



UNIVERSITÄT
KOBLENZ · LANDAU

Fachbereich 4: Informatik

Gestaltung von Innenräumen mit Hilfe eines hybriden AR-/VR-Systems

Bachelorarbeit

zur Erlangung des Grades Bachelor of Science (B.Sc.)
im Studiengang Computervisualistik

vorgelegt von
Benedikt Geiger

Erstgutachter: Prof. Dr.-Ing. Stefan Müller
(Institut für Computervisualistik, AG Computergraphik)

Zweitgutachter: Bastian Kraye, M.Sc.
(Institut für Computervisualistik, AG Computergraphik)

Koblenz, im Juni 2018

Zusammenfassung

Das Ziel dieser Bachelorarbeit bestand darin, die Verbindung zwischen den Technologien Augmented und Virtual Reality zu veranschaulichen und ein sinnvolles Zusammenspiel der beiden Darstellungsformen zu kreieren. Hierfür wurde eine Anwendung im Bereich der Innenarchitektur implementiert, bei welcher man einen Raumplan mittels Augmented Reality intuitiv gestalten und sich anschließend einen realitätsnahen Eindruck des eingerichteten Zimmers mit einer Virtual Reality Simulation machen kann. Auf Basis des nötigen Grundwissens wurde ein Konzept für dieses Projekt ausgearbeitet und anschließend mit verschiedenen Entwicklungssystemen realisiert. Diese Implementierung wurde im Rahmen einer Evaluationsreihe getestet und darauffolgend optimiert. Das Ergebnis bestätigt die Annahme, dass sich Augmented und Virtual Reality mit ihren jeweiligen Stärken evident miteinander verbinden lassen. Diese Arbeit ist sowohl für Studierende im Bereich Informatik als auch für Interessenten an innovativen Lösungen relevant.

Abstract

This Bachelor thesis illustrates the connection between the technologies Augmented and Virtual Reality and creates an expedient interdependency of the two forms of presentation. For this purpose, an application in the area of interior design has been implemented, where designing a room can be made more intuitive by using Augmented Reality, as it provides a realistic impression of the planned apartment with a Virtual Reality simulation. Based on the relevant knowledge, a project-concept has been drafted and realized by using several development systems. During a series of tests this implementation has been evaluated and subsequently optimized. The result confirms the assumption, that Augmented and Virtual Reality with their strengths can be combined to an evident solution. This thesis is relevant for computer science students as well as for people interested in innovative solutions.

Inhaltsverzeichnis

Abkürzungsverzeichnis	iii
Tabellenverzeichnis	iv
Abbildungsverzeichnis	v
1 Einleitung	1
1.1 Relevanz der Thematik	1
1.2 Zielsetzung und zentrale Fragestellung der Arbeit	2
1.3 Vorgehensweise und Aufbau der Arbeit	2
2 Begriffliche und theoretische Grundlagen	4
2.1 Extended Reality	4
2.1.1 Reality-Virtuality Continuum	4
2.1.2 Augmented Reality	5
2.1.3 Virtual Reality	10
2.1.4 Verbindung von Augmented und Virtual Reality	17
2.2 Computergrafik	19
2.2.1 Texturen und Materialien	19
2.2.2 Beleuchtung	20
2.3 Innenarchitektur	22
2.3.1 Richtlinien für die Platzierung von Einrichtungsgegenständen . .	23
2.3.2 Einrichtung von Hotelzimmern	24
3 Augmented und Virtual Reality im Bereich Innenarchitektur	26
3.1 Anforderungen und Kriterien an eine Augmented Reality Innenarchitektur-Anwendung	26
3.2 Anforderungen und Kriterien an eine Virtual Reality Innenarchitektur-Anwendung	27
4 Konzeption der Anwendung	30
4.1 Grundlegende Idee	30
4.2 Ziele und Anforderungen	31
4.3 Planung der Anwendung	33
4.3.1 Hardware- und Software-Setup	33
4.3.2 3D-Modellierung und Texturierung	34
4.3.3 Augmented Reality Teil	35
4.3.4 Übergang: Augmented Reality zu Virtual Reality	36
4.3.5 Virtual Reality Teil	36
4.3.6 Übergang: Virtual Reality zu Augmented Reality	37

5	Implementation der Anwendung	38
5.1	Setup	38
5.1.1	Hardware-Setup	38
5.1.2	Software-Setup	41
5.2	Darstellung der Objekte	43
5.2.1	Modellierung	43
5.2.2	Texturierung und Materialien	46
5.3	Augmented Reality Teil	47
5.3.1	Marker-Tracking und Anzeige der Augmented Reality Szene	47
5.3.2	User Interface	52
5.4	Übergang: Augmented Reality zu Virtual Reality	55
5.4.1	Basis für den Szenenwechsel	55
5.4.2	Bereitstellen der Augmented Reality Informationen für Virtual Reality	57
5.4.3	Auswahl einer Virtual Reality Fortbewegungsart	61
5.5	Virtual Reality Teil	62
5.5.1	Abrufen der Augmented Reality Informationen in Virtual Reality	63
5.5.2	Fortbewegung in Virtual Reality	65
5.5.3	Visuelle und akustische Darstellung der Szene	67
5.6	Übergang: Virtual Reality zu Augmented Reality	70
5.6.1	User Interface	70
5.6.2	Wechsel von Virtual zu Augmented Reality	70
6	Evaluation und Optimierung der Anwendung	72
6.1	Zielsetzung und Planung der Evaluation	72
6.2	Durchführung der Evaluation	73
6.3	Auswertung der Testergebnisse und Optimierung der Anwendung	74
7	Ausblick und Fazit	83
7.1	Ausblick	83
7.2	Fazit	85
A	Anhang	87
A.1	Evaluierungsbogen	87

Abkürzungsverzeichnis

XR	Extended Reality oder auch X-Reality
AR	Augmented Reality
VR	Virtual Reality
MR	Mixed Reality
UI	User Interface
HMD	Head-Mounted Display
FOV	Field of View
DOF	Degree of Freedom
SDK	Software Development Kit

Tabellenverzeichnis

1 Übersicht der relevanten Fortbewegungsarten	16
2 Multimodulares UI in der AR-Ansicht	36
3 Technische Daten von Oculus Rift CV1 und Logitech C310 HD Webcam	39
4 Überblick über AR-SDKs	42
5 Ursprüngliche und überarbeitete Sprachbefehle	74

Abbildungsverzeichnis

1	Reality-Virtuality Continuum	4
2	AR Video See-through am Beispiel eines Smartphones und Optical See-through am Beispiel einer AR-Brille	5
3	Kamera-Pose	7
4	Beziehung von Kamera- und Welt-Koordinatensystem	8
5	AR-Tracking-Pipeline	8
6	Tangible UI	10
7	Ray Casting	10
8	UI: Info-Bubble	10
9	VR-HMD am Einsatzbeispiel eines VR-Zeichenprogramms	11
10	VR-Stereo-Display	12
11	Stereo-Rendering mit Off-Axis Methode	12
12	Degrees of Freedom	13
13	VR-Fortbewegungsart: Room Scale	13
14	VR-Fortbewegungsart: Free Locomotion und Teleportation	14
15	VR-Fortbewegungsart: Walking in Place	15
16	UI: Game Element UI am Beispiel eines Flugsimulators	17
17	Hybrides AR/VR Video See-through HMD	18
18	Texture-Map: Metalness Map	20
19	Texture-Map: Specular Map	20
20	Texture-Map: Normal Map	20
21	Texture-Map: Ambient Occlusion Map	20
22	Lichtquelle: Point Light	21

23	Lichtquelle: Spot Light	21
24	Lichtquelle: Directional Light	22
25	Richtlinien für die Platzierung von Einrichtungsgegenständen	24
26	Einrichtungsgegenstand-Symbole	24
27	Kollaborative AR-Architektur-Anwendung	27
28	VR-Architektur-Anwendung	28
29	Konzeptzeichnung: Hybrides AR-/VR-HMD	33
30	Grundriss und Eindrücke des Marriott Residence Inn Standard Studio Room	34
31	Konzeptzeichnung: AR-Ansicht	35
32	FOV ohne und mit Weitwinkel-Objektiv	39
33	Aufbau des hybrides AR-/VR-HMDs	40
34	Aufbau der Arbeitsunterlage	41
35	Grundriss des virtuellen Hotelzimmers	44
36	Modell des virtuellen Hotelzimmers in AR und VR	45
37	3D-Modelle der Einrichtungsgegenständen	45
38	Erstellung eines Materials	47
39	3D-Modelle mit Materialien	47
40	Vuforia-Merkmalpunkte	48
41	Marker-Koordinatensystem	49
42	Szenen-Hierarchie im AR-Modus	51
43	Demonstration der AR-Szene	51
44	AR-Assistenzfunktion: Positionierungsvorschläge	53
45	AR-Assistenzfunktion: Grundriss-Ansicht	54
46	AR-Assistenzfunktion: Massangaben	55

47	reticleCursor1 und reticleCursor2	57
48	Verfärbung der Info-Bubble	58
49	Veranschaulichung der Informationsübergabe von AR zu VR	61
50	UI zur Auswahl einer VR-Fortbewegungsart	62
51	Demonstration der Informatinsübergabe von AR zu VR	64
52	Controller-Eingabe für Free Locomotion	66
53	Komponenten von Walking in Place	67
54	Szenen-Beleuchtung und Post Processing	68
55	Sicht-Vignette	69
56	Game Element UI für den Wechsel VR zu AR	70
57	Gegenüberstellung der Evaluationsergebnisse der Testreihen 1 und 2: Sprachsteuerung der AR-Assistenzfunktion	75
58	Grundriss- und komplette 2D-Ansicht	76
59	Gegenüberstellung der Evaluationsergebnisse der Testreihen 1 und 2: Ladezeit für den Wechsel AR zu VR	77
60	Digitale Anleitung in AR	78
61	Gegenüberstellung der Evaluationsergebnisse der Testreihen 1 und 2: Übereinstimmung der Einrichtung in AR und VR	79
62	Gegenüberstellung der Evaluationsergebnisse der Testreihen 1 und 2: Motion Sickness bei den VR-Fortbewegungsarten	81
63	Tool-Tip in VR	82

1 Einleitung

„We’re finally going to be free of the 2D monitor. It’s been a window into virtual reality that we’ve all looked into for 30 or 40 years.“ - Brendan Iribe

1.1 Relevanz der Thematik

Betrachtet man die Entwicklung der Interaktion zwischen Mensch und Computer, so lassen sich drei essentielle Transformationen herausstellen: Während in der Anfangszeit der Computer-Entwicklung in den 1970er Jahren Einzelplatzrechner und Mainframe-Systeme nur via Text-Eingaben bedient werden konnten, stellte das Graphical User Interface Mitte der 1980er Jahre aufgrund seiner intuitiven Benutzerfreundlichkeit einen wesentlichen Meilenstein dar. Mit der zunehmenden Verbreitung von Smartphones seit den 2000er Jahren erlangten nicht nur Touchscreen-Systeme an Bedeutung, sondern die Mensch-Maschine-Interaktion wurde weitreichend in den Alltag integriert (Scoble und Israel, 2017, S. 3).

Neben diesen drei Phasen lässt sich eine vierte, aufkommende Transformation ausmachen, welche vor allem durch zwei Technologien vorangetrieben wird: Augmented und Virtual Reality (Scoble und Israel, 2017, S. 3). Diese computerunterstützten Darstellungsformen haben nach Aussagen von Industrie-Analysten das Potenzial „[to be] the next technological megatrend [and] the next evolution of computing [...]“ (Tobak, Steve, 2015). Der Grund für diese Prognosen ist die durch diese Technologien vorangetriebene Revolution in der Informationsdarstellung und der Interaktion zwischen Mensch und Computer (Scoble und Israel, 2017, S. 7). So greift der Nutzer nicht mehr von außen auf die computergenerierten Darstellungen zu, sondern findet sich selbst in einer digital angereicherten Umgebung wieder, wobei die Grenzen zwischen Realität und Virtualität verschwinden (Scoble und Israel, 2017, S. 3, 4).

Die positiven Effekte dieser vierten Transformation zeichnen sich unter anderem durch eine verbesserte Kommunikation und Informationsübergabe, sowie einem Anstieg in Produktivität und Präzision aus (Scoble und Israel, 2017, S. 7). Dabei lassen sich verschiedene Aufgabenbereiche durch Augmented und Virtual Reality unterschiedlich gut lösen. So wird im Bereich Innenarchitektur Augmented Reality bevorzugt in der Planungs- und Design-Phase eines Innenraums eingesetzt (Wang, 2009, S. 316; Phan und Choo, 2010, S. 16), während sich Virtual Reality vor allem zur realitätsnahen Darstellung eines Gebäudes eignet (Campbell und Wells, 1994, S. 2). Mit dem Ziel das volle Potenzial der vierten Transformation auszuschöpfen, besteht die grundlegende Aufgabe für Entwickler darin, eine Symbiose von Augmented und Virtual Reality aufzubauen und die Stärken der beiden Technologien miteinander zu kombinieren (Billinghurst u. a., 2009, S. 6). Aufgrund der hohen Relevanz dieser Thematik und des damit einher-

gehenden Potenzials für Entwickler beschäftigt sich diese Arbeit mit der Gestaltung von Innenräumen mit Hilfe eines hybriden AR-/VR-Systems.

1.2 Zielsetzung und zentrale Fragestellung der Arbeit

Das Primärziel, die Beziehung zwischen Augmented Reality und Virtual Reality genauer zu beleuchten, kann in folgende Teilaufgaben untergliedert werden: Für das bessere Verständnis werden dem Leser die für die Arbeit relevanten theoretischen Grundlagen vermittelt. Anschließend werden anhand verschiedener Beispiele die Anforderungen und Kriterien an Augmented und Virtual Reality Anwendungen im Bereich Innenarchitektur erläutert. Die Kernaufgabe dieser Arbeit besteht darin, ein hybrides Augmented und Virtual Reality System zur Gestaltung von Innenräumen zu entwickeln. Neben der Konzeption wird die Implementation der Anwendung realisiert und die Ergebnisse dokumentiert. Anhand von Probandentests wird das fertiggestellte System evaluiert und anschließend etwaige Optimierungen vorgenommen. Abschließend werden alle gewonnenen Resultate in dieser Ausarbeitung präsentiert und zusammengefasst. Folgende Fragen gilt es dabei zu beantworten:

1. Was sind die technologischen Grundlagen von Augmented und Virtual Reality und wie hängen diese Technologien zusammen?
2. Was sind die Kriterien und Anforderungen an Augmented und Virtual Reality Anwendungen im Bereich Innenarchitektur?
3. Welches Konzept liegt der im Rahmen dieser Bachelorarbeit erstellten Anwendung zugrunde?
4. Welche Implementationsschritte wurden für die Entwicklung dieser Anwendung durchgeführt?
5. Zu welchen Ergebnissen führten die im Rahmen einer Evaluation durchgeführten Probandentests der Anwendung?
6. Welche Änderungen wurden zur Optimierung der Anwendung anhand der Evaluationsergebnisse durchgeführt?
7. Wie lassen sich die Ergebnisse der Arbeit zusammenfassen und welches Fazit kann man daraus ziehen?

1.3 Vorgehensweise und Aufbau der Arbeit

Um die im vorhergehenden Abschnitt dargestellten Fragen effektiv zu beantworten, ist folgender Aufbau für die Abhandlung festgelegt worden:

- Für ein besseres Verständnis der Arbeit werden im zweiten Kapitel die Grundlagen von X-Reality, Elementen aus der Computergrafik sowie Richtlinien der Innenarchitektur erklärt.

1. EINLEITUNG

- Kapitel drei stellt anhand verschiedener Beispiele die Anforderungen und Kriterien für Augmented und Virtual Reality Anwendungen im Bereich Innenarchitektur dar.
- Im vierten Kapitel wird die für die Anwendung grundlegende Konzeption erläutert.
- Die für die Ideen und Zielsetzungen von Kapitel vier durchgeführten Implementationsschritte werden im fünften Kapitel erklärt und veranschaulicht.
- Das sechste Kapitel geht auf die Durchführung und die Ergebnisse der Evaluation sowie auf die daraus resultierenden Optimierungsschritte ein.
- Auf Basis der in Kapitel sechs gewonnenen Ergebnisse werden in Kapitel sieben die vorgenommenen Optimierungsschritte der Anwendung dargelegt.
- Der finale Status und die gewonnenen Ergebnisse der Anwendung sind Themen des siebten Kapitels.
- Im letzten Kapitel werden die wichtigsten Erkenntnisse nochmals kurz zusammengefasst und Resümee über die in der Arbeit gewonnenen Ergebnisse und Schlussfolgerungen gezogen.

2 Begriffliche und theoretische Grundlagen

Mit dem Ziel, der Arbeit besser folgen zu können und eine Wissensbasis zu schaffen, werden in diesem Kapitel die theoretischen und begrifflichen Grundlagen zu den Bereichen Extended Reality, Computergrafik und Innenarchitektur erläutert.

2.1 Extended Reality

Unter dem von Charles Wyckoff geprägten Begriff Extended Reality oder auch X-Reality (XR) versteht man ein Kollektivum, unter welchem alle computergenerierten Darstellungsformen, wie Augmented Reality (AR) oder Virtual Reality (VR), zusammengefasst werden (Honkanen u. a., 2018, S. 19, 121; Mann u. a., 2018, S. 2). Im Folgenden sollen diese technischen Bezeichnungen anhand des Reality-Virtuality Continuums veranschaulicht und insbesondere auf die Systeme Augmented Reality und Virtual Reality sowie deren Beziehung eingegangen werden.

2.1.1 Reality-Virtuality Continuum

Das Reality-Virtuality Continuum (s. Abb. 1) von Professor Paul Milgram beschreibt verschiedene Wahrnehmungsstufen der Realität mit der zunehmenden Anreicherung virtueller Komponenten. Während die Real Environment eine vollkommen reale Umgebung ohne virtuelle Einflüsse darstellt, wird im Kontrast dazu die Virtual Environment komplett durch virtuelle und künstliche Elemente präsentiert. Zwischen den beiden Extrema gibt es verschiedene Stufen, die unter dem Sammelbegriff Mixed Reality definiert werden (Milgram u. a., 1995, S. 283). Während der Nutzer auf den ersten Stufen des Reality-Virtuality Continuums (auf dem Strahl links) seine reale Umgebung noch wahrnimmt und die digitalen Informationen eine Erweiterung zur Realität darstellen, wird hingegen auf den höheren Stufen des Spektrums (auf dem Strahl rechts) die Realität immer weiter ausgeblendet und durch virtuelle Objekte ersetzt (Milgram u. a., 1995, S. 283).



Abbildung 1: Reality-Virtuality Continuum

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

Die bekanntesten virtuellen Wahrnehmungsstufen des Reality-Virtuality Continuums stellen die Augmented und Virtual Reality dar.

2.1.2 Augmented Reality

Der 1992 von dem Forscher Tom Caudell geprägte und 1994 von Paul Milgram aufgegriffene Begriff Augmented Reality – zu Deutsch: Erweiterte Realität – beschreibt eine Zwischenstufe im Reality-Virtuality Continuum (Siltanen, 2012, S. 16). Hierbei wird die reale Welt mit virtuellen Informationen überlagert, sodass für den Benutzer Realität und digitale Informationen verschmelzen (Ludwig und Reimann, 2005, S. 3, 4). Für die Erweiterte Realität sind drei Charakteristiken entscheidend: Realität und Virtualität werden miteinander vereint, der Nutzer kann in Echtzeit interagieren und die dargestellten Informationen befinden sich im dreidimensionalen Raum (Siltanen, 2012, S. 17).

Bei der Realisierung von AR gibt es zwei gängige Anzeigarten: Bei einem Video See-through wird die reale Umgebung mit einer Kamera eingefangen und für den Benutzer mit den überlagerten virtuellen Informationen auf einem Display angezeigt. Diese Methode lässt sich leicht auf Mobile Devices implementieren und ist somit die derzeit am weitesten verbreitete Anzeige-Art für AR. Mit einem Optical See-through sieht der Anwender seine Umgebung direkt durch ein durchsichtiges Display, auf das die virtuellen Informationen zusätzlich projiziert werden. Letztere Anzeigart wird vor allem bei AR-Brillen verwendet (s. Abb. 2) (Milgram u. a., 1995, S. 284).



Abbildung 2: AR Video See-through am Beispiel eines Smartphones und Optical See-through am Beispiel einer AR-Brille (Lancaster, Luke, 2016; Fremont, 2017) [Foto]

Für eine korrekte Überlagerung von Realität und Virtualität müssen eine Vielzahl an Faktoren berücksichtigt werden. So ist die Position und Orien-

tierung der Kamera, welche unter dem Begriff Kamera-Pose zusammenfasst werden, entscheidend für die korrekte Positionierung, Skalierung und Ausrichtung der angezeigten virtuellen Objekte (Kanbara u. a., 2000, S. 255; Siltanen, 2012, S. 47). Die Berechnung der relativen Kamera-Pose wird als Tracking bezeichnet. Im Laufe der Zeit haben sich verschiedene Tracking-Verfahren entwickelt, welche jeweils Vor- und Nachteile bieten und daher entsprechend des jeweiligen Anwendungsbereichs ausgewählt werden sollten. Da viele AR-Anwendungen zur Darstellung des Bildes eine Kameraaufnahme verwenden, eignet sich für diesen Bereich das optische Tracking (Siltanen, 2012, S. 38). Beim optischen Tracking wird die Pose anhand von visuellen Informationen und Merkmalen bestimmt, wobei eine hohe Genauigkeit der Pose-Berechnung erzielt werden kann (Wang, 2009, S. 315; Siltanen, 2012, S. 39). Weitere Vorteile dieses Verfahrens sind, dass es resistent gegenüber magnetischen oder akustischen Störsignalen ist. Allerdings ist zu beachten, dass das optische Tracking auf den für die Kamera sichtbaren Bereich eingeschränkt ist und die Genauigkeit der Berechnungen durch störende Lichtquellen vermindert werden kann (Wang, 2009, S. 315).

Eine ebenso präzise sowie verhältnismäßig leicht zu implementierende Methode ist das Tracking von Markern (Siltanen, 2012, S. 39). Unter einem Marker im Kontext der AR wird ein vordefinierter Anhaltspunkt verstanden, der von einem Computer-System in einem Videobild unter Anwendung von Techniken der Bildverarbeitung, Mustererkennung und Bilderkennung identifiziert werden kann (Siltanen, 2012, S. 39). Ziel ist es, einen vordefinierten Marker anhand seiner Merkmalspunkte zu finden und die Kamera-Pose in Relation zu dem gefundenen Marker zu bestimmen. Der getrackte Marker kann anschließend mit virtuellen Informationen überlagert werden. Werden für eine Anwendung mehrere Marker verwendet, so spricht man von einem Dictionary (Garrido-Jurado u. a., 2014, S. 2280). Damit ein Marker möglichst gut detektiert werden kann, sollten folgende Kriterien beachtet werden: Da sich Schwarz-Weiß-Marker gut mittels Schwellwert-Extraktionen erkennen lassen, besitzt die Detektion dieses Marker-Typs eine verhältnismäßig hohe Fehlertoleranz bei schlechten und wechselhaften Lichtverhältnissen (Doerner u. a., 2013, S. 104, 105, 256). Des Weiteren steigt die Stabilität des Trackings mit der Anzahl der Merkmalspunkte pro Marker-Muster (Siltanen, 2012, S. 57). Da Ecken bei der Detektion von Markern als Knotenpunkte identifiziert werden, werden meist viereckige Marker verwendet (Siltanen, 2012, S. 40). Bei diesem Marker-Typ wird die Pose anhand der vier Eckpunkte bestimmt und der Marker anhand seines Zentrums durch die Berechnung der Homographie identifiziert (Koehler u. a., 2011, S. 37; Garrido-Jurado u. a., 2014, S. 2281).

Das Tracking eines AR-Markers verläuft im Allgemeinen wie folgt: Zunächst wird das Kamerabild vom System erfasst und mittels Thresholding und

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

Kantendetektion verarbeitet. Des Weiteren wird im Rahmen der Vorverarbeitung das Bild entzerrt, Ecken und Linien detektiert und das Bild in ein Intensitätsbild umgewandelt. Als nächstes gilt es, potenzielle Marker zu finden. Werden für die Anwendung viereckige Marker verwendet, so werden alle nicht-eckigen Formen sowie zu kleine Objekte als potenzielle Marker ausgeschlossen. Die in den Referenz-Layouts der Marker enthaltenen Identifikationspunkte werden gemäß der Formel $f : M_{ref} \rightarrow M_{hyp}$ auf die potenziell gefundenen Marker gemappt. Anschließend werden die entsprechenden Referenz-Marker identifiziert und decodiert (Koehler u. a., 2011, S. 38; Siltanen, 2012, S. 40 - 42). Im nächsten Schritt gilt es, die extrinsischen Kamera-Parameter, welche sich aus Kamera-Position und -Orientierung im Raum zusammensetzen, in Relation zu den gefundenen Markern zu bestimmen (s. Abb. 3) (Siltanen, 2012, S. 47).

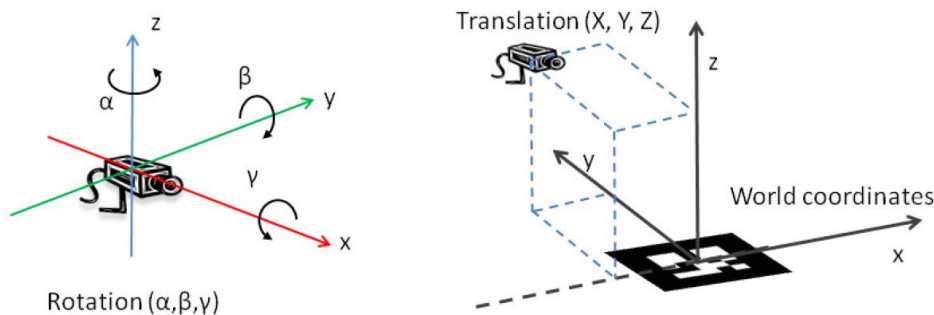


Abbildung 3: Kamera-Pose (Siltanen, 2012, S. 47) [Graphik]

Während dieses Vorgangs wird die Model-View Matrix berechnet. Mit Hilfe dieser Matrix wird die Position eines Punktes w im Welt-Koordinatensystem zu seinem korrespondierenden Punkt c im Kamera-Koordinatensystem transformiert. Damit wird das virtuelle Objekt korrekt im Kamera-Bild wiedergegeben (Kanbara u. a., 2000, S. 259). Zusammengesetzt ist die 4×4 Model-View Matrix M aus einer 3×3 Rotations-Matrix R und einem dreidimensionalen Translations-Vektor t (Kanbara u. a., 2000, S. 259), welche die relative Drehung und Translation des virtuellen Objekts in das Kamera-Koordinatensystem darstellen (Solem, 2012, S. 91, 92). Die Transformation setzt sich dabei aus homogenen Koordinaten zusammen.

$$M = \left[\begin{array}{ccc|c} \mathbf{R} & \mathbf{t} & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

Die gesuchte Position c eines virtuellen Objekt-Punktes im Kamera-Koordinatensystem ergibt sich aus dem Produkt der Model-View Matrix M und der Position des entsprechenden Punktes im Welt-Koordinatensystem w (s. Abb. 4) (Kanbara u. a., 2000, S. 259):

$$c = M * w$$

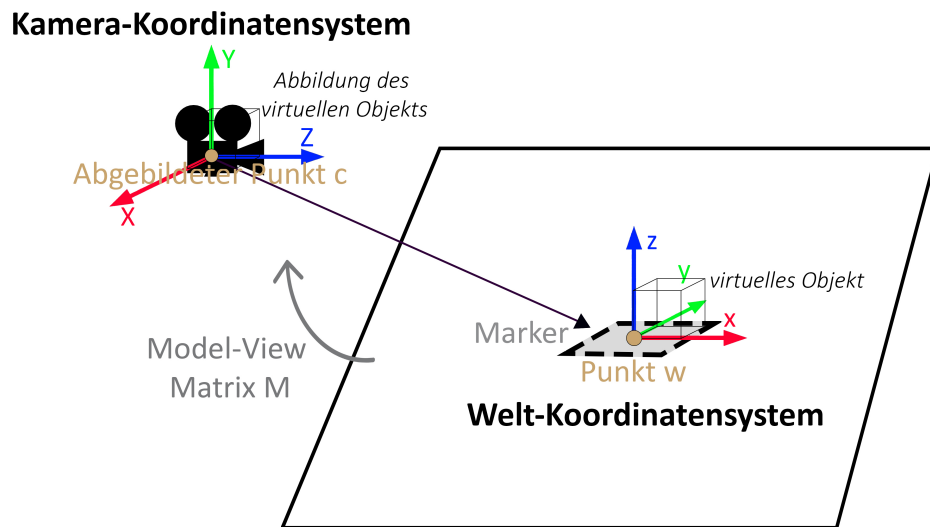


Abbildung 4: Beziehung von Kamera- und Welt-Koordinatensystem

Zur besseren Übersicht ist der Ablauf der AR-Tracking-Pipeline in der Grafik 5 skizziert:

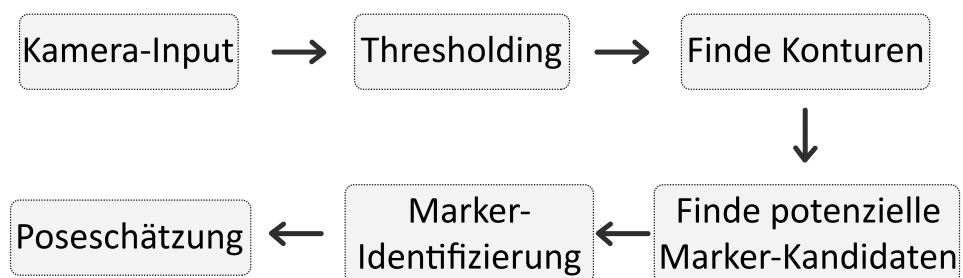


Abbildung 5: AR-Tracking-Pipeline

Dieses Vorgehen wird kontinuierlich wiederholt und stellt das Fundament für eine korrekte Überlagerung der Realität mit virtuellen Informationen dar. Potenziell auftretende Fehler bei der AR-Anzeige äußern sich meist durch das als Vertex Jitter bezeichnete Zittern der virtuellen Objekte oder deren

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

totalen Ausfall. Diese Fehler treten dadurch auf, dass die Kamera-Pose aufgrund schlechter Lichtverhältnisse, zu schneller Kamera-Bewegungen oder Verdeckung der Marker nicht exakt oder gar nicht bestimmt werden kann (Garrido-Jurado u. a., 2014, S. 2286).

Mit dem Ziel, den Nutzer nicht nur auf eine passive, beobachtende Rolle zu beschränken, gibt es viele verschiedene Interaktionsarten für AR-Anwendungen, welche unterschiedliche Aktionsmöglichkeiten haben (Poupyrev u. a., 2002, S. 44). Sowohl physische Komponenten, als auch die visuelle und auditive Wiedergabe sowie die Art der Interaktion sind für ein AR User Interface (UI) entscheidend (Billinghurst u. a., 2005, S. 17).

So ist die Interaktion mit realen Objekten, wie beispielsweise Markern, eine für den Menschen intuitive und natürliche Handlungsweise, welche das kontextuelle Verständnis für die Anwendung und deren Arbeitsfluss fördert (Slay u. a., 2001, S. 71; Poupyrev u. a., 2002, S. 45; Billinghurst u. a., 2009, S. 16): Soll beispielsweise das angezeigte virtuelle Objekt rotiert oder translatiert werden, so führt man die dementsprechende Handbewegung mit dem korrespondierenden realen Objekt aus. Zum Skalieren bewegt sich der Anwender auf das Objekt zu oder entfernt sich von diesem (Slay u. a., 2001, S. 71, 72). Anders als mit Eingabegeräten, wie beispielsweise Maus und Tastatur, muss diese Interaktion nicht erst erlernt werden. Des Weiteren können mehrere Handlungen und Aufgaben parallel mit beiden Händen ausgeführt werden (Kato u. a., 2000, S. 112; Billinghurst u. a., 2005, S. 18; Billinghurst u. a., 2009, S. 16). Interaktionen mit realen Objekten, welche auf virtuelle Bezugsobjekte übertragen werden, unterliegen dem UI-Oberbegriff *Tangible User Interface* (s. Abb. 6) (Billinghurst u. a., 2005, S. 17, 18). *Tangible User Interfaces* sind aufgrund ihrer Beziehung zu realen Objekten, welche *Tokens* genannt werden, Teil der MR und weit links in dem Reality-Virtuality Continuum angeordnet (Billinghurst u. a., 2001, S. 6; Billinghurst u. a., 2005, S. 18). Fundament für ein korrektes Mapping der virtuellen Informationen auf ihre realen Bezugsobjekte ist ein präzises Tracking-Verfahren. So bietet sich für diese UI-Art bei AR-Anwendungen optisches Tracking mit Markern an (Billinghurst u. a., 2005, S. 18). Weitere populäre Interaktions- und Selektions-Arten für AR-Applikationen stellen die Sprachsteuerung und das Ray Casting dar. Unter Letzterem wird das Aussenden eines virtuellen Strahls – typischerweise ausgehend vom Kopf oder den Händen – verstanden, um einen virtuellen Gegenstand auszuwählen (s. Abb. 7) (Slay u. a., 2001, S. 71).

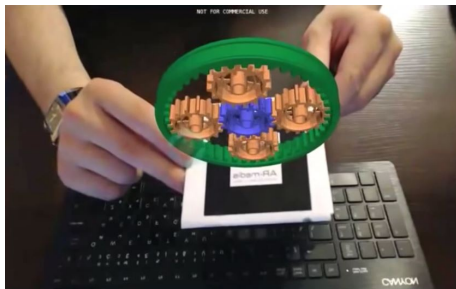


Abbildung 6: Tangible UI (Himix, 2015) [Foto]

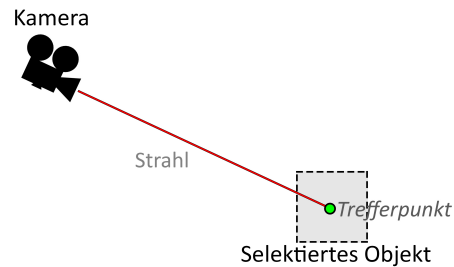


Abbildung 7: Ray Casting

Werden mehrere verschiedene Interaktionsarten innerhalb einer Anwendung vereinigt, so bilden diese ein multimodales UI (Billinghurst u. a., 2009, S. 17).

Ein für AR beliebtes UI-Element stellt die Info-Bubble dar. Diese zeigt dem Nutzer Informationen zu einem bestimmten Objekt virtuell an, um ihn bei seiner Aufgabe zu unterstützen und ein gutes Benutzererlebnis zu bieten (s. Abb. 8) (Bell u. a., 2001, S. 102; Billinghurst u. a., 2005, S. 17). Dabei soll die Beziehung zwischen Info-Bubble und dem jeweiligen Bezugsobjekt klar ersichtlich sein und das UI-Element andere relevante Informationen nicht verdecken (Bell u. a., 2001, S. 103).



Abbildung 8: UI: Info-Bubble

2.1.3 Virtual Reality

Im Gegensatz zur AR nimmt der Betrachter bei der Virtual Reality ausschließlich computergenerierte Informationen wahr. Das Ziel von VR ist es, eine immersive virtuelle Welt zu kreieren, mit der der Nutzer in Echtzeit interagieren kann (Boas, 2013, S. 1; Doerner u. a., 2013, S. 66). Immersion beschreibt dabei das „Eintauchen“ in die virtuelle Welt, sodass die Reize aus der realen Umgebung möglichst ausgeblendet werden und der Fokus auf der VR liegt. Die Immersion ist das Produkt der – typischerweise visuellen,

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

akustischen und motorischen – simulierten Sinneseindrücke und kann unter anderem durch ein großes Sichtfeld bzw. Field of View (FOV) und eine für den Nutzer natürliche Fortbewegungs- und Interaktions-Art gefördert werden (Doerner u. a., 2013, S. 46). Eine gute VR-Anwendung soll dem Nutzer nicht nur das Gefühl von Immersion, sondern auch Präsenz vermitteln. Im Kontext der VR bezeichnet die Präsenz das körperliche Gefühl, in der computergenerierten Simulation anwesend zu sein (Linowes, 2015, S. 52). Entscheidend für die Präsenz in der VR sind folgende Kriterien: Die unter dem Begriff der Ortsillusion zusammengefasste Darstellung der virtuellen Szene soll auf den Anwender möglichst real wirken. Des Weiteren sollen die als Plausibilitätsillusion beschriebenen Inhalte der VR-Anwendung möglichst natürlich wahrgenommen und das Interesse des Nutzers geweckt werden, sodass er sich in die VR involviert fühlt (Doerner u. a., 2013, S. 18). Während die Immersion technisch objektiv erklärt und implementiert werden kann, stellt die Präsenz eine gefühlsorientierte, subjektive Wahrnehmung dar (Linowes, 2015, S. 52).

Die gängigste VR-Anzeigeart stellt das Head-Mounted Display (HMD), ein am Kopf getragenes Wiedergabegerät, dar (s. Abb. 9) (Boas, 2013, S. 2).



Abbildung 9: VR-HMD am Einsatzbeispiel eines VR-Zeichenprogramms (Tintori, Ray, 2016) [Foto]

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

Die in dem Sichtvolumen bzw. View Frustum der virtuellen Kamera befindlichen Informationen werden dem Nutzer typischerweise mit Hilfe eines Stereo-Displays angezeigt (s. Abb. 10) (Doerner u. a., 2013, S. 129).



Abbildung 10: VR-Stereo-Display (Linowes, 2015, S. 54) [Foto]

Dabei wird für beide Augen ein individuelles Bild gerendert, deren Sichtbereiche sich überlappen und durch die dabei entstandene Parallaxe ein für den Nutzer räumlicher Eindruck der VR-Szene entsteht. Wichtig dabei ist, dass die View Frusta der virtuellen Kameras gesichert werden, sodass die Blickrichtung für beide Augen parallel bleibt und eine gemeinsame Projektionsebene entsteht (s. Abb. 11) (Doerner u. a., 2013, S. 130; Linowes, 2015, S. 52).

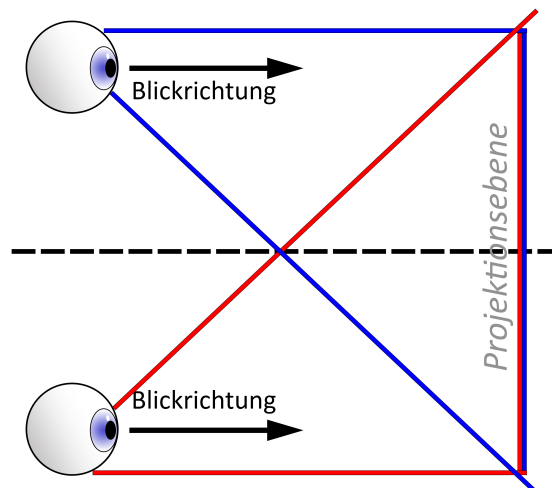


Abbildung 11: Stereo-Rendering mit Off-Axis Methode

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

Das HMD wird typischerweise mittels mehrerer externer Kameras optisch oder anhand der Beschleunigung in eine Richtung mittels interner Trägheitssensoren getrackt (Doerner u. a., 2013, S. 22; Linowes, 2015, S. 56). Ziel beim Tracking des HMDs ist es, Veränderungen der Orientierung und Position des Nutzers in Echtzeit in die VR zu übertragen. Die Bewegungsmöglichkeiten hängen von der Anzahl der im System vorhandenen Freiheitsgrade oder auch Degrees of Freedom (DOFs) ab. So besitzt ein starrer Körper im dreidimensionalen Raum sechs DOFs, die sich in die Translation in die drei verschiedenen Raumrichtungen und die Rotation um die drei Koordinatenachsen unterteilen (s. Abb. 12) (Doerner u. a., 2013, S. 98, 99).

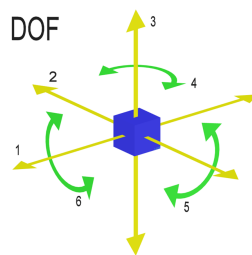


Abbildung 12: Degrees of Freedom (wiki2015Vrijheidsgraad) [Graphik]

Im Laufe der Zeit wurden viele unterschiedliche Fortbewegungsarten für VR entwickelt, welche auf die für diese Arbeit relevanten und gängigsten Methoden eingeschränkt werden sollen. Die für den Menschen Natürlichste ist die physische Fortbewegung. Bei dieser - im Kontext der VR auch als Room oder World Scale bezeichneten - Lokomotion bewegt sich der Anwender unter fortwährendem HMD-Tracking durch den realen und somit auch durch den virtuellen Raum (s. Abb. 13) (Bowman u. a., 2001, S. 7; Linowes, 2015, S. 11).



Abbildung 13: VR-Fortbewegungsart: Room Scale (Paligaru, Matt, 2018) [Foto]

Vorteile dieser Bewegungsart sind, dass aufgrund der natürlichen Handlung sowohl die Präsenz als auch die Abschätzung räumlicher Distanzen geför-

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

dert werden. Da die Größen des virtuellen und realen Raums miteinander übereinstimmen, müssen je nach Ausmaß der virtuellen Szene ausreichend freier Platz in der realen Umgebung oder zusätzliche Fortbewegungsgeräte, wie beispielsweise Laufbänder, vorhanden sein (Bowman u. a., 2001, S. 7).

Zwei weitere populäre Lokomotionsarten stellen die Fortbewegung mittels Hand-Eingabegeräten und Teleportation dar. Bei der VR-Fortbewegung mit Hand-Controllern, welche auch als Free Locomotion bezeichnet wird, manövriert der Nutzer mittels Tastendruck oder dem Bewegen von Analog-Sticks durch die virtuelle Umgebung (Slater u. a., 1995, S. 206; Bowman u. a., 2001, S. 7). Während bei der Free Locomotion die komplette Distanz zwischen Start- und Ziel-Punkt zurückgelegt wird, springt der Anwender bei der Teleportation-Methode direkt zu einem von ihm ausgewählten Ort (s. Abb. 14) (Linowes, 2015, S. 144). Da die Fortbewegung via Teleportation keine für den Menschen natürliche Handlungsweise darstellt, bietet sie nur eine geringe Präsenz. Die zurückgelegte Distanzen können schwerer eingeschätzt werden und es kann zudem zu Desorientiertheit in der virtuellen Umgebung kommen. Im Gegensatz zur Free Locomotion hat die Teleportation jedoch den Vorteil, dass sie sehr gut für Menschen mit einer Anfälligkeit für Motion Sickness geeignet ist. Motion Sickness bezeichnet Gefühle von Übelkeit, Unwohlsein oder Benommenheit, welche unter anderem durch eine Diskrepanz zwischen real ausgeführten und virtuell angezeigten Bewegungen verursacht werden können (Bhandari u. a., 2017, S. 2). Da bei der Free Locomotion zwar eine Fortbewegung in VR dargestellt wird, der Nutzer in der Realität jedoch auf einer Stelle bleibt, steigt das Risiko für den Anwender Motion Sickness zu verspüren. Die Teleportation wirkt diesem Übelkeitsgefühl entgegen, indem keine Bewegungen angezeigt werden.

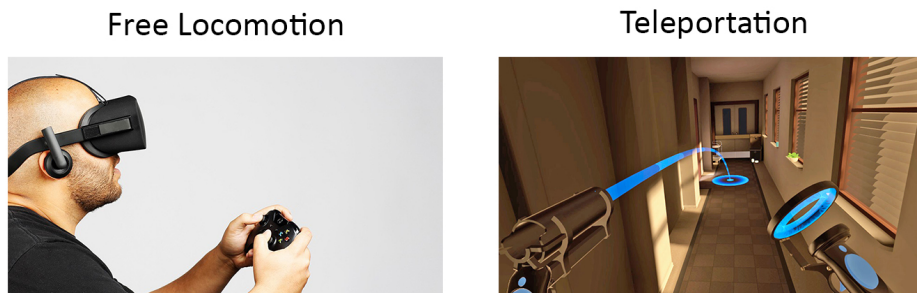


Abbildung 14: VR-Fortbewegungsart: Free Locomotion und Teleportation (Tuttle, Will, 2015; Buckley, Sean, 2016) [Foto]

Eine Bewegungsart, welche die Vorteile der dargestellten Fortbewegungstechniken vereint und deren Nachteile vermeidet, ist das Walking in Place (s. Abb. 15). Bei dieser VR-Lokomotion registriert das System, ob der Nutzer auf der Stelle läuft und führt in VR dementsprechend eine Fortbewegung

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

in Blickrichtung aus (Slater u. a., 1995, S. 201). Resultate dieser Fortbewegungsart sind, dass der Nutzer aufgrund der real ausgeführten Bewegungen ein hohes Präsenz-Gefühl in der VR erhält, jedoch – anders als bei Room Scale – in der realen Umgebung kein großer Bewegungsraum benötigt wird (Slater u. a., 1995, S. 214; Bhandari u. a., 2017, S. 1).



Abbildung 15: VR-Fortbewegungsart: Walking in Place (Lee u. a., 2017, S. 9)
[Foto]

Des Weiteren wird das räumliche Empfinden durch die angezeigte zurückgelegte Distanz zwischen Start- und Ziel-Punkt erhöht und dem Risiko für Motion Sickness durch die Reduktion der Diskrepanz zwischen real und virtuell ausgeführten Bewegungen entgegengewirkt (Slater u. a., 1995, S. 202; Tregillus und Folmer, 2016, S. 1). Bei der Realisierung der Walking in Place Lokomotionsart wird die Tatsache ausgenutzt, dass sich ein Mensch leicht auf und ab bewegt, wenn er auf der Stelle läuft. So wird von einem Tracking-System die Veränderung der Höhe des HMDs registriert und in einen entsprechenden virtuellen Schritt in Blickrichtung umgesetzt. Mit dem Ziel, Fehler bei der Schritterkennung zu vermeiden, wird eine Fortbewegung nur unter folgenden Bedingungen in VR durchgeführt:

- Die Veränderung der HMD- und somit der Nutzer-Höhe wird nur in einem Intervall bis zu 1,3cm als Schritt wahrgenommen. Damit sollen andere Bewegungen, wie beispielsweise Ducken oder Springen, als Schritt ausgeschlossen werden (Bhandari u. a., 2017, S. 4).
- Die Dauer eines Schrittes t_{step} muss in einem Zeitintervall I_{min} und I_{max} liegen. Nur wenn $t_{step} \geq I_{min}$ und $t_{step} \leq I_{max}$ ist, wird ein Schritt erkannt und in die VR übernommen (Bhandari u. a., 2017, S. 2).

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

- Für die VR-Anwendung wird eine minimale und maximale Fortbewegungsgeschwindigkeit V_{min} und V_{max} festgelegt. Die durchschnittliche Schrittgeschwindigkeit eines Menschen liegt bei 2m/s bzw. einer Frequenz von 1,4 Schritten pro Sekunde (Bhandari u. a., 2017, S. 4).

Aus diesen Faktoren und Einschränkungen ergibt sich folgende Formel, mit welcher man ermitteln kann, ob und wie schnell sich der Nutzer durch das auf der Stelle laufen in VR fortbewegt.

$$v = \frac{t_{step} - I_{min}}{I_{max} - I_{min}} * (V_{max} - V_{min}) + V_{min}$$

In Tabelle 1 sind die Vor- und Nachteile der einzelnen VR-Fortbewegungsarten zusammengefasst:

	Room Scale	Free Locomotion	Teleportation	Walking in Place
Präsenz	✓	✗	✗	✓
Einschätzung von Distanzen	✓	✓	✗	✓
Risiko für Motion Sickness	✓	✗	✓	✓

Tabelle 1: Übersicht der relevanten Fortbewegungsarten

Es gibt eine Vielzahl verschiedener VR-Anwendungsarten, welche sich hinsichtlich ihrer Interaktions- und Handlungsmöglichkeiten voneinander unterscheiden. So wird der Nutzer bei einer First-Person Experience in eine passive Beobachter-Rolle versetzt, in der er sich zwar frei im virtuellen Raum bewegen, aber – im Gegensatz zu einer Interactive Virtual Environment – nicht mit seiner virtuellen Umwelt interagieren kann. Social VR Anwendungen versetzen mehrere User gleichzeitig in die virtuelle Welt, wobei die gegenseitige Kontaktaufnahme unter den Nutzern im Mittelpunkt steht (Linowes, 2015, S. 10). Die Interaktion mit den in der virtuellen Szene befindlichen Elementen wird typischerweise mit Hilfe externer Eingabegeräte, wie beispielsweise Hand-Controllern, mittels Ray Casting oder Sprachsteuerung ausgeführt (Bowman u. a., 2001, S. 14). Um den Anwender bei seiner Aufgabe in der VR zu unterstützen und über die dafür nötigen Anwendungsmechaniken zu informieren, gibt es viele unterschiedliche UI-Elemente. Eine weit verbreitete UI-Anzeige ist der Reticle Cursor, ein im Sichtzentrum dargestelltes Fadenkreuz oder Pointer. Dieses UI-Element gibt den Sichtfokus des Users an und wird oftmals in Verbindung mit Ray Casting eingesetzt wird. Mit dem Ziel, die für den Kontext der Anwendung relevante Informationen darzustellen, werden neben Info-Bubbles oftmals Game Element UIs eingesetzt (s. Abb. 16). Game Element UIs unterscheiden sich von Info-Bubbles dahingehend, dass sich die angezeigten Benachrichtigungen in die

VR-Szene integrieren und Teil der virtuellen Umgebung sind. Somit wird – anders als beim Reticle Cursor oder der Info-Bubble – die Immersion in der VR gefördert (Linowes, 2015, S. 82).



Abbildung 16: UI: Game Element UI am Beispiel eines Flugsimulators (robbonline, 2009) [Foto]

Nachdem die Technologien Augmented und Virtual Reality vorgestellt und die relevanten theoretischen Grundlagen dargelegt wurden, soll im nächsten Schritt die Beziehung zwischen AR und VR genauer beleuchtet werden.

2.1.4 Verbindung von Augmented und Virtual Reality

Die verschiedenen Stufen des Reality-Virtuality Continuums eignen sich unterschiedlich gut für die Durchführung bestimmter Aufgabenbereiche. Während VR-Systeme dem Nutzer eine immersive Erfahrung bieten und einen Sachverhalt möglichst realistisch darstellen können, wird AR vor allem für Applikationen eingesetzt, bei denen der Nutzer sowohl reale als auch virtuelle Elemente im Blick haben soll und die direkte Zusammenarbeit mit anderen Personen eine wichtige Rolle spielt. Da sich ein Aufgabenbereich jedoch meist in mehrere Teilaufgaben untergliedert, die jeweils unterschiedliche Herangehensweisen erfordern, bietet es sich an, eine Anwendung für verschiedene XR-Stufen zu implementieren (Billinghurst u. a., 2001, S. 6).

Um dieses Vorhaben umsetzen zu können, muss zunächst eine technische Basis in Form eines geeigneten Wiedergabegeräts geschaffen werden. So kann beispielsweise mit Hilfe eines hybriden AR-/VR-HMDs sowohl zwischen der Real Reality, der Augmented Reality, als auch der Virtual Reality gewechselt werden (Karlsson und Olsson, S. 16). Um die Anforderungen der einzelnen Darstellungsformen zu erfüllen, muss die reale Umgebung aus der

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

Sicht des Nutzers in Echtzeit im HMD angezeigt und mit virtuellen Komponenten überlagert werden können (Steptoe, William, 2014; Van Gaalen und Zhang, 2016, S. 3). Da eine Verzögerung bei der visuellen Darstellung sowie eine niedrige Bildwiedergabe-Frequenz das Risiko für Motion Sickness erhöhen, muss das HMD über eine möglichst niedrige Latenz und eine Mindest-Framerate von 60Hz verfügen (Van Gaalen und Zhang, 2016, S. 3). Mit dem Ziel, die reale Umgebung für die Darstellung der Real und Augmented Reality einzufangen und für die VR ausblenden zu können, bietet sich ein Headset mit Video See-through an. Dabei dient die am HMD angebrachte Kamera dem Nutzer als digitales Auge (s. Abb. 17) (Edwards u. a., 1993, S. 224). Da der Nutzer die reale Umgebung ausschließlich über das Headset und der darin verbauten Displays wahrnimmt, sollte sich die applizierte Kamera an die Spezifikationen des HMDs und nicht an die Anforderungen der Augen richten (Steptoe, William, 2014). Hat das angezeigte Videobild der Kamera eine niedrigere Auflösung als die Displays des HMDs, erscheinen für den Nutzer Objekte weiter entfernt, als sie in Wirklichkeit sind, während eine im Vergleich zum Headset höhere Kameraauflösung nicht zur Geltung kommt (Drascic und Milgram, 1996, S. 129; Phan und Choo, 2010, S. 16).



Abbildung 17: Hybrides AR/VR Video See-through HMD (Bezmalinovic, Tomislav, 2017) [Foto]

Für die Anzeige der realen Umgebung muss das vom System eingefangene Kamerabild auf einer flachen Textur-Ebene (Plane) gerendert werden (Steed u. a., 2017, S. 232). Da das Kamera-FOV meist kleiner als der Sichtbereich des Menschen ist und sich die Kamera in einem bestimmten Abstand zu den Augen des Nutzers befindet, muss die Videotextur-Plane so zum Betrachter hin- oder wegtranslatiert werden, dass ein für den Nutzer angenehmer Eindruck der realen Umgebung entsteht (Drascic und Milgram, 1996, S. 131, 132). Des Weiteren muss das Koordinatensystem der Tracking-Vorrichtung mit dem Welt-Koordinatensystem sowie die Beziehung zwischen den virtuellen und realen Objekten zu ihrem jeweiligen Ursprung übereinstimmen, um im Rahmen der AR-Anzeige die virtuellen Objekte an ihrer korrekten

Position darzustellen (Bajura und Neumann, 1995, S. 53). Wechselt der Nutzer von der AR- in die VR-Anzeige, so wird die Kamera deaktiviert und die Anzeige der Videotextur-Plane durch eine ausschließlich virtuelle Umgebung ersetzt. Das Resultat ist eine Wiedergabe-Basis, mit welcher sowohl die Extrema Real Reality und Virtual Reality, als auch die MR-Zwischenstufe Augmented Reality erreicht werden können.

2.2 Computergrafik

In den letzten Abschnitten wurden verschiedene Darstellungsformen der XR anhand des Reality-Virtuality Continuum veranschaulicht und dabei genauer auf die Grundlagen der AR und VR und deren gemeinsame Beziehung eingegangen. Da für diese computergenerierten Systeme die visuelle Darstellung eine wichtige Rolle spielt, sollen im Folgenden die für diese Arbeit spezifischen Grundlagen aus der Computergrafik, wie Texturen und Materialien sowie Beleuchtung thematisiert werden.

2.2.1 Texturen und Materialien

Eine virtuelle Szene ist aus mehreren Komponenten zusammengesetzt, welche sich zum einen in für den Nutzer sichtbare, teils interaktionsfähige 3D-Objekte und zum anderen in unsichtbare, abstrakte Elemente unterteilen (Doerner u. a., 2013, S. 66). Die für den Anwender sichtbaren virtuellen Komponenten stellen meist Festkörpermodelle dar. Im Generellen besteht ein Festkörpermodell aus einem geschlossenen Polygonnetz, welches ein Volumen einschließt (Doerner u. a., 2013, S. 74).

Um eine Objektoberfläche visuell detailliert darzustellen, ohne jedoch das 3D-Modell an sich zu verändern, werden Texturen verwendet. Eine Textur ist ein Rasterbild, welches mit Hilfe von Texturkoordinaten auf eine Objektoberfläche gemappt wird (Doerner u. a., 2013, S. 66). Da Objektoberflächen unterschiedliche Eigenschaften besitzen, werden diese mittels Materialien genauer beschrieben. So wird Licht beispielsweise von metallisch glatten Objekten anders reflektiert, als von Festkörpermodellen mit rauen Oberflächen. Reflexionen des Umgebungslichts beeinflussen im Generellen die Grundfarbe und Helligkeit einer Modell-Außenseite (Doerner u. a., 2013, S. 75, 76). Mit dem Ziel, die Beschaffenheit von Objektoberflächen möglichst detailreich darzustellen, können Materialien aus verschiedenen Texture Maps aufgebaut sein. Die im Rahmen dieser Arbeit relevanten Texture Maps sollen im Folgenden erläutert werden.

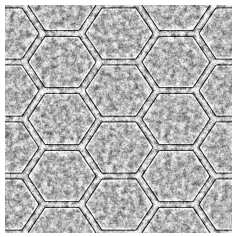


Abbildung 18: Texture-Map: Metalness Map

Metalness Maps sind Graukeil-Bilder, welche die Metallizität eines Objekts steuern. Während matte Oberflächen – dargestellt durch dunkle Bereiche – diffuse Reflexionen verursachen, wird das Licht bei glänzenden Objekten – dargestellt durch helle Bereiche – spiegelnd zurückgeworfen (s. Abb. 18) (Doerner u. a., 2013, S. 75, 76; Sketchfab, 2018).

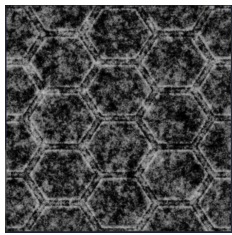


Abbildung 19: Texture-Map: Specular Map

Specular Maps regulieren ähnlich wie Metalness Maps die Glanzpunkte auf einer Objektoberfläche. Je satter diese (s)RGB-Texturen sind, desto glänzender erscheinen die entsprechenden Objekte (s. Abb. 19) (Sketchfab, 2018).

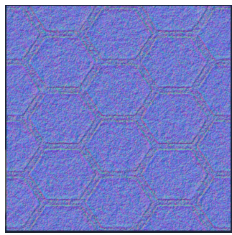


Abbildung 20: Texture-Map: Normal Map

Normal Maps werden durch farbliche Texturen repräsentiert, deren RGB-Werte in X-, Y- und Z-Koordinatenwerte übersetzt werden. Dadurch wird der Objektoberfläche ein scheinbares Höhenprofil verliehen (s. Abb. 20) (Sketchfab, 2018).

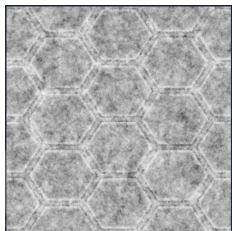


Abbildung 21: Texture-Map: Ambient Occlusion Map

Ambient Occlusion Maps bestimmen die Verschattung der Objektoberfläche. Je dunkler die Graukeil-Texturen sind, desto matter erscheint das Objekt (s. Abb. 21).

Des Weiteren bestimmt der Grad der Transparenz, ob und in welchem Maß das Licht durch die Objektoberfläche hindurchdringt. Neben reflektierenden Objektmodellen gibt es emittierende Oberflächen, welche von sich aus Licht mit einer bestimmten Farbe und Intensität aussenden (Doerner u. a., 2013, S. 75, 76).

2.2.2 Beleuchtung

Wie in dem letzten Abschnitt bereits skizziert, hat die Beleuchtung großen Einfluss auf die visuelle Darstellung virtueller Objekte. Da dieses Thema ausreichend Stoff für eine eigene Ausarbeitung bietet, sollen in diesem Ka-

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

pitel nur die für diese Arbeit relevanten Aspekte erläutert werden. So gibt es neben den lichtemittierenden Materialien verschiedene Beleuchtungsarten mit jeweils unterschiedlichen Charakteristiken.

Point Lights stellen kugelförmige Lichtquellen dar, welche Strahlen in alle Richtungen gleichmäßig aussenden (s. Abb. 22). Mit zunehmender Entfernung zu den Point Lights nimmt sowohl die Intensität als auch die Farbsättigung des Lichts ab (Doerner u. a., 2013, S. 84). Diese Beleuchtungsart eignet sich beispielsweise für die Darstellung von Lampen (Unity Technologies, 2018a).

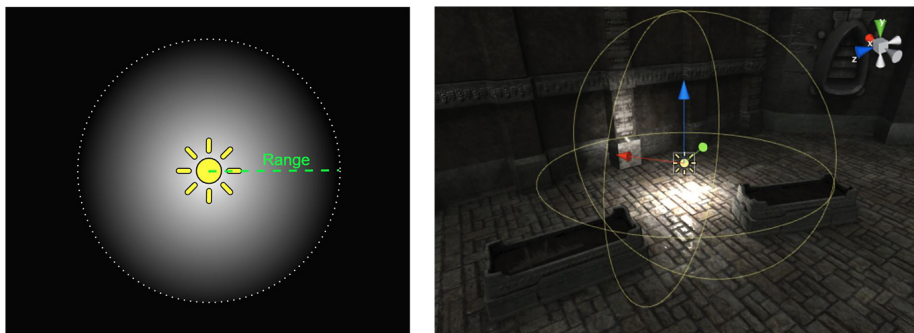


Abbildung 22: Lichtquelle: Point Light (Unity Technologies, 2018b) [Graphik, Foto]

Im Gegensatz zu einem Point Light geht von einem Spot Light ein Lichtkegel in eine bestimmte Richtung aus (s. Abb. 23). Je größer die Distanz zur Lichtquelle, desto mehr nehmen Farbsättigung und Intensität der Beleuchtung ab (Doerner u. a., 2013, S. 84). Spot Lights finden oftmals bei der Darstellung künstlicher Lichtquellen, wie beispielsweise Scheinwerfer, Verwendung (Unity Technologies, 2018a).

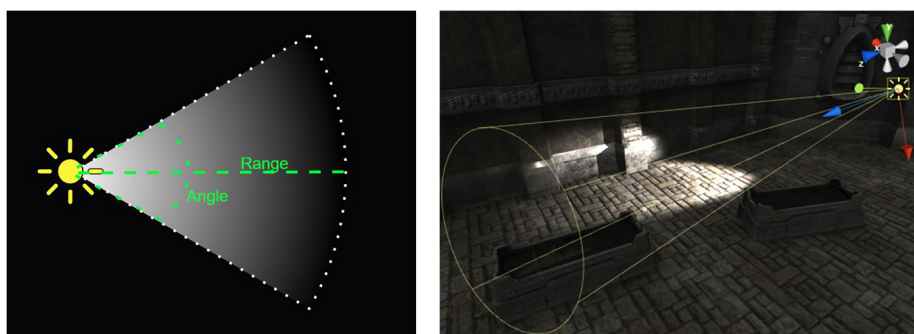


Abbildung 23: Lichtquelle: Spot Light (Unity Technologies, 2018b) [Graphik, Foto]

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

Anders als bei den Point und Spot Lights hat die Entfernung keine Auswirkung auf die Farbe und Intensität von Directional Lights (s. Abb. 24). Diese Beleuchtungsart sendet Lichtstrahlen in eine bestimmte Richtung aus, wobei die Lichtquelle unendlich weit entfernt erscheint. Aufgrund dieser Eigenschaften eignet sich das Directional Light besonders für die Darstellung natürlicher Lichtquellen, wie das Sonnenlicht (Unity Technologies, 2018a).

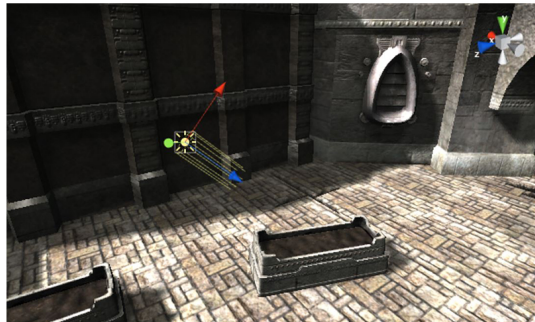
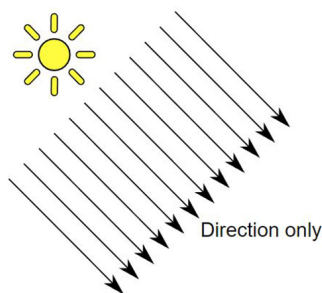


Abbildung 24: Lichtquelle: Directional Light (Unity Technologies, 2018b) [Graphik, Foto]

Die Beleuchtung in einer virtuellen Szene kann im Allgemeinen durch zwei verschiedene Techniken realisiert werden: Realtime und Baked Lighting. Bei der ersten Methode werden Beleuchtungseffekte in Echtzeit bestimmt, was vor allem bei beweglichen Objekten und dynamischen Szenenabläufen zum Tragen kommt. Kehrseite von Realtime Lighting ist jedoch, dass – je nach Komplexität der virtuellen Umgebung – diese Beleuchtungstechnik sehr rechenintensiv ist und Verschattungen nicht so realistisch dargestellt werden können wie mit Baked Lighting. Unter Baked Lighting wird die Berechnung von Lightmaps anhand der in der virtuellen Szene befindlichen Lichtquellen verstanden. Mit Hilfe dieser im Vorhinein erstellten Texturen werden Licht- und Schatten-Informationen auf den Objektoberflächen angezeigt. Da die Beleuchtung der einzelnen Objekte bereits zuvor bestimmt wird, ist diese Beleuchtungstechnik performanter als das Realtime Lighting. Allerdings können Beleuchtungseffekte beim Baked Lighting während des Programmablaufs nicht aktualisiert werden, was vor allem bei der realistischen Darstellung von Bewegungen einen Nachteil darstellt (Unity Technologies, 2018a).

2.3 Innenarchitektur

In den letzten Abschnitten wurden verschiedene XR-Stufen sowie Verfahren aus der Computergrafik erläutert, mit dem Ziel, eine theoretische Wissensbasis zur Erstellung eines hybriden AR-/VR-Systems zu schaffen. Da sich die im Rahmen dieser Arbeit erstellte Applikation auf die Gestaltung eines Hotelzimmers konzentriert, dürfen die dafür nötigen Grundlagen aus

der Innenarchitektur nicht außer Acht gelassen werden. So spielen im Kontext dieses Anwendungsbereichs die generellen Richtlinien zur Platzierung von Einrichtungsgegenständen, die Möblierung eines Hotelzimmers sowie die Einhaltung architektur-spezifischer Normen eine essentielle Rolle. Aus diesem Grund sollen diese Themen in den folgenden Kapiteln genauer beleuchtet und thematisiert werden.

2.3.1 Richtlinien für die Platzierung von Einrichtungsgegenständen

Mit einer Wohnung werden verschiedene Bedürfnisse und Kriterien assoziiert. Einerseits soll sie dem Bewohner Schutz und Geborgenheit bieten, andererseits soll durch die Wahl der Einrichtung ein Gefühl von Wohlempfinden und Vertrautheit vermittelt werden. Um diese Ziele erfüllen zu können, werden im Rahmen der Raumplanung sowohl funktionale als auch visuelle Kriterien betrachtet, wobei der Funktionalität eine erhöhte Priorität zukommt (Merrell u. a., 2011, S. 2; HAF Hamburger Institut fuer Fernstudien GmbH, 2018).

Die funktionalen Richtlinien bei der Einrichtung eines Wohnraums werden durch die menschliche Psychologie und Verhaltensweisen bedingt. Dabei müssen die Gesichtspunkte Konversation, Freiraum und die Beziehung der Einrichtungsgegenstände zueinander betrachtet werden. Die Möglichkeit, Gespräche miteinander führen zu können sowie deren Qualität hängt unter anderem von der Anordnung der Sitzgelegenheiten ab. Sitzplätze sollten so ausgerichtet werden, dass die Gesprächspartner Blickkontakt zueinander aufbauen können und die Sitzplätze für eine normale Gesprächslautstärke maximal 2,4m voneinander entfernt sind (Merrell u. a., 2011, S. 2, 3). Des Weiteren muss bei der Platzierung der Einrichtungsgegenstände auf den dabei belegten und freien Raum geachtet werden. So beeinflusst die Stellfläche, womit der von einem Einrichtungsgegenstand eingenommene Platz bezeichnet wird, sowohl die zur Durchquerung des Raums benötigte Verkehrsfläche als auch die zur Benutzung des Mobiliars abhängige Bedienfläche (Merrell u. a., 2011, S. 3; HAF Hamburger Institut fuer Fernstudien GmbH, 2018). Da Einrichtungsgegenstände oftmals in einem gewissen Kontext zueinanderstehen, sollten diese sinnvoll zueinander platziert werden (Merrell u. a., 2011, S. 3).

Die Einrichtung eines Innenraums beeinflusst neben der Funktionalität auch dessen visuelle Wahrnehmung. So sollte bei der Wahl der Möbelstücke darauf geachtet werden, dass diese visuell bzgl. Farbigkeit, Form und Ausgestaltung miteinander harmonisieren und im Raum so verteilt werden, dass sie einerseits zu den Wänden und andererseits zueinander ausgerichtet sind. Mit dem Ziel, den Blick des Betrachters zu lenken, sollte jeder Raum einen visuellen

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

Schwerpunkt, wie beispielsweise einen Fernseher oder ein Fenster besitzen (s. Abb. 25) (Merrell u. a., 2011, S. 2, 4).



Abbildung 25: Richtlinien für die Platzierung von Einrichtungsgegenständen (Merrell u. a., 2011, S. 87) [Graphik]

Im Bereich der Innenarchitektur haben sich allgemeine Darstellungen von Einrichtungsgegenständen manifestiert. Um die Kohärenz der Anwendung in diesem Bereich zu wahren, sollen die dabei relevanten Symbole entsprechend dargestellt werden (s. Abb. 26) (HAF Hamburger Institut fuer Fernstudien GmbH, 2018):

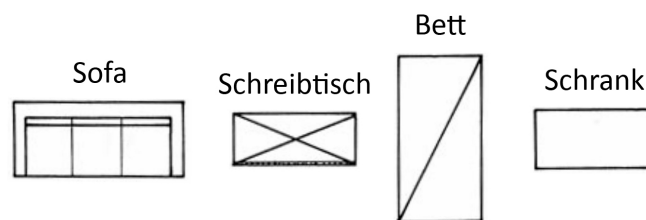


Abbildung 26: Einrichtungsgegenstand-Symbole (HAF Hamburger Institut fuer Fernstudien GmbH, 2018) [Graphik]

2.3.2 Einrichtung von Hotelzimmern

Auf Basis der im vorhergehenden Abschnitt gewonnenen Kenntnisse über die allgemeinen Richtlinien zur Innenraum-Einrichtung soll im Folgenden ge-

2. BEGRIFFLICHE UND THEORETISCHE GRUNDLAGEN

nauer auf die Kriterien zur Gestaltung von Hotelzimmern eingegangen werden. Im Allgemeinen untergliedern sich Hotels in temporäre und permanente Unterkünfte. Während temporäre Lokalitäten bei Veranstaltungen für einen gewissen Zeitraum aufgebaut werden, wird bei permanenten Hotels keine zeitliche Frist eingeplant (Mak, 2013, S. 22). Es gibt eine Vielzahl verschiedener Hotelzimmer-Arten, wie beispielsweise Touristen-, Standard- oder Luxus-Unterkünfte, welche sich in ihrer jeweiligen Größe und ihrem Komfort voneinander unterscheiden (Toth, 2008, S. 16). Vereint werden die einzelnen Räumlichkeiten in ihren allgemeinen Anforderungen, welche im Folgenden erläutert werden.

Einerseits fungiert ein Hotelzimmer als Schlafzimmer. Der Gast soll sich ausruhen, schlafen, schreiben und sich umkleiden können. Um diese Kriterien zu erfüllen, besteht ein Hotel-Schlafbereich grundsätzlich aus einem Bett, einem Schreibtisch und einem Kleiderschrank sowie weiteren Ablageflächen, Nachttischen oder Regalen (HAF Hamburger Institut fuer Fernstudien GmbH, 2018). Für ein gesteigertes Wohlbefinden des Hotelbesuchers werden die Wände des Schlafbereichs meist mit Tapeten verkleidet oder farbig gestrichen (Siguaw und Enz, 1999, S. 45).

Andererseits hat ein Hotelzimmer die Funktion als Wohnzimmer und soll dem Gast eine Möglichkeit zur Entspannung, Unterhaltung und Kommunikation mit anderen Personen bieten. Um diese Bedürfnisse erfüllen zu können, sollte die Zimmereinrichtung aus diversen Sitz- und Liegemöglichkeiten, Schränken sowie Ablageflächen bestehen (Merrell u. a., 2011, S. 1; HAF Hamburger Institut fuer Fernstudien GmbH, 2018). Typische Einrichtungsgegenstände in diesem Bereich sind unter anderem Sessel, Sofas, Sitzgruppen, Tische, Schränke und TV-Videomöbel (HAF Hamburger Institut fuer Fernstudien GmbH, 2018). Damit sich der Gast in seiner Unterkunft wohlfühlt, soll zudem bei der Wahl der Möbelstücke auf einen einheitlichen visuellen Stil geachtet werden (Merrell u. a., 2011, S. 1).

Neben dem Schlaf- und Wohnbereich, welche sich meist in getrennte Raunteile untergliedern, je nach Art des Hotelzimmers jedoch auch in einem Zimmer vereint sein können, ist das Badezimmer ein genereller Bestandteil von Hotels. Das Badezimmer soll von Gästen für Hygiene-Zwecke genutzt werden können. So besteht ein Hotel-Badezimmer aus einer Dusche oder Badewanne, einem Waschbecken, einer Toilette sowie einem Badschrank (Dolnicar und Otter, 2003, S. 6). Für einen visuell ansprechenden Stil werden Bestandteile des Badezimmers meist aus Marmor angefertigt (Siguaw und Enz, 1999, S. 45).

3 Augmented und Virtual Reality im Bereich Innenarchitektur

Das vorhergehende Kapitel setzte sich aus mehreren verschiedenen Themenbereichen zusammen. Neben Aspekten aus der XR wurden Verfahren aus der Computergrafik sowie Richtlinien aus der Innenarchitektur mit dem Ziel dargestellt, eine begriffliche und theoretische Basis für diese Arbeit zu schaffen. Für ein besseres Verständnis sollen neben diesen Grundlagen die Kriterien von Augmented und Virtual Reality Anwendungen im Bereich Innenarchitektur erläutert und anhand von Beispielen veranschaulicht werden.

3.1 Anforderungen und Kriterien an eine Augmented Reality Innenarchitektur-Anwendung

Innenarchitekten und Designer benutzen bei der Gestaltung von Innenräumen üblicherweise die Methode des systematischen Ausprobierens. Da dieses Verfahren viel Zeit in Anspruch nimmt, haben 3D-Applikationen die Aufgabe, den Nutzer durch die Positionierung von digitalen Einrichtungsgegenständen sowie durch funktionale und visuelle Ratschläge zu unterstützen und somit den Arbeitsprozess zu verkürzen und Ressourcen einzusparen (Woodward u. a., 2007, S. 536; Merrell u. a., 2011, S. 1, 2). Die Vorteile von AR gegenüber Monitor-Anwendungen bestehen darin, dass zum einen die Kommunikation und die Zusammenarbeit mehrerer User gefördert wird und zum anderen die Bedienung und Steuerung der Applikation intuitiv und leicht erlernbar ist (Kato u. a., 2000, S. 115). So werden bei AR-Anwendungen im Bereich Innenarchitektur oftmals table-top Tangible UIs verwendet: Mehrere Nutzer mit AR-HMDs befinden sich an einem Tisch, auf welchem die virtuellen Informationen zu einem geplanten Gebäude angezeigt werden. Die Einrichtungsgegenstände werden als virtuelle Informationen auf AR-Markern angezeigt, welche auf eine für den Menschen natürliche Weise angefasst und positioniert werden können. Während des Design-Prozesses kann sich der Nutzer mit den anderen Anwesenden unterhalten und beraten, wobei er gleichermaßen seine Gesprächspartner und die virtuellen Informationen im Blick hat. Der Tisch stellt dabei die gemeinsame Arbeitsfläche dar, auf welchem mit den virtuellen Objekten kollaborativ interagiert werden kann (s. Abb. 27) (Kato u. a., 2000, S. 111).

3. AUGMENTED UND VIRTUAL REALITY IM BEREICH INNENARCHITEKTUR



Abbildung 27: Kollaborative AR-Architektur-Anwendung (Studio 216, 2017) [Foto]

Für eine gute AR Innenarchitektur-Anwendung müssen verschiedene Anforderungen und Kriterien erfüllt werden. Zum einen ist es ratsam darauf zu achten, dass sich die Entwickler der Anwendung mit den Bereichen Innenarchitektur und Design auseinandersetzen und Architekten zu Rate ziehen, um alle für den Anwendungsbereich relevanten Funktionen und Komponenten zu integrieren (Wang, 2009, S. 311, 312). Zum anderen spielt die Interaktion in Echtzeit eine wichtige Rolle: So müssen für einen guten Arbeitsfluss die virtuellen Einrichtungsgegenstände in Echtzeit hinsichtlich ihrer Position, Ausrichtung und Skalierung verändert werden können (Phan und Choo, 2010, S. 17). Da bei der Entwurfs- und Gestaltungsphase eines Innenraums mehrere Architekten und Designer involviert sind, sollte die AR-Applikation kollaboratives Arbeiten ermöglichen und fördern (Wang, 2009, S. 317). Des Weiteren spielt der Detailgrad der 3D-Modelle eine wichtige Rolle, damit ein realitätsnaher Eindruck des Zimmers und der Einrichtungsgegenstände gewonnen werden kann. Um den Eindruck der Einrichtungsgegenstände so wirklichkeitsgetreu wie möglich wiederzugeben, sollten die einzelnen Modelle nicht mit Texten und Zusatzinformationen überladen und verdeckt werden, sondern gut sichtbar sein (Wang, 2009, S. 323).

3.2 Anforderungen und Kriterien an eine Virtual Reality Innenarchitektur-Anwendung

Im vorhergehenden Abschnitt wurde erläutert, dass Entwickler bei der Erstellung von AR-Applikationen im Bereich Innenarchitektur unter anderem auf den realitätsnahen Eindruck der Zimmer und der Einrichtungsgegenstände achten sollten. Diesem Aspekt kommt bei VR-Architektur-Anwendungen

3. AUGMENTED UND VIRTUAL REALITY IM BEREICH INNENARCHITEKTUR

eine essentielle Rolle zu. So kann sich ein Architekt mit Hilfe von VR in ein virtuelles Zimmer hineinversetzen und erhält bereits in der Planungsphase eine wirklichkeitsgetreue Sichtweise auf seine Ideen und Entwürfe, welche mit herkömmlichen Monitor-Anwendungen vergleichsweise wenig immersiv vermittelt werden können (s. Abb. 28) (Campbell und Wells, 1994, S. 2; Donath und Regenbrecht, 1996, S. 205). Im Allgemeinen werden VR Innenarchitektur-Anwendungen in den frühen Phasen des Architektur-Designs eingesetzt, um neue Ideen und Möglichkeiten kostengünstig und effizient auszuprobieren (Donath und Regenbrecht, 1995, S. 317; Donath und Regenbrecht, 1996, S. 201).



Abbildung 28: VR-Architektur-Anwendung (OConnell, Kim, 2016) [Foto]

Wie bereits erwähnt hat der Detailgrad der einzelnen 3D-Modelle Auswirkungen auf den Eindruck, den der Nutzer von einem Innenraum und den einzelnen Einrichtungsgegenständen erhält und auf Basis dessen Design-Entscheidungen getroffen werden (Campbell und Wells, 1994, S. 5; Wang, 2009, S. 323). Neben diesem Aspekt spielt die Fortbewegung eine wichtige Rolle für eine gute VR Innenarchitektur-Anwendung. So soll es dem Nutzer möglich sein, sich frei in dem virtuellen Raum zu bewegen und diesen zu erkunden, um bestimmte Komponenten ausführlich in Augenschein nehmen zu können (Campbell und Wells, 1994, S. 4, 6). Des Weiteren sollte die Art der VR-Fortbewegung so gewählt werden, dass der Nutzer einen guten Eindruck der räumlichen Distanzen erhält und somit die Immersion in dem virtuellen Zimmer gefördert wird (Donath und Regenbrecht, 1995, S. 314). Die für den User immersivste und somit für VR-Architektur-Anwendungen am besten geeignetste Lokomotions-Methode stellt das Room Scale dar, da der Nutzer durch die eigene Bewegung eine sehr gute räumliche Vorstellung für das Zimmer erhält (Donath und Regenbrecht, 1996, S. 202). Da jedoch meist nur ein limitierter Bewegungsraum gegeben ist, bietet sich die Implementation der Walking in Place Fortbewegungsart als Alternative an (Slater u. a., 1995, S. 206). Die Immersion ist der Aspekt, mit dem sich ein VR-System gegenüber herkömmlichen Monitor-Anwendungen abheben kann und sollte somit besonders bei der Entwicklung beachtet werden. Da

3. AUGMENTED UND VIRTUAL REALITY IM BEREICH INNENARCHITEKTUR

das Mittendrin-Gefühl in VR durch Einbrüche in der Performanz stark reduziert wird, sollte bei der Implementation Wert auf eine flüssige Darstellung der Szene geachtet werden, jedoch der Detailgrad dabei nicht darunter leiden (Campbell und Wells, 1994, S. 5). Ein weiteres Kriterium für eine gute VR Innenarchitektur-Anwendung ist, dass sich der User möglichst gut auf den virtuellen Innenraum konzentrieren kann. Aus diesem Grund sollte sich das UI auf die wesentlichsten Informationen beschränken und Szenen-Objekte nicht mit UI-Elementen verdeckt werden (Wang, 2009, S. 323).

4 Konzeption der Anwendung

Anhand der in den vorhergehenden Kapiteln thematisierten Grundlagen und Kriterien für AR und VR Applikationen im Bereich Innenarchitektur soll eine eigene Anwendung erstellt werden, welche die computergenerierten Darstellungsformen sinnvoll miteinander vereint. Dieses Kapitel befasst sich mit der Konzeption der Anwendung und geht sowohl auf deren grundlegende Idee als auch die dabei gesteckten Ziele und Anforderungen ein. Des Weiteren wird das Konzept für den Aufbau des Systems dargestellt.

4.1 Grundlegende Idee

Die Motivation ist ein System zu entwickeln, welches im Rahmen der im Kapitel 1 vorgestellten vierten Transformation die XR-Stufen *Augmented* und *Virtual Reality* sinnvoll in einen Kontext setzt und dabei die Stärken der beiden Technologien kombiniert. Da sowohl AR als auch VR bereits getrennt voneinander in dem Bereich Innenarchitektur Anwendung finden, jedoch bislang keine Architektur-Anwendung bekannt ist, bei der diese beiden Darstellungsformen kombiniert werden, soll diese Möglichkeit im Rahmen dieser Arbeit evaluiert werden. Das dabei gewählte Szenario ist die Einrichtung eines Hotelzimmers. Die Vorteile bestehen darin, dass beinahe jeder Mensch mit der Thematik Hotel vertraut ist und – anders als beispielsweise bei einem Wohnzimmer – durch die dabei einhergehende Vorgabe bestimmter Einrichtungsgegenstände ein funktionaler Rahmen gesteckt wird.

Wie in den letzten beiden Kapiteln erläutert, liegen die Stärken von AR in der für den Menschen natürlichen Interaktionsweise und den guten Möglichkeiten für kollaboratives Arbeiten. Aus diesem Grund eignen sich AR Innenarchitektur-Applikationen vor allem zur Anordnung von Einrichtungsgegenständen. Im Gegensatz dazu bietet VR in diesem Anwendungsbereich dem Nutzer die Möglichkeit, ein Zimmer bereits in der Planungsphase realitätsnah zu besichtigen. Grundlegende Idee dieser Arbeit ist es, die jeweiligen Stärken von AR und VR im Bereich Innenarchitektur in einer Applikation zu vereinen:

Der Nutzer trägt ein hybrides AR-/VR-HMD, mit welchem er fließend zwischen den beiden Realitätsstufen wechseln kann. Ausgangspunkt ist die AR-Ansicht, in welcher auf einer speziellen Arbeitsunterlage ein virtueller Innenraum in verkleinertem Maßstab aus der Vogelperspektive angezeigt wird. Der Anwender kann das Zimmer individuell einrichten, indem er verschiedene Möbelstücke mit Hilfe von Markern auf der Arbeitsunterlage platziert. Um einen realitätsnahen Eindruck von dem eingerichteten Innenraum zu gewinnen, kann der Nutzer von dem AR- in den VR-Modus wechseln und das Zimmer in der Normalsicht durchqueren. Will der Anwender weitere Ände-

rungen an der Inneneinrichtung vornehmen, so kann er wieder von der VR- in die AR-Sicht zurückwechseln.

4.2 Ziele und Anforderungen

Bei der Realisierung der im vorhergehenden Abschnitt dargelegten Idee sollen verschiedene Ziele und Anforderungen erfüllt werden. Kernaufgabe ist es, eine Anwendung im Bereich Innenarchitektur zu kreieren, die das Zusammenspiel von AR und VR evident und zweckvoll verbindet. Dabei soll die Applikation leicht und intuitiv bedient werden können und den Anwender bei seiner Aufgabe sowohl motivieren, als auch mit Hilfe unterstützender Funktionen eine positive User Experience bieten.

Für diesen Plan muss zunächst das nötige Fundament aus Hardware- und Software-Komponenten geschaffen werden. Hierfür soll mit den zur Verfügung stehenden Mitteln ein hybrides AR-/VR-HMD angefertigt werden, welches die Realität mit einem akzeptablen FOV in Echtzeit wiedergibt. Mit dem Ziel, einerseits Tracking-Fehler in AR zu vermeiden und andererseits die Positionierung der Marker zu erleichtern, soll eine Arbeitsunterlage als Unterstützungselement hergestellt werden. Die Größe der Unterlage soll so gewählt werden, dass sie genügend Platz für die einzelnen Marker bietet und sich komplett in das Sichtfeld des Anwenders integriert. Des Weiteren sollen die platzierten Marker bei allen Lichtverhältnissen von dem Tracking-System detektierbar sein und durch unbeabsichtigte leichte Stöße nicht verschoben werden. Für die Realisierung der Software soll eine Engine gewählt werden, welche sowohl die Implementation von AR- und VR-Anwendungen ermöglicht als auch durch notwendige Software Development Kits (SDKs) erweitert werden kann. Die 3D-Objekte sollen mit einem geeigneten Modellierungstool erstellt werden können.

Im Rahmen der 3D-Modellierung soll ein Hotelzimmer kreiert werden, welches sich an reale Vorbilder anlehnt und den entsprechenden Architektur-Normen entspricht. Die zur Verfügung stehenden Möbelstücke sollen sich in den Kontext eines Hotelzimmers einfügen. Dabei soll der Nutzer aus mehreren Modellen pro Einrichtungsgegenstand-Art auswählen können, um den Raum individuell zu gestalten. Die Objektoberflächen der einzelnen Modelle sollen möglichst detailliert dargestellt werden.

Als Basis für den AR-Modus soll das Kamerabild im HMD angezeigt und die virtuellen Informationen sollen in Echtzeit korrekt auf ihre korrespondierenden Marker gemappt werden. Um bei diesem Vorgang potenzielle Anzeigefehler zu vermeiden, sollen die Marker von dem System leicht detektiert und präzise getrackt werden. Die angezeigten virtuellen Objekte sollen jederzeit gut sichtbar sein und deren Größen in einem einheitlichen Maßstab

zueinanderstehen. Da die Interaktion bei der Platzierung der Einrichtungsgegenstände eine wichtige Rolle spielt, soll diese für den Nutzer möglichst natürlich und leicht zu erlernen sein. Mit dem Ziel, den Anwender bei seiner Aufgabe zu assistieren, sollen verschiedene UI-Elemente und Zusatzinformationen angezeigt werden: Zum einen soll sich der User – vor allem bei Interaktionsarten wie Ray Casting – stets im Klaren darüber sein, worauf sein Sichtfokus liegt. Zum anderen soll die Anwendung die Platzierung der Einrichtungsgegenstände erleichtern und das kontextuelle Verständnis des Nutzers für die Gestaltung des Hotelzimmers stärken. Die dabei dargestellten Informationen sollen jedoch nicht die Einrichtungsgegenstände verdecken.

Neben dem AR-Modus soll es dem Nutzer jederzeit möglich sein, das von ihm eingerichtete Hotelzimmer in VR besichtigen zu können. Dabei soll der Wechsel von AR zu VR intuitiv und kontrolliert durchgeführt werden können. Da jeder Mensch unterschiedlich empfindlich für Motion Sickness ist, soll die Möglichkeit bestehen, vor dem Wechsel in das virtuelle Hotelzimmer aus verschiedenen VR-Fortbewegungsarten auszuwählen, um das Risiko für Unwohlsein in der VR zu reduzieren. Die Kernaufgabe der Anwendung besteht darin, eine Brücke zwischen AR und VR zu schlagen. Hierfür soll die vom Nutzer in AR vorgenommene Einrichtung des Hotelzimmers korrekt in VR übertragen werden. Um den Nutzer in seiner VR-Erfahrung jedoch nicht zu beeinträchtigen, sollen Änderungen an den Einrichtungsgegenständen nicht in der VR vorgenommen werden, während dieser sich im VR-Modus befindet.

Bei der Wahl und Implementation der VR-Fortbewegungsarten müssen eine Vielzahl von Kriterien beachtet werden: So sollen die Lokomotionsarten einerseits das räumliche Empfinden des Nutzers fördern und eine gezielte Fortbewegung ermöglichen. Andererseits sollen die Fortbewegungsarten für Bewegungsräume realisierbar sein, die kleiner als das virtuelle Hotelzimmer sind. Die Bewegung im virtuellen Raum soll des Weiteren leicht erlernbar und physikalisch real nachvollziehbar sein. Da im VR-Modus der Eindruck des Hotelzimmers im Vordergrund steht, soll dieser nicht durch UI-Elemente gestört werden. Um die Immersion zu fördern und dem Anwender den Eindruck zu vermitteln, sich in einem Hotelzimmer zu befinden, soll sowohl auf einen visuell als auch akustisch hohen Detailgrad geachtet werden, der jedoch die Performanz der Applikation nicht beeinträchtigt.

Dem Nutzer soll es jederzeit möglich sein, von der VR- wieder in der AR-Ansicht zu wechseln. Die Interaktion soll dabei sowohl klar ersichtlich als auch leicht und kontrolliert durchführbar sein. Um den Arbeitsfluss nicht zu unterbrechen, sollen sich die Ladezeiten zwischen den einzelnen Modi in einem für den Nutzer akzeptablen zeitlichen Rahmen befinden.

4.3 Planung der Anwendung

Um die in den vorhergehenden Abschnitt gesteckten Ziele in die Realität umzusetzen, wurde ein Konzept entwickelt, welches den Aufbau und die einzelnen Komponenten der Anwendung erläutert und in diesem Kapitel dargestellt werden soll.

4.3.1 Hardware- und Software-Setup

Die Basis für das hybride AR-/VR-System zur Gestaltung von Innenräumen stellt eine VR-Brille dar, an deren Vorderseite eine Webcam mittels einer abnehmbaren Halterung angebracht ist (s. Abb. 29). Das VR-HMD verfügt sowohl über sechs DOFs, um eine Bewegung im Raum zu ermöglichen als auch über eine Mindest-Framerate von 60Hz, um das Risiko für Motion Sickness vorzubeugen. Die Auflösung und das FOV der verwendeten Kamera entsprechen den korrespondierenden Spezifikationen des VR-Headsets. Sowohl Webcam als auch HMD sind via USB- und HDMI-Kabel mit einem Computer verbunden, auf welchem die Anwendung ausgeführt wird.

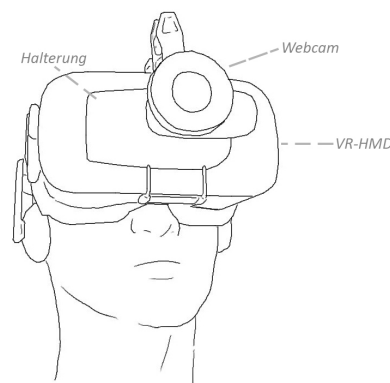


Abbildung 29: Konzeptzeichnung: Hybrides AR-/VR-HMD

Der Arbeitsbereich zur Nutzung der Applikation stellt eine quadratische Unterlage dar, welche sich im Abstand der geringsten durchschnittlichen Armlänge vollkommen in das FOV des angezeigten Kamerabildes einpasst. Um den Rand der Arbeitsunterlage ist ein weißes Leuchtband befestigt. Mit dieser Vorrichtung soll eine ausreichend gute Beleuchtung der Arbeitsfläche und der darauf platzierten Marker gegeben sein, um Trackingfehlern bei schwachen Lichtverhältnissen entgegenzuwirken. Damit die vom Nutzer platzierten AR-Marker nicht durch unbeabsichtigte leichte Stöße verschoben werden, haften diese magnetisch auf der Unterlage.

Der Kern des Software-Setups bildet eine Game Engine mit angebundener Programmierumgebung, welche mit einem – anhand von persönlichen

4. KONZEPTION DER ANWENDUNG

Erfahrungstests ausgewählten – AR-SDK erweitert wird. Für die Erstellung des virtuellen Hotelzimmers wird ein Modellierungs-Tool verwendet. Um in dem begrenzten zeitlichen Rahmen für die Implementation eine Vielzahl an Einrichtungs-Modellen in die Applikation einbinden und sich gleichzeitig auf die System-Programmierung fokussieren zu können, werden öffentlich zugängliche Modell-Datenbanken genutzt. Texturen und Materialien werden zum einen mit Hilfe von Bildbearbeitungsprogrammen und Texture-Map Rechnern gestaltet und zum anderen aus öffentlichen Online-Bibliotheken ausgewählt.

4.3.2 3D-Modellierung und Texturierung

Bei der Modellierung des Innenraums wird der Marriott Residence Inn Standard Studio Room als real existierende Vorlage für ein permanentes Standard-Hotelzimmer verwendet (s. Abb. 30). Diese Wahl wird durch mehrere Faktoren bedingt: Einerseits lassen sich viele Informationen über den Marriott Residence Inn Standard Studio Room finden, womit eine detaillierte und realitätsgetreue Darstellung des Hotelzimmers ermöglicht wird. Andererseits fördert der in mehrere Bereiche unterteilte und in einem neutralen Stil gehaltene Innenraum eine durchdachte Platzierung unterschiedlicher Einrichtungsgegenstände. Des Weiteren eignet sich der beinahe quadratische Aufbau des Zimmers für die virtuelle Anzeige auf der verwendeten Arbeitsunterlage (Mak, 2013, S. 27, 59, 60; Towneplace Suites Marriott, 2018).

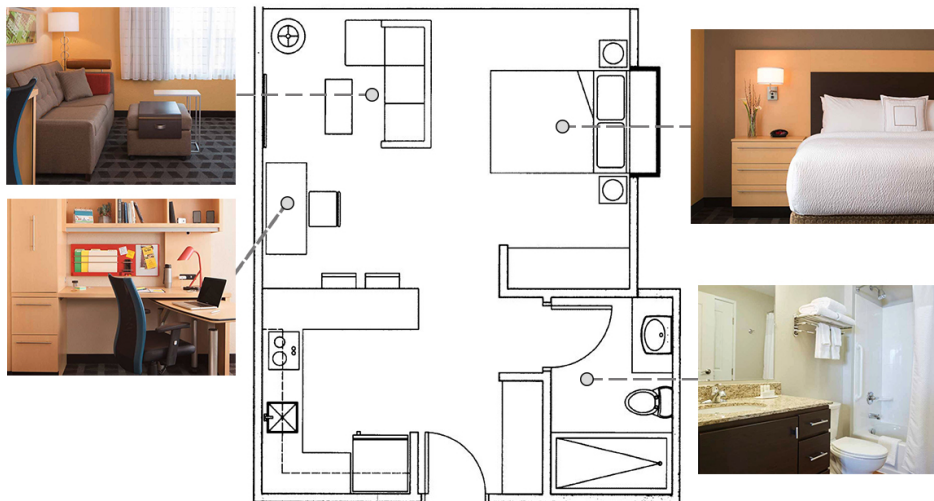


Abbildung 30: Grundriss und Eindrücke des Marriott Residence Inn Standard Studio Room (Towneplace Suites Marriott, 2018) [Foto, Graphik]

Für eine individuelle und kontextbezogene Einrichtung des Hotelzimmers stehen dem Nutzer unterschiedliche Bett-, (Kleider-)Schrank-, Sofa-, Sitz-

gruppen- und Schreibtisch-Arten zur Verfügung. Um einen möglichst hohen Detailgrad der Modelle zu erzeugen, werden die Objektoberflächen mit Materialien bestehend aus Texture-Maps dargestellt.

4.3.3 Augmented Reality Teil

Die AR-Ansicht stellt die Ausgangsbasis und zugleich den Editor-Modus der Anwendung dar (s. Abb. 31). Zur Realisierung des AR-Teils wird das vom System erfasste Kamerabild im VR-HMD angezeigt. Mit Hilfe von optischem Tracking werden vordefinierte Marker mit jeweils einem virtuellen Objekt überlagert, wobei das Hotelzimmer – anders als die Möbelstücke – fest auf der Arbeitsunterlage verankert ist und nicht vom Nutzer verändert werden kann.

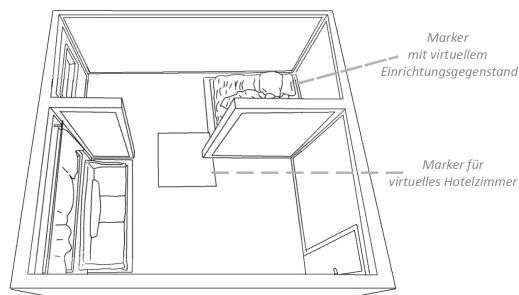


Abbildung 31: Konzeptzeichnung: AR-Ansicht

Mit dem Ziel, den Anwender möglichst gut bei der Einrichtung des Innenraums zu unterstützen, werden verschiedene Interaktionsarten mit ihren Stärken und Verwendungsmöglichkeiten in Form eines multimodalen UIs vereint. Für eine ebenso flüssige als auch leicht realisierbare Handlungsabfolge erfolgt die Positionierung und Ausrichtung der Einrichtungsgegenstände mittels eines intuitiv bedienbaren Tangible UIs. Mit Hilfe von Sprachbefehlen kann sich der User verschiedene Zusatzinformationen einblenden lassen, welche ihn bei seiner Aufgabe assistieren. So soll das System zum einen das kontextuelle Verständnis für die Anwendung mit Positionierungsvorschlägen nach funktionalen und visuellen Kriterien fördern und zum anderen dem Nutzer Anhaltspunkte bieten, wenn sich dieser bei der Platzierung der Möbelstücke unsicher ist. Um die Größenverhältnisse der einzelnen Objekte besser einschätzen zu können, lassen sich des Weiteren Maßangaben zu den einzelnen Einrichtungsgegenständen und Wänden des Hotelzimmers einblenden. Mittels einer weiteren Funktion wird das 3D-Modell des Hotelzimmers durch eine Grundriss-Ansicht ausgetauscht. Sinn dieser 2D-Darstellung ist es, Möbelstücke, welche von einer Zimmerwand verdeckt werden, sichtbar zu machen und somit alle platzierten Einrichtungsgegenstände gleichermaßen gut im Überblick zu behalten. Die dritte Interaktionsart stellt das Ray

Casting dar, mit der sich der Wechsel von AR und VR steuern lässt und im nächsten Abschnitt ausführlicher beschrieben wird (s. Tabelle 2).

Tangible UI	Sprachsteuerung	Ray Casting
Platzierung der Einrichtungsgegenstände	Anzeige von Positionierungsvorschlägen Anzeige der Grundriss-Ansicht Anzeige von Maß-Angaben	Wechsel von AR zu VR

Tabelle 2: Multimodulares UI in der AR-Ansicht

4.3.4 Übergang: Augmented Reality zu Virtual Reality

Der Nutzer hat jederzeit die Möglichkeit von der AR- in die VR-Ansicht zu wechseln und somit einen realitätsnahen Eindruck des Hotelzimmers und seiner Einrichtung zu gewinnen. Für einen kontrollierten Übergang teilt sich die Bedienung dabei in zwei Interaktionsschritte auf: Im Zentrum der Arbeitsunterlage wird eine virtuelle Info-Bubble angezeigt, mit deren Aktivierung via blickbasierten Ray Casting der Nutzer in den virtuellen Innenraum versetzt wird. Damit der Anwender beim Ray Casting weiß, auf welchem Objekt der Sicht-Fokus liegt, wird dieser durch einen Reticle Cursor dargestellt. Bei der Platzierung der Möbelstücke kann es vorkommen, dass die Info-Bubble unbeabsichtigt aktiviert und somit die AR-Ansicht verlassen wird. Um dies zu verhindern wird die Bedienung erst freigeschaltet, wenn der Nutzer mit einem Sprachbefehl seine Zustimmung gegeben hat.

Die Interaktion mittels Ray Casting ist für den Übergang von AR zu VR dahingehend essentiell, da die einzelnen auf der Arbeitsunterlage platzierten Marker optisch getrackt und somit deren Positionswerte und Ausrichtungen an die VR-Szene übergeben werden müssen. Anhand dieser Angaben wird die vom Nutzer in der AR-Ansicht vorgenommene Einrichtung in VR nachgebildet. Bevor der Anwender jedoch in das virtuelle Hotelzimmer versetzt wird, wird ihm zunächst ein Auswahl-Menü angezeigt, bei dem er eine von ihm präferierte VR-Fortbewegungsart auswählen kann.

4.3.5 Virtual Reality Teil

Sobald der Nutzer sich für eine Lokomotions-Methode entschieden hat, wird er in das von ihm individuell gestaltete Hotelzimmer versetzt und kann dieses frei aus der Normalansicht im Rahmen einer First-Person Experience erkunden. Für die Fortbewegung in VR stehen zwei verschiedene Methoden zur Verfügung: Free Locomotion und Walking in Place. Bei beiden Fortbewegungsarten wird das räumliche Empfinden gestärkt, welches eine wichtige Rolle für eine immersive Darstellung der virtuellen Umgebung spielt.

Des Weiteren lassen sich beide Lokomotionsarten in einem limitierten Bewegungsraum durchführen. Trotz des erhöhten Risikos für Motion Sickness bei Free Locomotion soll der Nutzer die Möglichkeit haben, zwischen den beiden VR-Fortbewegungsmethoden zu wählen, da die Steuerung mit dem Controller für viele Menschen vertraut ist und diese eine sehr präzise Lokomotion ermöglicht.

Neben der Lokomotion wird im Rahmen der Immersion auf eine möglichst realitätsnahe Darstellung der Szene geachtet. So spielt bei der visuellen Abbildung die virtuelle Szenen-Beleuchtung eine große Rolle, welche durch unterschiedliche Lichtquellen und Baked Lighting realisiert wird. Für die akustische Untermalung werden verschiedene Klangquellen mit Hintergrund- und Umgebungsgeräuschen in der VR-Szene implementiert.

4.3.6 Übergang: Virtual Reality zu Augmented Reality

Wie auch in der AR-Szene soll der User nach Belieben die Anwendungsansicht ändern und somit den VR-Modus verlassen können. Um einen flüssigen Arbeitsprozess zu erzeugen und auf die unterschiedlichen Präferenzen der Nutzer einzugehen, ist der Übergang mit zwei verschiedenen Interaktionsarten, nämlich mittels Tastendruck auf dem Controller oder per Sprachbefehl, möglich. Die Informationen über die Bedienung werden in Form eines Game Element UIs vermittelt. Mit der Wahl dieser UI-Art soll sich der Anwender ganz auf die Besichtigung des virtuellen Hotelzimmers konzentrieren können und nicht durch Anzeigen in seinem Sichtfeld abgelenkt werden. Sowohl bei dem Übergang von VR zu AR als auch in die andere Richtung wird auf möglichst geringe und für den Nutzer akzeptable Ladezeiten geachtet. Diese Erhöhung der Performanz wird durch Anpassungen bei der visuellen und akustischen Qualität erreicht, wobei der Detailgrad nicht unter diesen Vorkehrungen leiden soll.

5 Implementation der Anwendung

In den vorangegangenen Kapiteln wurde die Verbindung von Augmented und Virtual Reality genauer beleuchtet und ein Konzept für ein hybrides AR-/VR-System im Bereich Innenarchitektur vorgestellt. Auf dieser Basis wurde die geplante Anwendung in die Realität umgesetzt und die dabei vorgenommenen Implementationsschritte werden in den folgenden Abschnitten genau erläutert.

5.1 Setup

Wie bereits in Kapitel 4 beschrieben, muss für die Realisierung der Anwendung das nötige Setup aus Hardware- und Software-Komponenten gegeben sein. Die folgenden Unterkapitel beschreiben, wie das hybride AR-/VR-Wiedergabegerät und die für die Applikation nötige Arbeitsunterlage kreiert und welche Software-Programme und Webseiten für die Implementierung ausgewählt und integriert werden.

5.1.1 Hardware-Setup

Da sich zum Zeitpunkt der Projektdurchführung das Angebot für hybride AR-/VR-Wiedergabegeräte auf dem Markt auf wenige Modelle beschränkt ist, welche nicht für die private Nutzung angedacht sind, wird im Rahmen dieser Arbeit ein eigenes HMD erstellt, welches die beiden XR-Stufen vereint. Die Basis für dieses Headset stellt die VR-Brille Oculus Rift CV1 dar, welche über eine Auflösung von 1080 x 1200 Pixeln pro Auge, eine Bildwiederholfrequenz von 90Hz und einen diagonalen FOV von 110° verfügt. Für den Anwendungsaufbau werden vier externe Infrarot-Sensoren verwendet, welche zum Tracking der Oculus Rift dienen. Somit werden vom System sowohl Änderungen der Position des Headsets als auch dessen Ausrichtung erkannt, sodass sich der Nutzer mit sechs DOFs durch die VR bewegen kann (Martindale, Jon, 2018).

Damit die reale Umgebung eingefangen und in dem VR-HMD angezeigt werden kann, wird eine Kamera benötigt, welche folgende Kriterien erfüllen soll:

- Die Kamera soll mit einem USB-Kabel mit dem Computer verbunden und von einer Game Engine angesteuert werden können.
- Das Design der Kamera soll möglichst schmal sein, damit die Entfernung zwischen Kameralinse und den Augen des Anwenders möglichst gering ist.
- Des Weiteren sollten Auflösung und FOV von Kamera und VR-Headset möglichst übereinstimmen, um das räumliche Empfinden des Users nicht zu beeinträchtigen.

5. IMPLEMENTATION DER ANWENDUNG

- Damit eine flüssige Wiedergabe garantiert wird, soll die Kamera über eine Mindest-Bildwiederholrate von 60Hz verfügen (Steptoe, William, 2014).

Anhand dieser Anforderungen wird eine Logitech C310 HD Webcam mit einer Auflösung von 1280 x 960 Bildpunkten, einer Bildwiederholrate von 60Hz und einem diagonalen Sichtbereich von 60° ausgewählt (s. Tabelle 3) (Logitech, 2018).

	Oculus Rift CV1	Logitech C310 HD Webcam
Auflösung	1080 x 1200 Pixel pro Auge	1280 x 960 Pixel
Bildwiederholrate	90Hz	60Hz
Diagonales FOV	110°	60°
Verbindung mit Computer	HDMI 2.0, USB 3.0	USB 2.0

Tabelle 3: Technische Daten von Oculus Rift CV1 und Logitech C310 HD Webcam (Martindale, Jon, 2018; Logitech, 2018) [Foto]

Da bei dem FOV der Oculus Rift und der Logitech C310 eine Differenz von 50° besteht, wird die Webcam mit einem 110° Weitwinkel-Objektiv erweitert, sodass der Bildbereich der Kamera und des HMDs übereinstimmen. Der Unterschied zwischen den Kamera-Sichtbereichen ohne und mit Weitwinkelobjektiv wird in Abbildung 32 veranschaulicht. Die dargestellten Aufnahmen zeigen ein Koordinatensystem im Zentimeter-Maßstab, welches aus einer Entfernung von 61cm, was dem niedrigsten Wert der durchschnittlichen Armlänge entspricht, aufgenommen wurde (Pfeifer, Ralf, 2010).

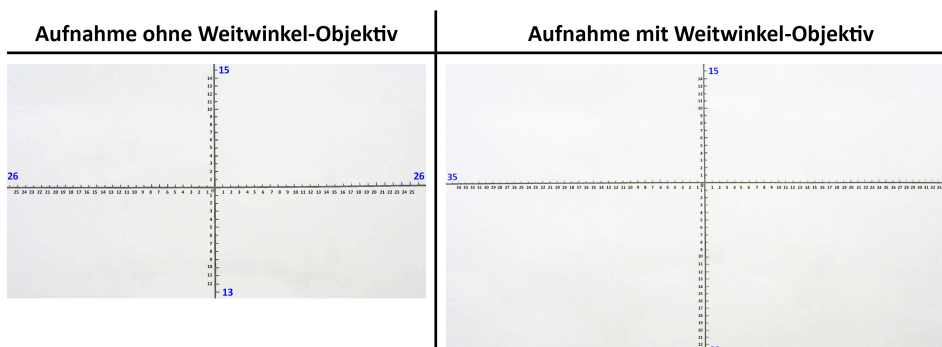


Abbildung 32: FOV ohne und mit Weitwinkel-Objektiv

5. IMPLEMENTATION DER ANWENDUNG

Da die Befestigung sowohl widerstandsfähig als auch abnehmbar sein soll, wird die Webcam horizontal an die Vorderseite des VR-HMDs mit Hilfe einer Tablet-Halterung montiert (s. Abb. 33).



Abbildung 33: Aufbau des hybrides AR-/VR-HMDs

Neben dem hybriden AR-/VR-HMD besteht das Hardware-Setup aus einer Arbeitsunterlage, auf welcher die AR-Marker bei der Ausführung der Applikation platziert werden. Die Größe der Unterlage wird anhand des Kamera-Sichtbereichs aus der Entfernung der niedrigsten durchschnittlichen Armlänge festgelegt. So beträgt die Seitenlänge der quadratischen Unterlage 50cm und ist somit aus einer Mindest-Entfernung von 61cm im Kamera-FOV vollkommen sichtbar. Mit dem Ziel, Tracking-Fehler durch schlechte Lichtverhältnisse zu vermeiden, wird die Arbeitsfläche von einem zehn Zentimeter hohen Rand umgeben, an welchem ein weißes Leuchtband befestigt ist. Mit dieser Vorrichtung wird die Unterlage von allen Seiten ausreichend gut beleuchtet. Damit die vom Nutzer platzierten AR-Marker nicht durch unbeabsichtigte leichte Stöße verschoben werden und somit Fehler bei der Einrichtung des Hotelzimmers entstehen, wird die Arbeitsunterlage mit einer Eisenfolie verkleidet und an den Unterseiten der Marker Magneten angebracht, was zu einer magnetischen Haftung der Marker auf der Unterlage

führt. Da eine Kamera eine einheitlich weiß gehaltene Fläche nicht fokussieren kann, wird der Boden der Arbeitsfläche mit schwarzem blickdichten Papier ausgelegt. Das Resultat der Arbeitsunterlage ist in Abbildung 34 zu sehen.

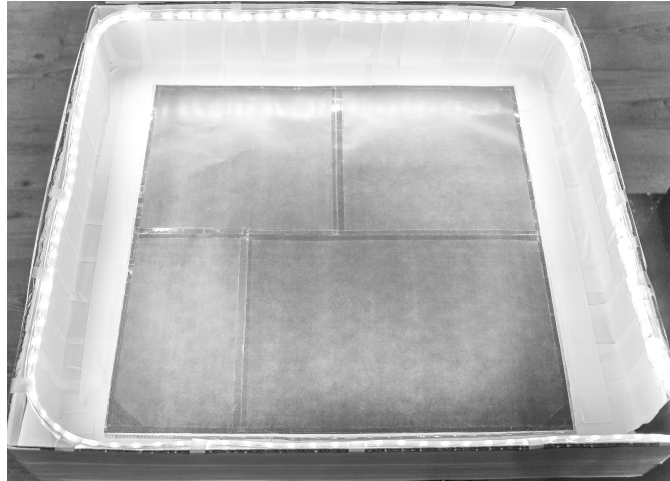


Abbildung 34: Aufbau der Arbeitsunterlage

Mit diesem Hardware-Setup wird die Wiedergabe von AR und VR ermöglicht sowie das Tracking und die Platzierung der AR-Marker optimiert.

5.1.2 Software-Setup

Neben den beschriebenen Hardware-Komponenten dienen verschiedene Software-Tools und Webseiten als Basis zur Implementierung der Anwendung. Für das Erstellen der einzelnen Szenen und der Programmierung der Skripts wird die Unity Game Engine in der Version 2017.3.0f3 mit Microsoft Visual Studio als Programmierumgebung verwendet. Gründe für die Wahl dieser Engine sind, dass Unity eine Vielzahl an Entwicklungsmöglichkeiten bietet, verschiedene AR- und VR- SDKs eingebunden werden können sowie die Implementierung der beiden XR-Darstellungsformen intern und durch externe Plugin Packages unterstützt wird (Kim u. a., 2014, S. 23; Linowes, 2015, S. 25; Unity Technologies, 2018a). Bei der Wahl des AR- SDKs werden die fünf gängigen Tools Vuforia, Arcolib, ARToolKit, Wikitude und XZIMG anhand der für die Applikation relevanten Kriterien untersucht: Das AR-SDK soll mit Unity Personal, welches die nicht-kommerzielle Standard-Version von Unity darstellt, kompatibel sein. Um die reale Umgebung erfassen zu können, soll eine externe Webcam angesteuert werden. Für die Implementierung eines Tangible UIs soll das AR-SDK des Weiteren möglichst fehlerrobustes Tracking von mindestens sieben gleichzeitig platzierten AR-Markern ermöglichen. Die persönlichen Erfahrungen mit den einzelnen AR-SDKs

5. IMPLEMENTATION DER ANWENDUNG

sowie Angaben von Entwicklern und externen Testberichten sind zur besseren Übersicht in der folgenden Tabelle festgehalten (s. Tabelle 4):

	VUFORIA	ARCOLIB	ARTOOLKIT	WIKITUDE	XZIMG
Ansteuerung einer USB-Webcam	<i>ja</i>	<i>ja</i>	<i>ja</i>	<i>nein</i>	/
Kompatibel mit Unity Personal	<i>ja</i>	<i>ja</i>	<i>ja</i>	/	<i>nein</i>
Optisches Tracking von Markern	<i>ja</i>	<i>ja</i>	<i>ja</i>	/	/
Bevorzugte Marker-Art	<i>schwarz-weiße Feature-Marker mit QR-Code ohne Rahmen</i>	<i>bunte Feature-Marker ohne Rahmen</i>	<i>schwarz-weiße Feature-Marker mit Rahmen</i>	/	/
Bewertung des Trackings	<i>gut: größtenteils stabiles Tracking, kein bis geringes Jittering, teilweise Verlust mancher Marker</i>	<i>mittel: nur teilweise stabiles Tracking, Jittering, Verlust mancher Marker</i>	<i>schlecht bis mittel: nur teilweise stabiles Tracking, Jittering, Kamera muss für Detektion sehr nah am Marker sein</i>	/	/
Anzahl der maximal simultan detektierbaren Marker	9	6	3	/	/

Tabelle 4: Überblick über AR-SDKs

Da Wikitude ausschließlich für Mobile Devices ausgelegt ist und keine Webcam ansteuern kann und XZIMG nur mit Unity Pro kompatibel ist, ließen sich keine Angaben zu deren Kriterien machen. Aufgrund der positiven Erfahrungen und Tests wird Vuforia als AR-SDK unter den drei verbliebenen Software-Tools ausgewählt. Da Oculus mit Oculus Virtual Reality (OVR) ein eigenes VR-SDK für Entwickler zur Verfügung stellt, welches in Unity eingebunden werden kann und eine Vielzahl an Beispiel-Szenen und Entwicklungsmöglichkeiten bietet, wird diese Software als VR-Plugin für die Engine verwendet. Für die Modellierung und Darstellung der Szenen-Objekte werden die Open Source 3D-Entwicklungs-Software Blender sowie die öffentlich zugänglichen Modell-Datenbanken turbosquid (www.turbosquid.com) und sketchfab (www.sketchfab.com) genutzt. Da neben den 3D-Modellen die Texturierung und Erstellung von Materialien eine wichtige Rolle bei der visuellen Darstellung spielen, werden die für die Anwendung benötigten 2D-Bilder mit Hilfe der Bildbearbeitungs-Software Photoshop CC angefertigt sowie von den Online-Textur-Archiven 3dtextures (www.3dtextures.me), cgbookcase (www.cgbookcase.com) und cc0textures (<http://cc0textures.com>) ausgewählt. Für das Kreieren von Texture-Maps, welche bei der Erstellung von Materialien Anwendung finden, wird der Online-Rechner NormalMap-Online (www.cpetry.github.io/NormalMap-Online) benutzt. Für den AR-Modus werden die Marker-Muster mit Hilfe der Webseite qrcode-generator (www.qrcode-generator.de) angefertigt. Des Weiteren werden frei zugängliche Audiodateien der Plattform freesound (www.freesound.org) für die Untermalung der virtuellen Szene verwendet.

Mit der Erstellung und Auswahl dieser Hardware- und Software-Komponenten ist das Fundament für die Realisierung der Anwendung geschaffen,

deren einzelne Implementationsschritte in den folgenden Abschnitten genauer beleuchtet werden.

5.2 Darstellung der Objekte

In den folgenden beiden Abschnitten wird auf die optische Darstellung der einzelnen Objekte im Rahmen der 3D-Modellierung, der Texturierung und der Erstellung von Materialien eingegangen.

5.2.1 Modellierung

Wie bereits in Kapitel 4.3.2 erläutert, wird für das in der Anwendung dargestellte Hotelzimmer der Marriott Residence Inn Standard Studio Room als reale Vorlage genommen. So wird der virtuelle Innenraum entsprechend dem Standard Studio Room mittels einer Trennwand und einem offenen Durchgang in zwei asymmetrisch große Bereiche aufgeteilt. Das Badezimmer liegt wie bei seinem realen Vorbild auf der rechten Seite des Zimmers und der Schlafbereich verfügt über einen visuellen Schwerpunkt, dargestellt durch ein Gemälde und davor positionierten Stehlampen. Im kleineren Raum wird der Blick des Betrachters auf eine große Fensterfläche gelenkt, durch welche er auf das Zentrum einer Altstadt sehen kann. Der Marriott Residence Inn Standard Studio Room wird dabei nicht eins zu eins als Vorlage in das virtuelle Modell überführt, sondern dem Anwendungsaufbau entsprechend angepasst. So reduziert sich die Größe des virtuellen Wohn- und Schlafbereichs gegenüber dem Original auf $28,614\text{m}^2$. Dadurch verringert sich der Skalierungsfaktor bei der Darstellung des virtuellen Hotelzimmers auf der Arbeitsunterlage. Als Resultat passt sich das virtuelle Modell gut in das Kamera-FOV ein und die Marker können ausreichend groß gestaltet werden, um eine gute Tracking-Qualität zu erzielen. Die Fläche des virtuellen Hotelzimmers ist somit um $5,614\text{m}^2$ größer als die eines Standard-Gästezimmers in den USA und wird in einem Maßstab von 1:13,75 auf die Arbeitsunterlage projiziert (Ellefsen u. a., 1979, S. 31). Eine weitere Änderung gegenüber dem Marriott Residence Inn Standard Studio Room ist, dass der virtuelle Innenraum über keinen Küchenbereich verfügt, da dieser wenig Möglichkeiten zur individuellen Platzierung von Möbelstücken bietet und nicht dem verbreiteten Standard eines Hotelzimmers entspricht. Mit dem Ziel, das virtuelle Hotelzimmer möglichst realistisch zu gestalten, entsprechen die einzelnen Komponenten realen Normen und Durchschnitts-Beschaffenheiten: So beträgt die Breite der Türen dem einflügeligen Drehtür-Standard entsprechend $0,8\text{m}$ und die Innen- und Außenwände sind gemäß der durchschnittlichen Breite von Ziegelsteinwänden $0,152\text{m}$ und $0,559\text{m}$ dick (Ellefsen u. a., 1979, S. 51; Ludwig, 2009, S. 11). Des Weiteren werden für eine erhöhte Immersion verschiedene Details, wie Gemälde, Teppiche, Gardinen, Decken- und Stehlampen, ein Kleiderständer, eine Blumenvase und diverse Badezimmer-

5. IMPLEMENTATION DER ANWENDUNG

artikel standardmäßig in das virtuelle Hotelzimmer eingefügt. Die Modellierung des Hotelzimmers wird in Blender vorgenommen, indem die Geometrie mehrerer Planes und 3D-Würfel im Edit-Modus transformiert und bearbeitet wird. Weitere Objektdetails werden aus Modell-Archiven eingefügt. Die Ergebnisse des Hotelzimmer-Modells sind in den Abbildungen 35 und 36 veranschaulicht.

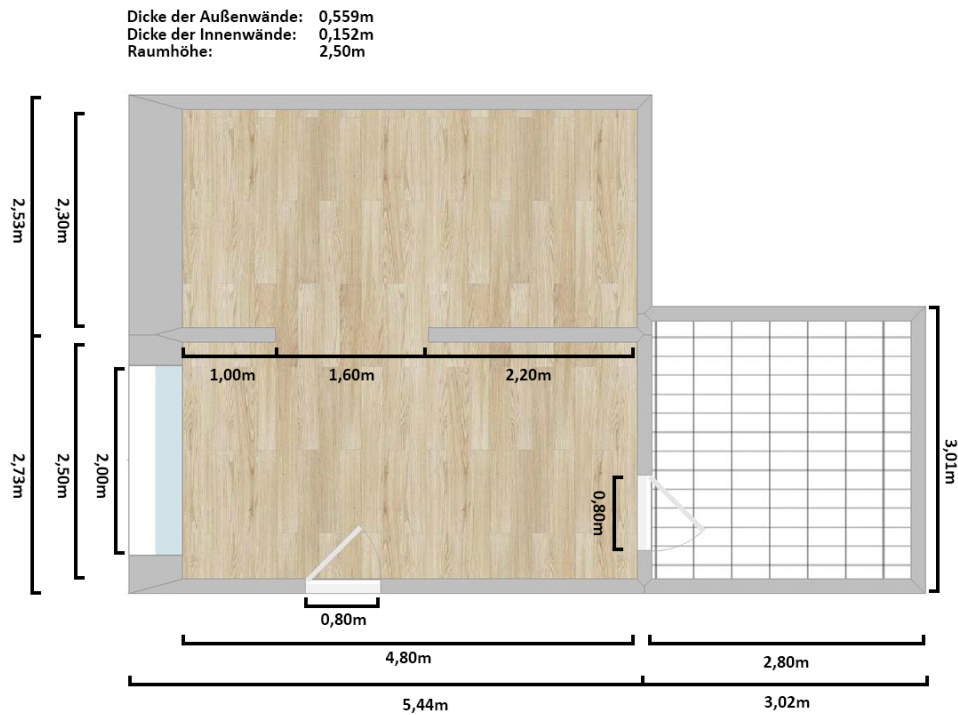


Abbildung 35: Grundriss des virtuellen Hotelzimmers

Da sich die Einrichtung des Hotelzimmers auf den Schlaf- und Wohnbereich konzentriert, werden in der AR-Ansicht, welche dem Editor-Modus entspricht, diese beiden Räume zur Möblierung angezeigt. Im VR-Modus ist es dem Nutzer zusätzlich möglich, das an das Hauptzimmer angeschlossene und nicht modifizierbare Badezimmer zu besichtigen, um einen guten Eindruck des gesamten Innenraums gewinnen zu können.

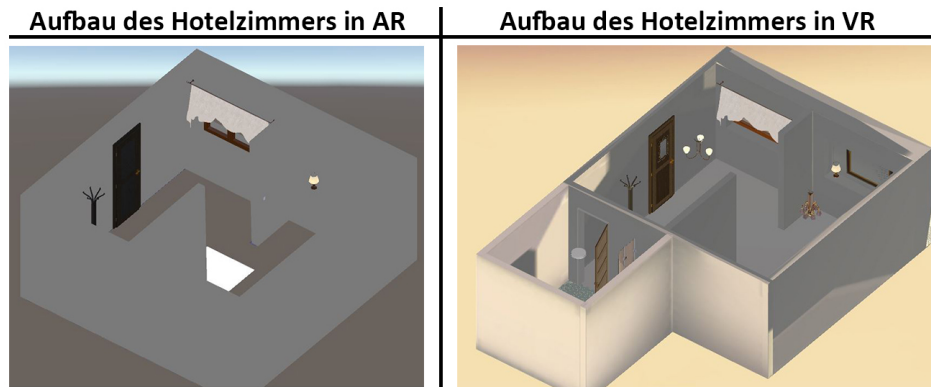


Abbildung 36: Modell des virtuellen Hotelzimmers in AR und VR

Neben dem virtuellen Hotelzimmer spielt die Wahl und Darstellung der einzelnen Einrichtungsgegenstände eine wichtige Rolle, da sich der Nutzer durch die von ihm individuell vorgenommene Gestaltung in dem Innenraum möglichst wohlfühlen soll (Siguaw und Enz, 1999, S. 44). So stehen dem Anwender bei der Wahl der Sitzgelegenheiten zwei stilistisch unterschiedliche Sofas und eine Sitzzecke zur Verfügung. Des Weiteren gibt es zwei Betten, die sich voneinander sowohl in ihrer Größe, ihrem Stil und Komfort abheben sowie drei unterschiedliche Schrank-Modelle und zwei Schreibtische (s. Abb. 37).

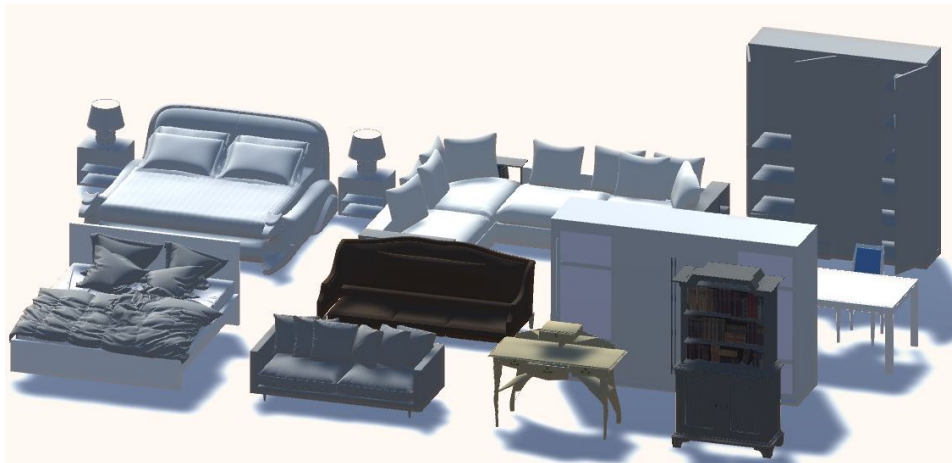


Abbildung 37: 3D-Modelle der Einrichtungsgegenständen

Das Hotelzimmer und die einzelnen Möbelstücke werden maßstabsgerecht in eine leere Unity-Szene eingefügt und es werden für die einzelnen Objekte Textur-Koordinaten erzeugt, welche Informationen darüber enthalten, wie Texturen auf die einzelnen Modelle projiziert werden.

5.2.2 Texturierung und Materialien

Mit dem Ziel, die Oberflächen der einzelnen Objekte möglichst detailliert darzustellen sowie visuelle UI-Anzeigen mit notwendigen Informationen zu erweitern, werden verschiedene Texturen mit Hilfe von Photoshop CC erstellt und aus Online-Archiven ausgewählt. Für ein besseres Verständnis der einzelnen UI-Elemente werden diese in den nächsten Kapiteln genauer behandelt. Bei der Wahl der Texturen für das Hotelzimmer wird auf einen einheitlich harmonischen und hellen Stil geachtet, der sich an dem Marriott Residence Inn Standard Studio Room anlehnt. So werden der Boden, Bilderrahmen und Türen im Wohn- und Schlafbereich von Holz-Texturen dominiert, die Wände erhalten einen grünen Anstrich und die Oberflächen der Zimmerdecke, Steh- und Deckenlampen, Teppiche und Gardinen werden durch verschiedene Muster dargestellt. Durch die helle Farbgestaltung, die große Fensterfassade und verschiedenen Gemälde soll ein positives Gefühl und Gemütlichkeit vermittelt werden (Siguaw und Enz, 1999, S. 44). Im Gegensatz dazu werden die Oberflächen des Badezimmers durch Marmor- und Keramik-Texturen in einem eher kalten und sterilen Stil präsentiert. Sowohl bei dem Badezimmerspiegel als auch bei dem Fenster im Hauptzimmer werden Texturen mit abgebildeten Staubpartikeln und Schlieren gewählt, um den Detailgrad und die Immersion in dem Innenraum zu erhöhen. Bei der Texturierung der einzelnen Einrichtungsgegenstände werden möglichst unterschiedliche Stile, wie unter anderem helle und dunkle Holzoberflächen, verschiedenfarbige Kissenbezüge, Lederverkleidungen und verschiedene Buchrücken gewählt. Damit soll eine individuelle Einrichtung für unterschiedliche Stilaffinitäten gewährleistet werden.

Da Objektaußenseiten durch die alleinige Darstellung von Texturen platt und unnatürlich wirken, wird für jedes Element ein Material erstellt, welches deren Oberflächenbeschaffenheit genauer beschreibt. So werden für jede Objektoberfläche Normal, Ambient Occlusion und Specular Maps generiert sowie abhängig von dem entsprechenden Element Metalness Maps ausgewählt, welche jeweils zu einem Material zusammengefasst werden. Weitere Einstellungsoptionen für die Materialien betreffen unter anderem die Materialglätte, Glanzpunkte und Reflexionen. Des Weiteren wird für jedes Material ein Shader bestimmt, welcher abhängig von den Einstellungen des Materials und der Umgebungsbeleuchtung die Farbwerte der einzelnen Pixel errechnet (Unity Technologies, 2018a). Bei jedem Material wird ein Rendering Modus angegeben, welcher bestimmt, wie transparent die jeweilige Objektoberfläche dargestellt wird. So sind bis auf Fenster, Lampenschirme und Gardinen alle Elemente des Hotelzimmers und der Einrichtungsgegenstände lichtundurchlässig. Eine gesonderte Rolle nehmen Lampen in der Szene ein, welche selbst Licht in einer bestimmten Farbe emittieren. Zur besseren Veranschaulichung des Vorgangs bei der Generierung von Objektoberflächen

5. IMPLEMENTATION DER ANWENDUNG

wird in Abbildung 38 ein Beispiel für die Texturierung und die Erstellung eines Materials dargestellt.

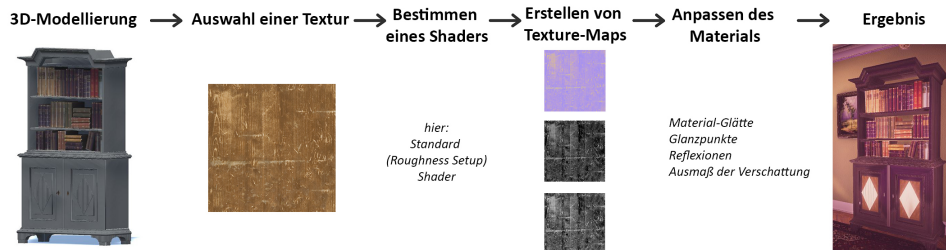


Abbildung 38: Erstellung eines Materials

Mit diesen Einstellungen soll ein visuell detaillierter Eindruck des Hotelzimmers und der einzelnen Einrichtungsgegenstände vermittelt werden (s. Abb. 39).



Abbildung 39: 3D-Modelle mit Materialien

5.3 Augmented Reality Teil

Nachdem die visuelle Darstellung der Objekte fertiggestellt und die Basis aus Hardware- und Software-Setup geschaffen wurde, kann die Implementierung der Anzeige der XR-Stufen durchgeführt werden. Dieses Kapitel beschreibt die im Rahmen dieser Arbeit vorgenommenen Schritte zur Realisierung des AR-Modus im hybriden AR-/VR-Headset.

5.3.1 Marker-Tracking und Anzeige der Augmented Reality Szene

Mit dem Ziel, die einzelnen Einrichtungsgegenstände mit Hilfe eines Tangible UIs in AR platzieren zu können, wird ein optisches Tracking-Verfahren mit vordefinierten Markern verwendet. Wie in Kapitel 5.1.2 erläutert, wird für die Realisierung der AR-Ansicht das von Qualcomm entwickelte AR-SDK Vuforia in Unity eingebunden. Vuforia verfügt über eine Vielzahl an AR-Implementierungsmöglichkeiten, für welche meist vorbestimmte 2D- oder

3D-Ziele eingesetzt werden. Bei der Realisierung des AR-Trackings mit Vuforia werden folgende Schritte durchgeführt, welche auf das in Kapitel 2.1.2 erläuterte Tracking-Verfahren aufbauen.

Zunächst wird ein Marker-Dictionary anhand der von Vuforia angegebenen Kriterien und selbst durchgeführten Erfahrungstests erstellt. Vuforia versteht unter einem Merkmalspunkt eine geometrische Form mit spitzem Winkel, wie beispielsweise Quadrate oder Rechtecke, welche sich in ihrem Grauwert möglichst von anderen Umrissen abhebt (s. Abb. 40) (Vuforia, 2018).

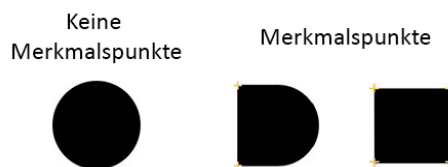


Abbildung 40: Vuforia-Merkmalspunkte (Vuforia, 2018) [Graphik]

So lassen sich rechteckige, rahmenlose Schwarz-Weiß-Marker, welche über möglichst viele Merkmalspunkte verfügen, sehr gut detektieren. Das für die Anwendung verwendete Dictionary besteht aus elf unterschiedlichen Markern, welche entsprechend der beschriebenen Kriterien mit einem QR-Code-Generator erstellt werden. Mit Ausnahme des für die Anzeige des Hotelzimmers verwendeten Markers werden die einzelnen Marker-Seitenlängen durch die Größen der jeweils korrespondierenden Möbelstücke in einem Maßstab von 1:13,75 bedingt. So variieren die verwendeten Marker-Seitenlängen zwischen sieben und 25 Zentimetern.

Nachdem eine Datenbank für die einzelnen AR-Marker erstellt und ein Vuforia-Lizenzschlüssel generiert wurde, wird in Unity eine neue Szene mit dem Index null erstellt, welche den AR-Modus der Applikation repräsentiert. **Szene 0** wird mit den einzelnen Komponenten für Vuforia und den erstellten 3D-Modellen erweitert. Damit AR-Marker getrackt werden können, wird die Standard Unity Kamera durch die in den Vuforia-Prefabs enthaltene **ARCamera** in der Szenen-Hierarchie ersetzt. Ein Prefab ist eine Objekt-Vorlage, von welcher beliebig viele Instanzen erzeugt werden können (Unity Technologies, 2018a). Die **ARCamera** verfügt über ihre eigene Szenen-Kamera sowie das **VuforiaBehaviour**-Skript. Mit Hilfe dieses Skripts wird das Tracking der einzelnen Marker gesteuert sowie das mit virtuellen Informationen überlagerte Kamerabild auf eine Plane gerendert und somit dem Nutzer angezeigt. Bei diesem Vorhaben werden folgende Einstellungen und Schritte mit Hilfe der **VuforiaBehaviour**-Komponente durchgeführt: Die für die Anwendung verwendete Webcam wird als Eingabegerät angegeben,

deren in Echtzeit übertragene Bilder in ein für die Programmierschnittstelle OpenGL ES (Open Graphics Library for Embedded Systems) kompatibles Format umgewandelt werden. Das Kamerabild wird mittels Thresholding und Kantendetektion verarbeitet und auf Rechteckformen untersucht. Die dabei gefundenen Merkmalspunkte werden mit den in der Datenbank gespeicherten Markern auf Gemeinsamkeiten untersucht und somit die Referenz-Marker detektiert. In einem weiteren Schritt wird die Pose für jeden gefundenen Marker berechnet und fortwährend in Echtzeit aktualisiert. Jeder Marker wird durch ein lokales Koordinatensystem definiert, welches in Abbildung 41 veranschaulicht ist (Vuforia, 2018).

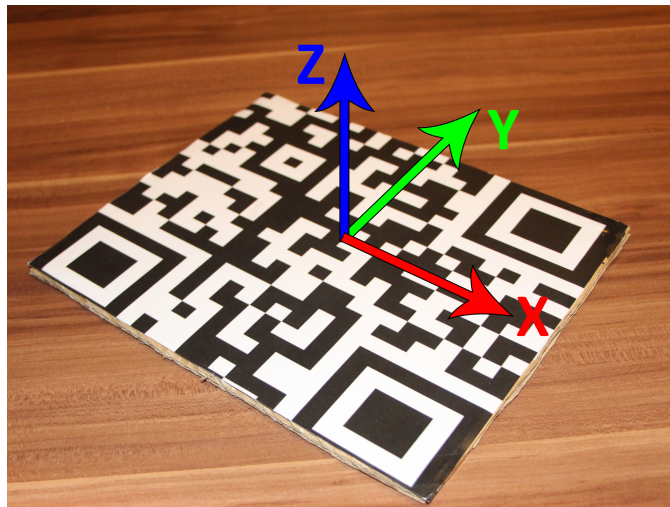


Abbildung 41: Marker-Koordinatensystem

Für die Anzeigart der AR-Szene wird eine digitale Brille mit Video See-Through eingestellt. Da die AR-Szene in einem VR-Headset wiedergegeben werden soll, wird das OVR-SDK integriert und das `OVRCameraRig` in die Szenen-Hierarchie eingefügt. Das `OVRCameraRig` enthält die beiden Skripte `OVRCameraRig` und `OVRManager`. Mit Hilfe des `OVRCameraRig`-Skripts wird das Kamerabild stereoskopisch im Headset angezeigt sowie das Oculus Rift Headset getrackt. Die `OVRManager`-Komponente stellt das Interface für das VR-HMD dar, mit welchem unter anderem das Tracking des Headsets auf drei oder sechs DOFs eingestellt werden kann. Für die Realisierung dieser Anwendung wird letztere Variante gewählt (Vuforia, 2018). Damit die AR-Szene im VR-Headset angezeigt werden kann, wird das `OVRCameraRig-GameObject` mit einem `GameObject TrackingSpace` erweitert. `TrackingSpace` erhält folgende Kinder: Es werden drei Szenen-Kameras hinzugefügt, wobei im VR-Headset zwei Kameras ihr Bild für jeweils ein Auge anzeigen und die dritte Kamera den Video-Stream für beide Augen rendert. Des Weiteren wird ein `GameObject`

`VuforiaCenterAnchor` als Kind von `TrackingSpace` hinzugefügt, welches die Positions- und Rotationswerte $(0, 0, 0)$ aufweist. In einem weiteren Schritt werden im `VuforiaBehaviour`-Skript der `ARCamera` die linke und rechte Kamera für das Video See-Through den jeweils korrespondierenden Kameras von `TrackingSpace` zugewiesen sowie `VuforiaCenterAnchor` als zentraler Ankerpunkt eingestellt. Für eine angenehme Abbildung der realen Umgebung und bessere Einschätzung der räumlichen Distanzen wird die Video-Plane so zum Betrachter translatiert, dass dieser eine möglichst natürliche Sicht im AR-Modus erhält. Mit diesen Vorkehrungen ist die Basis für eine korrekte Anzeige der AR-Szene geschaffen (Vuforia, 2018).

Um die erstellten AR-Marker mit ihren jeweils korrespondierenden virtuellen Objekten zu verbinden und diese anzeigen zu können, werden folgende Schritte angewandt: Die Szenen-Kamera von `TrackingSpace`, welche ihr Bild für beide Augen im Headset rendert, erhält ein Kind `TrackableParent`. Für jeden erstellten AR-Marker wird eine Instanz des Vuforia-Prefabs `ImageTarget` erstellt und als Kind von `TrackableParent` integriert. Im finalen Schritt werden den einzelnen `ImageTarget`-Instanzen jeweils ein AR-Marker zugewiesen und mit dem jeweils zugehörigen 3D-Modell erweitert. Der Aufbau der Szenen-Hierarchie ist für ein besseres Verständnis in Abbildung 42 veranschaulicht.

5. IMPLEMENTATION DER ANWENDUNG

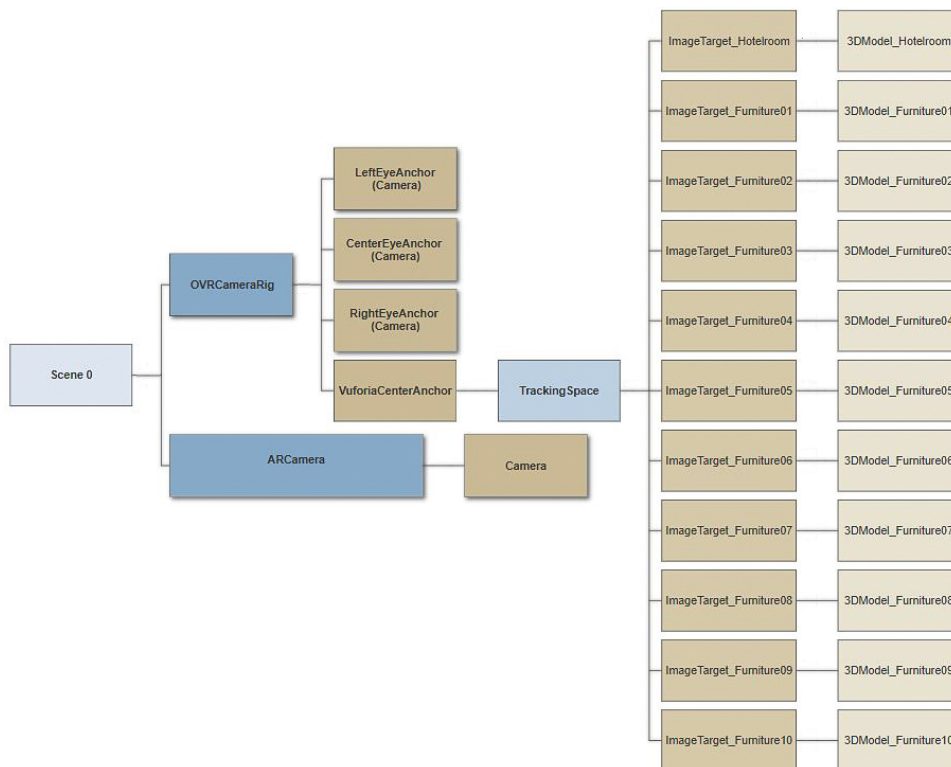


Abbildung 42: Szenen-Hierarchie im AR-Modus

Führt der Nutzer die Applikation aus, werden ihm die einzelnen virtuellen Objekte korrekt auf ihren korrespondierenden Markern im VR-HMD angezeigt (s. Abb. 43). Damit ist die Basis für die AR-Ansicht fertiggestellt.

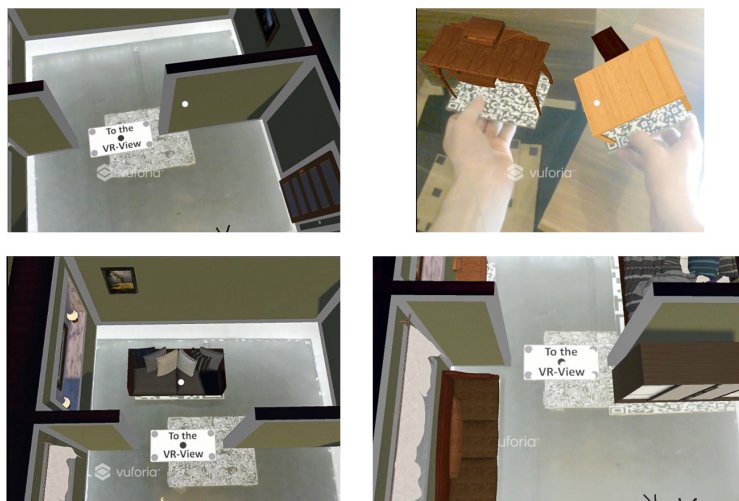


Abbildung 43: Demonstration der AR-Szene

5.3.2 User Interface

Mit den in Abschnitt 5.3.1 beschriebenen Implementierungsschritten ist es dem Nutzer möglich, verschiedene virtuelle Einrichtungsgegenstände in einem Hotelzimmer mittels eines Tangible UIs zu platzieren, wobei ihm die Objekte in AR mit Hilfe eines Video See-Through HMDs angezeigt werden. Um den Anwender bei der Platzierung der Einrichtungsgegenstände zu unterstützen, werden verschiedene Assistenzfunktionen in die Anwendung integriert.

Mit dem Ziel, dem User Anhaltspunkte zur Inneneinrichtung nach funktionalen und visuellen Kriterien zu geben, wird ein UI mit Positionierungsvorschlägen realisiert. Dafür werden zunächst Texturen für die Symbole der einzelnen Einrichtungsgegenstand-Arten entsprechend den Architektur-Normen mit Hilfe von Photoshop CC erstellt. In Unity wird als Kind von `ImageTarget_Hotelroom` ein neues `GameObject` mit dem Namen `FurniturePositionSuggestion` erstellt. `FurniturePositionSuggestion` erhält vier `Planes` als Kinder, welche durch die angefertigten Texturen der Einrichtungsgegenstand-Symbole erweitert werden. Auf Basis der in Kapitel 2.3 dargestellten Richtlinien und mit den Ratschlägen von Innenarchitekten wird eine Vorlage für Positionierungsvorschläge in dem Hotelzimmer gestaltet: So wird darauf geachtet, dass im Rahmen der funktionalen Anforderungen die einzelnen Möbelstücke so platziert werden, dass ausreichend Platz besteht, um diese nutzen und sich durch den Innenraum bewegen zu können. Des Weiteren stehen die Einrichtungsgegenstände in einem funktionalen Kontext zueinander: Das Bett und der Schreibtisch befinden sich im gleichen Raum und bilden somit den Schlafbereich des Zimmers. Der Wohnzimmerbereich, welcher der Kommunikation und Unterhaltung dient, besteht aus einem Sofa und einem Schrank. Für die Einhaltung visueller Aspekte werden die einzelnen Möbelstücke entlang den Hotelzimmerwänden ausgerichtet.

Damit sich der Nutzer die Positionierungsvorschläge jederzeit anzeigen lassen kann, wird ein Skript namens `AR_FurniturePositionSuggestion` erstellt und als Komponente von `ImageTarget_Hotelroom` hinzugefügt. Ziel dieses Skripts ist es, mit Hilfe von Sprachbefehlen das `GameObject FurniturePositionSuggestion`, deren Kinder die positionierten Symbole der Einrichtungsgegenstände darstellen, nach Belieben ein- und auszublen- den. Für die Implementierung dieses Skripts wird auf die Klassen `UnityEngine.Windows.Speech` und `System.Linq` zugegriffen, welche es der Applikation ermöglichen, Spracheingaben via Mikrofon zu registrieren (Microsoft Corporation, 2018). So wird ein neues Sprachbefehl-Dictionary erstellt, welches die Befehle „Suggestions On“ und „Suggestions Off“ enthält. Eine Funktion wird implementiert, welche die vom Mikrofon

5. IMPLEMENTATION DER ANWENDUNG

erfasste Sprache in Text umwandelt und mit den im Sprachbefehl-Dictionary enthaltenen Befehlen vergleicht. Zu Beginn befindet sich das `GameObject FurniturePositionSuggestion` außerhalb des Sichtfelds des Nutzers. Wird der Befehl „Suggestions On“ von dem System erkannt, wird die Position des `GameObjects` so verändert, dass sich die Texturen mit den Vorschlägen zur Positionierung der Einrichtungsgegenstände an ihren vordefinierten Stellen im virtuellen Hotelzimmer befinden. Der Nutzer erhält mit dieser Funktion Anhaltspunkte zur Positionierung der Möbelstück-Arten (s. Abb. 44). Möchte der Anwender die UI-Elemente ausblenden, so werden diese mit dem Befehl „Suggestions Off“ wieder aus dem FOV des Users translatiert.

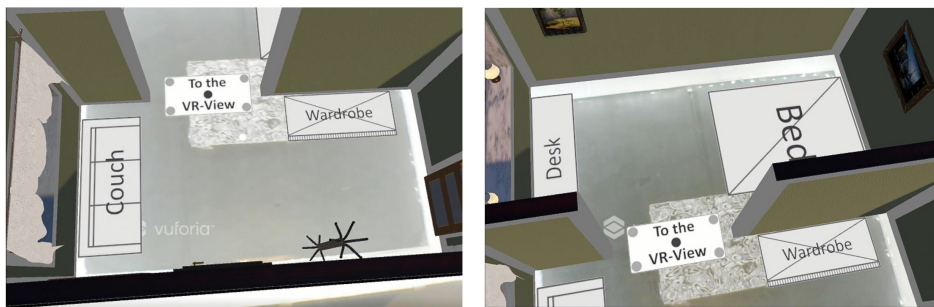


Abbildung 44: AR-Assistenzfunktion: Positionierungsvorschläge

Da es vorkommen kann, dass platzierte Möbelstücke von Zimmerwänden oder größeren Einrichtungsgegenständen verdeckt werden, soll eine Hilfsfunktion implementiert werden, um dieses Problem zu lösen. Ziel ist es, ein UI zu erschaffen, mit welchem der Nutzer in eine Grundriss-Ansicht wechseln kann und somit alle 3D-Objekte in 2D angezeigt werden. Für die Realisierung dieses Vorhabens werden für das Hotelzimmer und die einzelnen Einrichtungsgegenstände Texturen erstellt, welche die jeweiligen Objekte ohne räumliche Struktur darstellen. Für jedes `ImageTarget` wird ein weiteres `GameObject` als Kind generiert, welches Planes mit den korrespondierenden 2D-Texturen enthält. Analog der Hilfsfunktion zur Anzeige der Positionierungsvorschläge wird ein Skript als Komponente von `ImageTarget_Hotelroom` erstellt. So wird ebenfalls ein Sprachbefehl-Dictionary generiert, welches die Schlagworte „Two D“ und „Three D“ enthält.

5. IMPLEMENTATION DER ANWENDUNG

Zu Beginn befinden sich die 3D-Objekte an ihren vorgesehenen Positionen auf den Markern, während sich die 2D-Texturen außerhalb des Nutzer- FOVs befinden. Wird der Sprachbefehl „Two D“ erkannt, so werden die lokalen Positionen der dreidimensionalen Modelle und Texturen vertauscht. Damit erhält der Nutzer eine 2D-Ansicht der einzelnen AR-Objekte, wodurch verdeckte Möbelstücke sichtbar werden (s. Abb. 45). Mit dem Keyword „Three D“ wird wieder zu der 3D-Perspektive zurückgeschaltet.

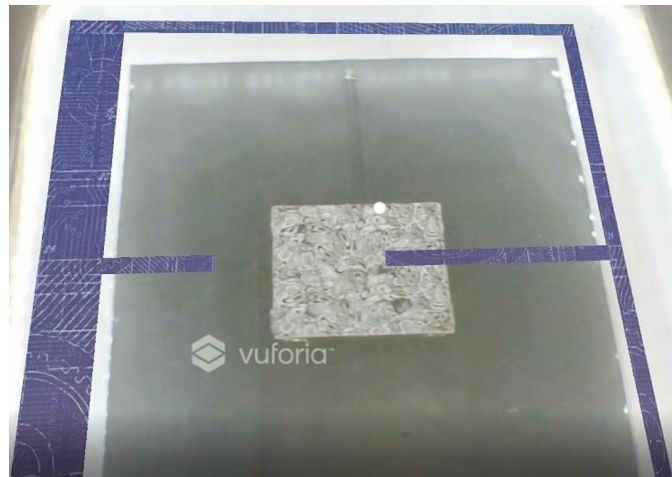


Abbildung 45: AR-Assistenzfunktion: Grundriss-Ansicht

Ist die Grundriss-Ansicht aktiviert, so wird eine öffentliche Variable `twoD` vom Typ `bool` auf `true` gesetzt. Andernfalls hat `twoD` den Wert `false`. Dieser Implementierungsschritt ist für die dritte und letzte AR-Assistenzfunktion von Relevanz.

Um die Größenverhältnisse der einzelnen Objekte besser einschätzen zu können, soll es dem User jederzeit möglich sein, sich Maßangaben zur Länge und Breite der einzelnen Einrichtungsgegenstände sowie Hotelzimmerwände anzeigen zu lassen. Im Rahmen der 3D-Modellierung wurden für jedes Objekt Angaben zur entsprechenden realen Größe festgehalten. Diese Informationen werden – nach dem gleichen Vorgehen wie in den vorherigen Abschnitten – als Texturen dargestellt und als Kinder der GameObjects `SizesX`, wobei `X` für das jeweilige 3D-Objekt steht, hinzugefügt. Ein Skript wird implementiert, welches es dem Anwender ermöglicht, über den Befehl „Sizes On“ die einzelnen Objekt-Maße ein- und über „Sizes Off“ auszublenden. Die Maßangaben werden dabei so dargestellt, dass sie für den Nutzer stets gut ersicht-lich sind. So erscheinen die Objektgrößen in der 3D-Ansicht am obersten Punkt der Modelle und in der 2D-Ansicht auf einer Ebene mit den jeweiligen Texturen. Für die Realisierung dieses Ziels wird die – im Rahmen der vorhergehenden Hilfsfunktion implementierten – Variable `twoD` verwendet.

Anhand von `twoD` erhält das Skript `AR_Sizes` Informationen darüber, ob der Nutzer die 2D- oder 3D-Perspektive ausgewählt hat und entsprechend werden die Maßangaben positioniert (s. Abb. 46). Des Weiteren wird die Positionierung der Objektgrößen reguliert, wenn zwischen den beiden Perspektiven hin und her geschaltet wird, während die Maßangaben aktiviert sind.

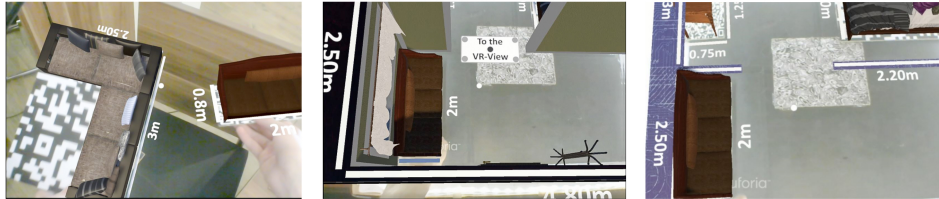


Abbildung 46: AR-Assistenzfunktion: Massangaben

5.4 Übergang: Augmented Reality zu Virtual Reality

Mit dem Abschluss des vorhergehenden Kapitels sind alle Basis- und Assistenzfunktionen für den AR-Modus der Applikation implementiert. Gemäß dem Ablauf der Anwendung konzentrieren sich die folgenden Abschnitte auf den Übergang von AR zu VR, welche eine der herausforderndsten Kernkomponenten der Arbeit darstellt.

5.4.1 Basis für den Szenenwechsel

Um von Augmented zu Virtual Reality wechseln zu können, müssen zunächst die dafür erforderlichen Unity-Szenen existieren. Neben der AR-Szene, welche den Index null trägt, werden zwei weitere Szenen mit den Indizes eins und zwei erstellt. Dabei stellt **Szene 2** den VR-Modus und **Szene 1** einen Zwischenbereich dar, in welchem der Nutzer eine von ihm präferierte VR-Fortbewegungsart auswählen kann. Der Aufbau von **Szene 1** wird in Kapitel 5.4.3 genauer erläutert. Ziel ist es, mit Hilfe eines UIs von **Szene 0** zu **Szene 1** und nach der Wahl einer Lokomotions-Methode zum VR-Modus zu gelangen. Das dafür nötige UI wird im Folgenden genauer behandelt.

Zunächst wird eine Textur erstellt, welche als Oberfläche für eine Info-Bubble dienen soll. Die Textur wird mit einem Informationstext und fünf Kreisen in den Ecken und dem Mittelpunkt erweitert. Die Info-Bubble wird in Unity mit Hilfe einer neuen Plane, auf welcher die Textur dargestellt wird, erstellt. Das UI-Element soll vom Nutzer via Ray Casting aktiviert werden können. Aus diesem Grund wird die Info-Bubble so im virtuellen Hotelzimmer platziert, dass sie stets gut sichtbar ist, jedoch nicht die Platzierung der Einrichtungsgegenstände behindert.

Um dem User beim Ray Casting Rückmeldung darüber zu geben, ob das UI-Element ausgewählt wird, wird ein neues Skript mit dem Namen `GazePlaneColor` als Komponente der Info-Bubble hinzugefügt. Ziel dieses Skripts ist es, dass sich die ursprünglich weiße Info-Bubble in einem fließenden Übergang immer dunkler verfärbt, je länger sie im Sicht-Fokus des Nutzers liegt. Wird die UI-Anzeige nicht via Ray Casting ausgewählt, so behält sie ihre Ausgangsfarbe bei. Um diese Ziele in die Realität umzusetzen, wird zunächst eine Variable `gazeTimer` erstellt, welche angibt, wie lange die Info-Bubble angeschaut wird. Zu Beginn ist `gazeTimer` auf null gesetzt. Des Weiteren wird ein RGB-Farbwert `greyDark` festgelegt, welchen das Anzeige-Element nach einer bestimmten Zeit annehmen soll. In den nächsten Schritten wird die aktuelle Position der Szenen-Kamera gespeichert und ausgehend von diesem Punkt ein Strahl definiert, welcher in Echtzeit mit der Veränderung der Kamera aktualisiert wird. Mit Hilfe einer Abfrage wird erkannt, ob der Strahl die Info-Bubble trifft. Für die Zeitspanne, in der eine Kollision zwischen Strahl und UI-Element registriert wird, erhöht sich der Wert von `gazeTimer` um die gemessene Dauer in Echtzeit. Gleichzeitig wird bei der Anzeige der Info-Bubble zwischen weiß und dem eingangs definierten Farbwert `greyDark` interpoliert, wodurch ein fließender Übergang entsteht. Nach einer Dauer von drei Sekunden hat die UI-Anzeige den Wert `greyDark` erreicht, welche ebenfalls der Zeitspanne zur Aktivierung des UIs für den Szenenwechsel darstellt.

Für ein besseres UI-Feedback soll dem Nutzer mit Hilfe eines Reticle Cursors angezeigt werden, worauf sein Sichtfokus liegt. Zu diesem Zweck wird eine neue Textur kreiert, welche einen weißen Kreis mit schwarzer Umrandung darstellt und an einer festen Position im Mittelpunkt des Nutzer-Sichtfeldes angebracht ist. Der Reticle Cursor wird dabei so skaliert und vom Nutzer wegtranslatiert, dass dieser den Anwender nicht stört und gut sichtbar bleibt. Damit der Reticle Cursor nicht von virtuellen Objekten verdeckt wird, wenn sich der User diese von Nahem ansieht, soll mit Hilfe eines weiteren Skripts Abhilfe geschaffen werden. Wie bei `GazePlaneColor` wird bei der neuen Implementierung ein Strahl ausgehend von der Kamera definiert und registriert, ob dieser auf ein virtuelles Objekt trifft. Ist dies der Fall, so wird die Distanz zwischen Kamera und Kollisionsobjekt gemessen. Unterschreitet der gemessene Wert eine vordefinierte Schwelle, so wird der Reticle Cursor näher zum Nutzer translatiert. Somit bleibt das UI-Element für den Anwender immer gut sichtbar.

Da bei dem Übergang von AR zu VR die Sichtbarkeit der platzierten Marker eine wichtige Rolle spielt, um die Positionswerte und Ausrichtungen der Einrichtungsgegenstände korrekt übergeben zu können, wird eine Hilfsanzeige implementiert. So wird ein neues UI mit Namen `reticleCursor2` mit Hilfe einer Textur erstellt, welche aus vier Kreisen besteht. Diese Textur befindet

sich zu Beginn außerhalb des User-Sichtfelds. Das Skript `GazePlaneColor` wird so erweitert, dass `reticleCursor2` im Nutzer-FOV erscheint, sobald die Info-Bubble angeschaut wird. Bewegt der Nutzer seinen Kopf so, dass die Punkte von `reticleCursor2` mit den Kreisen auf der Info-Bubble grob übereinstimmen, kann sich der Anwender sicher sein, dass alle Marker gut getrackt werden und eine korrekte Informationsübergabe erfolgt (s. Abb. 47)

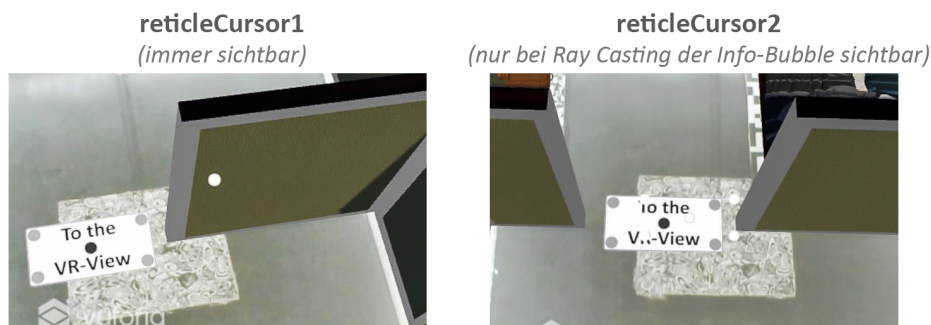


Abbildung 47: `reticleCursor1` und `reticleCursor2`

5.4.2 Bereitstellen der Augmented Reality Informationen für Virtual Reality

Nachdem die Basis für den Wechsel von dem AR- in den VR-Modus geschaffen wurde, soll der Übergang zwischen den Szenen implementiert werden. Dabei ist die Übergabe der relevanten Informationen entscheidend für die VR-Ansicht. Wird die Info-Bubble mittels Ray Casting aktiviert, werden alle Informationen über die Einrichtung des Hotelzimmers gespeichert und an die VR-Szene übergeben. Im VR-Modus werden die übermittelten Werte geladen und entsprechend die Szene aufgebaut.

Für die Realisierung des Wechsels der beiden XR-Stufen wird ein neues Skript mit dem Namen `TransitionARtoVR` erstellt, welches zur besseren Veranschaulichung mit Grafiken und Code-Fragmenten erläutert wird. Zu Beginn der Implementierung wird eine Instanz der Klasse und eine öffentliche `bool`-Variable mit dem Namen `ready` erstellt. Des Weiteren werden unterschiedliche Variablen deklariert, welche mit den zu übergebenden Informationen gefüllt werden sollen. So gibt es für das Hotelzimmer und jeden Einrichtungsgegenstand Vektoren für deren jeweilige Positionswerte und Ausrichtungen.

5. IMPLEMENTATION DER ANWENDUNG

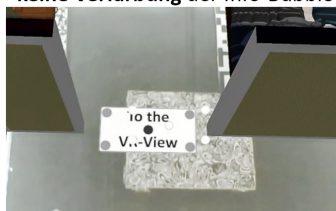
```
1 public Vector3 imageTarget_hotelroom_pos;  
2 public Vector3 imageTarget_furniture01_pos;  
3 ...  
4 public Vector3 imageTarget_furniture10_pos;  
5  
6 public float imageTarget_hotelroom_eulerAngleY;  
7 public float imageTarget_furniture01_eulerAngleY;  
8 ...  
9 public float imageTarget_furniture10_eulerAngleY;
```

In dem vorangegangenen Kapitel wurde die Info-Bubble vorgestellt, welche das UI für den Szenenwechsel darstellt. Da es bei der Einrichtung des Hotelzimmers im AR-Modus vorkommen kann, dass dieses UI-Element unbeabsichtigt aktiviert wird, soll eine kontrollierbarere Steuerung implementiert werden. So wird in `TransitionARtoVR` ein neues Sprach-Dictionary mit dem Keyword „Ready“ kreiert und folgende Schritte vorgenommen. Wird der Sprachbefehl „Ready“ von dem System registriert, so wird die Variable `ready` auf `true` gesetzt.

Auf diese Implementierung folgen zwei Schritte:

Erstens: Das Skript `GazePlaneColor`, welches der Feedback-Steuerung der Info-Bubble dient, greift auf die Variable `ready` zu und fragt deren Wert ab. Wurde `ready` auf `true` gesetzt, so wird die Interaktion via Ray Casting aktiviert und die Info-Bubble verfärbt sich, wenn sie im Sichtmittelpunkt des Nutzers liegt (s. Abb. 48). Wurde hingegen der Sprachbefehl nicht erkannt, so ist die Interaktion via Ray Casting für das UI-Element deaktiviert und der User kann die einzelnen Objekte betrachten, ohne darauf achten zu müssen, ob sein Sichtfokus auf die Info-Bubble fällt.

Keine Registrierung des **Sprachbefehls** und somit
keine Verfärbung der Info-Bubble



Registrierung des **Sprachbefehls** und somit
Verfärbung der Info-Bubble



Abbildung 48: Verfärbung der Info-Bubble

5. IMPLEMENTATION DER ANWENDUNG

Zweitens: Das Skript `TransitionARtoVR` erhält eine Variable `countDown` mit dem Initial-Wert drei. Wurde der Befehl „Ready“ erkannt und fokussiert der Anwender die Info-Bubble, so reduziert sich `countDown` für die Dauer der Ray Casting Zeit.

```
1 if (currentButton != null && ready == true)
2 {
3     countDown -= Time.deltaTime;
4     ...
```

Ist `countDown` bei null angekommen, so werden die eingangs deklarierten Variablen für die Informationsübergabe mit den Positionswerten und Ausrichtungen der jeweiligen Objekte in Relation zu ihrem Parent befüllt.

```
1 if (countDown <= 0.0f)
2 {
3
4     imageTarget_hotelroom_pos = transform.InverseTransformPoint(
5         ↪ GameObject.Find("imageTarget_hotelroom").transform.
6         ↪ localPosition);
7     imageTarget_furniture01_pos = GameObject.Find("furniture01").
8         ↪ transform.localPosition;
9     ...
10    imageTarget_furniture10_pos = GameObject.Find("furniture01").
11        ↪ transform.localPosition;
12
13    imageTarget_hotelroom_eulerAngleY = GameObject.Find("
14        ↪ imageTarget_hotelroom ").transform.localEulerAngles.y;
15    ImageTarget_furniture01_eulerAngleY = GameObject.Find("furniture01"
16        ↪ ).transform.localEulerAngles.y;
17    ...
18    imageTarget_furniture10_eulerAngleY = GameObject.Find("furniture10"
19        ↪ ).transform.localEulerAngles.y;
20    ...
```

Da nur die Einrichtungsgegenstände in VR dargestellt werden sollen, welche vom Nutzer im AR-Modus platziert wurden, müssen zwei Fehlerquellen ausgeschlossen werden. Das erste potenzielle Problem betrifft Möbelstücke, welche vom User nicht verwendet und somit nicht getrackt wurden. Nicht-getrackte Marker können daran erkannt werden, dass sie automatisch die Initial-Positionswerte (0, 0, 0) erhalten. Anhand dieses Merkmals werden die nicht platzierten Einrichtungsgegenstände identifiziert und somit nicht in der VR-Szene angezeigt. Die zweite Fehlerquelle bezieht sich auf Möbelstücke, welche zunächst im AR-Modus platziert, jedoch vor der Aktivierung der Info-Bubble wieder weggelegt wurden. Ursache für die fehlerhafte Informationsübergabe dieser Einrichtungsgegenstände ist, dass automatisch jede zuletzt getrackte Marker-Position und -Ausrichtung gespeichert und somit an die VR-Szene übergeben wird. Gelöst wird dieser Bug durch das Ausnutzen folgender Feststellung: Solange ein Marker getrackt wird, verändert sich stetig dessen Position und Ausrichtung in Relation zur Kamera. Mit

5. IMPLEMENTATION DER ANWENDUNG

Hilfe dieses Merkmals werden Marker erkannt, welche bei der Aktivierung der Info-Bubble nicht getrackt werden und entsprechend werden die korrespondierenden Einrichtungsgegenstände nicht in VR übernommen.

Nachdem alle potenziellen Fehlerquellen ausgeschlossen wurden, kann die Informationsübergabe an die VR-Szene erfolgen. Die Positionen und Ausrichtungen der vom Nutzer platzierten Einrichtungsgegenstände werden in der eingangs erzeugten Klassen-Instanz gespeichert und stehen für die Nutzung in der VR-Szene bereit. Anschließend wird der AR-Modus verlassen und *Szene 1* geladen.

```
1 TransitionARtoVR.transitionARtoVRInstance.imageTarget_hotelroom_pos =
  ↳ imageTarget_hotelroom_Pos;
2 TransitionARtoVR.transitionARtoVRInstance.imageTarget_furniture01_pos =
  ↳ imageTarget_furniture01_pos;
3 ...
4 TransitionARtoVR.transitionARtoVRInstance.imageTarget_furniture10_pos =
  ↳ imageTarget_furniture10_pos;
5
6 TransitionARtoVR.transitionARtoVRInstance.
  ↳ imageTarget_hotelroom_eulerAngleY =
  ↳ imageTarget_hotelroom_eulerAngleY;
7 TransitionARtoVR.transitionARtoVRInstance.imageTarget_furniture01 =
  ↳ imageTarget_furniture01_eulerAngleY;
8 ...
9 TransitionARtoVR.transitionARtoVRInstance.imageTarget_furniture10 =
  ↳ imageTarget_furniture10_eulerAngleY;
10
11 SceneManager.LoadScene(1);
```

Zur besseren Übersicht wird der Informations-Transfer von AR zu VR in der nachfolgenden Grafik veranschaulicht (s. Abb. 49).

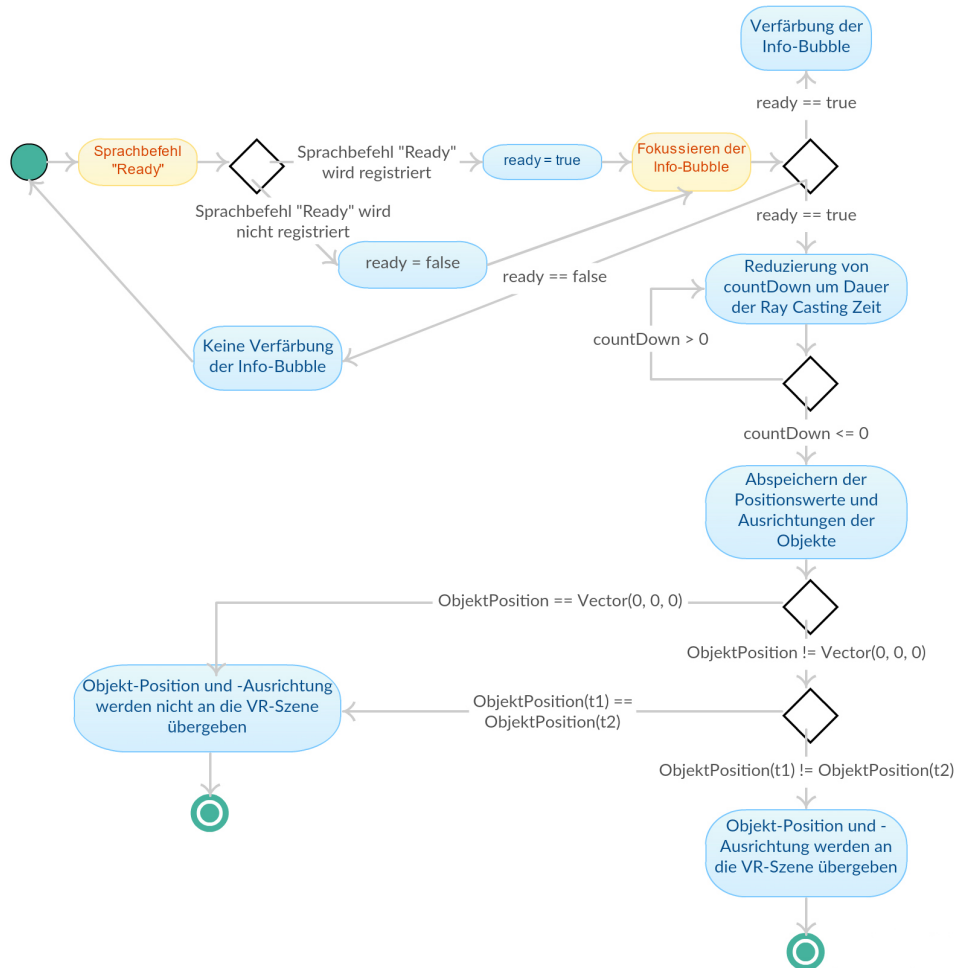


Abbildung 49: Veranschaulichung der Informationsübergabe von AR zu VR

5.4.3 Auswahl einer Virtual Reality Fortbewegungsart

Mit dem Laden von **Szene 1** wird die Webcam deaktiviert und der Nutzer befindet sich in einem vollkommen computergenerierten Bereich, in welchem er eine von ihm präferierte VR-Fortbewegungsart auswählen kann. Der Szenenaufbau ist schlicht gehalten, um sich auf das Optionsmenü, welches den Kern der Szene darstellt, konzentrieren zu können. Da der Nutzer aus dem AR-Modus einen gesenkten Blick auf die Arbeitsunterlage gewöhnt ist, wird er mit einem Text aufgefordert, geradeaus und somit das Auswahlmenü anzuschauen. Damit soll einer anfänglichen Desorientiertheit in der neuen Umgebung entgegengewirkt werden. Das UI für die Lokomotionsmethoden

besteht aus zwei nebeneinander angeordneten Info-Bubbles sowie einem Informationstext, welcher Aufschluss über die beiden selektierbaren Optionen gibt. Für eine bessere Veranschaulichung der VR-Fortbewegungsarten Free Locomotion und Walking in Place werden diese mit Hilfe der jeweiligen Aufschrift und einem Symbol auf den Auswahl-Buttons repräsentiert (s. Abb. 50).



Abbildung 50: UI zur Auswahl einer VR-Fortbewegungsart

Wie in der AR-Ansicht kann der User die Info-Bubbles und somit seine bevorzugte VR-Fortbewegungsart via Ray Casting auswählen. Dabei erhält er sowohl über den Reticle Cursor in seinem Sichtmittelpunkt sowie durch die Verfärbung der jeweils selektierten UI-Elemente Rückmeldung zu seiner Interaktion. Für die Übergabe der nötigen Informationen von `Szene 1` an den VR-Modus wird ein neues Skript erstellt, welches in seinem Aufbau der Codierung von `TransitionARtoVR` ähnelt. Dabei werden zwei öffentliche `bool`-Variablen `decisionFreeLocomotion` und `decisionWalkingInPlace` deklariert, welche den Initial-Wert `false` haben. Sobald der User eine der beiden Info-Bubbles aktiviert, wird entsprechend `decisionFreeLocomotion` oder `decisionWalkingInPlace` auf `true` gesetzt, während die andere Variable ihren Wert `false` beibehält. Die beiden Variablen werden in einer Klassen-Instanz gespeichert und stehen für weitere Verwendungen bereit. Abschließend wird `Szene 2` und somit der VR-Modus geladen.

5.5 Virtual Reality Teil

Anders als der AR-Modus fokussiert sich der VR-Part nicht auf die Interaktion mit den einzelnen Szenen-Objekten, sondern dient im Rahmen einer First-Person Experience einer realitätsnahen Besichtigung des individuell eingerichteten Hotelzimmers. Für dieses Vorhaben wird auf die übergebenen Informationen aus der AR-Szene zugegriffen und diese möglichst immersiv dargestellt. Die Schritte zur Realisierung dieses Vorhabens werden in diesem Kapitel erläutert.

5.5.1 Abrufen der Augmented Reality Informationen in Virtual Reality

Damit eine sinnhaltige und wirksame Beziehung zwischen den beiden XR-Stufen geschaffen wird, müssen die von der AR-Szene bereitgestellten Informationen korrekt aufgegriffen und im Kontext der VR verarbeitet werden. Zu diesem Zweck enthält **Szene 2** das Hotelzimmer und dieselben Einrichtungsgegenstände wie in der AR-Ansicht. Die Kernaufgabe besteht darin, die von dem Nutzer in AR vorgenommene Einrichtung identisch in den Innenraum zu übernehmen. Für die Realisierung dieses Ziels erhält jedes virtuelle Möbelstück ein eigenes Skript mit dem Namen `PositionerObjectVR_X`, wobei `X` ein Platzhalter für den jeweiligen Namen des Einrichtungsgegenstands ist. Zu Beginn jedes Skripts wird die Ausrichtung des Hotelzimmers und des jeweiligen Einrichtungsgegenstands aus der Klassen-Instanz geladen. Da sich die übergebenen Ausrichtungen der Möbelstücke in einem Bereich von -180° bis 180° befinden, werden diese Werte zunächst in das Intervall 0° bis 360° überführt. Anschließend wird für jeden Einrichtungsgegenstand die Ausrichtung in Relation zum Hotelzimmer bestimmt.

Die Positionierung der Einrichtungsgegenstände verläuft etwas komplizierter. Da sich – wie in Kapitel 5.2.1 erläutert – die 3D-Modelle des Hotelzimmers für den AR- und den VR-Modus voneinander unterscheiden, müssen die übergebenen Möbelstück-Positionen an diesen Umstand angepasst werden. Des Weiteren besitzen die einzelnen 3D-Modelle aufgrund ihrer unterschiedlichen Originalgrößen verschiedene Skalierungs-Faktoren, welche sich ebenfalls auf die Positionierung auswirken. Um diesen Fehlerquellen entgegenzuwirken, werden folgende Implementierungsschritte vorgenommen:

Ziel ist es, für jedes Mobiliar einen Vektor

`PosEinrichtungsgegenstand = (x-Position, y-Position, z-Position)`

zu bestimmen. Da die AR-Marker am Boden der Arbeitsunterlage platziert werden, befinden sich auch in VR alle Einrichtungsgegenstände auf dem Hotelzimmerboden. Aus diesem Grund gilt stets

`PosEinrichtungsgegenstand = (x-Position, 0, z-Position)`.

Als nächstes sollen die x- und z-Positionswerte ermittelt werden. Dabei werden die unterschiedlichen Skalierungsfaktoren der 3D-Modelle mit Hilfe zweier Variablen `xShift1` und `zShift1` ausgeglichen. Da das Modell des Hotelzimmers in VR, anders als zu seinem Gegenstück in AR, über ein Badezimmer verfügt, muss eine Korrektur bei der Translation in x- und z-Richtung erfolgen. Dazu werden die Variablen `xShift2` und `zShift2` deklariert. Die Bestimmung der vier Variablen-Werte erfolgt durch ein investigatives Verfahren, bei welchem die Positionen der Marker in den Ecken des AR-Hotelzimmers auf die korrespondierenden Punkte in der VR-Darstellung gemappt werden. Mit Hilfe der folgenden Formel werden die x- und z-Positionswerte der Einrichtungsgegenstände berechnet.

5. IMPLEMENTATION DER ANWENDUNG

```
1 PosHotelroom =  
2 (  
3 TransitionARtoVR.transitionARtoVRInstance.ImageTarget_Hotelroom_Pos.x,  
4 0,  
5 TransitionARtoVR.transitionARtoVRInstance.ImageTarget_Hotelroom_Pos.z  
6 )  
7  
8 PosFurniture =  
9 (  
10 xShift1 * (-(PosHotelroomX - xShift2 + (PosFurnitureX - PosHotelroomX))  
11 ↪ ,  
12 0,  
13 zShift1 * (-(PosHotelroomZ - zShift2 + (PosFurnitureZ - PosHotelroomZ))  
14 )
```

Dieses Vorgehen wird für alle in der AR-Szene platzierten Einrichtungsgegenstände vorgenommen. Nach Durchführung dieser Implementierungsschritte werden sowohl die Ausrichtungen als auch Positionen der einzelnen Möbelstücke korrekt in die VR übertragen. Die erfolgreiche Informationsübergabe soll anhand der Abbildung 51 demonstriert werden.

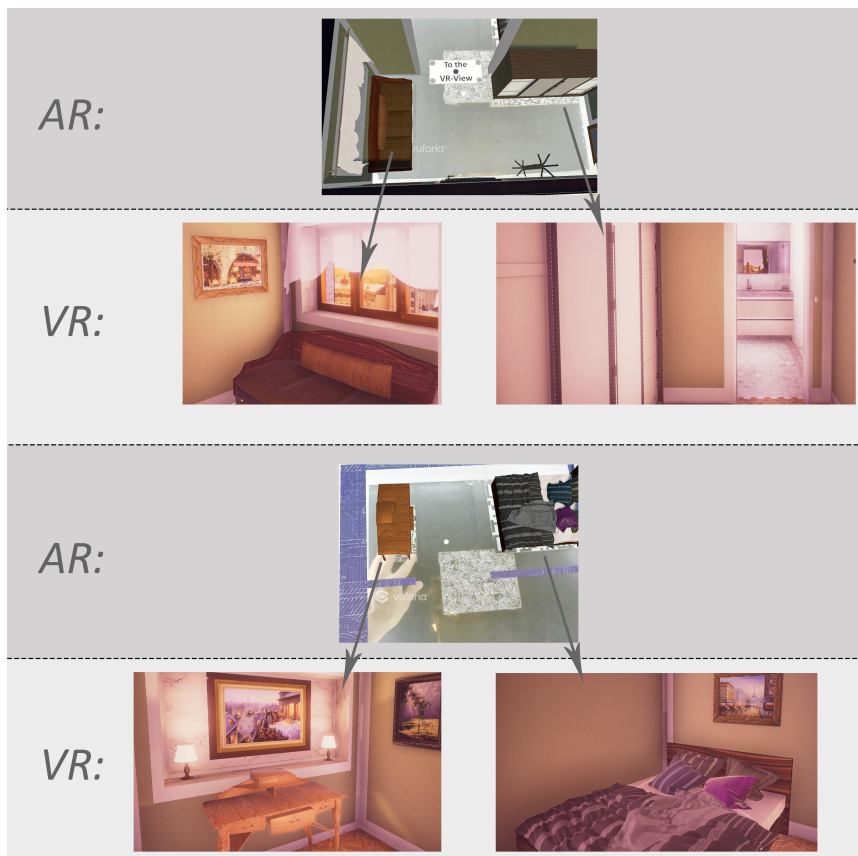


Abbildung 51: Demonstration der Informationsübergabe von AR zu VR

5.5.2 Fortbewegung in Virtual Reality

Nachdem die Übergabe der Informationen von AR zu VR implementiert wurde, soll der Nutzer das virtuelle Hotelzimmer mit seiner zuvor ausgewählten VR-Fortbewegungsart begehen und sich einen realitätsnahen Eindruck seiner Einrichtung machen können. Dafür muss zunächst die nötige Basis geschaffen werden. So wird ein neues `GameObject` mit dem Namen `VRUser` erstellt, welches eine Szenen-Kamera als Kind besitzt und die Sicht des Nutzers repräsentiert. `VRUser` wird vor die Eingangstür des Innenraums platziert und die Kamera so eingestellt, dass sie sich auf einer für den Menschen natürlichen Höhe befindet und das eingefangene Bild stereoskopisch im VR-HMD wiedergibt. Damit die Lokomotion in der VR physikalisch ermöglicht wird und sich der Anwender nicht durch Szenen-Objekte hindurchbewegen kann, erhalten alle 3D-Modelle sowie der `VRUser` Collider. Unter einem Collider versteht man ein unsichtbares Polygonnetz, welches verschiedenen Formen, wie z.B. Quader, Kugeln, oder Zylinder annehmen kann. Wird ein Szenen-Objekt von einem – seiner geometrischen Form entsprechend – geeigneten Collider umgeben, so kann es physikalisch auf andere Kollisionsobjekte reagieren (Unity Technologies, 2018a). Mit dieser Grundlage können die beiden Fortbewegungsarten Free Locomotion und Walking in Place implementiert werden.

Damit sich der Anwender mittels eines Hand-Controllers durch die virtuelle Szene bewegen kann, müssen zunächst die einzelnen Steuerelemente mit Variablen versehen werden. Dafür werden in den Unity-Einstellungen Eingabe-Belegungen für die horizontalen und vertikalen Achsen des linken und rechten Controller-Sticks vorgenommen und anschließend ein neues Skript mit dem Namen `VRLocomotion_FreeLocomotion` als Komponente von `VRUser` erstellt. Um die Free Locomotion Methode nutzen zu können, wird der Wert der übergebenen Variable `decisionFreeLocomotion` abgefragt. Hat sich der Anwender in `Szene 1` für die Steuerung mit Hand-Controller entschieden, so ist `decisionFreeLocomotion true` und der Lokomotions-Typ damit aktiviert. Mit Hilfe der Eingabe-Belegungen kann stets die aktuelle Richtung eines Analog-Sticks abgefragt werden. So wird implementiert, dass sich `VRUser` nach vorne oder hinten in der virtuellen Umgebung bewegt, wenn die vertikale Ausrichtung des linken Analog-Sticks größer oder kleiner null ist. Wird ein positiver oder negativer Wert für die horizontale Achse des linken Sticks registriert, so erfolgt eine Fortbewegung nach links bzw. rechts. Für die Fortbewegungsgeschwindigkeit wird ein Wert gewählt, welcher ein für den Menschen normales Schritt-Tempo darstellt und man somit präzise und kontrolliert durch das virtuelle Hotelzimmer manövrieren kann.

5. IMPLEMENTATION DER ANWENDUNG

Als Zusatz zu dem 360° Tracking, welches der Oculus Rift inhärent ist, hat der Nutzer die Möglichkeit, mit Hilfe des rechten Controller-Sticks die Szenen-Kamera auf der Horizontalen nach Belieben auszurichten (s. Abb. 52).



Abbildung 52: Controller-Eingabe für Free Locomotion

Im Gegensatz zur Free Locomotion fällt die Implementierung der Walking in Place Fortbewegungsart etwas komplizierter aus. Das GameObject `VRUser` erhält ein weiteres Skript mit dem Namen `VRLocomotion_WalkingInPlace`, welches in den folgenden Schritten implementiert wird. Wie bei der Lokomotion mit Controller kann der Nutzer die Walking in Place Methode nur verwenden, wenn er sich für diese Lokomotionsart in der vorhergehenden Szene entschieden hat. Da die Veränderung der Nutzerhöhe beim auf der Stelle Laufen eine entscheidende Rolle für diese VR-Fortbewegungsart spielt, wird zu Beginn die Höhe des VR-Headsets in einer Variable `heightCurrent` gespeichert sowie weitere Datenspeicher mit den Namen `heightPrevious`, `heightDelta` und `heightDeltaMax` deklariert. Dabei entspricht `heightDelta` der Differenz zwischen der aktuell gemessenen Nutzerhöhe `heightCurrent` und dem zuvor festgehaltenen Wert `heightPrevious`. Wie in Kapitel 2.1.3 erläutert, verändert sich die Höhe eines Menschen beim Laufen bis zu 1,3cm. Dieser Wert wird in der Variable `heightDeltaMax` festgehalten. Des Weiteren werden auf Basis persönlicher Tests Werte für die minimale und maximale Fortbewegungsgeschwindigkeit sowie Schrittdauer bestimmt. Die Schrittdauer ergibt sich aus der Division von `heightDelta` und der dabei verstrichenen Zeit. Ein Schritt wird dann vom System registriert, wenn `heightDelta < heightDeltaMax` ist und sich die Schrittdauer innerhalb des zuvor festgelegten Intervalls befindet. Wird ein Schritt erkannt, so wird anhand der in Kapitel 2.1.3 erläuterten Formel die Fortbewegungsgeschwindigkeit in der VR ermittelt (s. Abb. 53). Dabei bewegt sich der Nutzer in Blickrichtung fort und kann sein Tempo durch die Frequenz seiner Schritte regeln. Registriert das Skript keinen Schritt, so hat die

Fortbewegungsgeschwindigkeit den Wert null.

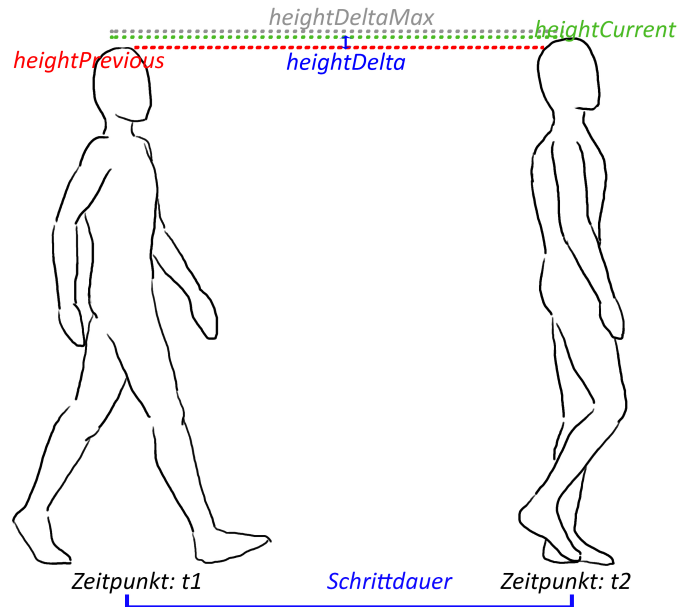


Abbildung 53: Komponenten von Walking in Place

5.5.3 Visuelle und akustische Darstellung der Szene

Neben der Fortbewegung soll die visuelle Darstellung und akustische Untermalung der VR-Szene die Immersion erhöhen und eine für den Nutzer angenehme Atmosphäre vermitteln. Der Schauplatz des Hotelzimmers orientiert sich dabei an eine französische Altstadt an einem Sommerabend. Um dies zu erreichen, wird eine Skybox, welche einen roten Abendhimmel darstellt, in die Szene integriert.

Für eine möglichst detaillierte Beleuchtung werden verschiedene Lichtquellen in der virtuellen Umgebung platziert. So befindet sich außerhalb des Hotelzimmers ein Directional Light, welches mit einem gelb-orangen Farbton durch die Fensterfassade scheint und das Sonnenlicht repräsentiert. Des Weiteren wird ein Lens Flare hinter der Fensterscheibe platziert, welcher den realistischen Eindruck des Sonnenlichts verstärken soll. Die einzelnen Decken- und Stehlampen werden durch Point Lights dargestellt, welche im Hauptraum des Zimmers einen warmen und im Badezimmer einen eher kalten Farbton haben. Alle Lichtquellen bis auf das Point Light im Badezimmer werden auf Baked Lighting gesetzt. Somit werden vorberechnete Lightmaps generiert, welche im Zusammenhang mit den eingangs erstellten Materialien in einer detaillierten Beleuchtung und Verschattung der Objektflächen resultieren. Mit dem Ziel, eine flackernde Glühbirne zu simulieren, wird für

5. IMPLEMENTATION DER ANWENDUNG

das Point Light im Badezimmer die Beleuchtungstechnik Realtime gewählt und ein neues Skript für die Lichtquelle erstellt. Bei der Codierung wird die Lichtfarbe der Deckenlampe kontinuierlich mit einem zufallsgenerierten Wert multipliziert und in Echtzeit ausgegeben. Mit diesem kreierten Flimmer-Effekt und den anderen Lichtquellen soll die Immersion in der VR gesteigert werden.

Der visuelle Eindruck der Szene wird mit Hilfe des frei zugänglichen Post Processing Stack PlugIns für Unity optimiert. So werden mit Hilfe dieser Erweiterung folgende Schritte vorgenommen: Mit Hilfe von Antialiasing wird eine höhere Kantenglättung der Objekte erzeugt und die Szenen-Modelle mittels Ambient Occlusion möglichst realistisch verschattet dargestellt. Des Weiteren wird durch Blooming ein dezent überstrahlter und weicher visueller Effekt der Umgebung generiert. Ein wichtiges Element des Post Processing Stack PlugIns stellt das Color Grading dar, mit welchem die optische Darstellung der virtuellen Umgebung durch verschiedene Farbtöne verändert werden kann. Für die Applikation werden die Einstellungen beim Color Grading so gewählt, dass die Abendstimmung im Hotelzimmer unterstrichen wird. Die Resultate der Szenenbeleuchtung und des Post Processings sind in Abbildung 54 veranschaulicht.

Vergleich:



Abbildung 54: Szenen-Beleuchtung und Post Processing

5. IMPLEMENTATION DER ANWENDUNG

Mit Hilfe des Post Processing Stack PlugIns lässt sich nicht nur die visuelle Darstellung der Szene optimieren, sondern das Risiko für Motion Sickness reduzieren. Nach einer Untersuchung der Columbia University kann die Gefahr für Unwohlsein in der virtuellen Welt durch die Einschränkung des Sichtbereichs eingedämmt werden. Der Grund dafür ist, dass der Nutzer durch ein verringertes FOV nicht mit visuellen Informationen überreizt wird und sich besser auf bestimmte Umgebungsdetails fokussieren kann (Irving, Michael, 2016). Aus diesem Grund wird mit Hilfe des PlugIns eine Vignette erzeugt, welche die Risiken für Motion Sickness reduzieren jedoch nicht vom Nutzer als störend empfunden werden soll (s. Abb. 55).



Abbildung 55: Sicht-Vignette

Um den Detailgrad in der virtuellen Szene weiterhin zu erhöhen, werden Partikelsysteme in dem Hotelzimmer platziert. Mit Hilfe von Partikelsystemen können flexible, nicht-greifbare Objekte, wie beispielsweise Regen oder Rauch kreiert werden (Unity Technologies, 2018a). So werden in dem virtuellen Innenraum drei Partikelsysteme gleichmäßig verteilt und unter anderem Einstellungen an den Partikel-Größen, -Geschwindigkeiten, -Farben und -Formen vorgenommen. Mit diesen Implementierungsschritten sollen Staubpartikel in dem virtuellen Hotelzimmer dargestellt und die Immersion des Nutzers erhöht werden.

An den Wänden des Zimmers befinden sich mehrere Bilderrahmen mit Gemälden von Paris. Mit dem Ziel, die Atmosphäre einer französischen Innenstadt zu unterstreichen, werden mehrere Audio-Quellen mit unterschiedlichen Lautstärken in der virtuellen Szene platziert. So dringen durch das Fensters französische Straßenmusik und die Stimmen von Passanten. Befindet sich der Nutzer vor der Eingangstür, so vernimmt er dumpfe Gespräche von anderen Hotelgästen. Im Badezimmer wird das Flimmern der Deckenlampe durch das Geräusch von Glühbirnen-Surren untermalt. Des Weiteren wird mit Hilfe eines Skripts der Ton von Kirchenglocken in einem Zeitintervall von 70 Sekunden wiederholt.

5.6 Übergang: Virtual Reality zu Augmented Reality

Nachdem sich der Anwender einen realitätsnahen und immersiven Eindruck seines eingerichteten Innenraums gemacht hat, soll ihm die Möglichkeit geboten werden, wieder in den AR-Modus zurückzukehren und Änderungen am Mobiliar vorzunehmen zu können. In diesem Kapitel wird das dafür implementierte UI und der Szenenwechsel erläutert.

5.6.1 User Interface

Bei der Gestaltung des UIs wird einerseits das Ziel verfolgt, den Nutzer über die Steuerung des Wechsels zum AR-Modus zu informieren, andererseits die Immersion nicht durch Anzeige-Elemente zu stören. Aus diesem Grund fällt die Wahl auf ein Game Element UI, welches sich in die virtuelle Umgebung einpasst. Dabei wird das Modell eines Türschilds in die VR-Szene integriert und mit einer Textur erweitert. Die Textur zeigt ein Stück Papier, auf welchem die Steuerung mittels Controller- und Sprach-Eingabe dargestellt wird (s. Abb. 56).

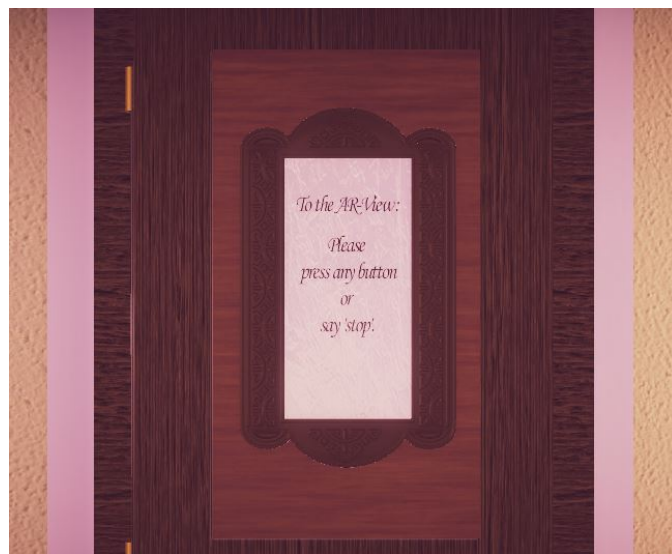


Abbildung 56: Game Element UI für den Wechsel VR zu AR

5.6.2 Wechsel von Virtual zu Augmented Reality

Wie im vorhergehenden Abschnitt skizziert, kann der Nutzer per Tastendruck auf dem Controller oder mit Hilfe eines Sprachbefehls in die AR-Ansicht zurückwechseln. Ähnlich wie bei der Implementierung von Free Locomotion werden für den Szenenübergang neue Eingabe-Belegungen in den Unity-Einstellungen für alle Controller-Buttons vorgenommen. Mit Hilfe eines neuen Skripts, welche als Komponente von `VRUser` erstellt wird, wird

5. IMPLEMENTATION DER ANWENDUNG

die Betätigung einer Taste erkannt und somit der AR-Modus geladen. Des Weiteren kann der Nutzer auch über den Sprachbefehl „Stop“ die VR-Szene verlassen. Grund für die Implementierung zweier Interaktionsarten ist, dass der Nutzer, wenn er sich für die Walking in Place Fortbewegungsart entschieden hat, wahrscheinlich keinen Controller in der Hand führt. Des Weiteren soll mit dem multimodularen UI auf die unterschiedlichen Präferenzen der Anwender eingegangen und somit die User Experience gefördert werden. Mit der Codierung des Übergangs von VR zu AR ist die Anwendung bereit, um in Form einer Evaluation getestet werden.

6 Evaluation und Optimierung der Anwendung

Die erstellte Anwendung soll im Rahmen einer Evaluation genauer untersucht werden. Dabei fokussiert sich dieses Kapitel auf die Vorbereitung und Durchführung der Probandentests, die dabei gewonnenen Ergebnisse und die darauf basierende Optimierung der Anwendung.

6.1 Zielsetzung und Planung der Evaluation

Mit der Evaluation der Applikation werden mehrere unterschiedliche Ziele verfolgt. So besteht die Kernaufgabe darin, zu untersuchen, wie gut sich Augmented und Virtual Reality miteinander verbinden lassen und wie sinnvoll das Zusammenspiel der beiden Darstellungsformen im Rahmen einer Innenarchitektur-Anwendung ist. In diesem Kontext soll die Eignung von Augmented Reality bei der Einrichtung eines Zimmers und die realitätsnahe Besichtigung dieses Raums mit Virtual Reality getestet werden. Indem die Resonanz der Probanden zu einzelnen Funktionen und Elementen der Applikation eingeholt wird, sollen potenzielle Schwachstellen herausgefunden, Optimierungsstrategien erarbeitet und diese in die Realität umgesetzt werden.

Um sich ein klares Bild über die einzelnen Komponenten der Applikation machen zu können, wird ein Evaluationsbogen mit insgesamt 37 Fragen zu verschiedenen Bereichen erstellt. So wird der AR-Modus auf seine Bedienung und UI, die Qualität des Trackings, die Funktionalität der Assistenzfunktionen sowie die Wiedergabe des Kamerabildes im VR-Headset untersucht. Um potenzielle Schwachstellen bei dem Übergang von AR zu VR herauszustellen, wird die Performanz beim Szenenwechsel, das dafür verwendete UI und die Genauigkeit der Informationsübergabe getestet. Im Bereich des VR-Modus werden Fragen zu den einzelnen Fortbewegungsarten sowie der Immersion in der virtuellen Umgebung gestellt und anschließend der Übergang von VR zu AR hinsichtlich Performanz und UI thematisiert. Um den Grundgedanken der Arbeit zu hinterfragen, wird auf die Verbindung von AR und VR im Kontext ihrer Funktionsbereiche in der Anwendung eingegangen. Für die Messung der einzelnen Themen wird eine Likert-Skala gewählt. Dabei stehen pro Frage fünf Antwortoptionen zur Verfügung, welche von absoluter Ablehnung bis zu vollkommener Zustimmung reichen. Mit diesem Vorgehen soll die Einstellung der Testperson zu den thematisierten Faktoren möglichst exakt eingefangen werden (Statista, 2018). Der für die Probandentests erstellte Evaluationsbogen ist im Anhang A.1 beigefügt.

6.2 Durchführung der Evaluation

Insgesamt nahmen an der Evaluation acht Testpersonen, wobei es sich um drei Frauen und fünf Männer im Alter von 22 bis 25 Jahren handelte, in zwei Versuchsreihen teil. Der erste Feldversuch wurde mit sechs Probanden in einem Zeitraum von drei Tagen durchgeführt. Auf Basis der dabei gewonnenen Ergebnisse konnten Fehlerquellen ausgemacht und erste Optimierungsschritte an der Anwendung eingeleitet werden. Mit dem Ziel, die Eignung der vorgenommenen Implementierungen zu überprüfen, folgte eine weitere Untersuchung mit zwei Testpersonen an zwei unterschiedlichen Tagen. Nachdem alle Versuche ausgewertet wurden, wurde die Applikation entsprechend der Meinungen und Ratschläge der Probanden überarbeitet.

Der Ablauf der einzelnen Tests war dabei wie folgt:

Zu Beginn jedes Versuchs wurde die Testperson mit Hilfe eines kurzen Demonstrationsvideos der Anwendung und einer schriftlichen Anleitung in die Thematik eingeführt. Anschließend wurde die Applikation gestartet. Um die Testpersonen nicht in ihrer Nutzererfahrung zu beeinträchtigen, wurde diesen ausreichend Zeit gegeben, sich mit dem hybriden AR-/VR-HMD und den einzelnen Funktionen der Anwendung vertraut zu machen sowie das Hotelzimmer individuell einzurichten. Die Nutzungsdauer der Applikation variierte zwischen 20 und 35 Minuten. Im Anschluss füllten die Probanden den Evaluationsbogen aus und äußerten teils Anmerkungen und Verbesserungsvorschläge. Dabei konnten verschiedene Ergebnisse und Erkenntnisse gewonnen werden, welche im nächsten Abschnitt genau behandelt werden.

6.3 Auswertung der Testergebnisse und Optimierung der Anwendung

Im Rahmen des AR-Modus konnten folgende Meinungen eingeholt und Messwerte festgehalten werden: Die Interaktion mit dem Tangible UI konnte von den Testern intuitiv und präzise durchgeführt werden und wurde daher im Fragebogen durchschnittlich mit der höchsten Akzeptanz-Stufe beurteilt. Die Anzeige der realen Umgebung im hybriden AR-/VR-Headset stieß hingegen auf gemischte Resonanzen. So konnten sich manche Probanden schnell mit der AR-Ansicht im HMD zurechtfinden, während andere zunächst etwas Eingewöhnungszeit benötigten. Durchschnittlich erzielten Fragen zur AR-Ansicht in der VR-Brille gute bis teils mittelmäßige Bewertungen. Die Qualität des AR-Trackings erhielt sehr gute bis gute Beurteilungen und war bis auf vereinzelte kurze Aussetzer bei größerer Entfernung zum Marker kontinuierlich stabil. Die einzelnen Assistenzfunktionen wurden von den Testern durchschnittlich als gute bis sehr gute Erweiterung angesehen. Dabei wurde im Verhältnis die Grundriss-Ansicht am meisten und die Positionierungsvorschläge am wenigsten genutzt, da die Nutzer eine individuelle Einrichtung bevorzugten. Während der Durchführung des ersten Feldversuchs konnten einige Sprachbefehle erst nach mehrmaligem Wiederholen vom System registriert werden. So kam es vor allem bei mehrsilbigen Wörtern sowie bei „th“-, „s“- und „z“-Lauten aufgrund ihrer phonetischen Ähnlichkeit zu Problemen. Aus diesem Grund wurde die Sprachsteuerung entsprechend der Tabelle 5 überarbeitet.

Funktion	Ursprünglicher Sprachbefehl	Neuer Sprachbefehl
Anschalten der Positionierungs-Vorschläge in AR	„Recommendations On“	„Plan On“
Ausschalten der Positionierungs-Vorschläge in AR	„Recommendations Off“	„Plan Off“
2D-Ansicht des Hotelzimmers in AR	„Two D“	„Layout On“
2D-Ansicht aller Objekte in AR		„Layout All“
3D-Ansicht aller Objekte in AR	„Three D“	„Layout Off“
Anschalten der Maß-Angaben der Objekte in AR	„Sizes On“	„Metrics On“
Ausschalten der Maß-Angaben der Objekte in AR	„Sizes Off“	„Metrics Off“

Tabelle 5: Ursprüngliche und überarbeitete Sprachbefehle

Mit den vorgenommenen Implementierungsschritten konnte im Rahmen der zweiten Testreihe eine durchweg sehr gute Bedienbarkeit erzielt werden. Die Resonanzen der Nutzer auf die ursprüngliche und überarbeitete Sprachsteuerung sind in den folgenden Diagrammen festgehalten (s. Abb. 57).

6. EVALUATION UND OPTIMIERUNG DER ANWENDUNG

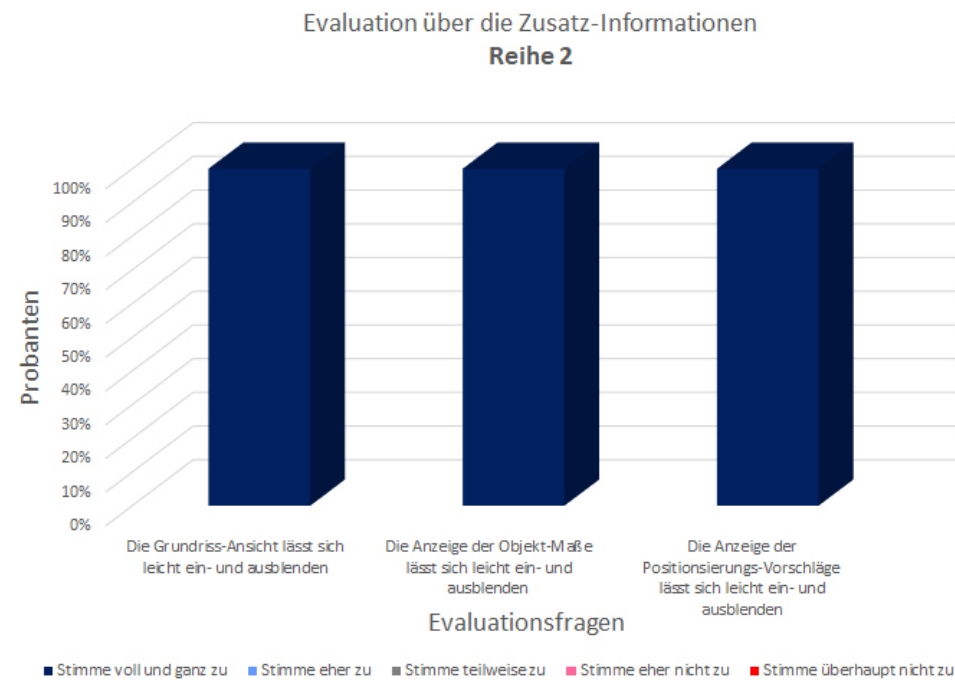
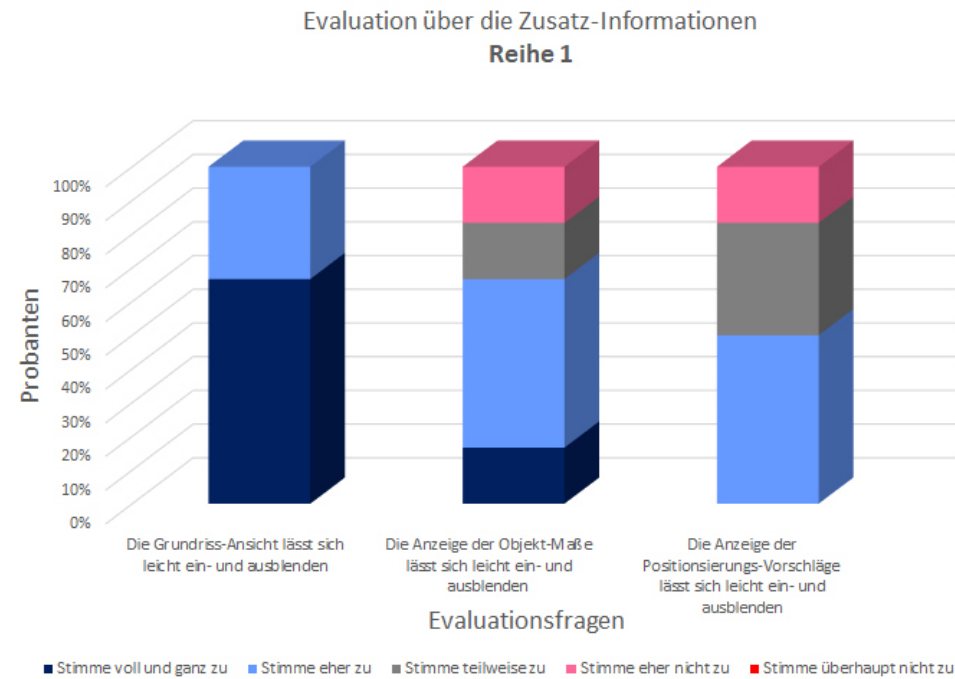


Abbildung 57: Gegenüberstellung der Evaluationsergebnisse der Testreihen 1 und 2: Sprachsteuerung der AR-Assistenzfunktion

6. EVALUATION UND OPTIMIERUNG DER ANWENDUNG

Des Weiteren merkten drei Probanden der ersten Versuchsreihe an, dass sie sich neben der Grundriss-Ansicht eine weitere Funktion wünschen würden, in welcher das Hotelzimmer in 2D und die einzelnen Einrichtungsgegenstände in 3D dargestellt werden, sodass Objekte durch die Hotelzimmerwände nicht mehr verdeckt werden, der 3D-Eindruck der Modelle jedoch zugleich nicht verloren geht. Diesem Anliegen wurde nachgegangen und eine entsprechende Assistenzfunktion implementiert (s. Abb. 58).

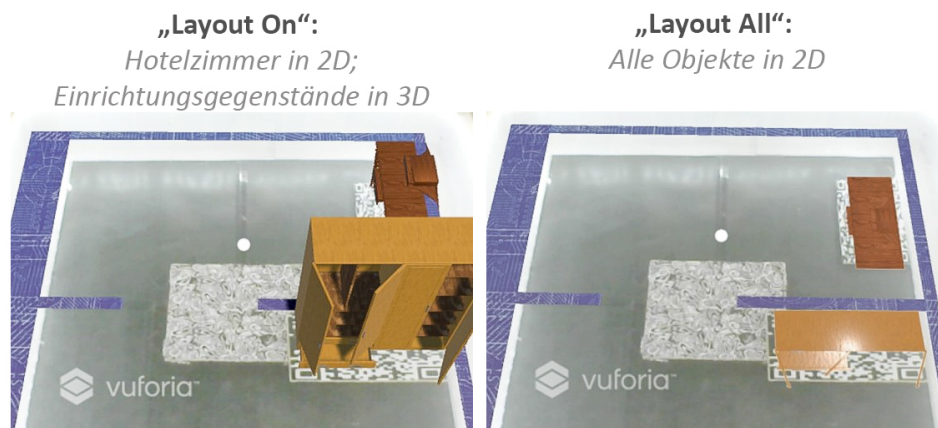


Abbildung 58: Grundriss- und komplette 2D-Ansicht

Der nächste Teil des Fragebogens thematisierte den Übergang von AR zu VR. In diesem Kontext wurde die Ladezeit zwischen **Szene 1** und **2** von den Testern der ersten Versuchsreihe durchschnittlich nur als teilweise akzeptabel befunden. Mit dem Ziel, die Performanz der Applikation zu erhöhen, wurde die Qualität von weniger dominanten Audioquellen in der VR-Szene verringert und die Sound-Dateien so eingestellt, dass diese erst während des Szenenablaufs dekomprimiert werden. Mit diesen Implementierungsschritten konnte die Ladezeit von anfänglich elf auf fünf Sekunden reduziert werden. Die Optimierung der Anwendung wirkte sich positiv auf das Feedback der Tester aus, wie anhand der folgenden Grafik zu sehen ist (s. Abb. 59).

6. EVALUATION UND OPTIMIERUNG DER ANWENDUNG

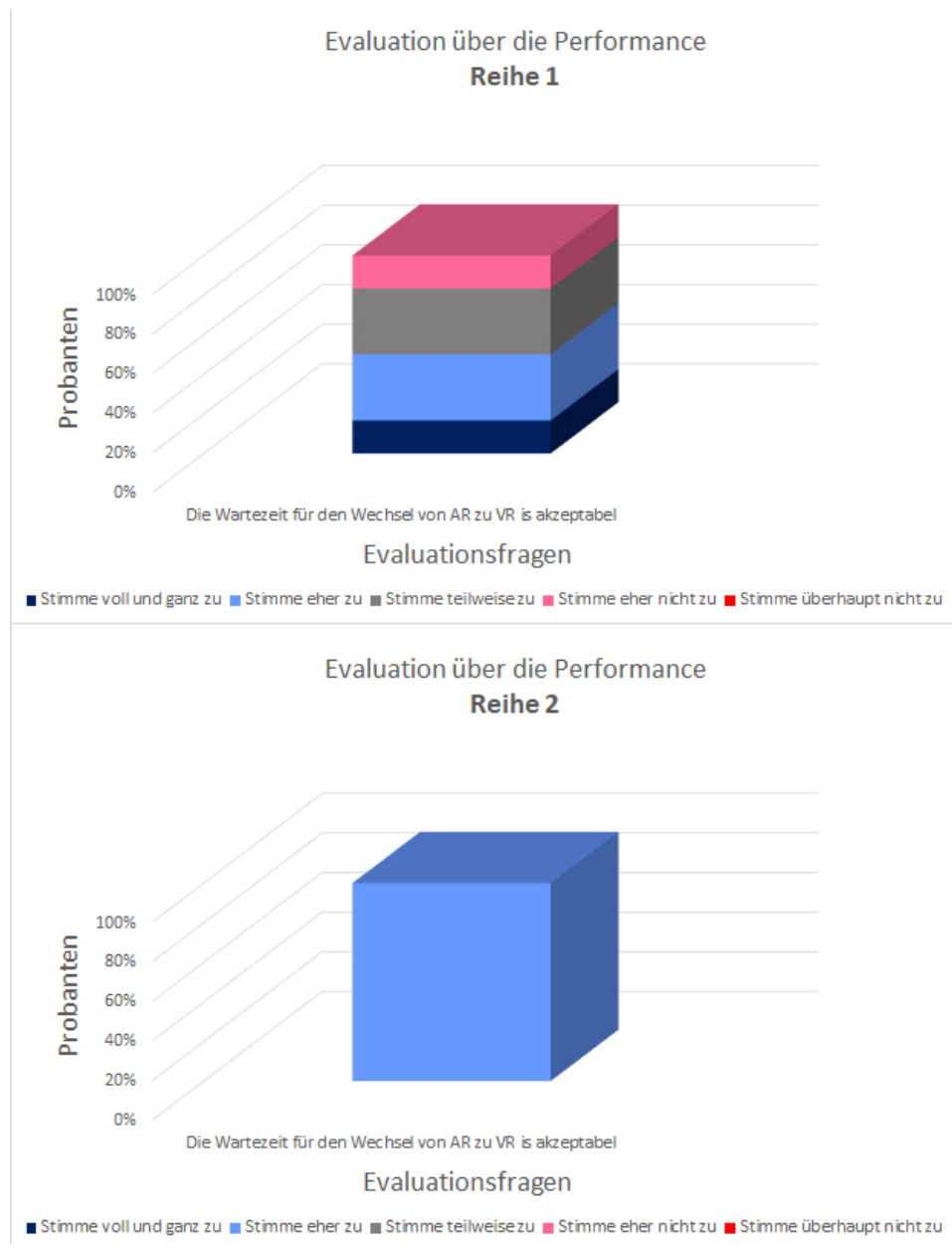


Abbildung 59: Gegenüberstellung der Evaluationsergebnisse der Testreihen 1 und 2: Ladezeit für den Wechsel AR zu VR

6. EVALUATION UND OPTIMIERUNG DER ANWENDUNG

Die Fragen zur Bedienbarkeit des Szenenwechsels wurden durchschnittlich mit der höchsten bis zweit-höchsten Akzeptanzstufe beurteilt. So wurde das UI für den Wechsel von AR zu VR und bei der Wahl der VR-Fortbewegungsart durchweg als sehr positiv und anwenderfreundlich wahrgenommen. Da der ausgedruckte Informationstext zu der Handhabung der Anwendung durch die Kamera-Ansicht nicht optimal lesbar war, wurde zusätzlich eine digitale Version erstellt, welche sich mit Hilfe eines Markers anzeigen lässt und somit sehr gut lesbar ist (s. Abb. 60).

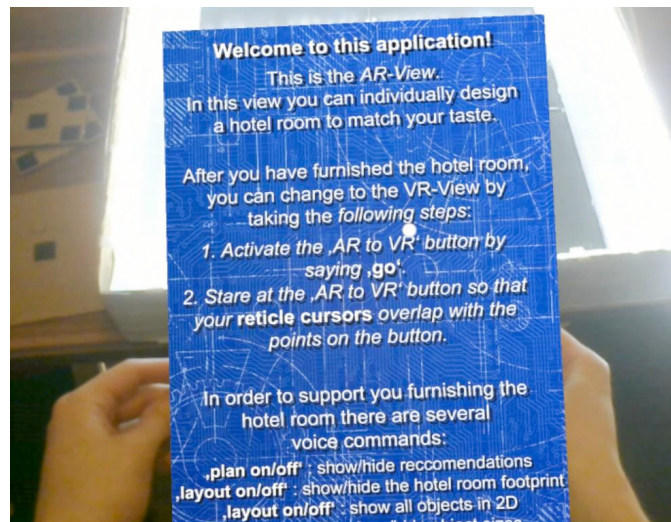


Abbildung 60: Digitale Anleitung in AR

Die Übergabe der Ausrichtungen der Einrichtungsgegenstände von AR zu VR verlief durchweg korrekt. Jedoch kam es bei einem Möbelstück vor, dass dessen Position in AR und VR nicht ganz übereinstimmte. Dieser Fehler wurde behoben, sodass sowohl die Ausrichtungen als auch Positionswerte aller Einrichtungsgegenstände korrekt von AR nach VR transferiert werden. Die Optimierung der Anwendung zeigte sich in den Nutzertests, welche in den folgenden Diagrammen veranschaulicht werden (s. Abb. 61).

6. EVALUATION UND OPTIMIERUNG DER ANWENDUNG

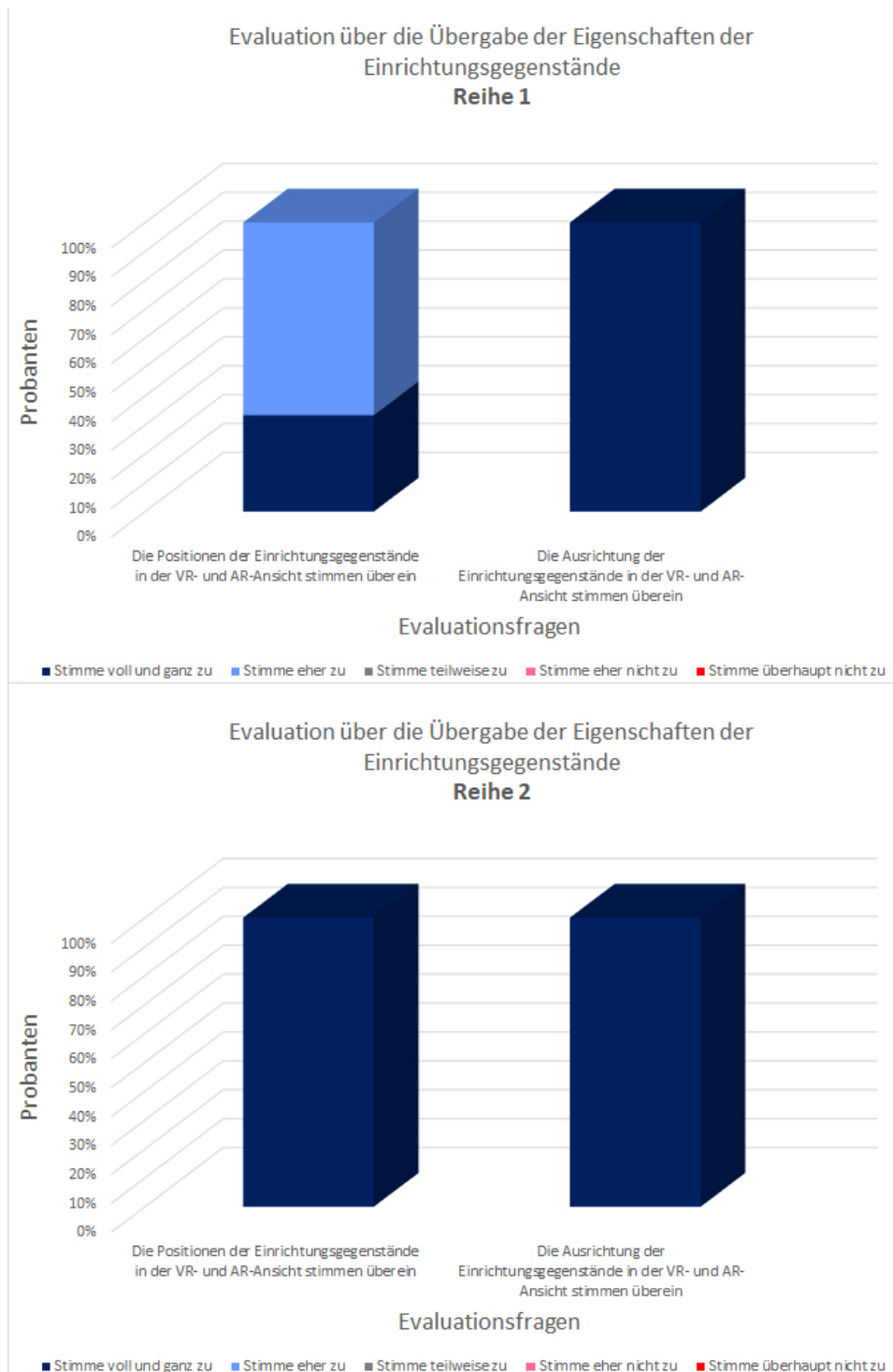


Abbildung 61: Gegenüberstellung der Evaluationsergebnisse der Testreihen 1 und 2: Übereinstimmung der Einrichtung in AR und VR

Bei der Auswertung der Fragen zum VR-Modus stellte sich heraus, dass die Free Locomotion Fortbewegungsart am liebsten genutzt wurde. So wurde die Steuerung mit dem Controller als präziser, eingänglicher und angenehmer empfunden als die Alternative mit Walking in Place. Im Gegensatz dazu wurde Walking in Place im Bereich Motion Sickness besser bewertet. So gaben die Tester des ersten Feldversuchs an, dass sie mit dieser Fortbewegungsart weniger Anzeichen von Unwohlsein verspürten als mit Free Locomotion. Durchschnittlich wurden beide Lokomotions-Methoden von den Testern der ersten Versuchsreihe mit der zweit-höchsten Zufriedenheitsstufe valuiert, wobei Walking in Place etwas besser abschnitt. Um die Anzeichen von Motion Sickness weiter zu minimieren, wurde die Vignette in der Nutzer-Ansicht vergrößert. Mit dieser Vorkehrung konnte bei beiden Fortbewegungsarten im Rahmen des zweiten Feldversuchs eine sehr positive Resonanz erzielt werden (s. Abb. 62).

6. EVALUATION UND OPTIMIERUNG DER ANWENDUNG

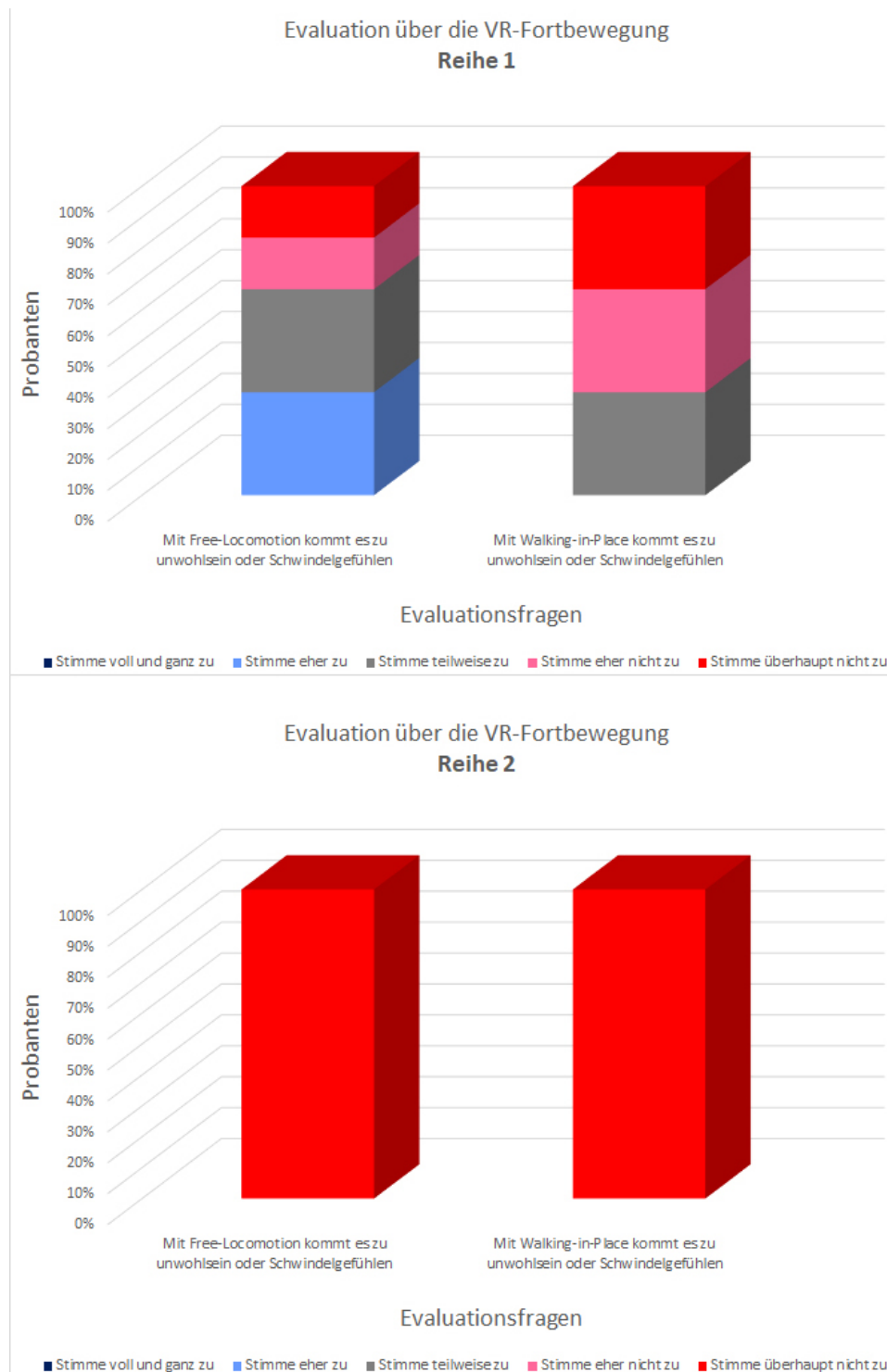


Abbildung 62: Gegenüberstellung der Evaluationsergebnisse der Testreihen 1 und 2: Motion Sickness bei den VR-Fortbewegungsarten

6. EVALUATION UND OPTIMIERUNG DER ANWENDUNG

Bei Fragen zu der Immersion in der VR-Szene wurden sowohl bei der visuellen als auch bei der akustischen Darstellung des Hotelzimmers durchweg sehr positive Bewertungen erzielt.

Der vorletzte Teil des Evaluationsbogens behandelte den Übergang von VR zu AR. Dabei wurde die Ladezeit von allen Probanden mit der höchsten Akzeptanz-Stufe beurteilt. Komplikationen gab es hingegen bei der Bedienung. So wurde das Game Element UI, welches anzeigt, wie man vom VR-zum AR-Modus zurückgelangt, von den Testern teilweise nicht sofort gefunden. Aus diesem Grund wurde ein Tool-Tipp implementiert, welcher dem Nutzer zu Beginn der VR-Szene in seinem Sichtfeld angezeigt wird und über die Interaktion informiert (s. Abb. 63). Mit diesem Vorgehen ist klar ersichtlich, wie die Steuerung des Szenenwechsels funktioniert.

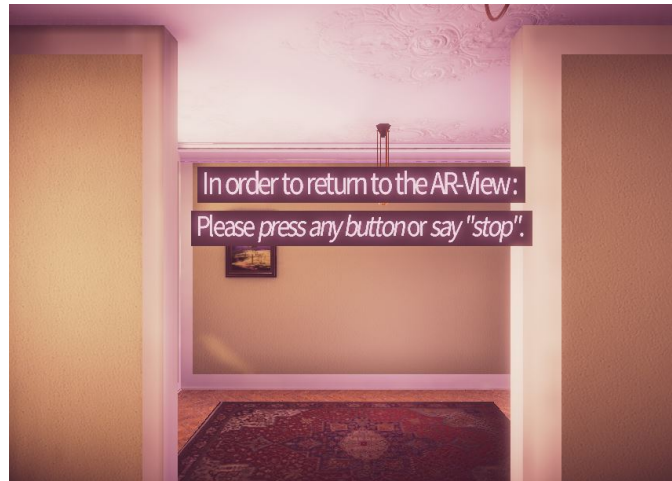


Abbildung 63: Tool-Tip in VR

Sowohl alle Probanden der ersten als auch der zweiten Testreihe gaben an, dass die Applikation sehr gut und sinnvoll das Zusammenspiel von AR und VR vereint. Bei der Frage, inwiefern die Einrichtung eines Innenraums mittels AR sinnvoll ist, stimmten zwei Personen eher und die restlich sechs Tester voll und ganz zu. Die realitätsnahe Besichtigung des Hotelzimmers mit VR wurde durchweg mit der höchsten Akzeptanzstufe bewertet.

7 Ausblick und Fazit

Mit dem Ziel die Symbiose von Augmented und Virtual Reality zu untersuchen, wurde in dieser Abhandlung ein hybrides AR-/VR-System im Bereich Innenarchitektur entwickelt, evaluiert und optimiert. In diesem finalen Kapitel wird ein kurzer Ausblick auf weitere Implementierungsmöglichkeiten für die Anwendung vorgestellt und die in der Arbeit erworbenen Kenntnisse und Schlussfolgerungen zusammengefasst.

7.1 Ausblick

Auf Basis der Evaluationsergebnisse wurden offene Schwachstellen der Applikation herausgestellt und weitestgehend optimiert sowie auf die Anregungen und Meinungen der Probanden eingegangen. Dieser Abschnitt soll weitere Verbesserungsmöglichkeiten und Ideen beleuchten, welche die Verbindung von AR und VR darüber hinaus stärken können.

Neben der bereits durchgeführten Evaluation, welche aufgrund ihrer geringen Teilnehmerzahl keine repräsentative Studie darstellt, können weitere Probandentests vorgenommen werden. Dabei sollte die Intention verfolgt werden, neue Erkenntnisse über die Eignung bereits realisierter Funktionen, die Wünsche und Anforderungen der Nutzer an die Anwendung sowie potenzielle neue Ideen und Implementierungsmöglichkeiten zu erfassen. Da die Zielgruppe der Applikation Innenarchitekten und Raumgestalter darstellt, sollten primär die Meinungen und Ratschläge dieser Berufsgruppe eingeholt werden. Durch die Umsetzung der dabei gewonnenen Eindrücke und Erfahrungen kann die Qualität der Anwendung gesteigert werden.

Nach einer gewissen Eingewöhnungszeit war es jeder Testperson möglich, das Hotelzimmer mit Hilfe des hybriden AR-/VR-HMDs individuell und präzise einzurichten. Um die Dauer zur Eingewöhnung an das Video See-through HMD zu verkürzen und ein möglichst angenehmes Bedienerlebnis zu schaffen, muss die für den Menschen natürliche Sicht optimal simuliert werden. Für dieses Vorhaben müssen verschiedene Modifikationen am HMD vorgenommen werden. Zum einen muss eine zweite Kamera integriert werden, damit das bisher implementierte monoskopische System zu einer stereoskopischen Ansicht erweitert wird und der Nutzer somit einen räumlichen Eindruck seiner Umgebung erhält. Dabei müssen beide Kameras nach der Off-Axis Methode ausgerichtet werden, sodass keine vertikale Parallaxe auftritt. Mit diesem Vorgehen erhält der Nutzer eine dreidimensionale Sicht auf die reale Umgebung und kann Tiefeninformationen besser verarbeiten. Zum anderen sollten sowohl die Kameras als auch die Displays im HMD das für den Menschen natürliche FOV von 220° möglichst abdecken (Steptoe, William, 2014). Um sowohl dem Screen Door Effekt entgegenzuwirken als

auch das Tracking weiter entfernter AR-Marker zu optimieren, müssen die Kameras und Displays über eine erhöhte Auflösung verfügen. Die für das menschliche Auge wahrnehmbare Grenze liegt dabei bei 17000 x 12000 Pixeln (Horsky, Jan, 2017). Wie in Kapitel 5.1.1 beschrieben, gibt es zum Zeitpunkt der Projektdurchführung kaum für die private Nutzung verfügbare hybride AR-/VR-Wiedergabegeräte. Das HMD, welches den beschriebenen Anforderungen zum Zeitpunkt der Arbeit am ehesten gerecht wird, ist die im April 2018 erschienene HTC Vive Pro, welche über zwei an der Vorderseite integrierte Kameras, einem FOV von 110° und einer Auflösung von 1440 x 1600 Pixeln pro Auge verfügt (HTC Corporation, 2018).

Wie in Kapitel 3.1 erläutert, liegt einer der Vorteile von AR-Anwendungen im Bereich Innenarchitektur darin, dass die Kommunikation mehrerer Nutzer gefördert wird, welche gemeinsam an der Einrichtung eines Zimmers beteiligt sind. So könnte die im Rahmen der Abhandlung implementierte Applikation für die Zusammenarbeit mit anderen Personen erweitert werden. Um dieses Ziel zu realisieren, müsste mindestens ein weiteres hybrides AR-/VR-HMD für die Anwendung zur Verfügung stehen. Für die Interaktion mit anderen Nutzern könnten weitere Funktionen implementiert werden. So könnte beispielsweise ein Anwender im VR-Modus den Innenraum besichtigen und sich mit einer weiteren Person, welche das Hotelzimmer aus der AR-Ansicht sieht, über diverse Einrichtungs-Möglichkeiten beraten. Dabei könnte dem Nutzer, der das Hotelzimmer aus der AR-Sicht wahrnimmt, die Position des Users, der sich im VR-Modus befindet, als Markierung im virtuellen Innenraum angezeigt werden. Somit wüssten die Beobachter stets, wo sich der Nutzer in der VR-Ansicht aktuell befindet. Des Weiteren könnte dem Nutzer im VR-Modus eine Funktion zur Verfügung stehen, mit welcher er seine virtuelle Besichtigung aufzeichnen und mit anderen Personen teilen kann.

Um die individuelle Einrichtung des Zimmers zu fördern, könnten für jeden Einrichtungsgegenstand verschiedene Farben und Designs hinzugefügt werden. So könnte der Anwender beispielsweise mit Hilfe eines Sprachbefehls oder durch die Betätigung eines virtuellen Buttons fließend zwischen den einzelnen Stilrichtungen eines Möbelstücks hin- und herschalten. Somit würde nicht nur die Gestaltung des Innenraums, sondern auch der Einrichtungsgegenstände in gewissem Maße ermöglicht werden.

7.2 Fazit

Im Rahmen dieser Arbeit wurde die Verbindung von Augmented und Virtual Reality durch die Konzeption, Implementation und Evaluation eines hybriden AR-/VR-Systems zur Gestaltung von Innenräumen eruiert. Dem Leser wurden dabei die technologischen Grundlagen der beiden XR-Stufen erläutert und die einzelnen Schritte bei der Realisierung der Anwendung präsentiert. Die wesentlichen Schlussfolgerungen sollen im Folgenden zusammengefasst werden:

AR und VR eignen sich für verschiedene Anwendungsbereiche unterschiedlich gut. Während sich die Vorteile von AR vor allem in einer natürlichen und intuitiven Interaktion sowie einer geförderten Kommunikation zeigen, lassen sich mit VR verschiedene Szenarien immersiv und realitätsnah darstellen. Vereint werden beide Technologien darin, dass die Grenzen zwischen Realität und Virtualität verschwinden, sodass sich digitale Informationen wie reale Objekte verwenden lassen. Da dabei kein materielles Inventar benötigt wird, können die beiden Darstellungsformen besonders gut in der Planungsphase ressourcenabhängiger Projekte, wie der Gestaltung eines Innenraums, integriert werden. Für die Realisierung eines hybriden AR-/VR-Systems muss ein Wiedergabegerät als Basis dienen, welches beide XR-Stufen gleichermaßen gut unterstützt. Um eine optimale Nutzererfahrung zu schaffen, muss die Anwendung von Grund auf für beide Darstellungsformen konzipiert werden. Um bestehende Anforderungen bestmöglich erfüllen zu können, muss der Entwickler über ein Grundverständnis für den jeweiligen Anwendungsbereich verfügen und im optimalen Fall Vertreter aus dem jeweiligen Spezialgebiet hinzuziehen. Bei dem Übergang zwischen AR und VR sollte besonders auf einen korrekten Informationsaustausch sowie einen fließenden Wechsel und ein intuitiv bedienbares UI geachtet werden. Da beide Medien noch am Anfang ihrer Entwicklung stehen, sollten verschiedene Funktionen mit Probanden getestet sowie neue Ideen und Meinungen berücksichtigt werden.

Zusammenfassend lässt sich festhalten, dass AR und VR für die Interaktion zwischen Mensch und Computer eine essentielle Rolle spielen. Dabei werden bestehende Technologien nicht zwingend ersetzt, jedoch ergeben sich neue Möglichkeiten, Informationen darzustellen und Anwendungsbereiche zu optimieren.

„There’s decades of innovations ahead. We’re at the very beginning, where it’s just at the stage where we can bring in consumers [but] there’s so much further to go from there.“ - Brendan Iribe

A Anhang

A.1 Evaluierungsbogen

		Stimme voll und ganz zu	Stimme eher zu	Stimme teilweise zu	Stimme eher nicht zu	Stimme überhaupt nicht zu
1. AR-Teil						
1.1. UI / Bedienung						
1.1.1.	Es ist leicht, die Einrichtungsgegenstände im AR-Teil an einer gewünschten Stelle zu positionieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.1.2.	Es ist leicht zu erkennen, worauf der Sicht-Fokus liegt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.2. AR-Ansicht im VR-Headset						
1.2.1.	Die Umgewöhnung von der normalen Sicht zu der AR-Ansicht im Headset ist leicht	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.2.2.	In der AR-Ansicht kommt es zu Unwohlsein oder Schwindelgefühlen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.2.3.	Die Koordination der Hände in der AR-Ansicht ist leicht	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.3. AR-Tracking						
1.3.1.	Es kommt oft zu Ausfällen bei der Anzeige der virtuellen Objekte auf den AR-Markern	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.3.2.	Die virtuellen Objekte sind deutlich zu sehen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.4. Zusatz-Informationen						
1.4.1.	Die Grundriss-Ansicht ist eine sinnvolle Erweiterung, um verdeckte Objekte anzuzeigen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.4.2.	Die Grundriss-Ansicht lässt sich leicht ein- und ausblenden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.4.3.	Die Anzeige der Objekt-Maße verbessert die Einschätzung für die Objekt-Größen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.4.4.	Die Anzeige der Objekt-Maße lässt sich leicht ein- und ausblenden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.4.5.	Die Anzeige der Positionierungs-Vorschläge bietet nützliche Ratschläge	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.4.6.	Die Anzeige der Positionierungs-Vorschläge lässt sich leicht ein- und ausblenden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

		Stimme voll und ganz zu	Stimme eher zu	Stimme teilweise zu	Stimme eher nicht zu	Stimme überhaupt nicht zu
2. Übergang AR zu VR						
2.1. Performance						
2.1.1.	Die Wartezeit für den Wechsel von AR zu VR ist akzeptabel	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.2. UI / Bedienung						
2.2.1.	Die Funktionalität und Handhabung des „AR zu VR“-Button ist leicht verständlich	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.2.2.	Es ist klar ersichtlich, auf welchem Auswahl-Button der Sicht-Fokus liegt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.2.3.	Die Auswahl-Button sind intuitiv bedienbar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.2.4.	Der „VR-Fortbewegungsart“-Dialog ist intuitiv bedienbar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.2.5.	Die Funktionsweisen der verschiedenen VR-Fortbewegungsarten sind verständlich	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.3. Übergabe der Eigenschaften der Einrichtungsgegenstände						
2.3.1.	Die Positionen der Einrichtungsgegenstände in der VR- und AR-Ansicht stimmen überein	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.3.2.	Die Ausrichtungen der Einrichtungsgegenstände in der VR- und AR-Ansicht stimmen überein	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

		Stimme voll und ganz zu	Stimme eher zu	Stimme teilweise zu	Stimme eher nicht zu	Stimme überhaupt nicht zu
3. VR-Teil						
3.1. VR-Fortbewegung						
3.1.1.	Man gewöhnt sich schnell an die Free-Lo-motion-Fortbewegungsart	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.1.2.	Man gewöhnt sich schnell an die Walking-In-Place-Fortbewegungsart	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.1.3.	Mit Free-Lo-motion ist eine gezielte Fortbewegung möglich	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.1.4.	Mit Walking-In-Place ist eine gezielte Fortbewegung möglich	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.1.5.	Mit Free-Lo-motion ist es angenehm, sich fortzubewegen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.1.6.	Mit Free-Lo-motion kommt es zu Unwohlsein oder Schwindelgefühlen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.1.7.	Mit Walking-In-Place ist es angenehm, sich fortzubewegen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.1.8.	Mit Walking-In-Place kommt es zu Unwohlsein oder Schwindelgefühlen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.2. Immersion						
3.2.1.	Es ist klar ersichtlich, dass man sich in einem Hotelzimmer befindet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.2.2.	Die visuelle Darstellung der Objekte ist für die Immersion ausreichend	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.2.3.	Die akustische Untermalung der VR-Szene ist für die Immersion ausreichend	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

		Stimme voll und ganz zu	Stimme eher zu	Stimme teilweise zu	Stimme eher nicht zu	Stimme überhaupt nicht zu
4. Übergang VR zu AR						
4.1. Performance						
4.1.1.	Der Wechsel von VR zu AR geht schnell	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.2. UI / Bedienung						
4.2.1.	Es ist klar ersichtlich, wie man zur AR-Sicht zurück wechselt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.2.2.	Die Auswahl-Button sind intuitiv bedienbar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

		Stimme voll und ganz zu	Stimme eher zu	Stimme teilweise zu	Stimme eher nicht zu	Stimme überhaupt nicht zu
5. Intention der Anwendung						
5.1.	Die Anwendung vereinigt sinnvoll das Zusammenspiel von AR und VR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.2.	Die Einrichtung eines Hotelzimmers lässt sich mit AR sinnvoll realisieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.3.	Die Besichtigung des zuvor eingerichteten Hotelzimmers lässt sich mit VR immersiv darstellen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Literatur

- [wiki2015Vrijheidsgraad] *Vrijheidsgraad*. <https://nl.wikipedia.org/wiki/Vrijheidsgraad>. – Online; Abgerufen am 9. Juni 2018
- [Bajura und Neumann 1995] BAJURA, Michael ; NEUMANN, Ulrich: Dynamic registration correction in video-based augmented reality systems. In: *IEEE Computer Graphics and Applications* 15 (1995), Nr. 5, S. 52–60
- [Bell u. a. 2001] BELL, Blaine ; FEINER, Steven ; HÖLLERER, Tobias: View management for virtual and augmented reality. In: *Proceedings of the 14th annual ACM symposium on User interface software and technology* ACM (Veranst.), 2001, S. 101–110
- [Bezmalinovic, Tomislav 2017] BEZMALINOVIC, TOMISLAV: *ZED Mini macht aus Oculus Rift und HTC Vive Mixed-Reality-Brillen*. <https://vrodo.de/die-zed-mini-macht-aus-oculus-rift-und-htc-vive-mixed-reality-brillen/>. 2017. – Online; Abgerufen am 3. Juni 2018
- [Bhandari u. a. 2017] BHANDARI, Jiwan ; TREGILLUS, Sam ; FOLMER, Eelke: Legomotion: scalable walking-based virtual locomotion. In: *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology* ACM (Veranst.), 2017, S. 18
- [Billinghurst u. a. 2005] BILLINGHURST, Mark ; GRASSET, Raphael ; LOOSER, Julian: Designing augmented reality interfaces. In: *ACM Siggraph Computer Graphics* 39 (2005), Nr. 1, S. 17–22
- [Billinghurst u. a. 2009] BILLINGHURST, Mark ; KATO, Hirokazu ; MYOJIN, Seiko: Advanced interaction techniques for augmented reality applications. In: *International Conference on Virtual and Mixed Reality* Springer (Veranst.), 2009, S. 13–22
- [Billinghurst u. a. 2001] BILLINGHURST, Mark ; KATO, Hirokazu ; POUPYREV, Ivan: The magicbook-moving seamlessly between reality and virtuality. In: *IEEE Computer Graphics and applications* 21 (2001), Nr. 3, S. 6–8
- [Boas 2013] BOAS, YAGV: Overview of virtual reality technologies. In: *Interactive Multimedia Conference* Bd. 2013, 2013
- [Bowman u. a. 2001] BOWMAN, Doug A. ; KRUIJFF, Ernst ; LAVIOLA JR, Joseph J. ; POUPYREV, Ivan: An introduction to 3-D user interface design. In: *Presence: Teleoperators & Virtual Environments* 10 (2001), Nr. 1, S. 96–108

- [Buckley, Sean 2016] BUCKLEY, SEAN: *Why 'teleportation' makes sense in virtual reality.* <https://www.engadget.com/2016/10/07/why-teleportation-makes-sense-in-virtual-reality/?guccounter=1>. 2016. – Online; Abgerufen am 3. Juni 2018
- [Campbell und Wells 1994] CAMPBELL, Dace A. ; WELLS, Maxwell: A critique of virtual reality in the architectural design process. In: *University of Washington HITL Technical Report R-94 3* (1994)
- [Doerner u. a. 2013] DOERNER, R ; BROLL, W ; GRIMM, P ; JUNG, B: *Virtual und Augmented Reality.* 2013
- [Dolnicar und Otter 2003] DOLNICAR, Sara ; OTTER, Th: Which hotel attributes matter? A review of previous and a framework for future research. (2003)
- [Donath und Regenbrecht 1995] DONATH, Dirk ; REGENBRECHT, Holger: VRAD (Virtual Reality Aided Design) in the early phases of the architectural design process. (1995)
- [Donath und Regenbrecht 1996] DONATH, Dirk ; REGENBRECHT, Holger: Using virtual reality aided design techniques for three-dimensional architectural sketching. In: *Design Computation, Collaboration, Reasoning, Pedagogy. ACADIA Conference Proceedings, 1996*, S. 201–212
- [Drascic und Milgram 1996] DRASCIC, David ; MILGRAM, Paul: Perceptual issues in augmented reality. In: *Stereoscopic displays and virtual reality systems III* Bd. 2653 International Society for Optics and Photonics (Veranst.), 1996, S. 123–135
- [Edwards u. a. 1993] EDWARDS, Emily K. ; ROLLAND, Jannick P. ; KELLER, Kurtis P.: Video see-through design for merging of real and virtual environments. In: *Virtual Reality Annual International Symposium, 1993., 1993 IEEE* IEEE (Veranst.), 1993, S. 223–233
- [Ellefsen u. a. 1979] ELLEFSEN, Richard ; KUHN, Thomas ; SPRING, Gary ; WICKS, LE: Characteristics of Urban Terrain / SAN JOSE STATE UNIV CA CARTREMS LAB. 1979. – Forschungsbericht
- [Fremont 2017] FREMONT: *ASUS ZenFone AR on Sale in the United States.* [//www.asus.com/us/News/OuJFodpfkQvxrjOC](http://www.asus.com/us/News/OuJFodpfkQvxrjOC). 2017. – Online; Abgerufen am 9. Juni 2018
- [Garrido-Jurado u. a. 2014] GARRIDO-JURADO, Sergio ; MUÑOZ-SALINAS, Rafael ; MADRID-CUEVAS, Francisco J. ; MARÍN-JIMÉNEZ, Manuel J.: Automatic generation and detection of highly reliable fiducial markers under occlusion. In: *Pattern Recognition* 47 (2014), Nr. 6, S. 2280–2292

- [HAF Hamburger Institut fuer Fernstudien GmbH 2018] HAF HAMBURGER INSTITUT FUER FERNSTUDIEN GMBH: *Raumgestaltung Innenarchitektur*. <http://www.raumgestaltung-innenarchitektur.de/raumgestaltung-innenarchitektur/planen.php>. 2018. – Online; Abgerufen am 10. Januar 2018
- [Himix 2015] HIMIX: *Augmented Reality Tutorial No. 12: Augmented Reality using AR-media plugin*. <http://www.himix.lt/augmented-reality/augmented-reality-using-armedia/>. 2015. – Online; Abgerufen am 8. Juni 2018
- [Honkanen u. a. 2018] HONKANEN, Lotta u. a.: *Enhancing Historical Attractions Through Augmented Reality*. (2018)
- [Horsky, Jan 2017] HORSKY, JAN: *When will Screen-Door-Effect go away?* <https://blog.infinite.cz/when-will-screen-door-effect-go-away-d8591f598927>. 2017. – Online; Abgerufen am 3. Juni 2018
- [HTC Corporation 2018] HTC CORPORATION: *Vive Pro*. <https://www.vive.com/de/product/vive-pro/>. 2018. – Online; Abgerufen am 3. Juni 2018
- [Irving, Michael 2016] IRVING, MICHAEL: *Restricting field of view to reduce motion sickness in VR*. <https://newatlas.com/columbia-university-vr-motion-sickness/43855/>. 2016. – Online; Abgerufen am 1. Juni 2018
- [Kanbara u. a. 2000] KANBARA, Masayuki ; OKUMA, Takashi ; TAKEMURA, Haruo ; YOKOYA, Naokazu: A stereoscopic video see-through augmented reality system based on real-time vision-based registration. In: *Virtual Reality, 2000. Proceedings. IEEE* IEEE (Veranst.), 2000, S. 255–262
- [Karlsson und Olsson] KARLSSON, Erik ; OLSSON, Henrik: Vehicle sensor data real time visualizer.
- [Kato u. a. 2000] KATO, Hirokazu ; BILLINGHURST, Mark ; POUPYREV, Ivan ; IMAMOTO, Kenji ; TACHIBANA, Keihachiro: Virtual object manipulation on a table-top AR environment. In: *Augmented Reality, 2000. (ISAR 2000). Proceedings. IEEE and ACM International Symposium on Ieee* (Veranst.), 2000, S. 111–119
- [Kim u. a. 2014] KIM, Sung L. ; SUK, Hae J. ; KANG, Jeong H. ; JUNG, Jun M. ; LAINE, Teemu H. ; WESTLIN, Joonas: Using Unity 3D to facilitate mobile augmented reality game development. In: *Internet of Things (WF-IoT), 2014 IEEE World Forum on IEEE* (Veranst.), 2014, S. 21–26

- [Koehler u. a. 2011] KOEHLER, Johannes ; PAGANI, Alain ; STRICKER, Didier: Detection and identification techniques for markers used in computer vision. In: *OASIS-OpenAccess Series in Informatics* Bd. 19 Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (Veranst.), 2011
- [Lancaster, Luke 2016] LANCESTER, LUKE: *Microsoft HoloLens brings mixed reality to Australia and UK*. <https://www.cnet.com/uk/news/you-can-now-preorder-microsoft-hololens-in-six-more-countries-sort-of/>. 2016. – Online; Abgerufen am 9. Juni 2018
- [Lee u. a. 2017] LEE, Jiwon ; KIM, Mingyu ; KIM, Jinmo: A Study on immersion and VR sickness in walking Interaction for immersive virtual reality applications. In: *Symmetry* 9 (2017), Nr. 5, S. 78
- [Linowes 2015] LINOWES, Jonathan: *Unity virtual reality projects*. Packt Publishing Ltd, 2015
- [Logitech 2018] LOGITECH: *HD Webcam C310*. http://support.logitech.com/en_us/product/hd-webcam-c310/specs. 2018. – Online; Abgerufen am 5. Januar 2018
- [Ludwig und Reimann 2005] LUDWIG, Christine ; REIMANN, Christian: Augmented reality: Information at focus. In: *C-Lab Report* 4 (2005), Nr. 1
- [Ludwig 2009] LUDWIG, D.I.: *Plandarstellung gemaess OENORM A 6240-2*. Universitaet Innsbruck, 2009
- [Mak 2013] MAK, Hanson Hang S.: *Hotel Architecture for the Creative Class*, Carleton University, Dissertation, 2013
- [Mann u. a. 2018] MANN, Steve ; FURNESS, Tom ; YUAN, Yu ; IORIO, Jay ; WANG, Zixin: All Reality: Virtual, Augmented, Mixed (X), Mediated (X, Y), and Multimeditated Reality. In: *arXiv preprint arXiv:1804.08386* (2018)
- [Martindale, Jon 2018] MARTINDALE, JON: *Oculus Rift vs. HTC Vive*. <https://www.digitaltrends.com/virtual-reality/oculus-rift-vs-htc-vive/>. 2018. – Online; Abgerufen am 27. Mai 2018
- [Merrell u. a. 2011] MERRELL, Paul ; SCHKUFZA, Eric ; LI, Zeyang ; AGRAWALA, Maneesh ; KOLTUN, Vladlen: Interactive furniture layout using interior design guidelines. In: *ACM Transactions on Graphics (TOG)* 30 (2011), Nr. 4, S. 87
- [Microsoft Corporation 2018] MICROSOFT CORPORATION: *Development*. <https://docs.microsoft.com/en-us/windows/mixed-reality/>. 2018. – Online; Abgerufen am 29. Mai 2018

- [Milgram u. a. 1995] MILGRAM, Paul ; TAKEMURA, Haruo ; UTSUMI, Aki-
ra ; KISHINO, Fumio: Augmented reality: A class of displays on the reality-
virtuality continuum. In: *Telem manipulator and telepresence technologies*
Bd. 2351 International Society for Optics and Photonics (Veranst.), 1995,
S. 282–293
- [OConnell, Kim 2016] OCONNELL, KIM: *Vier Tipps fuer die ersten*
Anwendungen von Virtual Reality in der Architektur. [https://www.
autodesk.de/redshift/virtual-reality-in-architecture](https://www.autodesk.de/redshift/virtual-reality-in-architecture). 2016. –
Online; Abgerufen am 4. Juni 2018
- [Paligaru, Matt 2018] PALIGARU, MATT: *How To Get*
The Most from Virtual and Augmented Reality. [https:
//blog.bestbuy.ca/wearable-technology/virtual-reality/
get-virtual-augmented-reality](https://blog.bestbuy.ca/wearable-technology/virtual-reality/get-virtual-augmented-reality). 2018. – Online; Abgerufen am
3. Juni 2018
- [Pfeifer, Ralf 2010] PFEIFER, RALF: *Koerperlaengen.* [http://www.
arsmartialis.com/index.html?name=http://www.arsmartialis.
com/technik/laenge/laenge.html](http://www.arsmartialis.com/index.html?name=http://www.arsmartialis.com/technik/laenge/laenge.html). 2010. – Online; Abgerufen am 29.
Januar 2018
- [Phan und Choo 2010] PHAN, Viet T. ; CHOO, Seung Y.: Interior design
in augmented reality environment. In: *International Journal of Computer*
Applications 5 (2010), Nr. 5
- [Poupyrev u. a. 2002] POUPYREV, Ivan ; TAN, Desney S. ; BILLINGHURST,
Mark ; KATO, Hirokazu ; REGENBRECHT, Holger ; TETSUTANI, Nobuji:
Developing a generic augmented-reality interface. In: *Computer* 35 (2002),
Nr. 3, S. 44–50
- [robbonline 2009] ROBBONLINE: *Flight Simulator X 2014 FSX HD - Ma-*
nual Landing Boeing 737-800 in snowy whether (cockpit view). [https:
//www.youtube.com/watch?v=oGGG9oTamkI](https://www.youtube.com/watch?v=oGGG9oTamkI). 2009. – Online; Abgerufen
am 9. Juni 2018
- [Scoble und Israel 2017] SCOBLE, Robert ; ISRAEL, Shel: The fourth trans-
formation. In: *How Augmented Reality and Artificial Intelligence Change*
Everything, Patrick Brewster (2017)
- [Siguaw und Enz 1999] SIGUAW, Judy A. ; ENZ, Cathy A.: Best practi-
ces in hotel operations. In: *Cornell Hotel and Restaurant Administration*
Quarterly 40 (1999), Nr. 6, S. 42–53
- [Siltanen 2012] SILTANEN, Sanni: *Theory and applications of marker-based*
augmented reality. VTT, 2012

- [Sketchfab 2018] SKETCHFAB: *Materials (PBR)*. <https://help.sketchfab.com/hc/en-us/articles/204429595-Materials-PBR->. 2018. – Online; Abgerufen am 22. Mai 2018
- [Slater u. a. 1995] SLATER, Mel ; USOH, Martin ; STEED, Anthony: Taking steps: the influence of a walking technique on presence in virtual reality. In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 2 (1995), Nr. 3, S. 201–219
- [Slay u. a. 2001] SLAY, Hannah ; PHILLIPS, Matthew ; VERNIK, Rudi ; THOMAS, Bruce: Interaction modes for augmented reality visualization. In: *Proceedings of the 2001 Asia-Pacific symposium on Information visualisation-Volume 9* Australian Computer Society, Inc. (Veranst.), 2001, S. 71–75
- [Solem 2012] SOLEM, Jan E.: *Programming Computer Vision with Python: Tools and algorithms for analyzing images*. Ö'Reilly Media, Inc., 2012
- [Statista 2018] STATISTA: *Statistik-Lexikon: Definition Likert-Skala*. https://de.statista.com/statistik/lexikon/definition/82/likert_skala/. 2018. – Online; Abgerufen am 2. Juni 2018
- [Steed u. a. 2017] STEED, Anthony ; ADIPRADANA, Yonathan W. ; FRISTON, Sebastian: The AR-Rift 2 Prototype. In: *Virtual Reality (VR), 2017 IEEE* IEEE (Veranst.), 2017, S. 231–232
- [Step toe, William 2014] STEPTOE, WILLIAM: *AR-Rift*. <http://willstep toe.com/post/66968953089/ar-rift-part-1>. 2014. – Online; Abgerufen am 05. Januar 2018
- [Studio 216 2017] STUDIO 216: *Virtual and Mixed Reality meets Architecture and Design*. <http://www.studio216.com/blog/2017/2/1/mixed-reality-architecture-and-design>. 2017. – Online; Abgerufen am 4. Juni 2018
- [Tintori, Ray 2016] TINTORI, RAY: *Tilt Brush - Painting from a New Perspective*. <https://vimeo.com/166560235>. 2016. – Online; Abgerufen am 9. Juni 2018
- [Tobak, Steve 2015] TOBAK, STEVE: *Why Virtual Reality Will Be Bigger Than Smartphones*. <http://www.foxbusiness.com/features/2015/06/11/why-virtual-reality-will-be-bigger-than-smartphones.html>. 2015. – Online; Abgerufen am 2. Juni 2018
- [Toth 2008] TOTH, Alfred: Grundzuege einer Semiotik des Hotelgewerbes. In: *Electronic Journal for Mathematical Semiotics*, <http://www.mathematicalsemiotics.com/pdf/Sem.%20des%20Hotelgewerbes.pdf> (2008)

- [Towneplace Suites Marriott 2018] TOWNEPLACE SUITES MARRIOTT: *Suites*. <http://towneplacesuites.marriott.com/suites/>. 2018. – Online; Abgerufen am 25. Mai 2018
- [Tregillus und Folmer 2016] TREGILLUS, Sam ; FOLMER, Eelke: Vr-step: Walking-in-place using inertial sensing for hands free navigation in mobile vr environments. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* ACM (Veranst.), 2016, S. 1250–1255
- [Tuttle, Will 2015] TUTTLE, WILL: *Xbox and Oculus Partner to Change the Face of Virtual Reality*. <https://news.xbox.com/en-us/2015/06/11/xbox-xbox-and-oculus-partner-to-change-the-face-of-virtual-reality/>. 2015. – Online; Abgerufen am 3. Juni 2018
- [Unity Technologies 2018a] UNITY TECHNOLOGIES: *Documentation*. <https://docs.unity3d.com/Manual>. 2018. – Online; Abgerufen am 22. Mai 2018
- [Unity Technologies 2018b] UNITY TECHNOLOGIES: *Types of light*. <https://docs.unity3d.com/Manual>. 2018. – Online; Abgerufen am 3. Juni 2018
- [Van Gaalen und Zhang 2016] VAN GAALEN, R ; ZHANG, I: *Augmented Reality in a Virtual Reality Setup*. (2016)
- [Vuforia 2018] VUFORIA: *Integrating Gear VR and the AR/VR Sample in Unity 5.3 and above*. <https://library.vuforia.com/articles/Solution/Integrating-Gear-VR-and-the-AR-VR-Sample-in-Unity-5-3-and-above>. 2018. – Online; Abgerufen am 6. März 2018
- [Wang 2009] WANG, Xiangyu: *Augmented reality in architecture and design: potentials and challenges for application*. In: *International Journal of Architectural Computing* 7 (2009), Nr. 2, S. 309–326
- [Woodward u. a. 2007] WOODWARD, Charles ; LAHTI, Jani ; RÖNKKÖ, Jukka ; HONKAMAA, Petri ; HAKKARAINEN, Mika ; JÄPPINEN, Jani ; RAINIO, Kari ; SILTANEN, Sanni ; HYVÄKKÄ, Jouko: *Case Digitalo-A range of virtual and augmented reality solutions in construction application*. In: *24th W78 Conference, Maribor*. In: *24th W* Bd. 78, 2007, S. 529–540