort und Bedienbarkeit. Hier liegt die Abweichung bei max. 7 %. Beim Kriterium Sichtfeld weicht der AHP 16,14 % gegenüber der Nutzwertanalyse ab. Beim Inbetriebnahme beträgt die Abweichung 29,03 % von der Nutzwertanalyse gegenüber des AHP. Für das Kriterium Dokumentation weicht der AHP um 17,46 % von der Nutzwertanalyse ab. Beim Kriterium Qualität der Immersion zeigt sie eine Abweichung von 21,80 % der Nutzwertanalyse gegenüber des AHP. Bei der Benutzerfreundlichkeit dreht sich die Abweichung, hier weicht der AHP gegenüber der Nutzwertanalyse um 25,13 % ab. Beim Kriterium Mobilität des Systems ergibt sich eine relativ geringe Abweichung von 10,57 % der Nutzwertanalyse gegenüber des AHP. Im Gegensatz dazu weicht beim Kriterium Open Source Software der AHP um starke 41,30 % von der Nutzwertanalyse ab. In Bezug auf das Kriterium Open Hardware liegt eine Abweichung der Nutzwertanalyse gegenüber des AHP von 19,04 % vor. Beim Kriterium Verfügbare Software ergibt sich die stärkste Abweichung des Vergleiches: Sie beträgt 90 %. Somit empfanden die befragten Personen das Kriterium bei der Befragung mit dem AHP als beinahe nur halb so wichtig als bei der Befragung mit der Nutzwertanalyse. Hinsichtlich des Kriteriums Entwicklung für die Plattform ergab sich eine Abweichung von 44,25 % der Nutzwertanalyse gegenüber des AHP. Wie die oben genannten Abweichungen entstehen, lässt sich nur schwer nachvollziehen. Theoretisch müsste sich die befragte Person in beiden Verfahren genau gleich entschieden haben. Eine mögliche Ursache könnte jedoch die Verwendung verschiedener Entscheidungsarten sein. Wie bereits erwähnt, wurde bei der Nutzwertanalyse eine binäre Entscheidung angewendet, die keine Aussage über die Stärke der Präferenz des einen Kriteriums gegenüber dem anderen Kriterium zulässt. Im Gegenteil dazu verwendet der AHP eine präzise Skala, die solche Aussagen zulässt. Somit können die Entscheidungen innerhalb der AHP als differenzierter als die der Nutzwertanalyse betrachtet werden.

Im nächsten Abschnitt werden die Entscheidungsalternativen anhand der gewichteten Kriterien bewertet.

8.5 Bewertung der Entscheidungsalternativen

In diesem Abschnitt werden die drei Head-Mounted Displays Rift von Oculus, Zeiss VR One und Dark Shader von ZAAK ZURICH BERLIN – Cardboard Kits Europe anhand der gewichteten Kriterien bewertet.

Für die Bewertung der Alternativen anhand der Kriterien kommt eine 10er Skala zum Einsatz, siehe Figure 28.

Punkte	Bedeutung
0	Kriterium ist nicht erfüllt/nicht nützlich
1–3	Kriterium ist unzureichend und nur mit erheblichen Mängeln erfüllt/nur bedingt nützlich
4–6	Kriterium ist hinreichend, aber mit Mängeln erfüllt/nützlich
7–9	Kriterium ist in gutem Umfang erfüllt/sehr nützlich
10	Kriterium ist in sehr gutem Umfang erfüllt/außerordentlich nützlich

Figure 28: Punktwertekorridore auf einer 10er Skala [29] S. 17

Bei der Bewertung des Gewichts ist die Dark Shader mit Abstand die leichteste Brille bei einem Gewicht von 68 Gramm, während die Oculus Rift 453g und die Zeiss VR One 413g wiegt. Dabei ist jedoch noch in Betracht zu ziehen, dass bei der Zeiss VR One und der Dark Shader noch das Gewicht des betriebsnotwendigen Smartphones zu ergänzen ist. Somit erhält die Dark Shader 10 Punkte, die Oculus Rift 6 Punkte und die Zeiss VR One 5 Punkte. Letztere bildet mit Smartphone das schwerste Gesamtsystem.

Bewertet man die Polsterung, fällt auf, dass die Dark Shader über keinerlei Polsterung verfügt und somit 0 Punkte erhält. Bei Oculus Rift und Zeiss VR One kommt eine hochwertige Polsterung zum Einsatz die ein angenehmes Tragegefühl vermittelt. Beide erhalten daher 10 Punkte.

In Sachen Displayauflösung sind die Ergebnisse der Dark Shader und Zeiss VR One abhängig vom verwendeten Smartphone. Allerdings bieten die meisten Smartphones aktuell im Durchschnitt eine genauso hohe Auflösung, wie die 1920 mal 1080 Pixel der Oculus Rift. Somit erhalten alle drei HMDs dieselbe Punktzahl von 7 Punkten. Diese wurde bewusst unterhalb der maximal Punktzahl angesetzt, da für ein Konsumieren von VR Inhalten ohne das Erkennen von einzelnen Bildpunkten eine höhere Auflösung nötig ist.

Für das Kriterium Qualität der grafischen Texturen sind die HMDs von der ihnen zur Verfügung stehenden Prozessor- und Grafikleistung abhängig. Dieses ist für die Oculus Rift wesentlich höher als für die anderen beiden HMDs, da sie die Leistung eines Desktop PC abgreift, während die anderen beiden abhängig vom Smartphone sind. Folglich wird die Oculus Rift mit zehn Punkten bewertet und die anderen beiden mit fünf Punkten, da die Leistung der Smartphones dennoch für viele Anwendungen ausreicht.

Bei der Bewertung der Präzision des Trackings ergibt sich eine weitere Überlegenheit der Oculus Rift gegenüber der beiden anderen HMDs. Grund dafür ist, dass die Oculus Rift neben den integrierten Sensoren, über welche auch die Smartphones und damit die anderen beiden HMDs verfügen, eine Infrarotkamera nutzt, um die Position und Orientierung zu bestimmen. Somit wird eine höhere Präzision erreicht, was einer Punktzahl von zehn Punkten entspricht, wobei die beiden anderen HMDs fünf Punkte erhalten, da die Smartphonesensoren ein noch ausreichend präzises Tracking erlauben.

Für das Kriterium FPS ergibt sich eine weitere Abhängigkeit der beiden Cardboard HMDs vom der Rechenleistung des Smartphones. Darüber hinaus werden die FPS auch von der Software limitiert. Die Oculus Rift unterstützt eine Bildwiederholungsrate von 75 Hz, während die Mehrheit der aktuellen Smartphones nicht mehr als 60 Bildern pro Sekunde zur Verfügung stellen. Optimal wäre eine Bildwiederholungsrate über 90 Bildern pro Sekunde. Folglich erhält die Oculus Rift sechs Punkte und die beiden anderen HMDs vier Punkte.

Die Oculus Rift bietet genauso wie die Zeiss VR One ein Sichtfeld von 100 Grad, die Dark Shader ein Sichtfeld von 90 Grad. Da das menschliche Auge ein horizontales Sichtfeld von bis zu 200 Grad besitzt, erhalten Oculus Rift und Zeiss VR One acht Punkte und Dark Shader sieben Punkte.

Die Inbetriebnahme gestaltet sich für die beiden Google Cardboard HMDs leicht und geht zügig vonstatten. Smartphone einlegen, Google Cardboard App installieren, QR-Code des HMD einscannen und schon sind alle nötigen Einstellungen abgeschlossen. Bei der Oculus Rift ist der Prozess aufwendiger, da zuerst das HMD und dann die Infrarot Kamera mit dem PC verbunden werden müssen, um anschließend die Treiber installieren zu können. Außerdem müssen alle nötigen Einstellungen

vom Nutzer selbst durchgeführt werden. Folglich erhalten die Cardboard HMDs zehn und die Oculus Rift vier Punkte.

Die Bedienbarkeit fällt bei allen drei HMDs leicht. Die Produkte und die entsprechende Software sind intuitiv zu bedienen, wobei die Oculus Rift Software etwas mehr Einarbeitungszeit benötigt, was wiederum dem größeren Umfang geschuldet ist. In Punkten ausgedrückt erhalten die Cardboard HMDs zehn Punkte und die Oculus Rift acht.

In Sachen Dokumentation liegt die Dark Shader weit vor den beiden anderen HMDs. Grund dafür ist, dass sie ein Nachbau des ursprünglichen Google Cardboard ist, bei welchem es sich um ein Open Source Projekt handelt und somit sämtliche Information öffentlich sind. Bei der Oculus Rift finden sich auch viele Information auf der Herstellerseite und im Internet, allerdings nicht im selben Umfang wie bei der Dark Shader. Am geringsten ist der Umfang bei der Zeiss VR One. Die einzige Informationsquelle ist hier die Herstellerseite, welche sehr unübersichtlich gestaltet ist. So dauerte es im konkreten Fall beispielsweise 30 Minuten, um herauszufinden, welche Smartphones von der Zeiss VR One unterstützt werden. Die Zeiss VR One erhält einen Punkt, die Oculus Rift vier und die Dark Shader zehn.

Beim Kriterium Kosten liegt die Dark Shader ebenfalls weit vor den beiden anderen HMDs. Mit einem Kaufpreis von 10 € weist sie mit Abstand das beste PreisLeistungsverhältnis auf. Im Gegensatz dazu hat die Zeiss VR One das schlechteste. Sie kostet mit 100 € das zehnfache der Dark Shader und bietet keine zusätzlichen Features, sondern nur hochwertigere Materialien, welche aber keinesfalls den zehnfachen Preis rechtfertigen. Das Oculus Rift DK 2 kostet 350 € und bietet damit für den gebotenen Preis eine Leistung, die in einem vernünftigen Rahmen liegt. Dabei muss allerdings bedacht werden, dass für die Dark Shader und die Zeiss VR One zusätzlich ein Smartphone benötigt wird. Für eine Bildungseinrichtung bietet es sich an, die Smartphones der Studenten zu nutzen, da in der Regel entsprechende Smartphones bereits vorhanden sind und somit der Anschaffungspreis wie oben erläutert 10 € bzw. 100 € beträgt. Die Dark Shader erhält zehn Punkte, die Oculus Rift fünf und die Zeiss VR One null.

Was die Robustheit betrifft, kann die Dark Shader nicht mit den beiden anderen HMDs mithalten. Sie besteht nur aus Pappkarton, während die anderen beiden aus

stabilen Kunststoff hergestellt sind. Wie langlebig der Pappkarton tatsächlich gegenüber dem Kunststoff ist, müsste mit Hilfe eines Langzeittestes festgestellt werden. Aus aktueller Sicht ergibt sich eine Punktzahl von drei für die Dark Shader und zehn für die anderen beiden HMDs.

Die Bewertungen für die Hauptkriterien Tragekomfort, Benutzerfreundlichkeit und Qualität der Immersion ergibt sich aus den Bewertungen deren Unterkriterien.

Beim Kriterium Mobilität des Systems schneidet die Oculus Rift am schlechtesten ab, da sie stets mit einem leistungsstarken PC oder Notebook verbunden werden muss und erhält somit null Punkte. Für die beiden anderen wird bekanntlich nur ein Smartphone benötigt, was zu einer hohen Mobilität und damit zehn Punkten führt.

Das Kriterium Open Source Software muss für alle drei HMDs mit null Punkten bewertet werden, da weder Oculus noch Google den Quellcode, der zum Betreiben der HMD genutzt wird, offenlegt.

In Sachen Open Hardware verhält sich die Dark Shader vorbildlich. Bei Google können alle technischen Zeichnungen zu jeder Komponente heruntergeladen werden, was zehn Punkte rechtfertigt. Bei Zeiss VR One ist es nicht möglich umfangreiche, technische Informationen zur Hardware, wie beispielsweise den Linsen, ausfindig zu machen, womit sich null Punkte ergeben. Bei der Oculus Rift finden sich keine tiefgehenden, offiziellen Informationen zur Hardware auf der Herstellerseite. Jedoch können über andere Quellen, wie beispielsweise iFixit (siehe Kapitel 4), umfangreiche Informationen zur Hardware beschafft werden. Sie erlangt daher noch drei Punkte.

Leider ist es für das Kriterium verfügbare Software nicht möglich offizielle Zahlen aus den Google Playstore bzw. dem Oculus Store einzusehen. Allerdings ist anzunehmen, dass mehr Software für die Cardboard HMDs vorhanden ist, als für die Oculus Rift. Grund dafür ist der geringere Preis, die frühere Verfügbarkeit und folglich die höhere Verbreitung der Geräte. In Punkten ausgedrückt erhält die Oculus Rift vier Punkte und die anderen beiden HMDs sieben.

Beim Kriterium Entwicklung für die Plattform schneiden alle drei HMDs ähnlich gut ab, da sowohl von Google als auch von Oculus viele Informationen für die Entwicklung mit den Grafik Engines als auch mit der gerätespezifischen SDK zur Verfügung gestellt werden. Allerdings kann die Entwicklung für Google Cardboard als

etwas weniger aufwendig gesehen werden, da diese mit Hilfe von Android Studio Apps entwickelt werden und in Sachen Appentwicklung viele Developer schon Erfahrung mit Android Studio haben. Somit erhalten die Cardboard HMDs acht Punkte und die Oculus Rift sechs.

8.5.1 Gewichtete Bewertung der Entscheidungsalternativen mit der Nutzwertanalyse

Dieser Abschnitt beinhaltet die Gewichtung der Bewertungen aus dem vorherigen Abschnitt der Nutzwertanalyse. Tabelle 5 zeigt die Bewertungen und die anschließende Multiplikation mit den Gewichten. Am Ende der Tabelle befindet sich die Summe über die gewichtete Bewertung für jede der drei HMDs.

			Rift von Oculus		Zeiss VR One		Dark Shader	
	Kriterien	Gewichtung	Bewertung	gew. Bewertung	Bewertung	gew. Bewertung	Bewertung	gew. Bewertung
Tragekomfort	Gewicht	66,67%	6,00	4,00	5,00	3,33	10,00	6,67
	Polsterung	33,33%	10,00	3,33	10,00	3,33	0,00	0,00
			Summe	7,33		6,67		6,67
Qualität der Immersion	Displayauflösung	14,09%	7,00	0,99	7,00	0,99	7,00	0,99
	Qualität der Texturen	14,77%	10,00	1,48	5,00	0,74	5,00	0,74
	Präzession des Trackings	34,90%	10,00	3,49	5,00	1,74	5,00	1,74
	FPS	17,45%	6,00	1,05	4,00	0,70	4,00	0,70
	Sichtfeld	18,79%	8,00	1,50	8,00	1,50	7,00	1,32
			Summe	8,50		5,67		5,48
Benutzerfreundlichkeit	Inbetriebnahme	27,27%	4,00	1,09	10,00	2,73	10,00	2,73
	Bedienbarkeit	61,36%	8,00	4,91	10,00	6,14	10,00	6,14
	Dokumentation	11,36%	4,00	0,45	1,00	0,11	10,00	1,14
			Summe	6,45		8,98		10,00
Hauptkriterien	Kosten	9,42%	5,00	0,47	0,00	0,00	10,00	0,94
	Robustheit	8,66%	10,00	0,87	10,00	0,87	3,00	0,26
	Tragekomfort	10,64%	7,33	0,78	6,67	0,71	6,67	0,71
	Qualität der Immersion	13,53%	8,50	1,15	5,67	0,77	5,48	0,74
	Benutzerfreundlichkeit	13,07%	6,45	0,84	8,98	1,17	10,00	1,31
	Mobilität des Systems	8,36%	0,00	0,00	10,00	0,84	10,00	0,84
	Open Source Software	9,12%	0,00	0,00	0,00	0,00	0,00	0,00
	Open Hardware	6,08%	3,00	0,18	0,00	0,00	10,00	0,61
	Verfügbare Software	12,31%	4,00	0,49	7,00	0,86	7,00	0,86
	Entwicklung für diese Plattform	8,81%	6,00	0,53	8,00	0,71	8,00	0,71
	Summe gew. Bewertungen			49,90		48,55		51,27

Tabelle 5: Gewichtete Bewertung und Ergebnisse aus der Nutzwertanalyse

Folglich handelt es sich bei der HMD Dark Shader mit einer Summe von 51,27 Punkte um die für die Fragestellung am besten geeignete Alternative. Auf Platz zwei befindet sich die Oculus Rift mit einer Summe von 49,90 Punkten. An letzter Stelle steht die Zeiss VR One mit einer Summe von 48,55 Punkten.

8.5.2 Gewichtete Bewertung der Entscheidungsalternativen mit dem AHP

Dieser Abschnitt beinhaltet die Gewichtung der Bewertungen aus Abschnitt 8.5 mit dem AHP. Er zeigt tabellarisch die Bewertungen und die anschließende Multiplikation mit den Gewichten. Am Ende der Tabelle befindet sich die Summe über die gewichtete Bewertung für jede der drei HMDs.

			Rift von Oculus		Zeiss VR One		Dark Shader	
Kriterien	Gewichtung	Bewertung	gew. Bewertung	Bewertung	gew. Bewertung	Bewertung	gew. Bewertung	
Trage- komfort	Gewicht	75,00%	6,00	4,50	5,00	3,75	10,00	
	Polsterung	25,00%	10,00	2,50	10,00	2,50	0,00	
		Summe		7,00		6,25		
Qualität der Immersion	Displayauflösung	10,83%	7,00	0,76	7,00	0,76	7,00	
	Qualität der Texturen	12,52%	10,00	1,25	5,00	0,63	5,00	
	Präzession des Trackings	30,09%	10,00	3,01	5,00	1,50	5,00	
	FPS	17,92%	6,00	1,07	4,00	0,72	4,00	
	Sichtfeld	21,83%	8,00	1,75	8,00	1,75	7,00	
		Summe		7,84		5,35		
Benutzer- freund- lichkeit	Inbetriebnahme	21,14%	4,00	0,85	10,00	2,11	10,00	
	Bedienbarkeit	65,51%	8,00	5,24	10,00	6,55	10,00	
	Dokumentation	13,35%	4,00	0,53	1,00	0,13	10,00	
		Summe		6,62		8,80		
Haupt- kriterien	Kosten	9,86%	5,00	0,49	0,00	0,00	10,00	
	Robustheit	9,28%	10,00	0,93	10,00	0,93	3,00	
	Tragekomfort	10,06%	7,00	0,70	6,25	0,63	7,50	
	Qualität der Immersion	11,11%	7,84	0,87	5,35	0,59	5,13	
	Benutzerfreundlichkeit	16,35%	6,62	1,08	8,80	1,44	10,00	
	Mobilität des Systems	7,56%	0,00	0,00	10,00	0,76	10,00	
	Open Source Software	12,88%	0,00	0,00	0,00	0,00	0,00	
	Open Hardware	5,11%	3,00	0,15	0,00	0,00	10,00	
	Verfügbare Software	6,48%	4,00	0,26	7,00	0,45	7,00	
	Entwicklung für diese Plattform	6,11%	6,00	0,37	8,00	0,49	8,00	
	Summe gew. Bewertungen			47,78		46,09		

Tabelle 6: Gewichtete Bewertung und Ergebnisse aus dem AHP

In Summe ergeben sich für die Dark Shader 51,70, für die Oculus Rift 47,78 und für die Zeiss VR One 46,09 Punkte. Vergleicht man nun die Ergebnisse aus den beiden Verfahren miteinander, so ergeben sich, abhängig vom Verfahren, keine Unterschiede in der Rangfolge. Der Sieger aus der Nutzwertanalyse ist auch der Sieger aus dem AHP. Allerdings ist das Ergebnis aus dem AHP ein deutlicheres als jenes der Nutzwertanalyse.

8.6 Fazit der Anschaffungsempfehlung

Den Ergebnissen aus AHP und Nutzwertanalyse entsprechend empfiehlt es sich für eine Bildungseinrichtung von den vorgestellten Head-Mounted Displays die Dark Shader von ZAAK ZURICH BERLIN – Cardboard Kits Europe zu verwenden. Sie bietet das Optimum aus der Summe der bewerteten Kriterien.

Allerdings sei an dieser Stelle erwähnt, dass durch die Befragung der Personen das Gewicht für Kosten und Robustheit nur durchschnittlich ausgefallen ist. Für Bildungseinrichtungen erscheint eine höhere Gewichtung sinnvoll. Angenommen man würde die Kosten und die Robustheit überdurchschnittlich gewichten, beispielsweise mit 15 % bis 20 %, würde sich jedoch an der eigentlichen Empfehlung auch nichts ändern. Die Dark Shader wäre weiterhin die beste Alternative, obwohl sie über die geringste Robustheit verfügt. Auf Grund ihres niedrigen Preises kann die niedrige Robustheit kompensiert werden. So wäre eine fünfmalige Anschaffung trotz geringer Robustheit immer noch eine wirtschaftlich bessere Entscheidung, als auf Grund der Robustheit eine andere Brille zu konsumieren.

8.7 Zusammenfassung

In diesem Kapitel wurden die vorgestellten HMDs hinsichtlich einer Anschaffungsempfehlung miteinander verglichen. Dabei wurden die Verfahren Nutzwertanalyse und Analytical Hierarchy Process angewandt. Für die Gewichtung der Kriterien wurden fünfzehn Personen befragt. Die Daten der Personen wurden mit Hilfe von Boxplot Diagrammen visualisiert und interpretiert. Als das für die Fragestellung am besten geeignete HMD hat sich die Dark Shader erwiesen.

9 Konzipierung einer Laborübung zur HMD Einführung und Entwicklung

Dieses Kapitel behandelt die Erstellung einer Laborübung zur Einführung und Entwicklung einer HMD. Zu Beginn wird die Motivation für die Laborübung erläutert und die genutzten Technologien vorgestellt. Anschließend wird die Laborübung an sich vorgestellt und auf die Programmierung eingegangen. Danach erfolgt die Evaluation der Laborübung.

9.1 Motivation für die Laborübung

Dieser Abschnitt stellt die Motivation für die Erstellung der Laborübung da. Ziel ist es, eine Laborübung zum Thema Head-Mounted Displays und Virtual Reality zu erstellen. Der Umfang der Laborübung beträgt zwei Semesterwochenstunden, was drei tatsächlichen Stunden entspricht. Die Laborübung soll die Teilnehmer dazu befähigen den aktuellen Stand der Technik eines HMD und dessen Funktionsweise zu erlernen sowie ihre Einsatzgebiete und Vor- und Nachteile zu kennen. Weiterhin sollen die Laborteilnehmer dazu in der Lage sein, ein HMD zu bedienen. Außerdem soll das Wissen vermittelt werden, wie man das HMD als Ein- und Ausgabegerät für eine Webanwendung nutzen kann. Zusätzlich sollen Teilnehmer die Fähigkeit erwerben, Virtual Reality Inhalte, die mit dem HMD konsumiert werden können, zu programmieren. Im Zusammenhang mit dieser Programmierung sollen die Teilnehmer folgende Punkte erlernen und praktisch umsetzen können:

1. Die Sensorwerte eines HMD auslesen und mit einer Webanwendung verarbeiten
2. Das Ausgeben von Daten und Bildern auf dem HMD-Display, welche von der Webanwendung bereitgestellt werden
3. Durchführung eines bidirektionalen Datenaustausches zwischen der Webanwendung und dem HMD

Bidirektionaler Datenaustausch bedeutet, dass der Nutzer eine Eingabe macht und das HMD in Verbindung mit der Webanwendung die Eingabe verarbeitet. Der Eingabe entsprechend wird eine Ausgabe berechnet, die dem Nutzer anschließend auf dem HMD-Display dargestellt wird.

Neben der Entwicklung der eigentlichen Inhalte der Laborübung, gilt es auch, alle nötigen Dokumente zu erstellen, die für die Durchführung der Laborübung notwendig sind.

9.2 Vorstellung der genutzten Technologien

Dieser Abschnitt stellt die Technologien vor, die genutzt wurden, um die Laborübung zu realisieren. Die Übung wurde so konzipiert, dass sie mit allen drei HMDs, die innerhalb dieser Arbeit vorgestellt wurden, ausgeführt werden kann. Um diese Systemunabhängigkeit zu gewährleisten, wurde softwareseitig der Framework A-Frame von Mozilla verwendet, welcher bereits im Abschnitt 7.2.4 vorgestellt wurde. Um die Sensorwerte der jeweiligen HMD anzeigen zu können, wurde die WebVR API, welche ebenfalls von Mozilla stammt, verwendet. Die WebVR API erfüllt dabei folgenden Aufgaben: Zunächst wird mit Hilfe der Funktion *navigator.getVRDevices()* geprüft, welches HMD mit dem Computer verbunden ist. Die Funktion gibt das Objekt *PositionSensorVRDevice* zurück. Das Objekt *PositionSensorVRDevice* gibt nun die Eigenschaft-Wert Paare zurück, die in Abbildung 29 zu sehen sind.

```
PositionSensorVRDevice { hardwareUnitId: "0x100", deviceId: "0x101",
  deviceName: "Oculus VR HMD (Sensor)" }
```

Abbildung 29: Objekt *PositionSensorVRDevice*

Somit erkennt die API, dass die Rift von Oculus mit dem PC verbunden ist. Gleichzeitig wird noch eine ID für die Oculus ausgegeben. Als nächstes werden mit dem Objekt *VRPositionState* die Eigenschaft-Wert Paare der Sensoren des HMD ausgegeben, die in Abbildung 30 zu sehen sind.

```
VRPositionState { timeStamp: 90117.06471220135, hasPosition: false, position:
  null, linearVelocity: DOMPoint, linearAcceleration: DOMPoint, hasOrientation:
  true, orientation: DOMPoint, angularVelocity: DOMPoint, angularAcceleration:
  DOMPoint }
```

Abbildung 30: Objekt *VRPositionState*

Mit Hilfe der Eigenschaft *timeStamp* wird der Zeitpunkt der Abfrage dargestellt. Die Eigenschaften *hasPosition* und *position* geben an, ob der Sensor aktuell über eine gültige Position verfügt und *position* gibt den Wert dieser Position als 3D Vektor wieder. In dem in Abbildung 30 vorliegenden Fall kann die Hardware keine gültige Position ermitteln, womit der Wert von *position* null beträgt. Die Eigenschaften *linearVelocity* und *linearAcceleration* geben die aktuelle lineare Geschwindigkeit und die aktuelle lineare Beschleunigung des Sensors aus. Dabei wird kein konkreter Zahlenwert angezeigt, sondern der Wert *DOMPoint*. Bei einem

DOMPoint handelt es sich um einen Punkt innerhalb eines dreidimensionalen Koordinatensystems. Die Zahlenwerte des *DOMPoint* sind an dieser Stelle für die Laborübung nicht wesentlich, weshalb sie auch nicht abgefragt werden. Das gleiche gilt für die Eigenschaften *angularVelocity* und *angularAcceleration*. Sie geben die aktuelle Winkelgeschwindigkeit und die aktuelle Winkelbeschleunigung als *DOMPoint* an. Die Eigenschaften *hasOrientation* und *orientation* verhalten sich ähnlich zu den Eigenschaften *hasPosition* und *position*. Mit ihnen wird ermittelt, ob der Sensor über eine gültige Orientierung verfügt. Falls ja, wird diese als *DOMPoint* ausgegeben. Diese *DOMPoint* ist für die Laborübung interessant, da er die Position des Sensors relativ zum HMD anzeigt und somit die Orientierung des HMD im Raum bestimmt werden kann. Um die Zahlenwerte des *DOMPoint* abzufragen, wird mit der Eigenschaft *posState.orientation.x* die X-Koordinate des Punktes bestimmt, *posState.orientation.y* fragt die Y-Koordinate ab, *posState.orientation.z* die Z-Koordinate. Abbildung 31 zeigt die Ausgabe der Koordinaten des *DOMPoint* in der Konsole des Browsers.

```
-0.027186119928956032
-0.0038459498900920153
-0.0025371804367750883
```

Abbildung 31: Koordinaten des *DOMPoint*

Abbildung 32 beinhaltet den entsprechenden Programmiercode der WebVR API für den oben beschriebenen Vorgang.

```

49 <script>
50   var posPara = document.querySelectorAll('p')[0];
51   var orientPara = document.querySelectorAll('p')[1];
52   var timePara = document.querySelectorAll('p')[2];
53   var gHMD, gPositionSensor;
54   navigator.getVRDevices().then(function(devices) {
55     for (var i = 0; i < devices.length; ++i) {
56       if (devices[i] instanceof HMDVRDevice) {
57         gHMD = devices[i];
58         break;
59       }
60     }
61     if (gHMD) {
62       for (var i = 0; i < devices.length; ++i) {
63         if
64           (devices[i] instanceof PositionSensorVRDevice &&
65            devices[i].hardwareUnitId === gHMD.hardwareUnitId) {
66           gPositionSensor = devices[i];
67           break;
68           Position
69         }
70       }
71       console.log (gPositionSensor);
72       var posState = gPositionSensor.getState();
73       console.log (posState);
74
75       console.log (posState.orientation.x);
76       console.log (posState.orientation.y);
77       console.log (posState.orientation.z);
78
79     }
80   });
81 </script>

```

Abbildung 32: Programmiercode WebVR API

Somit lässt sich mit der WebVR API die mit dem PC verbundene Hardware erkennen und die Werte der Sensoren der Hardware auslesen.

9.3 Programmierung der Webanwendung

Dieser Abschnitt behandelt die Inhalte der Webanwendung. Im Kern besteht die Webanwendung aus verschiedenen Stationen. Jede Station vermittelt Kenntnisse über A-Frame. Der Laborteilnehmer kann sich dabei frei in der virtuellen Welt bewegen und von einer Station zur anderen laufen. Vor jeder Station erhält der Teilnehmer ein Dokument, welches die Station beschreibt und ihm das Wissen vermittelt, dass er zur Programmierung der Station benötigt. Es gibt insgesamt fünf Stationen.

9.3.1 Erste Station

Die erste Station dient als Einführung für die Laborteilnehmer. Die Station wird bearbeitet, sobald innerhalb der Laborübung das theoretische Wissen über die HMDs wie Bedienung, Funktionsweise und Einsatzgebiete vermittelt wurde. Zuerst wird dem Teilnehmer erklärt, aus welchen grundlegenden Elementen jede Anwendung, die mit Hilfe von A-Frame realisiert wurde, besteht. Diese sind Szene, Licht, Kamera und geometrische Objekte. Die Szene dient als Hauptelement jeder Anwendung. In ihr befinden sich alle anderen Elemente der Anwendung [32]. Die Kamera bestimmt dabei, aus welcher Perspektive die Szene wahrgenommen wird [33]. Geometrische Objekte sind der eigentliche Inhalt einer Szene. Dabei sind sämtliche Formen wie Dreiecke, Vierecke, Kegel usw. möglich. Durch die Kombination von grundlegenden geometrischen Objekten entstehen neue geometrische Objekte. So kann beispielsweise durch eine Kombination von Dreiecken und Vierecken ein Baum entstehen [34]. Das Licht beleuchtet die Szene und die geometrischen Objekte innerhalb jener. Es wird zwischen verschiedenen Arten von Licht, wie beispielsweise „spot“ oder „ambient“, unterschieden. Licht des Typs „spot“ beleuchtet eine bestimmte Position innerhalb der virtuellen Welt und beeinflusst somit nur Objekte, welche sich an dieser Position befinden. Licht des Typs „ambient“ beleuchtet die gesamte Szene und beeinflusst dadurch jedes Objekt welches sich innerhalb der Szene befindet [35]. Nachdem der Laborteilnehmer weiß, welche grundsätzlichen Elemente jede Szene in A-Frame hat, erfolgt das erstmalige Bauen

einer eigenen Szene. Bei der ersten Szene handelt es sich um Station Eins. Inhalt dieser Station soll ein roter Würfel sein. Außerdem soll der Teilnehmer einen Cursor programmieren, welcher im später Verlauf der Laborübung benötigt wird, um bestimmte Objekte anvisieren zu können.

Um starten zu können, muss der Teilnehmer zu allererst eine HTML Datei anlegen. Diese Datei wird typischerweise „index.html“ genannt. Um Zugriff auf die Funktionen von A-Frame zu haben, muss im „head“ der HTML Datei das Script von A-Frame eingebunden werden. Dies funktioniert mit Hilfe des `<script>` Tags und einem Link auf die Datei. Im „body“ der Datei kann nun mit dem Tag `<a-scene>` die Szene für Station Eins erstellt werden. Es ist wichtig, dass die Tags auch wieder geschlossen werden. Dies geschieht mit einem `/`, welches nach dem `<` Zeichen eingefügt wird, wie beispielsweise `</script>` oder `</a-scene>`. Nun kann mit Hilfe von `<a-box color=“red“></a-box>` ein Würfel innerhalb der Szene erstellt werden. Es ist wichtig, dass dem Würfel mit Hilfe der Komponente `color`, welche den Wert `red` hat, eine rote Farbe hinzugefügt wurde. Anderenfalls wäre der Würfel nicht sichtbar, da dessen Farbe dann weiß ist, genauso wie der Hintergrund der Szene. Ruft man die HTML Datei nun mit Hilfe eines Browsers auf, ergibt sich das Bild, welches in Abbildung 33 zu sehen ist.

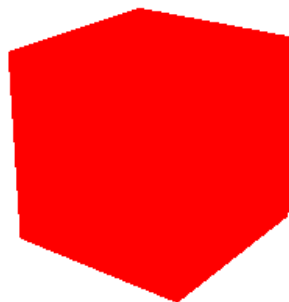


Abbildung 33: Roter Würfel

Den entsprechenden Programmiercode zeigt Abbildung 34.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <script src="https://rawgit.com/aframevr/aframe/master/dist/aframe.min.js">
5    </script>
6    <title>Station 1</title>
7
8  </head>
9  <body>
10   <a-scene>
11     <a-box color="red"> </a-box>
12   </a-scene>
13 </body>
14 </html>

```

Abbildung 34: Programmiercode roter Würfel

Aktuell kann nicht jeder Browser Virtual Reality Inhalte mit Hilfe eines HMDs darstellen. Dafür werden Entwicklerversionen von Google Chrome oder Mozilla Firefox benötigt. Allerdings kann jeder Browser die dreidimensionale Welt so darstellen, wie sie auf den Bildern dieses Abschnitts zu sehen sind.

Auffallend ist, dass bereits Inhalte dargestellt werden können, ohne dass eine Kamera oder ein Licht sich innerhalb der Szene befinden, obwohl Kamera und Licht als erforderliche Grundelemente für jede A-Frame Szene genannt wurden. Der Grund dafür ist, dass A-Frame automatisch ein voreingestelltes Licht in jeder Szene generiert, ohne dass dieses Licht explizit im Programmiercode vorhanden sein muss [35]. Bei diesem voreingestellten Licht handelt es sich um den Typ „*ambient*“ welches alle Objekte mit weisem Licht beleuchtet. Ergänzt man den Code in Abbildung 34 um die Zeile „`<a-light color="red" position="0 1 0"></a-light>`“, so ergibt sich folgendes Bild beim Ausführen der Anwendung , siehe Abbildung 35.

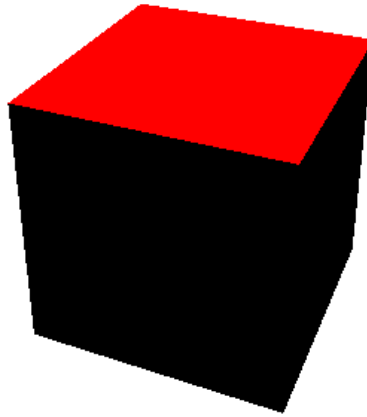


Abbildung 35: Roter Würfel mit rotem Licht

Abbildung 36 zeigt den entsprechenden Programmiercode zu Abbildung 35.

```

11 <body>
12   <a-scene>
13     <a-box color="red"> </a-box>
14     <a-light color="red" position="0 1 0"></a-light>
15   </a-scene>
16 </body>

```

Abbildung 36: Programmiercode roter Würfel mit rotem Licht

Wie man sieht, ist bei diesem Würfel nur die obere Seite rot, die restlichen Seiten sind schwarz. Grund dafür ist das Licht, welches die Komponente „*position*“ mit dem Wert „0 1 0“ hat. Diese Komponente legt die Position des Lichtes innerhalb der virtuellen Welt fest. Die erste Zahl bestimmt die Position auf der X-Achse, die zweite auf der Y-Achse und die dritte auf der Z-Achse. A-Frame nutzt für die Positionierung der Objekte ein Koordinaten System, dessen Achsenverlauf in Abbildung 37 zu sehen ist.

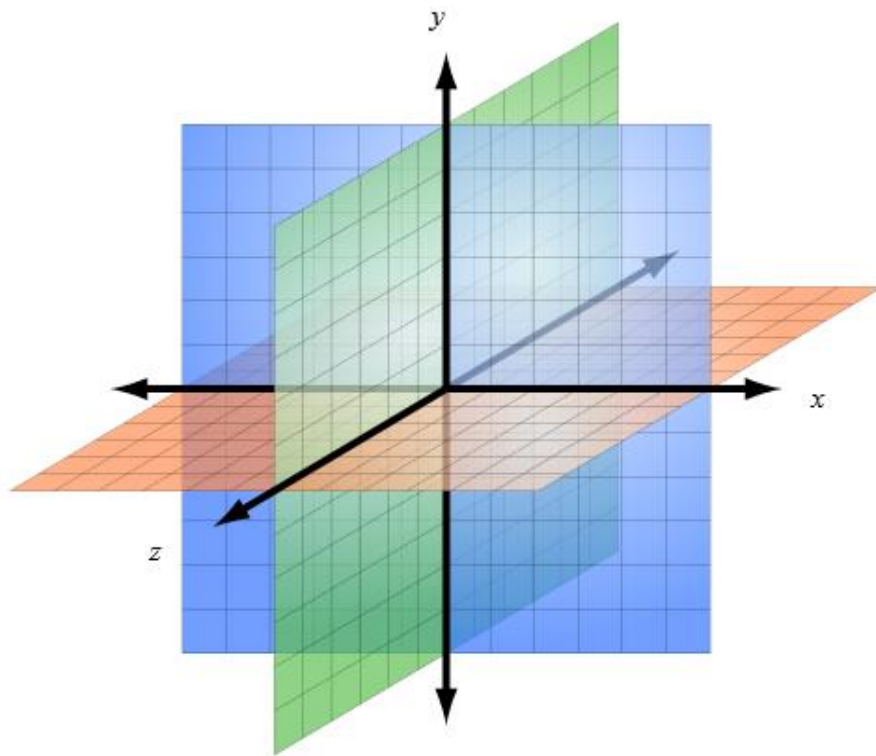


Abbildung 37: Koordinatensystem von A-Frame [36]

Die Bezeichnung der Achse steht dabei jeweils an der vom Nullpunkt ausgehenden positiven Skala der Achse.

Somit befindet sich das Licht direkt oberhalb des Würfels, welcher automatisch die Komponenten „*position*“ mit dem Wert „0 0 0“ besitzt, auch wenn dies nicht im Programmiercode von Abbildung 36 sichtbar ist. A-Frame positioniert jedes Objekt automatisch auf dieser Position, so lange keine andere programmiert wurde. Da sich nun das Licht direkt senkrecht oberhalb des Würfels befindet, wird auch nur seine Oberseite beleuchtet, was erklärt warum nur die Oberseite und nicht auch die anderen Seiten rot sind. Jene werden also nicht beleuchtet und somit schwarz dargestellt.

Ebenfalls automatisch wird von A-Frame eine Kamera erstellt, solange diese nicht programmiert wird. Diese Kamera befindet sich an der Position „0 1.8 5“ [33]. Würde man nun den Programmiercode um die Zeile „`<a-camera>`“ erweitern, ohne der Kamera die Komponenten „*position*“ mit einem bestimmten Wert zuweisen, würde der Würfel nicht mehr zu sehen sein. Dies liegt nicht daran,

dass der Würfel verschwunden ist, vielmehr befindet sie die Kamera innerhalb des nun nicht mehr sichtbaren Würfels.

Als letzte Teilaufgabe der Station Eins integriert der Teilnehmer durch die Zeile „`<a-cursor color="blue"></a-cursor>`“ einen blauen Cursor in der Anwendung. Wichtig ist dabei, dass sich diese Zeile innerhalb der `<a-camera>` Tags befindet, sodass der Cursor seine Position mit der Kamera ändert. Der Cursor erfüllt im späteren Verlauf wichtige Aufgaben.

Der vollständige Programmiercode für Station Eins ist in Abbildung 38 zu sehen.

```

10 </head>
11 <body>
12 <a-scene>
13
14   <a-camera position="0 1.8 4">
15     <a-cursor color="blue"></a-cursor>
16   </a-camera>
17   <a-light color="red" position="0 1 0"></a-light>
18   <a-box color="red"> </a-box>
19
20 </a-scene>
21 </body>

```

Abbildung 38: Vollständiger Programmiercode Station Eins

Das Ergebnis von Station Eins wird in Abbildung 39 dargestellt.

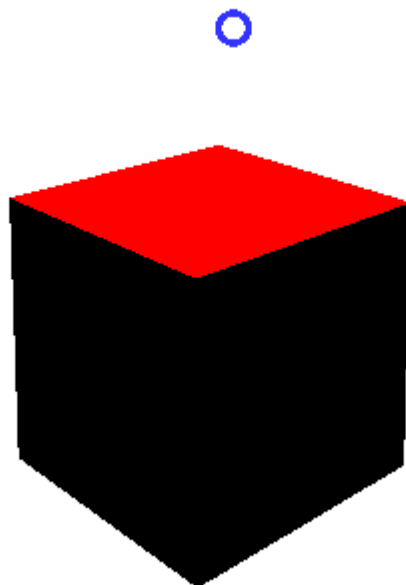


Abbildung 39: Vollständige Station Eins

Der Würfel, der Einfluss des Lichtes und der blaue Cursor sind gut sichtbar.

9.3.2 Zweite Station

In Station Zwei programmiert der Teilnehmer zuerst verschiedene geometrische Objekte. Diese sind ein gelber Quader, ein roter Würfel, ein grüner Zylinder, eine blaue Kugel und ein oranger Kegel. Über die Komponenten „*position*“ werden sie an verschiedenen Stellen platziert, um gut sichtbar zu sein. Mit Hilfe von geometrischen Komponenten wie „*width*“, „*height*“ oder „*depth*“ werden die Proportionen der Objekte verändert. Die drei genannten Komponenten beziehen sich auf den gelben Quader [37], wogegen bei der blauen Kugel die entsprechende Komponente „*radius*“ wäre [38]. Zusätzlich werden manche Objekte mit Hilfe der Komponenten „*rotation*“ innerhalb der Szene gedreht. Die Komponente „*rotation*“ besitzt ähnlich wie die Komponente „*position*“ als Wert drei Zahlen. Diese Zahlen können Werte von 0 bis 360 annehmen. Sie geben die Drehung um die jeweilige Achse an, die erste Zahl um die X-Achse, die zweite Zahl um die Y-Achse und die dritte Zahl um die Z-Achse. Somit beschreibt die Komponente „*rotation*“ die Orientierung eines Objektes innerhalb der Szene [39].

Nachdem alle Objekte nun entsprechend platziert und orientiert wurden, kann ihnen eine Animation hinzugefügt werden. Eine Animation wird mit Hilfe der Tags „*<a-animation></a-animation>*“ hinzugefügt, wobei wichtig ist, dass der Tag für die Animation zwischen den beiden Tags des jeweiligen Objekts steht, das animiert werden soll. Es gibt verschiedene Arten von Animationen. Der Typ „*rotation*“ bewirkt, dass sich das Objekt um seine Achse dreht. Der Typ „*position*“ sorgt dafür, dass sich das Objekt zwischen zwei auswählbaren Positionen bewegt. Der rote Würfel wird so animiert, dass er sich um seine eigene Y-Achse dreht, der grüne Zylinder bewegt sich zwischen zwei Punkten auf der Y-Achse und die blaue Kugel zwischen zwei Punkten auf X-Achse. Der entsprechende Programmiercode ist in Abbildung 40 zu sehen.

```

22 <a-cube color="red" rotation="0 45 0">
23   <a-animation
24     attribute="rotation"
25     repeat="indefinite"
26     to="0 360 0"
27   ></a-animation>
28 </a-cube>
29
30 <a-cylinder color="green" position="3 -0.5 0">
31   <a-animation
32     attribute="position"
33     from="-12 6 -12" to="8 3 -2"
34     dur="1500" repeat="indefinite"
35     fill="forwards"
36     direction="alternate"
37   ></a-animation>
38 </a-cylinder>
39
40 <a-sphere color="blue" position="-3 0 0">
41   <a-animation
42     attribute="position"
43     from="-3 0 0" to="-3 3 0"
44     dur="1500" repeat="indefinite"
45     fill="forwards"
46     direction="alternate"
47   ></a-animation>
48 </a-sphere>

```

Abbildung 40: Programmiercode Animationen

Im Folgenden werden nun die verschiedenen Komponenten, welche innerhalb der „`<a-animation></a-animation>`“ Tags stehen, beschrieben. Die Komponente „`attribute`“ legt, wie bereits beschrieben, den Typ der Animation fest. Mit „`repeat`“ wird angegeben, wie oft sich die Animation wiederholen soll. Die Komponente „`from`“ legt die Startposition fest. Ist diese nicht explizit festgelegt, wird sie automatisch auf die Position des Objektes gelegt. Mit „`to`“ wird der Zielpunkt der Animation festgelegt. Durch „`dur`“ wird festgelegt, wie viele Millisekunden die Animation andauert. Die Komponente „`fill`“ beschreibt das Verhalten des Objektes, nachdem die Animation ausgeführt wurde. Der Wert „`forwards`“ sorgt dafür, dass nach Ende der Animation der Wert der Komponente „`to`“ mit dem Wert der Komponente „`from`“ getauscht wird. Die Komponente „`direction`“ bewirkt, dass im ersten Durchlauf die Animation von der Position beschrieben durch „`from`“ zu Position beschrieben durch „`to`“ stattfindet. Anschließend erfolgt die Animation genau anders herum, von „`to`“ nach „`from`“, während sie im nächsten Animationszyklus wieder von „`from`“ nach „`to`“ durchlaufen wird. Das heißt, der Startpunkt wird nach

jedem Zyklus getauscht. Somit wird mit Hilfe der Komponenten „*fill*“ und „*direction*“ dafür gesorgt, dass sich bei der Animation das Objekt stets zwischen zwei Punkten hin und her bewegt und nicht nach dem Ende der Animation zum Anfangspunkt zurückspringt [40].

Als letzte Teilübung der Station Zwei wird dem gelben Quader und dem orangenen Kegel die „*look-at*“ Komponente hinzugefügt. Als Wert wird ein anderes Objekt innerhalb der Szene eingetragen, im vorliegenden Fall „*a-cylinder*“. Dadurch richten sich der Quader und der Kegel entlang der Bewegung des Zylinders aus. Sie blicken dem Zylinder hinterher, während dieser sich zwischen zwei Punkten auf der X-Achse bewegt. Abbildung 41 zeigt den entsprechenden Programmiercode.

```

48 <a-box position="-1 1 0"
49   rotation="45 45 0"
50   color="yellow"
51   width="1" height="1" depth="0.5"
52   look-at="a-cylinder"
53 ></a-box>
54 <a-cone position="2 0 0"
55   rotation="0 -45 90"
56   color="tomato"
57   radius-bottom="0.5"
58   radius-top="0.2"
59   look-at="a-cylinder"
60 ></a-cone>

```

Abbildung 41: Quader und Zylinder mit "look-at" Komponente

Es besteht auch die Möglichkeit „*a-camera*“ als Wert einzutragen, wodurch die Objekte immer dem Nutzer der Anwendung hinterherschauen würden.

9.3.3 Dritte Station

Während der Stationen Eins und Zwei wurden für Objekte stets Tags genutzt, die das Objekte bereits beschreiben. Beispielsweise wird mit den Tag „*<a-cylinder></a-cylinder>*“ ein Zylinder erstellt. Diese beschreibenden Tags werden „Primitives“ genannt. Primitives dienen als Vereinfachung der eigentlichen API von A-Frame. Im Kern nutzt A-Frame ein sogenanntes „entity-component-system“. Dieses System funktioniert derart, dass jedes Objekt im Programmiercode mit dem Tag „*<a-entity></a-entity>*“ angelegt wird. Dabei spielt es keine Rolle, ob das Objekt

ein Viereck, ein Würfel, eine Kugel oder sonst eine beliebige Form annehmen soll [41]. Möchte man nun mit Hilfe des „entity-component-system“ einen roten Würfel programmieren, sieht der Code im Vergleich zum Code, der für die Programmierung per Primitive benötigt wird, wie folgt aus.

```

14 <!--Würfel mit Primitive-->
15 <a-box
16     color="red"
17     position="0 0 0"
18 ></a-box>
19 <!--Würfel mit entity-component system-->
20 <a-entity
21     geometry="primitive: box"
22     material="color:red"
23     position="2 0 0"
24 ></a-entity>

```

Abbildung 42: Vergleich Primitives vs. entity-component-system

Abbildung 42 zeigt, dass bei der Verwendung des „entity-component-system“ wesentlich mehr Code programmiert werden muss, um das gleiche Ergebnis zu erhalten. Fraglich ist jedoch, wo dabei der Vorteil des „entity-component-system“ liegt. Dadurch, dass jedes Objekt im Grunde eine „<a-entity>“ ist, lässt sich jedes sehr leicht erweitern. Man kann der „<a-entity>“ einfach eine neue Komponente hinzufügen. Es können Komponenten, die eigentlich für ein anderes Objekt geschrieben wurden, jedem beliebigen Objekt hinzugefügt werden. Dadurch lassen sich Objekte leicht in ihrem Aussehen, Verhalten und in ihrer Funktion verändern. Außerdem erlaubt das „entity-component-system“ Komponenten von unabhängigen Entwicklern zu nutzen, die nicht standardmäßig in der A-Frame API enthalten sind. Dadurch wäre es möglich, dem roten Würfel eine Komponente hinzuzufügen, welche von einem anderen A-Frame Nutzer entwickelt wurde, was beispielsweise den Würfel nach dem Ablauf einer bestimmten Zeit explodieren lässt [42].

Mit Station Drei wird dem Laborteilnehmer anschließend gezeigt, wie man das „entity-component-system“ verwendet. Bevor der eigentliche Inhalt der Station betrachtet wird, bedarf es noch der Erläuterung eines weiteren Systems von A-Frame, das sogenannte „asset management system“. Dieses System hat die Funktion Bilder, Videos, Audiodateien oder 3D Modelle vorzuladen, bevor die A-Frame Szene gerendert wird. Dadurch wird gewährleistet, dass während des Renderings alle Daten vorhanden sind, was der Stabilität der Anwendung zu Gute kommt.

```

53 <!--preloading pictures, audio etc.-->
54 <a-assets>
55   
56   
57   
58   
59   <audio id="laugh" src="laugh.mp3"></audio>
60   
61 </a-assets>

```

Abbildung 43: Asset management system

Abbildung 43 zeigt das „asset management system“ innerhalb des Programmiercodes. Dabei wird jede Datei, ihrer Art entsprechend, mit einem Tag versehen, wie beispielsweise `` für Bild oder `<audio>` für eine Audio-Datei. Zusätzlich wird jede Datei mit einer ID versehen, die frei festlegbar ist. Mit Hilfe dieser ID kann die Datei später in einem Objekt eingebunden werden. Das Verwenden des „asset management system“ ist nicht zwingend notwendig, es können Dateien auch direkt über Links in Objekte eingebunden werden, allerdings bringt das System die oben genannten Vorteile mit [43].

Zusätzlich ergeben sich aus der dritten Station folgende Neuerungen für den Labor Teilnehmer: Das Einbinden von Videos, Bildern und 3D Modellen und das Auslösen eines Events mit Hilfe des Cursors.

Die dritte Station startet damit, dass der Teilnehmer zu allererst den Boden erstellen muss, auf dem sich die Station befindet. Dieser war bisher stets vorgegeben. Zuerst wird ein Objekt mit „`<a-entity></a-entity>`“ erzeugt. Die Komponente „*geometry*“ mit den Werten "*primitive: circle; radius: 20*" verleiht dem Objekt ein kreisförmiges Aussehen mit dem Radius von 20 Metern. Die Komponente „*material*“ mit den Werten "*src: #waldboden; repeat:20 20*" passt die Oberfläche des Objektes so an, dass sie dem Bild mit der ID „waldboden“ entspricht. Der Wert „*repeat: 20 20*“ gibt an, wie oft das Bild auf der Obefläche des Objektes in Richtung der X-Achse und der Y-Achse wiederholt wird. Zusätzlich wird das Objekt nun entsprechend positioniert und orientiert, um als Boden der Station zu dienen. Abbildung 44 zeigt den entsprechenden Programmiercode.


```

268 <!--floor-->
269 <a-entity
270   geometry="primitive: circle; radius: 20"
271   material="src: #waldboden; repeat:20 20"
272   position="0 -0.2 0"
273   rotation="-90 0 0"
274 ></a-entity>

```

Abbildung 44: Programmiercode Boden dritte Station

Nachdem der Boden programmiert wurde, soll nun ein 3D Model integriert werden. Dies ist kann einfach durch folgenden Code geschehen.

```

276 <a-entity >
277   <a-collada-model src="trees.dae"></a-collada-model>
278 </a-entity>

```

Abbildung 45: Einbinden des 3D Models

Wie Abbildung 45 zeigt, ist es dabei wichtig den `<a-collada-model>` Tag innerhalb des `<a-entity>` Tags zu setzen [44].

Als nächstes gilt es nun ein Video in die Anwendung zu integrieren, wobei wieder das „asset management system“ genutzt wird, wie Abbildung 46 zeigt.

```

280 <!--preloading video-->
281 <a-assets>
282   <video id="video" src="video3.mp4"></video>
283 ></a-assets>

```

Abbildung 46: Vorladen des Videos

Anschließend wird das Video wie in Abbildung 47 gezeigt eingebunden.

```

289 <!--video-->
290 <a-entity
291   geometry="primitive: plane ; width: 2.5; height: 1.4"
292   material="color:#6E6E6E"
293   position="-4 0.7 -5.075"
294   rotation="0 180 0">
295
296   <a-event name="mouseenter" material="src: #video"></a-event>
297
298 </a-entity>

```

Abbildung 47: Einbinden des Videos

Innerhalb des `<a-entity>` Tags für das Video wird ein `<a-event></a-event>` Tag programmiert. Die Komponente „name“ legt dabei den Typ des Events fest, der Wert „mouseenter“ bedeutet, dass das Event ausgelöst wird, sobald der Cursor der Anwendung das Objekt anvisiert. Mit der Komponente „material“ und dem Wert „src: # video“ wird das Video eingebunden. Abbildung 48 zeigt die fertiggestellte

Station Drei. Der Wald welcher mit dem 3D Modell erzeugt wurde, der Waldboden und das Video in der Mitte sind gut erkennbar.

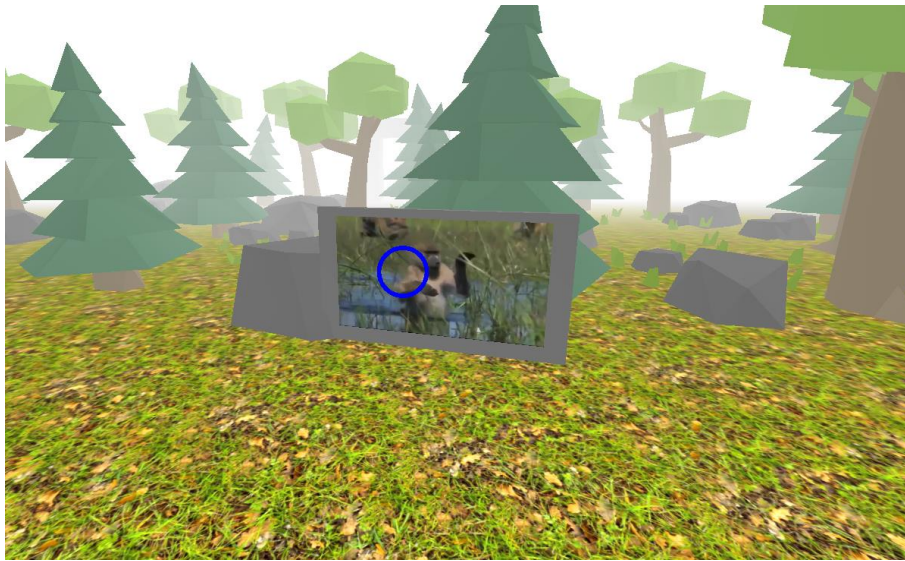


Abbildung 48: Fertiggestellte Station Drei

9.3.4 Vierte Station

Die vierte Station bringt als neuen Tag `<a-curvedimage>` mit sich. Dabei handelt es sich um Bilder, welche gebogen sind, so dass jeder Bildpunkt die gleiche Entfernung zum Betrachter hat. Diese macht die Beobachtung einfacher, als bei einem normalen Bild [45].

Außerdem führt diese Station die Komponente „*scale*“ ein, welche in jedem Objekt eingefügt werden kann. Die Komponente bewirkt ein Schrumpfen oder Vergrößern des Objektes. Die Komponente „*scale*“ mit dem Wert „2 2 2“ bewirkt, dass das zugehörige Objekte in der X-Achse, der Y-Achse und der Z-Achse um den Skalierungsfaktor zwei vergrößert wird [46].

Als weitere Besonderheit hat die vierte Station ein 3D Model, auf deren Oberfläche sich der Körperscan eines Mannes befindet. Ansonsten werden nochmals die Kenntnisse von der vorherigen Station genutzt, wie etwa das Einbinden von Daten in Objekte mit Hilfe des „entity-component-system“, um diese Station fertig zu stellen. Abbildung 49 zeigt die fertiggestellte Station Vier.

Diese Station ist von einem der Beispiele der Webseite von A-Frame inspiriert. Der Link zum Beispiel lautet: <https://aframe.io/examples/showcase/shopping/>.



Abbildung 49: Vierte Station

9.3.5 Fünfte Station

Bei der fünften Station kommt erneut eine weitere Form der Interaktion zwischen Teilnehmer und A-Frame Anwendung hinzu. Dabei wird ein Würfel integriert, welcher durch das Anklicken des Teilnehmers zu lachen beginnt. Abbildung 50 zeigt den entsprechenden Programmiercode.

```

315 <!--cube with on-click tickle-->
316 <a-entity
317   geometry="primitive:box"
318   material="src:#tickle"
319   sound="src: laugh.mp3; on: click"
320
321 >/a-entity>

```

Abbildung 50: Programmiercode für Würfel mit Lachen

Abbildung 51 zeigt die fertiggestellte fünfte Station.

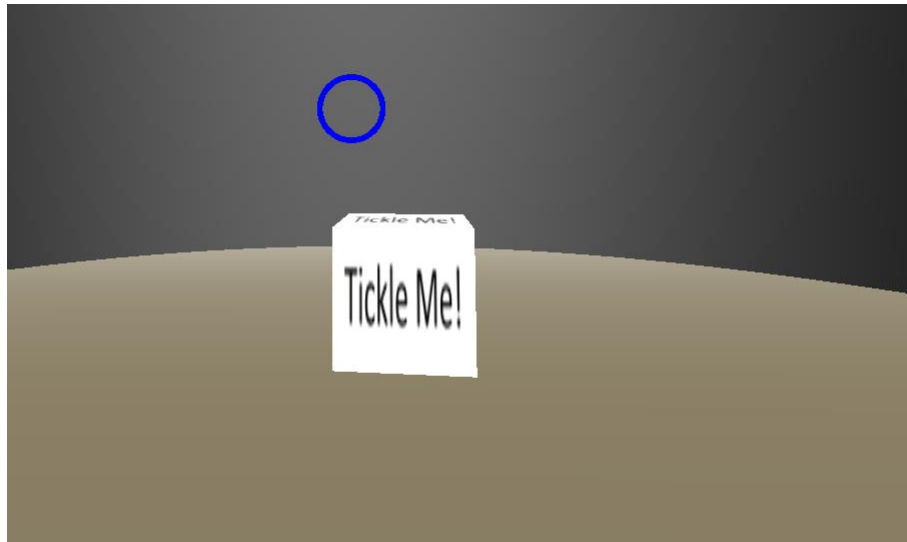


Abbildung 51: Fünfte Station

9.3.6 Andere Inhalte der Laborübung

Neben den eigentlichen Stationen enthält die Laborübung noch Verbindungswege mit Toren und Beschriftung zwischen den einzelnen Stationen. Die Beschriftung wird durch eine Komponente realisiert, die nicht standardmäßig Teil der A-Frame API ist. Sie nennt sich „a-frame-text-component“ und ist unter diesem Link zu finden: <https://github.com/ngokevin/aframe-text-component>. Um die Komponente nutzen zu können, muss innerhalb des HTML Dokuments im `<head>` Bereich das entsprechende Skript eingefügt werden. Abbildung 52 zeigt den Weg zur vierten Station mit dem oben beschriebenen Tor samt Beschriftung.

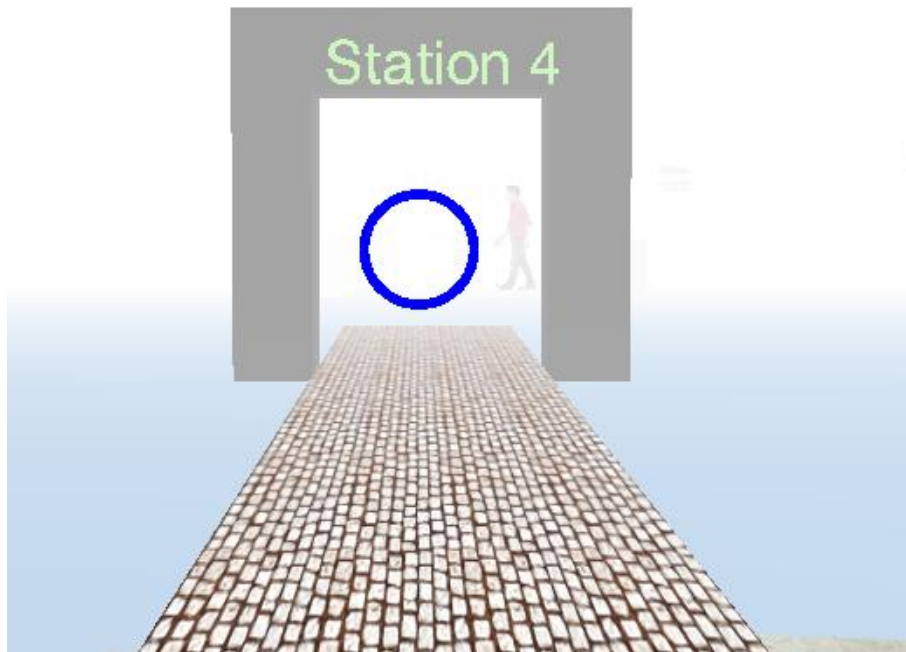


Abbildung 52: Tor mit Beschriftung

Zusätzlich wird innerhalb der Anwendung ein User Interface angezeigt, welches Informationen bereitstellt, wie beispielsweise Bilder pro Sekunde mit der die Anwendung läuft oder Anzahl der Objekte in der Anwendung bzw. Zeit in Millisekunden, bis die Anwendung geladen ist. Abbildung 53 zeigt das User Interface.

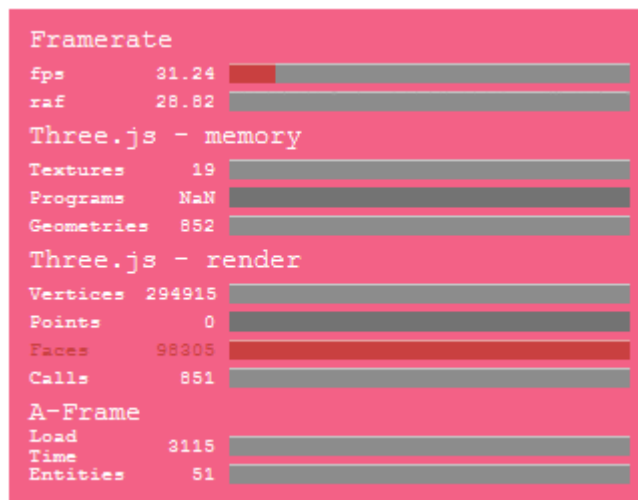


Abbildung 53: User Interface der Laborübung

Im Programmiercode kann das User Interface aktiviert werden indem dem `<a-scene>` Tag die Komponente „stats“ mit dem Wert „true“ hinzugefügt wird [47].

Zusätzlich wurde der Kern API von A-Frame eine Funktion hinzugefügt, welche dafür sorgt, dass bei jedem Bild, das gerendert wird, die aktuelle Position der Kamera angezeigt. Die Funktion nennt sich „*logging*“ und der entsprechende Programmiercode sieht wie folgt aus.

```

31 <script>
32   /* funtion to post position of camera on each frame*/
33   (AFRAME.aframeCore || AFRAME).registerComponent('logging', {
34     init: function () {},
35     update: function () { this.tick(); },
36     tick: function () {
37       var camera = this.el.querySelector('[camera]');
38       console.log(camera.getAttribute('position'));
39       var p = camera.getAttribute('position');
40       document.getElementById('cameraposition').innerHTML =
41         'X: ' + p.x + ' Y: ' + p.y + ' Z: ' + p.z;
42     },
43   },
44   remove: function () {}
45   });
46 </script>

```

Diese Funktion wurde mit Hilfe von JavaScript in die HTML Datei integriert. Um die Funktion innerhalb der Anwendung zu verwenden wird das `<a-scene>` Tag um die Komponente „*logging*“ erweitert. Dadurch wird die Funktion ausgeführt. Ein Wert für die Komponenten ist nicht nötig. Die durch die Funktion ermittelte Position wird dem Nutzer der Anwendung neben dem User Interface eingeblendet, siehe Abbildung 54.

X: -4.500042471118718 Y: 1 Z: 0.2615423283453465

Abbildung 54: Anzeige der Kameraposition

Die komplette Laborübung ist auf Github verfügbar. Der Link lautet: <https://github.com/Nicklas1989/Stationen>. Dort sind der komplette Programmiercode, sowie alle verwendeten Dateien einsehbar und können heruntergeladen werden.

9.4 Zusammenfassung

Dieses Kapitel behandelt die Laborübung zum Thema Head-Mounted Displays. Es werden die Anforderungen an die Übung und die genutzten Technologien vorgestellt. Anschließend wird der Inhalt der Laborübung erläutert und darauf eingegangen, wie diese mit Hilfe von A-Frame programmiert wurden. Die Laborübung ist stufenweise aufgebaut und besteht aus insgesamt fünf Stationen.

10 Evaluation der Laborübung

In diesem Kapitel soll über die Gestaltung der Laborübung diskutiert werden. Aus Sicht des Autors eignet sich der Aufbau in einzelne Station gut, um die Laborteilnehmer langsam an das neue Thema heranzuführen. Dadurch entsteht keine Überforderung bzw. zu viel Input auf einmal. Durch den stufenweisen Aufbau der Laborübung werden Inhalte aus früheren Stationen in der aktuellen Station wiederholt, was dabei hilft, dass Wissen zu festigen. Weiterhin ist es sinnvoll, in den ersten beiden Übungen „primitives“ von A-Frame zu verwenden, da diese das erzeugte Objekt gut beschreiben und den Programmieraufwand möglichst geringhalten. So muss sich der Teilnehmer erst zu einem späteren Zeitpunkt mit dem komplexen „entity-component-system“ beschäftigen, zu dem er die Grundzüge von A-Frame bereits verinnerlicht hat. Zusätzlich wird dem Teilnehmer mit der in Abschnitt 9.3.6 erwähnten Positionsanzeige der Kamera eine Hilfestellung an die Hand gegeben, um sich so innerhalb der Laborübung zurechtzufinden. Zuletzt hat der Teilnehmer die Möglichkeit alle verwendeten Daten der Laborübung über Github einzusehen, was die Übung transparent macht. Außerdem kann er dadurch Verbesserungsvorschläge und seine Meinung zur Übung einbringen.

Eine Evaluation durch Befragung der Laborteilnehmer kann natürlich erst zu einem späteren Zeitpunkt erfolgen, da die Übung noch nicht in der Praxis durchgeführt wurde.

11 Ausblick

In diesem Kapitel werden zukünftige Herausforderungen für Hersteller von HMDs sowie technologische Neuerung zum Thema Virtual Reality und Augmented Reality gezeigt. Dabei handelt es sich um Prototypen, die von einer Marktreife mehr oder weniger weit entfernt sind.

Die Nutzung von Eye-Tracking innerhalb eines HMD gilt dabei als eine der Schlüsselkomponenten von HMDs der nächsten Generation. „Unter Eye-Tracking bezeichnet man allgemein die Verfolgung der Blickrichtung des menschlichen Auges. Eye-Tracking kann man wörtlich mit Blickregistrierung übersetzen.“ [17] S. 117. Somit kann festgestellt werden, welcher Bereich auf dem Display des HMD vom Nutzer fokussiert wird. Dieser Bereich wird nun in hoher Auflösung dargestellt. Im Gegensatz dazu werden umliegende Punkte in geringer Auflösung dargestellt, da dies für das menschliche Auge nicht auffällig ist. Dadurch können Anwendungen mit weniger Rechenleistung betrieben werden, als das bei HMDs ohne Eye-Tracking der Fall ist. Außerdem können Eye-Tracking Systeme zur Steuerung von Anwendungen genutzt werden. Eye-Tracking Systeme bieten auch die Möglichkeit, eine glaubwürdige Tiefenwirkung zu erzeugen. So können fokussierte Bereiche scharf dargestellt werden, während Bereiche im Hintergrund verschwommen dargestellt werden [48].

Ein weiter Trend in Sachen Virtual Reality ist es, neben dem Kopf, Händen bzw. den Augen, noch weitere Teile des Körpers zur Steuerung innerhalb von virtuellen Welten zu nutzen. Durch diese Eingabegeräte soll der Grad der Immersion, die der Nutzer erfährt, stark erhöht werden. So entwickelt beispielsweise das Unternehmen *AxonVR* eine Art Ganzkörperanzug. Dadurch lassen sich Bewegungen von Armen bzw. Beinen des Nutzers in die virtuelle Welt übertragen. Außerdem arbeitet *AxonVR* an einem Ausgabegerät, welches sowohl die Oberflächenbeschaffenheit als auch die entsprechende Oberflächentemperatur simulieren kann. Dadurch soll die Interaktion mit virtuellen Objekten wesentlich überzeugender werden [49].

Ein weiterer Trend geht hin zu Stand-alone HMDs. Diese HMDs benötigen weder einen PC noch ein Smartphone, um betrieben zu werden. Vielmehr sind alle nötigen Komponenten, wie etwa CPU und GPU, im HMD integriert. Dadurch entstehen

mobile Systeme die ausschließlich für Virtual Reality bzw. Augmented Reality entwickelt wurden. Aktuell gibt es Gerüchte, dass Google zeitnah eine Stand-alone HMD vorstellt [50].

In Sachen Augmented Reality gibt es zum aktuellen Zeitpunkt noch keine HMDs die marktreif sind. Das aktuell am weitesten entwickelte Augmented Reality HMD ist sicherlich die *Hololens* von Microsoft. Entwickler können für 3000 \$ ein Exemplar der Hololens bestellen, allerdings nur in den USA bzw. Kanada [51].

12 Fazit

Der Fokus dieser Bachelorarbeit liegt auf der Anschaffungsempfehlung der vorgestellten Head-Mounted Displays und der Schaffung einer Laborübung welche die Head-Mounted Displays im Kontext eine Bildungseinrichtung nutzt.

Von den Head-Mounted Displays hat sich letztendlich das Dark Shader von ZAAK ZURICH BERLIN – Cardboard Kits Europe durchgesetzt. Für die Gewichtung der Kriterien, welche am Ende zu dieser Entscheidung geführt haben, wurden fünfzehn Personen befragt. An dieser Stelle wäre sicherlich eine höhere Anzahl an befragten Personen wünschenswert gewesen um aussagekräftigere Boxplot Diagramme zu gewährleisten. Ergänzend wäre auch die Einbeziehung des Head-Mounted Displays von OSVR interessant, das zum Zeitpunkt dieser Arbeit jedoch nicht verfügbar war. Dieses HMD wird auf ähnlich hohem technischen Niveau produziert wie die Oculus Rift, verfolgt aber konsequent den Gedanken eines Open Source Projekt [52].

Die Laborübung wurde mit Hilfe verschiedener Webtechnologien umgesetzt. Die wichtigste Rolle spielt dabei die Auszeichnungssprache A-Frame. Sie wird auch das Medium sein, mit dem sich die Teilnehmer der Laborübung auseinandersetzen. Dafür wurde versucht, den Teilnehmern durch den stufenweisen Aufbau der Übung ein möglichst angenehmer Einstieg zu gewährleisten. Im Anschluss an diese Arbeit gilt es natürlich die Durchführung der Laborübung weiter auszugestalten und mit Hilfe von Evaluationen diese an die individuellen Bedürfnisse hinsichtlich der praktischen Ausführung der Lernenden anzupassen.

13 Literaturverzeichnis

- [1] DÖRNER, Ralf (Hrsg.); BROLL, Wolfgang (Hrsg.); GRIMM, Paul (Hrsg.); JUNG, Bernhard (Hrsg.): *Virtual und Augmented Reality (VR / AR) : Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Berlin, Heidelberg : Imprint: Springer Vieweg, 2013 (EXamen.press)
- [2] PAUL SORENE: *Jaron Lanier's EyePhone: Head And Glove Virtual Reality In The 1980s*. URL <http://flashbak.com/jaron-laniers-eyephone-head-and-glove-virtual-reality-in-the-1980s-26180/> – Überprüfungsdatum 2016-03-20
- [3] BROLL, Wolfgang: Augmentierte Realität. In: DÖRNER, Ralf; BROLL, Wolfgang; GRIMM, Paul; JUNG, Bernhard (Hrsg.): *Virtual und Augmented Reality (VR / AR) : Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Berlin, Heidelberg : Imprint: Springer Vieweg, 2013 (EXamen.press), S. 241–294
- [4] BRILL, Manfred: *Virtuelle Realität*. Berlin, Heidelberg : Springer-Verlag, 2009 (Informatik im Fokus)
- [5] RADEMACHER, Martin H.: *Virtual Reality in der Produktentwicklung : Instrumentarium zur Bewertung der Einsatzmöglichkeiten am Beispiel der Automobilindustrie*. Wiesbaden : Springer Vieweg, 2014 (SpringerLink : Bücher)
- [6] PARISI, Tony: *Learning virtual reality : Developing immersive experiences and applications for desktop, Web, and mobile*. Sebastopol CA : O'Reilly, 2015
- [7] DÖRNER, Ralf ; STEINCKE, Frank: Wahrnehmungsaspekte von VR. In: DÖRNER, Ralf; BROLL, Wolfgang; GRIMM, Paul; JUNG, Bernhard (Hrsg.): *Virtual und Augmented Reality (VR / AR) : Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Berlin, Heidelberg : Imprint: Springer Vieweg, 2013 (EXamen.press), S. 33–64
- [8] GRIMM, Paul ; HEROLD, Rigo ; REINERS, Dirk ; CRUZ-NEIRA CAROLINA: VR-Ausgabegeräte. In: DÖRNER, Ralf; BROLL, Wolfgang; GRIMM, Paul; JUNG, Bernhard (Hrsg.): *Virtual und Augmented Reality (VR / AR) : Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Berlin, Heidelberg : Imprint: Springer Vieweg, 2013 (EXamen.press), S. 127–156

- [9] SCHNEID, Marcus: *Entwicklung und Erprobung eines kontaktanalogen Head-up-Displays im Fahrzeug*. München, Technische Universität München. Dissertation. 2008-07-01
- [10] TÖNNIS, Marcus: *Augmented Reality : Einblicke in die Erweiterte Realität*. Berlin : Springer, 2010 (Informatik im Fokus)
- [11] ARNOLD, Volker ; DETTMERING, Hendrik ; ENGEL, Torsten ; KARCHER, Andreas: *Product Lifecycle Management beherrschen : Ein Anwenderhandbuch für den Mittelstand*. Dordrecht : Springer, 2011
- [12] BAUMEISTER, Mika: *Audi VR Experience: Wunsch-Autos schon vor dem Kauf testen*. URL <http://www.mobilegeeks.de/artikel/audi-vr-experience/>. – Aktualisierungsdatum: 2015-12-28 – Überprüfungsdatum 2015-12-28
- [13] MARCUS YAM: *Oculus Rift - The Past, Present, And Future Of VR And AR: The Pioneers Speak*. URL <http://www.tomshardware.com/reviews/ar-vr-technology-discussion,3811-4.html> – Überprüfungsdatum 2016-04-08
- [14] iFIXIT: *Oculus Rift Development Kit 2 Teardown*. URL <https://de.ifixit.com/Teardown/Oculus+Rift+Development+Kit+2+Teardown/27613> – Überprüfungsdatum 2016-04-08
- [15] SVEN MILKER: *Bewegungserkennung mit Smartphones mittels deren Sensoren*. Koblenz, Universität Koblenz-Landau. Bachelorarbeit. 2012-07-20
- [16] MOZILLA DEVELOPER NETWORK: *Keep it level: responding to device orientation changes*. URL https://developer.mozilla.org/en-US/Apps/Fundamentals/gather_and_modify_data/responding_to_device_orientation_changes. – Aktualisierungsdatum: 2016-02-29 – Überprüfungsdatum 2016-04-11
- [17] GRIMM, Paul ; HEROLD, Rigo ; HUMMEL, Johannes ; BROLL, Wolfgang: VR-Eingabegeräte. In: DÖRNER, Ralf; BROLL, Wolfgang; GRIMM, Paul; JUNG, Bernhard (Hrsg.): *Virtual und Augmented Reality (VR / AR) : Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Berlin, Heidelberg : Imprint: Springer Vieweg, 2013 (EXamen.press), S. 97–126
- [18] ZAAK.IO: *Dark Shader - Cardboard Kits Europe*. URL <http://shop.zaak.io/products/dark-shader-google-cardboard-kit-europe> – Überprüfungsdatum 2016-04-20

- [19] RIPTON, J. T.: *Techradar.com: Google Cardboard : Everything you need to know*. URL <http://www.techradar.com/news/phone-and-communications/mobile-phones/google-cardboard-everything-you-need-to-know-1277738> – Überprüfungsdatum 2016-04-21
- [20] ZEISS VR One. URL <http://zeissvrone.tumblr.com/#we-are-one> – Überprüfungsdatum 2016-04-21
- [21] ZEISS VR One : Photo. URL <http://zeissvrone.tumblr.com/image/124226881037> – Überprüfungsdatum 2016-04-21
- [22] JASER, Michael: *Evaluation, Bewertung und Implementierung verschiedener Cross-Platform Development Ansätze für Mobile Internet Devices auf Basis von Web-Technologien*. Augsburg, Hochschule Augsburg. Bachelorthesis. 2011-03-21. URL <http://cross-mobile-apps.de/files/bachelorthesis-michael-jaser.pdf> – Überprüfungsdatum 2016-04-23
- [23] WebGL - OpenGL ES 2.0 for the Web. URL <https://www.khronos.org/webgl/> – Überprüfungsdatum 2016-01-03
- [24] WebVR - Bringing Virtual Reality to the Web. URL <https://webvr.info/>. – Aktualisierungsdatum: 2016-04-19 – Überprüfungsdatum 2016-04-25
- [25] *three.js - documentation - Manual - Creating a scene*. URL http://threejs.org/docs/index.html#Manual/Introduction/Creating_a_scene. – Aktualisierungsdatum: 2015-10-17 – Überprüfungsdatum 2016-01-04
- [26] *Auszeichnungssprache — Enzyklopaedie der Wirtschaftsinformatik*. URL <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/technologien-methoden/Sprache/Auszeichnungssprache> – Überprüfungsdatum 2016-05-15
- [27] A-Frame. URL <https://aframe.io/docs/guide/>. – Aktualisierungsdatum: 2016-04-24 – Überprüfungsdatum 2016-04-26
- [28] BUHR, Mathias ; PFEIFFER, Thies ; REINERS, Dirk ; CRUZ-NEIRA, Carolina ; JUNG, Bernhard: Echtzeitaspekte von VR-Systemen. In: DÖRNER, Ralf; BROLL, Wolfgang; GRIMM, Paul; JUNG, Bernhard (Hrsg.): *Virtual und Augmented Reality (VR / AR) : Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Berlin, Heidelberg : Imprint: Springer Vieweg, 2013 (EXamen.press), S. 194–240

- [29] KÜHNAPFEL, Jörg B.: *Nutzwertanalysen in Marketing und Vertrieb*. Wiesbaden : Springer-Gabler, 2014 (essentials)
- [30] SAATY, Thomas L. ; VARGAS, Luis Gonzalez: *Models, methods, concepts & applications of the analytic hierarchy process*. 2. ed. New York : Springer, 2012 (International series in operations research & management science 175)
- [31] PETERS, Malte L. ; ZELEWSKI, Stephan: *Analytical Hierarchy Process (AHP) – dargestellt am Beispiel der Auswahl von Projektmanagement-Software zum Multiprojektmanagement*. URL https://www.pim.wiwi.uni-due.de/uploads/tx_itochair3/publications/bericht14.pdf – Überprüfungsdatum 2016-02-24
- [32] *A-Frame*. URL <https://aframe.io/docs/core/scene.html>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-02
- [33] *A-Frame*. URL <https://aframe.io/docs/components/>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-02
- [34] *A-Frame*. URL <https://aframe.io/docs/components/geometry.html>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-02
- [35] *A-Frame*. URL <https://aframe.io/docs/components/light.html>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-02
- [36] *File:3D coordinate system.svg - Wikimedia Commons*. URL https://commons.wikimedia.org/wiki/File:3D_coordinate_system.svg. – Aktualisierungsdatum: 2016-04-28 – Überprüfungsdatum 2016-05-03
- [37] *A-Frame*. URL <https://aframe.io/docs/primitives/a-box.html>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-04
- [38] *A-Frame*. URL <https://aframe.io/docs/primitives/a-sphere.html>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-04
- [39] *A-Frame*. URL <https://aframe.io/docs/components/rotation.html>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-04
- [40] *A-Frame*. URL <https://aframe.io/docs/core/animations.html>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-04
- [41] *A-Frame*. URL <https://aframe.io/docs/primitives/>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-04

- [42] *A-Frame*. URL <https://aframe.io/docs/core/>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-04
- [43] *A-Frame*. URL <https://aframe.io/docs/core/asset-management-system.html>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-04
- [44] *A-Frame*. URL <https://aframe.io/docs/components/collada-model.html>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-04
- [45] *A-Frame*. URL <https://aframe.io/docs/primitives/a-curvedimage.html>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-04
- [46] *A-Frame*. URL <https://aframe.io/docs/components/scale.html>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-04
- [47] *A-Frame*. URL <https://aframe.io/docs/components/stats.html>. – Aktualisierungsdatum: 2016-04-29 – Überprüfungsdatum 2016-05-05
- [48] *Fove: Head-mounted Display mit Eye Tracking bei Kickstarter - Golem.de*. URL <http://www.golem.de/news/fove-head-mounted-display-mit-eye-tracking-bei-kickstarter-1505-114169.html> – Überprüfungsdatum 2016-05-18
- [49] *Home*. URL <http://axonvr.com/#virtual-reality-you-can-feel>. – Aktualisierungsdatum: 2016-05-18 – Überprüfungsdatum 2016-05-18
- [50] GEIGER, Caren Stella: *Google I/O 2016 Vorschau: Android N, VR und mehr heute auf der Entwickler-Keynote*. URL http://www.chip.de/news/Google-IO-2016-Vorschau-Android-N-VR-und-mehr-heute-auf-der-Entwickler-Keynote_93994723.html. – Aktualisierungsdatum: 2016-05-18 – Überprüfungsdatum 2016-05-18
- [51] MICROSOFT: *Microsoft HoloLens*. URL <https://www.microsoft.com/microsoft-hololens/en-us> – Überprüfungsdatum 2016-05-18
- [52] *Razer / OSVR - Open-Source Virtual Reality for Gaming*. URL <http://www.osvr.org/>. – Aktualisierungsdatum: 2016-06-03 – Überprüfungsdatum 2016-06-08

14 Anhang

Anhang 1: Vorlage zur Gewichtung aller Hauptkriterien mit der Paarvergleichsmethode	115
Anhang 2: Vorlage zur Gewichtung der Unterkriterien des Tragekomforts	116
Anhang 3: Vorlage zur Gewichtung der Unterkriterien der Qualität der Immersion	117
Anhang 4: Vorlage zur Gewichtung der Unterkriterien der Benutzerfreundlichkeit	118
Anhang 5: Durchgeführte Gewichtung aller Hauptkriterien	119
Anhang 6: Durchgeführte Gewichtung der Unterkriterien des Tragekomforts...	120
Anhang 7: Durchgeführte Gewichtung der Unterkriterien der Qualität der Immersion	121
Anhang 8: Durchgeführte Gewichtung der Unterkriterien der Benutzerfreundlichkeit	122
Anhang 9: Vorlage zur Gewichtung der Unterkriterien des Tragekomforts mit dem AHP	123
Anhang 10: Vorlage zur Gewichtung der Unterkriterien der Qualität der Immersion mit dem AHP	124
Anhang 11: Vorlage zur Gewichtung der Unterkriterien der Benutzerfreundlichkeit mit dem AHP	125
Anhang 12: Vorlage zur Gewichtung der Hauptkriterien mit dem AHP	126
Anhang 13: Durchgeführte Gewichtung der Unterkriterien des Tragekomforts mit dem AHP	127
Anhang 14: Durchgeführte Gewichtung der Unterkriterien der Qualität der Immersion mit dem AHP	128
Anhang 15: Durchgeführte Gewichtung der Unterkriterien der Benutzerfreundlichkeit mit dem AHP	129
Anhang 16: Durchgeführte Gewichtung der Hauptkriterien mit dem AHP	130

Anhang 17: Programmiercode vierte Station	131
---	-----

Anhang 1: Vorlage zur Gewichtung aller Hauptkriterien mit der Paarvergleichsmethode

Gewichtungsgesamt												
		1	2	3	4	5	6	7	8	9	10	
					Qualität							
		Kosten	Robustheit	Tagekomfort	der Immersion	Benutzerfreundlichkeit	Mobilität des Systems (Kabel, Pc notwendig)	Open Source Software	Open Hardware	Verfügbare Software	Entwicklung für diese Plattform	Summe
Anschaffung als auch evtl.												
laufende Kosten	Kosten											
Langlebigkeit der Brille	Robustheit											
siehe Unterkriterien	Tagekomfort											
siehe Unterkriterien	Qualität der Immersion											
siehe Unterkriterien	Benutzerfreundlichkeit											
Ist ein PC nötig?												
Wird die Bewegungsfreiheit durch Kabel eingeschränkt?												
Ist der Software Code öffentlich zugänglich?	Mobilität des Systems											
Werden die verwendeten, technischen Komponenten vom Hersteller offgelegt?	Open Source Software											
Anzahl und Art	Open Hardware											
Wie leicht ist eine native Entwicklung möglich?	Verfügbare Software											
	Entwicklung für diese Plattform											

Anhang 2: Vorlage zur Gewichtung der Unterkriterien des Tragekomforts

Gewichtung Tragekomfort				
			1	2
			Gewicht	Polsterung
1	Gewicht der Brille	Gewicht		
2	Wie bequem ist die Polsterung?	Polsterung	0	
			Summe	
			0	

Anhang 3: Vorlage zur Gewichtung der Unterkriterien der Qualität der Immersion

Gewichtung Qualität der Immersion						
		1	2	3	4	5
		1	Qualität der grafischen Texturen	Präzession des Trackings	FPS	Sichtfeld
1	Wie aufwändig ist die Grafik der Simulation	Displayauflösung				
2	Wie präzise kann die Simulation meinen Kopfbewegungen folgen					
3	Präzession des Trackings					
4	Bilder pro Sekunde	FPS				
5	Welches Sichtfeld bietet die Darstellung durch die Brille	Sichtfeld				
						Summe

Anhang 4: Vorlage zur Gewichtung der Unterkriterien der Benutzerfreundlichkeit

Gewichtung Benutzerfreundlichkeit					
			1	2	3
			Inbetriebnahme	Bedienbarkeit	Dokumentation
	Wie leicht fällt die erste Inbetriebnahme	Inbetriebnahme			
1	Inbetriebnahme				
	Wie leicht fällt die der Umgang bzw. die Handhabung mit der Brille	Bedienbarkeit			
2	Brille		0		
	Welche Hilfestellung wird vom Hersteller bereitgestellt	Dokumentation			
3	Hersteller bereitgestellt		0	0	
					Summe
					0
					0
					0

Anhang 5: Durchgeführte Gewichtung aller Hauptkriterien

Gewichtungsgesamt												
		1	2	3	4	5	6	7	8	9	10	
					Qualität der Immersion	Benutzerefreundlichkeit	Mobilität des Systems (Kabel, PC notwendig)	Open Source Software	Open Hardware	Verfügbare Software	Entwicklung für diese Plattform	Summe
Anschaffung als auch evtl. 1 laufende Kosten 2 Langlebigkeit der Brille 3 siehe Unterkriterien 4 siehe Unterkriterien 5 siehe Unterkriterien		Kosten	Robustheit	Tragekomfort								
			1	1	4	5	1	1	1	1	1	7
		1		2	4	5	2	2	8	9	2	4
		1	2		4	5	3	3	8	9	3	3
Ist ein PC nötig? Wird die Bewegungsfreiheit durch Kabel 6 eingeschränkt?		4	4	4		4	4	4	4	4	4	9
		5	5	5	4		6	7	8	9	10	3
Ist der Software Code öffentlich?												
7 zugänglich?		1	2	3	4	7	5		8	9	10	1
Werden die verwendeten, technischen Komponenten vom 8 Hersteller offgelegt?												
		1	8	8	4	8	5	8		8	8	6
9 Anzahl und Art		1	9	9	4	9	5	9	8		9	5
Wie leicht ist eine native Entwicklung 10 möglich?		1	2	3	4	10	5	10	8	9		2

Anhang 6: Durchgeführte Gewichtung der Unterkriterien des Tragekomforts

Gewichtung Tragekomfort				
			1	2
			Gewicht	Polsterung
1	Gewicht der Brille	Gewicht		1
2	Wie bequem ist die Polsterung?	Polsterung	1	
			Summe	
				1
				0

Anhang 7: Durchgeführte Gewichtung der Unterkriterien der Qualität der Immersion

Gewichtung Qualität der Immersion						
	1	2	3	4	5	
		Qualität der grafischen Texturen				
1	Displayauflösung	1	3	4	1	Summe
Wie aufwändig ist die Grafik der Simulation	Qualität der grafischen Texturen		3	4	5	
Wie präzise kann die Simulation meinen Kopfbewegungen folgen	Präzession des Trackings		4	3		
4 Bilder pro Sekunde	FPS				4	
Welches Sichtfeld bietet die Darstellung						
5 durch die Brille	Sichtfeld					

Gewichtung Benutzerfreundlichkeit						
			1	2	3	
			Inbetriebnahme	Bedienbarkeit	Dokumentation	Summe
	Wie leicht fällt die erste Inbetriebnahme			2	2	0
	Wie leicht fällt die der Umgang bzw. die Handhabung mit der Brille				2	2
	Welche Hilfestellung wird vom Hersteller bereitgestellt	Dokumentation	2	2		0

Anhang 9: Vorlage zur Gewichtung der Unterkriterien des Tragekomforts mit dem AHP

CR Value = #DIV/0!			
Pairwise comparisons			
Item Number	Item Number		
	Item Description	Gewicht	
1	Gewicht der Brille	1,00	Polsterung
2	Wie bequem ist die Polsterung?		1,00
	Sum	1,00	

Anhang 10: Vorlage zur Gewichtung der Unterkriterien der Qualität der Immersion mit dem AHP

		CR Value =		-0,893		Reasonable Consistency	
		Pairwise comparisons					
Item Number	Item Number	Item Description	1	2	3	4	5
		Item Description	1	2	3	4	5
1		Displayauflösung	1,00				
	Wie aufwändig ist die Grafik der Simulation	Qualität der Texturen		1,00			
2	Wie präzise kann die Simulation meinen Kopfbewegungen folgen	Präzession des Trackings			1,00		
		FPS				1,00	
4	Bilder pro Sekunde						
	Welches Sichtfeld bietet die Darstellung durch die Brille	Sichtfeld					1,00
5		Sum	1,00				

Anhang 11: Vorlage zur Gewichtung der Unterkriterien der Benutzerfreundlichkeit mit dem AHP

		Pairwise comparisons		CR Value = -1,724	Reasonable Consistency	
Item Number		Item Number	Item Description	1	2	3
1	Wie leicht fällt die erste Inbetriebnahme	Inbetriebnahme		1,00		
2	Wie leicht fällt die der Umgang bzw. die Handhabung mit der Brille	Bedienbarkeit			1,00	
3	Welche Hilfestellung wird vom Hersteller bereitgestellt	Dokumentation				1,00
		Sum		1,00		

Anhang 13: Durchgeführte Gewichtung der Unterkriterien des Tragekomforts mit dem AHP

CR Value = $\frac{\sum \text{DIV}}{0!}$				
Pairwise comparisons				
Item Number	Item Number	Item Description	Gewicht	Polsterung
1	1	Gewicht der Brille	1,00	0,50000
2	2	Wie bequem ist die Polsterung?	2,00	1,00
5	5	Sum	3,00	1,50

Anhang 14: Durchgeführte Gewichtung der Unterkriterien der Qualität der Immersion mit dem AHP

		CR Value = 0,090		Reasonable Consistency			
		Pairwise comparisons					
Item Number	Item Number		1	2	3	4	5
	Item Description	Displayauflösung		Qualität der Texturen	Präzession des Trackings	FPS	Sichtfeld
1		Displayauflösung	1,00	0,50000	0,33333	0,50000	2,00000
	Wie aufwändig ist die Grafik der Simulation						
2	Wie präzise kann die Simulation meinen Kopfbewegungen folgen		2,00	1,00	0,16667	0,25000	0,50000
3	Präzession des Trackings		3,00	6,00	1,00	2,00000	4,00000
4	Bilder pro Sekunde Welches Sichtfeld bietet die Darstellung durch die Brille		2,00	4,00	0,50	1,00	3,00000
5	Sichtfeld		0,50	2,00	0,25	0,33	1,00
5	Sum		8,50	13,50	2,25	4,08	10,50

Anhang 15: Durchgeführte Gewichtung der Unterkriterien der Benutzerfreundlichkeit mit dem AHP

CR Value = 0,000			Reasonable Consistency		
Pairwise comparisons					
Item Number	Item Number	Item Description	1	2	3
1	1	Inbetriebnahme	1,00	0,16500	1,00000
2	2	Wie leicht fällt die der Umgang bzw. die Handhabung mit der Brille	6,06	1,00	6,00000
3	3	Welche Hilfestellung wird vom Hersteller bereitgestellt	1,00	0,17	1,00
5	5	Sum	8,06	1,33	8,00

Anhang 16: Durchgeführte Gewichtung der Hauptkriterien mit dem AHP

		Pairwise comparisons		Reasonable Consistency							
Item Number	Item Description	1	2	3	4	5	6	7	8	9	10
Anschaffung als auch evtl.	Kosten		Robustheit								
1 laufende Kosten	Kosten	1,00	4,00000								8,00000
2 Langlebigkeit der Bauteile	Robustheit	0,25	1,00	5,00000	0,33000						6,00000
3 siehe Unterkriterien	Tragekomfort	0,20	1,00	2,00000	0,25000	0,50000	8,00000	2,00000	3,00000		2,00000
4 siehe Unterkriterien	Qualität der Immersion	3,03	4,00	6,00	0,16667	0,25000	0,25000	0,12500	0,50000	9,00000	7,00000
5 siehe Unterkriterien	Benutzerefreundlichkeit	2,00	4,00	5,00	1,00	2,00000	9,00000	4,00000	5,00000	8,00000	6,00000
Ist ein PC nötig?					0,50	1,00	8,00000	3,00000	4,00000	9,00000	
Wird die Bewegungsfreiheit durch Kabel eingeschränkt?											
6 eingeschränkt?	Mobilität des Systems	0,13	8,00	4,00	0,11	0,13	1,00	0,12500	0,12500	0,50000	3,00000
Ist der Software Code öffentlich?											
7 zugänglich?	Open Source Software	0,50	8,00	4,00	0,25	0,33	8,00	1,00	2,00000	8,00000	6,00000
Werden die verwendeten, technischen Komponenten vom Hersteller offengelegt?											
8 Anzahl und Art	Open Hardware	0,33	2,00	3,03	0,20	0,25	8,00	0,50	1,00	0,25000	0,33000
9 Wie leicht ist eine native Entwicklung möglich?	Verfügbare Software	0,11	0,13	0,33	0,13	0,11	2,00	0,13	4,00	1,00	2,00000
10	Entwicklung für diese Plattform	0,13	0,17	0,50	0,14	0,17	0,33	0,17	3,03	0,50	1,00
	Sum	7,67	31,79	30,86	3,08	4,94	44,71	11,29	22,99	47,25	41,33

Anhang 17: Programmiercode vierte Station

```

228 <a-assets>
229   <a-asset-item id="why-male-models" src="man.dae"></a-asset-item>
230 </a-assets>
231
232 <a-entity id="model" position="0 0 -2">
233   <a-animation attribute="rotation" from="0 -30 0" to="0 330 0" dur="15000" easing="linear" repeat="infinite"></a-animation>
234   <a-collada-model position="- .35 0 .55" rotation="0 -20 0" scale="1.5 1.5 1.5" src="#why-male-models"></a-collada-model>
235   <a-image src="radial-shadow-2.png" rotation="90 0 0" scale="0.5 0.5 0.5"></a-image>
236 </a-entity>
237
238 <a-curvedimage id="mozvr-logo" src="mozvr.png" radius="5.7" theta-length="36" height="1" position="0 2.6 0" opacity="0.5">
239   <a-animation attribute="rotation" from="0 10 0" to="0 200 0" begin="500" dur="1000"></a-animation>
240 </a-curvedimage>
241
242 <a-image id="price" src="price.png" width="7" height="3.5" scale="0.2 0.2 0.2">
243   <a-animation attribute="position" from="0 2.8 -6" to="2.25 2.8 -6" begin="1000" dur="1000"></a-animation>
244 </a-image>
245
246 <a-cylinder id="goggles" color="#101010" height="0.02" radius="0.8">
247   <a-animation attribute="rotation" from="-270 0 0" to="-90 0 0" dur="750" begin="1000" fill="both"></a-animation>
248   <a-animation attribute="position" from="8 0 -9" to="8 3.5 -9" dur="750" begin="1000" fill="both"></a-animation>
249   <a-image src="goggles.png" width="2" height="1" rotation="90 0 0" position="0 -.05 0" scale=".4 .4 .4"></a-image>
250 </a-cylinder>
251
252 <a-curvedimage id="stereoscopic-fall-collection-text" src="fall.png"
253   radius="5.7" theta-length="18" height=".45" position="0 0.9 0" scale=".4 .4 .4">
254   <a-animation attribute="rotation" from="0 180 0" to="0 210 0" begin="750" dur="1000"></a-animation>
255 </a-curvedimage>
256
257 <a-curvedimage id="shoes" src="shoes.png" radius="5.7" theta-length="18" height=".8" position="0 0.9 0" scale=".4 .4 .4">
258   <a-animation attribute="rotation" from="0 180 0" to="0 130 0" begin="750" dur="1000"></a-animation>
259 </a-curvedimage>
260
261 <a-entity>
262   <a-cylinder position="0 0.5 0" radius="4" height="1.6" side="back" open-ended="true" color="#E5E5E5"></a-cylinder>
263 </a-entity>

```