



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorthesis

Nils Ulbricht

Einsatzmöglichkeiten und Potenzial der virtuellen
Realität im Web am Beispiel der
Beleuchtungsindustrie

Nils Ulbricht

Einsatzmöglichkeiten und Potenzial der virtuellen
Realität im Web am Beispiel der
Beleuchtungsindustrie

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung
im Studiengang Informations- und Elektrotechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Heike Neumann
Zweitgutachter : Dr. C. Claudius J. Noack

Abgegeben am 20. Februar 2018

Nils Ulbricht

Thema der Bachelorthesis

Einsatzmöglichkeiten und Potenzial der virtuellen Realität im Web am Beispiel der Beleuchtungsindustrie

Stichworte

Virtuelle Realität, WebVR, Beleuchtung, räumlicher Ton, HTML, Javascript, Browser, Relux, A-Frame

Kurzzusammenfassung

Diese Arbeit beschäftigt sich mit der virtuellen Realität. Dabei wird ein besonderer Fokus auf die Beleuchtung, deren Planung und die Umsetzung einer virtuellen Realität im Web-Browser mit dem Framework A-Frame gelegt. Zudem wird sich mit dem wirtschaftlichen Potenzial der virtuellen Realität für die Industrie auseinander gesetzt.

Nils Ulbricht

Title of the paper

Application possibilities and potential of virtual reality in the web, using the example of the lighting industry

Keywords

Virtual Reality, WebVR, illumination, spatial audio, HTML, Javascript, browser, Relux, A-Frame

Abstract

This thesis deals with virtual reality. A special focus is placed on lighting, its planning and the implementation of a virtual reality in the web browser with the framework A-Frame. In addition, the economic potential of virtual reality for industry is examined.

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Bachelorarbeit motiviert und unterstützt haben.

Zuerst möchte ich mich bei der Claus & Sylvia Stäcker Stiftung für die Zuwendung erkenntlich zeigen. Außerdem möchte ich mich bei Prof. Dr. Heike Neumann und Herrn Dr. Claudius Noack für die Betreuung dieser Arbeit bedanken. Des Weiteren möchte ich mich bei Jens Heinrich, Luca Kiehn, Marietta Kappler, Vanessa Weissbrod und allen weiteren Angestellten der NORKA für die Unterstützung und Hilfestellungen bedanken.

Ergänzend bedanke ich mich bei den Unternehmen Deutsche Bahn AG, Deutsche See GmbH und Relux Informatik AG für die Bereitstellung der 360 Grad Inhalte, die im Rahmen dieser Arbeit verwendet wurden.

Abschließend möchte ich mich bei meinen Lieben bedanken, da sie mir während des gesamten Studiums Rückhalt gegeben sowie mir dieses Studium ermöglicht haben.

Nils Ulbricht
10.02.2018

Inhaltsverzeichnis

Tabellenverzeichnis	7
Abbildungsverzeichnis	8
Listings	9
1 Einführung	10
1.1 Vorstellung NORKA	10
1.2 Vorstellung virtuelle Realität	10
2 Theorie	11
2.1 Virtual Reality	11
2.2 Augmented Reality	12
2.3 Vestibuläres System	13
3 Analyse	15
3.1 Heutige Nutzung von virtueller Realität	15
3.2 Wirtschaftlichkeit	16
3.3 Ungenutztes Potenzial von virtueller Realität	17
3.4 VR in der Lichtbranche	18
3.5 Virtuelle Realität als Publikumsmagnet	18
3.6 Auswahl des HMD zur Darstellung der virtuellen Realität	18
4 Projektbeschreibung	20
4.1 Produktentwicklungstheorie	20
4.2 Idee	21
4.3 Systemvoraussetzungen	21
4.4 WebVR	22
4.5 A-Frame	22
5 Entwurf	23
5.1 Gestaltung	23
5.1.1 Infowindow	23
5.1.2 Verweis	23

5.1.3	Auswahlmenü	23
5.2	Klangwahrnehmung	25
5.3	Visuelle Wahrnehmung	26
6	Realisierung	28
6.1	Positionsbestimmung einer Leuchte	28
6.2	Leuchteninformation und Verweis	29
6.2.1	Informationen über Leuchten	29
6.2.2	Verweis auf Leuchten des gleichen Typs	30
6.3	Erstellung eines 360° Panoramas mit Relux	31
6.4	Koordinatenumrechnung	32
6.5	Darstellung des Auswahlmenüs	32
6.6	Klangberechnung	34
6.7	Mobile Geräte	35
6.8	Messe Version der WebVR Anwendung	36
6.8.1	Position der Seite	36
6.8.2	Erkennung des Browsers	39
6.8.3	Tablet	39
6.9	Externe Steuerung	39
6.9.1	Tablet-Steuerung	39
6.9.2	Tastatur-Steuerung	41
6.10	Statistische Auswertung der Auswahl der Benutzer	42
7	Tests	44
7.1	Testumgebung	44
7.2	Funktionstests und Fehlerbehandlungen	45
7.3	Anwendertest	46
8	Evaluation	48
8.1	Vergleich mit dem Entwurf	48
8.2	WebVR fähige Geräte	49
8.3	Mögliche Erweiterungen	51
9	Fazit	52
	Literaturverzeichnis	53
	Anhang	55
	Abkürzungsverzeichnis	56
	Index	57

Tabellenverzeichnis

3.1	HMD-Vergleich	19
6.1	Tastenbelegung für Tastatursteuerung	42
7.1	Testsysteme	45
7.2	Auswertung des Anwendertests in Sekunden	47
8.1	Anteil Smartphones	49
8.2	Anteil Rechner	51

Abbildungsverzeichnis

2.1	AR am Beispiel der App Pokémon Go	13
3.1	Virtuelle Realität als Verkaufshilfe (JP-Performance, 2017)	16
3.2	Prognose 2020: Umsatz durch VR pro Branche	17
5.1	Knopf zum Öffnen des Auswahlmenüs	24
5.2	Auswahlmenü mit ausgewähltem Punkt	25
5.3	Anordnungsbeispiel für virtuelle Lautsprecher	26
6.1	Informationen mit Verweis auf Leuchte	30
6.2	Darstellung des Auswahlmenüs mit verschiedener Anzahl von Anwendungs- gebieten	33
6.3	Realisierung des Auswahlmenüs mit ausgewähltem Anwendungsgebiet	34
6.4	Ansicht beider Augen	38
6.5	Umsetzung des Wiedergabebildschirms	38
6.6	Ansicht der externen Website zum Steuern	40
6.7	Statistik Website für Messen	43
8.1	Verlauf der Android Versionen (Computerbild, 2018)	50

Listings

6.1	Funktion die auf Leuchten des gleichen Modells verweist	30
6.2	Verschiebung des Body für die Messe	37
6.3	Socket-Verbindungsherstellung HTML	40
6.4	Socket-Verbindungsherstellung Javascript	41
6.5	Erkennung eines Tastendrucks am Beispiel der Taste „M“	41

1 Einführung

1.1 Vorstellung NORKA

Die *NORKA (Norddeutsche Kunststoff- und Elektrogesellschaft Stäcker mbH & Co. KG)* wurde 1948 von Walter Stäcker gegründet. Heute wird das Unternehmen in der dritten Generation von Herrn Martin C. Stäcker geführt. Die Verwaltung des Unternehmens ist mit ca. 50 Mitarbeitern in Hamburg ansässig, während die Produktion mit ca. 200 Mitarbeitern in Hülssen etwa 15 km südlich von Verden(Aller) beheimatet ist. NORKA produziert Leuchten für schwierige Umgebungsbedingungen, wie große Hitze, extreme Kälte oder chemische Belastungen zum Beispiel durch Chlor oder Ammoniak. Das Produktprogramm von NORKA umfasst ungefähr 4000 Leuchten

1.2 Vorstellung virtuelle Realität

Virtual Reality (VR) ist die Vortäuschung einer anderen Realität, die von Computern umgesetzt wurde und keine realen Gegenstände beinhaltet. Meistens werden sogenannte VR-Brillen in Verbindung mit Hand-Controllern benutzt, um dem Anwender einen Aufenthalt in einer irrealen Welt vorzutäuschen. Die computergestützte Grafiksimulation ist dabei nicht statisch. Das heißt, dass sich der Anwender in einem bestimmten Rahmen bewegen kann. So wird beispielsweise bei einer 90 Grad Drehung des Kopfes nach links simultan das angezeigte Bild in der VR-Brille geändert. Da besonders die Aktualisierung des angezeigten Bildes latenzarm verlaufen muss, ist es von hoher Wichtigkeit, dass ein Computer mit großer Rechenleistung und einer guten Grafikkarte zum Einsatz kommt.

2 Theorie

2.1 Virtual Reality

VR ist eine künstlich erzeugte, virtuelle Umgebung, in welcher es keine Interaktion mit der realen Umgebung gibt. Das heißt, dass im Optimalfall keine Eindrücke von außen durchdringen, die verhindern dass die virtuelle Realität mit allen Sinnen wahrgenommen wird. Je besser das gelingt, desto größer ist die Immersion. Dafür werden sogenannte Head-Mounted-Display (HMD)s eingesetzt, die auf den Kopf gesetzt werden und entweder mittels Computer ein sich bewegendes Bild in den Displays des HMD erzeugen, oder bei denen das Bild durch das Display eines eingesetzten Smartphones erzeugt wird. Der Begriff Immersion beschreibt den Effekt, der dem Bewusstsein vortäuscht, sich in einer anderen Welt zu befinden. Eine hundertprozentige Immersion ist technisch zurzeit nicht machbar, da dafür alle Sinne des Menschen getäuscht werden müssten. Aktuell wird daran gearbeitet die Immersion zu verbessern, beispielsweise mit Geräten, die man auf die Nase setzt und die Düfte in die virtuelle Realität bringen.

Um eine gute Immersion zu erreichen, sollte sichergestellt sein, dass ausreichend Bilder pro Sekunde (englisch: Frames per Second (FPS)) wiedergegeben werden können, damit für das menschliche Auge der Eindruck einer fließenden Bewegung entsteht. Optimal ist dabei ein Wert von ≥ 90 FPS. Zudem muss eine sehr hohe Auflösung benutzt werden, damit keine Fliegengittereffekt in dem HMD zu sichtbar ist. Dabei handelt es sich um eine Art Raster, die der Anwender eines HMD sehen kann. Der Effekt entsteht durch die kleine Distanz zwischen Auge und Display und wird zusätzlich durch die Linsen verstärkt. Es wird daher empfohlen, Displays mit mindestens 4K (3840 x 2160 Pixel) Auflösung pro Auge und einer hohen Pixeldichte einzusetzen. Weiterhin muss ein großes Blickfeld (englisch: Field of View (FOV)) bereitgestellt werden, damit nicht das Gefühl eines eingeschränkten Blickfelds entsteht.

Die Anfänge von VR sind auf das Jahr 1932 zurückzuführen, als Edwin Land die „Polarisator-Brille“ konstruierte, welche noch heute ermöglicht, 3D Filme anzugucken (VR-Nerds, 2017). Der nächste große Schritt für VR erfolgte Mitte der 1970er Jahre mit Myron Krueger, der den „Video Place“ entwickelte. Beim Video Space wurde eine virtuelle Realität durch einen Raum erzeugt, dessen Wände mit Monitoren ausgestattet waren. Durch die Monitore wurde

kein Eingabegerät seitens des Nutzers benötigt. 1994 wurde mit der „Forte VFX1“ das erste HMD vorgestellt. Das HMD bot eine Auflösung von 320 x 200 Pixel, konnte eine Drehung des Kopfes erkennen und hatte zwei Kopfhörer sowie ein Mikrofon. Zudem kostete das HMD mit 1800 Mark soviel wie ein damaliger Computer.

2000 wurde für die US-Airforce ein HMD entwickelt, welches eine Auflösung von 1920 x 1080 Pixeln erreichte, jedoch war dieses nicht käuflich und weitere Details wurden nicht an für die Öffentlichkeit freigegeben. 2012 wurde mit der „Oculus Rift“ die erste VR-Brille vorgestellt, die so wie heutige VR-Brillen aufgebaut ist

2.2 Augmented Reality

Augmented Reality (AR) ist die Vermischung einer virtuellen mit der realen Welt. Eine Möglichkeit dies umzusetzen ist, dass die reale Welt mit einer Kamera erfasst wird und die virtuelle Welt, bzw. die virtuellen Objekte nachträglich in das von der Kamera erfasste Bild eingefügt werden. Dieses Bild wird dann in Echtzeit an den Benutzer ausgegeben. So ist es beispielsweise möglich, dass sich zwei Personen gegenüber sitzen und dank eines HMD eine Partie Schach gegeneinander spielen können, ohne über reale Schachfiguren zu verfügen. Die vorhandene Realität wird genutzt und mit Bausteinen einer virtuellen Realität ergänzt.

Ein weiteres und zudem sehr bekanntes Beispiel für AR ist die Smartphone App „Pokémon Go“, welche im Sommer 2016 veröffentlicht und zwischenzeitlich von 30 Millionen Benutzern mindestens einmal pro Tag verwendet wurde. Der Benutzer musste dabei die App öffnen und sich außerhalb von Gebäuden bewegen um zufällige Pokémon fangen oder gegen diese mit seinen bereits gefangenen Pokémon kämpfen zu können. Dabei wurde auf die Kamera des Smartphones, bei eingeschalteter AR-Funktion, zugegriffen und dem Bild ein virtuelles Pokémon hinzugefügt (siehe Abbildung 2.1).

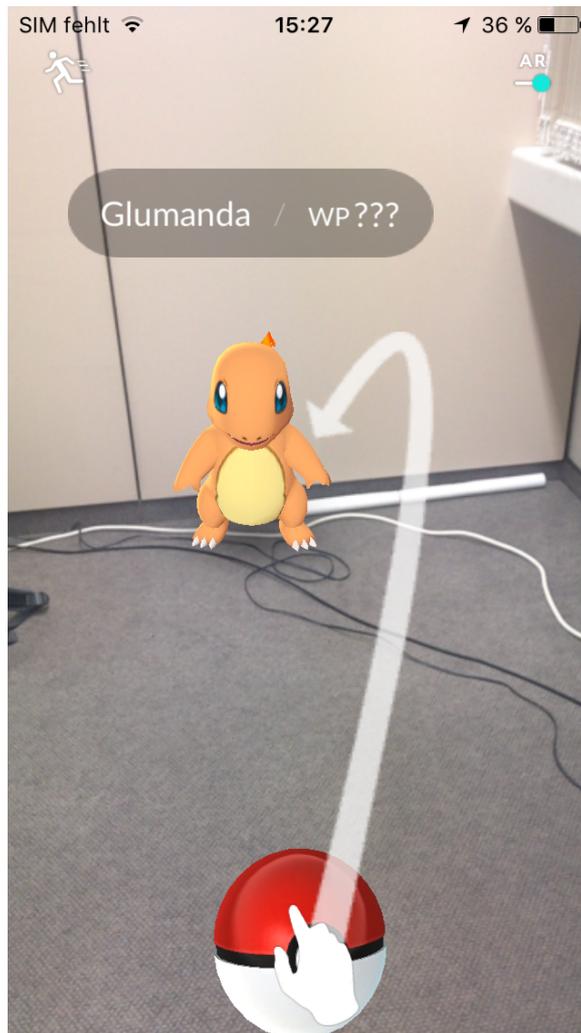


Abbildung 2.1: AR am Beispiel der App Pokémon Go

2.3 Vestibuläres System

Das vestibuläre System im menschlichen Körper dient der Orientierung beziehungsweise der Gleichgewichtsfindung. Zum vestibulären System gehört das Gleichgewichtsorgan, welches im Innenohr des Menschen sitzt (Behrends und Alumni, 2016). Es ist mit Muskeln und Sehnen im gesamten Körper verknüpft. Im Zusammenspiel mit dem Seh- und Tastsinn ermöglicht es uns, ohne Nachdenken das Gleichgewicht halten zu können. Wenn die

einzelnen Apparate zur Steuerung des Gleichgewichts widersprüchliche Signale senden, kann dies zu Schwindel führen. Ein Beispiel dafür ist die Seekrankheit, die im Inneren eines Schiffes auftreten kann, wenn der Gleichgewichtssinn eine schiefe Lage des Körpers meldet, die Augen jedoch dieses nicht bestätigen können. Für die Augen ist die Schräglage nicht erkennbar, da sich das Schiff genauso dreht wie der Körper.

VR kann zu einer Störungen des vestibulären Systems beitragen und somit zu VR-Krankheit (englisch: Motion-Sickness) führen. VR-Krankheit hat dabei die gleichen Symptome wie See- beziehungsweise Flugkrankheit. So kann die Nutzung einer VR-Brille zu Unwohlsein, Schwindel, erhöhtem Puls oder Schweißausbrüchen führen. Sobald eines dieser Symptome auftritt sollte die VR-Brille unverzüglich abgesetzt werden.

3 Analyse

In diesem Kapitel wird das Thema VR in der Industrie analysiert. Dabei wird ein spezieller Fokus auf die heutige Nutzung, die Wirtschaftlichkeit, ungenutztes Potenzial und die Auswahl eines passenden HMD gelegt.

3.1 Heutige Nutzung von virtueller Realität

Bei der Analyse der heutigen Nutzung von VR ist zwischen der Vermarktung für private und geschäftliche Konsumenten zu unterscheiden. VR für den privaten Bereich wird momentan hauptsächlich von der Spiele- und Unterhaltungsindustrie genutzt (Statista, 2018). In anderen Branchen wird VR zurzeit noch selten genutzt. Positive, zu erwähnende VR-Projekte sind zum Beispiel die vom Kraftfahrzeughersteller AUDI und dem Einrichtungshaus IKEA. Bei AUDI kann mittlerweile bei jedem zweiten Händler das Auto durch den Kunden selbst konfiguriert und das Resultat anschließend mit Hilfe einer VR-Brille sowohl von innen als auch von außen begutachtet werden (AUDI, 2017). Außerdem bietet IKEA ein VR-Projekt an, bei dem man in einer Küche steht und die vorhandenen Möbel anders gestalten, oder mit Controllern Hackfleischbällchen braten kann. Aus wirtschaftlicher Sicht fehlt noch die Funktion die angeschauten Möbel direkt online zu bestellen. Zudem kann man das VR-Projekt von AUDI nur im Autohaus benutzen, wodurch weniger Interessenten angesprochen werden, als wenn eine Benutzung über Smartphone möglich wäre. An diesen Beispielen wird deutlich, welches Potenzial VR mit sich bringt und wie ein potenzieller Kunde die Produkte virtuell erleben und auf die Varianten aufmerksam gemacht werden kann. Andererseits zeigen diese Beispiele auch, dass noch viel mehr im Bereich VR möglich und noch Verbesserungspotenzial gegeben ist.



Abbildung 3.1: Virtuelle Realität als Verkaufshilfe (JP-Performance, 2017)

In der Industrie wird VR schon deutlich länger als für private Endverbraucher genutzt. So wurde in den 1990er Jahren das System Cave Automatic Virtual Environment (CAVE) eingesetzt um virtuelle Welten in einem Raum durch Wandprojektionen darstellen zu können. Dieses System war technisch sehr aufwendig und teuer, hatte aber den Vorteil, dass der Benutzer kein HMD benutzen musste. CAVE wurde besonders zur Forschung in der Industrie, zum Beispiel in der Automobil- und der Flugzeugindustrie genutzt. Heute hat CAVE keine verbreitete Verwendung mehr, da es durch mobile und leistungsfähigere VR-Systeme ersetzt wurde.

3.2 Wirtschaftlichkeit

Experten schätzen, dass mit VR im Jahr 2017 ein weltweiter Umsatz von 10 Milliarden USD erzielt wurde. Bis 2020 wird ein Anstieg dieses Umsatzes auf 30 Milliarden USD prognostiziert. Wobei 2020 der größte Anteil auf die Spieleindustrie mit 14 Milliarden fallen würde, gefolgt von der Hardwarebranche mit 6 Milliarden und der Filmbranche mit 5,1 Milliarden (siehe Abbildung 3.2)(Digi-Capital, 2016).

Zudem wird geschätzt, dass der noch niedrigere Umsatz von AR ebenfalls ansteigt und im Jahr 2019 bereits den VR-Umsatz übertrifft.

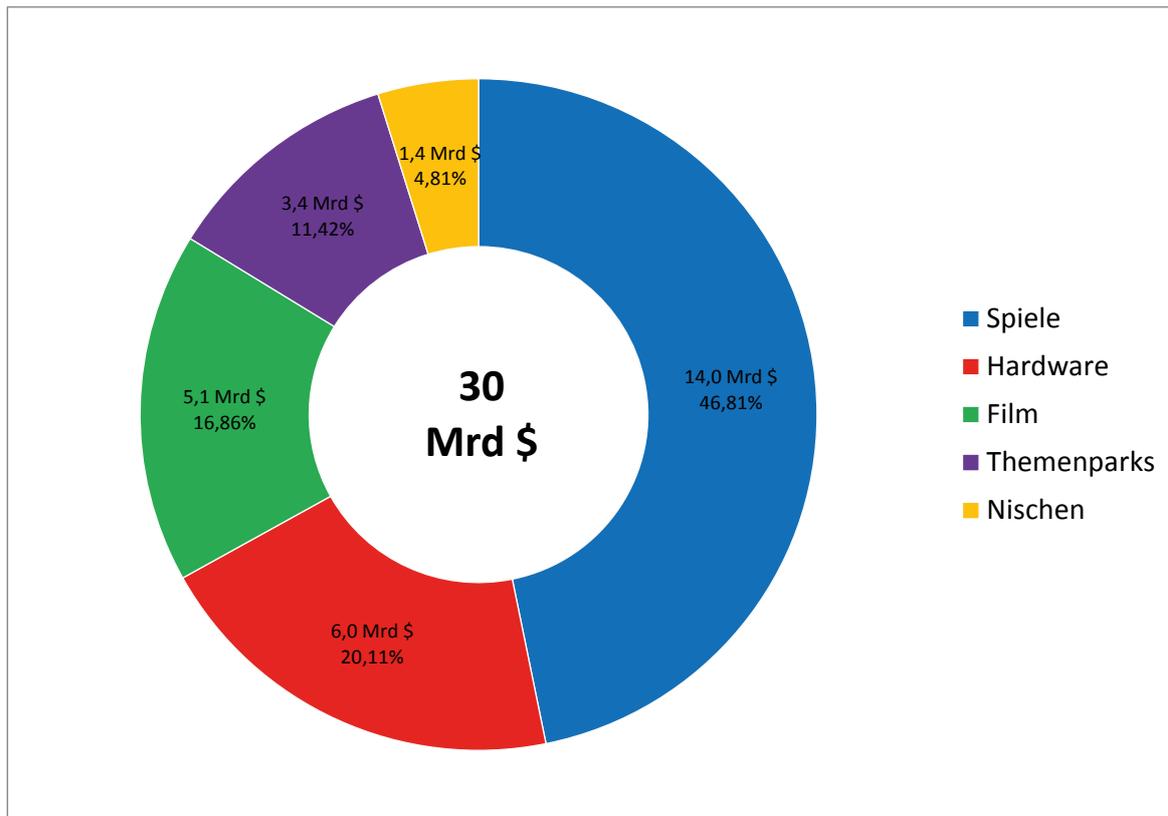


Abbildung 3.2: Prognose 2020: Umsatz durch VR pro Branche

3.3 Ungenutztes Potenzial von virtueller Realität

VR ermöglicht Firmen, ihre Produkte den Kunden und Interessenten virtuell vorzustellen, ohne dass der Kunde dafür den Ort wechseln muss um diese in ihrer Anwendung zu sehen. Die meisten Firmen haben 3D Modelle von ihren Produkten, die sie mit minimalem Aufwand den Kunden und Interessenten durch eine VR-Anwendung zur Verfügung stellen können. Durch diese VR-Anwendung kann der Kunde verschiedene Spezifikationen betrachten und entscheiden, welche Variante am besten gefällt. Die virtuelle Betrachtung und Auswahl unterstützt und fördert eine Kaufentscheidung erheblich.

Bei einer AR-Anwendung, kann der Kunde das gewünschte Produkt sogar virtuell in der Umgebung platzieren, wo es sich befinden soll. Das ist besonders bei Layout- und Designaspekten von großem Vorteil.

3.4 VR in der Lichtbranche

In der Lichtbranche bietet beispielsweise „WeCon Solution GmbH“ eine Lichtplanung in VR an (WeCon, 2018). Mit aufgesetzter VR-Brille kann der Anwender verschiedene Alternativen ausprobieren und entscheiden, welche Bereiche eines Raums stärker oder weniger stark beleuchtet werden sollen. Somit kann eine individuellere Ausleuchtung gewährleistet werden.

3.5 Virtuelle Realität als Publikumsmagnet

Dieser Abschnitt beschäftigt sich damit, inwieweit VR sich als Publikumsmagnet eignet, und warum.

Im Zeitalter der Digitalisierung gibt es eine hohe Anzahl von Menschen, die sehr aufgeschlossen gegenüber neuen Technologien sind und dazu neigen, diese zu testen. Da VR eine relativ neue Technologie ist, möchten viele Menschen sie selber ausprobieren. Auf Messen wird VR daher zur Zeit häufig eingesetzt um Interesse und Aufmerksamkeit bei dem vorbeigehenden Publikum zu wecken und Kontakt mit potenziellen Kunden oder Geschäftspartnern herzustellen. Gleichzeitig präsentiert sich die ausstellende Firma dadurch als ein modernes und innovatives Unternehmen. Häufig fehlt in diesen VR-Anwendungen jedoch der Produktbezug. Die Interessenten werden mit der VR-Anwendung an sich begeistert, aber ohne weitere Verknüpfung zu den Produkten des Ausstellers. Deswegen ist es von hoher Wichtigkeit, einen starken Bezug zu den Produkten und ihren Besonderheiten in der VR-Umgebung herzustellen, sodass die Produkte beim potenziellen Kunden im Gedächtnis hängen bleiben. Da viele VR-Anwender die VR-Anwendung selbst betrachten wollen, ohne eine direkte Anweisung worauf ihre Aufmerksamkeit zu richten ist, ist es die Aufgabe des Entwicklers der VR-Anwendung, zu überlegen, wie der Anwender bestimmte und interessante Stellen anvisiert, ohne dass Hinweise ihn dabei stören. Der Anwender soll das Gefühl haben, dass es seine eigene Entscheidung ist, auf die Stellen zu gucken, die vom Hersteller für Interaktionen vorhergesehen sind (MDS, 2017).

3.6 Auswahl des HMD zur Darstellung der virtuellen Realität

Auf dem Markt gibt es unterschiedliche HMD-Modelle, mit verschiedenen Vorteilen und Nachteilen. Eine kurze Übersicht darüber ist in Tabelle 3.1 zu finden.

Tabelle 3.1: HMD-Vergleich

Modell	Auflösung horizontal	Auflösung vertikal	FOV	Gewicht
Oculus Rift	2160 px	1200 px	110 °	500 g
HTC Vive	2160 px	1200 px	110 °	600 g
HTC Vive Pro	2880 px	1600 px	110 °	0 g
Playstation VR	1920 px	1080 px	100 °	610 g
LG 360 VR	960 px	720 px	80 °	100 g
Samsung Gear VR	2560 px	1440 px	96 °	318 g

Für dieses Projekt wurde das Modell „HTC Vive“ ausgewählt, da es zwei integrierte Displays besitzt und nicht zusätzlich ein passendes Smartphone benötigt wird. Die zwei Displays haben eine Auflösung von 1080 x 1200 Pixel pro Auge, was einer gesamten Auflösung von 2160 x 1200 Pixel entspricht und eine Bildwiederholrate von 90 Hz und ein FOV von 110 Grad. Außerdem besitzt die HTC Vive eine sehr gute und bessere Standortbestimmung, als die des Modells „Oculus Rift“, was durch 2 Basis-Stationen, die diagonal gegenüber im Raum angebracht werden, gewährleistet wird. Die Basisstationen erkennen mittels Laserstrahlen die Position des HMD im Raum. Zusätzlich ist eine Kamera und ein Mikrofon eingebaut. Wenn neben dem visuellen auch das akustische VR-Erlebnis genutzt werden soll, ist entweder ein Kopfhörer am HMD anzuschließen, oder der optional erhältliche „Deluxe Audio Strap“ zu kaufen, welcher zudem einen besseren Tragekomfort gewährt. Der verfügbare Aktionsbereich beträgt maximal 5 mal 5 Meter mit genau 2 Basisstationen.

Mitte Januar 2018 wurde das weiterentwickelte Modell „HTC Vive Pro“ vorgestellt, welches über eine Auflösung von 1440 mal 1600 Pixel pro Auge verfügt, was einer gesamten Auflösung von 2880 mal 1600 Pixel entspricht. Gegenüber dem vorher beschriebenen Modell verfügt das neue Modell über einen eingebauten Kopfhörer und 2 Kameras. Es hat ein Mikrofon, die gleiche Bildwiederholrate und das gleiche FOV wie die HTC Vive. Für 2018 sind noch neue Basisstationen und eventuell neue Controller mit einem größeren Funktionsumfang angekündigt. Mit den neuen Basisstationen kann dann ein Aktionsbereich von 10 mal 10 Metern abgedeckt werden, allerdings werden dazu 4 Basisstationen der neuen Generation erforderlich sein. Das neue Modell ist kompatibel mit den alten Basisstationen. Dadurch kann das alte Modell durch die neuere Version ersetzt werden, ohne dass gleich in ein komplettes Set investiert werden muss. Zudem wird im Laufe des Jahres ein kabelloser Adapter lieferbar sein, welcher für beide Modelle der „HTC Vive“ benutzt werden kann. Durch diesen Adapter kann in Zukunft ein immersiveres VR-Erlebnis gewährt werden, da der Anwender nicht mehr darauf Acht geben muss, dass sich die Kabel verfangen und sich dadurch freier bewegen.

4 Projektbeschreibung

In diesem Kapitel wird das Projekt in verschiedenen Teilen erklärt.

4.1 Produktentwicklungstheorie

Für die Softwareentwicklung gibt es verschiedene Vorgehensweisen, von denen die Wesentlichen im folgenden Absatz vorgestellt werden. Anschließend wird eine Vorgehensweise für das Projekt definiert.

Das erste Vorgehensmodell ist das Wasserfallmodell, welches durch aufeinander folgende Projektphasen organisiert. Die Phasen haben feste Start- beziehungsweise Endpunkte. Vorteile dieses Modells sind, dass es strukturiert ist und man einfach Meilensteine festlegen kann. Nachteil ist, dass es keine zeitliche Flexibilität bietet.

Extreme Programming ist ein agiles Vorgehensmodell zu Softwareentwicklung, welches es ermöglicht während der Entwicklung schnell auf sich ändernde Anforderungen zu reagieren. Dadurch bietet dieses Modell den Vorteil, dass Risiken einer Fehlentwicklung minimiert werden können. Zudem werden frühe und regelmäßige Tests durchgeführt, damit das Produkt bei Fertigstellung allen Anforderungen des Kunden erfüllt und aus Sicht des Programmierers einwandfrei ist.

Cleanroom Engineering ist eine Entwicklungsmethode, die dem Wasserfallmodell ähnelt. Jedoch werden bei diesem Modell die Phasen von unterschiedlichen Projektteams durchgeführt, wodurch es für diese Softwareentwicklung nicht geeignet ist, da die zu Software von einer Person entwickelt werden soll.

Das Projekt wird nach dem Vorgehensmodell Extreme Programming entworfen, weil der Kunde (in diesem Fall NORKA) zu Anfang des Projektes noch nicht die kompletten Anforderungen im Detail festgelegt hatte und diese sich erst im Laufe der Entwicklung ergeben. Dadurch sind während des Entwicklungszyklus kontinuierliche Rücksprachen mit NORKA notwendig und unnötige Programmierarbeiten und Fehlentwicklungen können vermieden werden.

4.2 Idee

Ziel des Projekts ist auf der einen Seite durch die VR-Anwendung einen Blickfang auf der Messe zu generieren, um die Aufmerksamkeit eventueller Kunden zu erreichen und mit ihnen in Kontakt zu kommen. Auf der anderen Seite soll gezeigt werden, wie eine in 3D geplante Lichtumgebung realisiert werden kann. Dazu soll die 3D Lichtplanung und das realisierte Projekt mit einem HMD betrachten werden können. Das realisierte Projekt soll mit einer 360 Gradkamera gefilmt werden und mit einem kurzen Ausschnitt klarstellen, welchen Umgebungsbedingungen die Leuchten ausgesetzt sind.

Weiterhin soll eine leicht erweiterbare VR-Anwendung erstellt werden, die an potenzielle Kunden geschickt werden kann, um den Kunden eine bessere Visualisierung der Lichtsituation mit den potenziell verwendeten Leuchten zu bieten, als dies bei klassischen Anwendungen der Fall ist. Diese Anwendung soll beim Kunden funktionieren, ohne dass dieser noch zusätzliche Installationen durchführen muss.

4.3 Systemvoraussetzungen

Für das Modell HTC Vive gibt es folgende Mindestanforderungen (Vive, 2017) an die Computer-Hardware:

- Prozessor: gleichwertig Intel Core i5-4590 oder AMD FX 8350, bzw. besser
- Grafikkarte: NVIDIA GeForce GTX 1060 oder AMD Radeon RX 480, oder hochwertiger
- Speicher: 4 GB Random Access Memory (RAM)
- Videoausgang: 1x High Definition Multimedia Interface (HDMI) 1.4-Anschluss oder DisplayPort 1.2 oder neuer
- USB: 1x USB 2.0-Anschluss oder neuer
- Betriebssystem: Windows 7 SP1, Windows 8.1 oder Windows 10

Für dieses Projekt wurde der Computer selbst konfiguriert, damit er die VR-Präsentationen auf Messen in naher Zukunft gewährleistet. Darum wurde als Prozessor der Intel Core i7 7700K, als Grafikkarte die NVIDIA GeForce GTX 1060 mit 6GB Video-RAM, eine SSD mit 250 GB als Speichermedium und ein Arbeitsspeicher mit 16 GB DDR4 verbaut.

Zu Beginn des Projekts, als der vorher beschriebene Computer noch nicht zur Verfügung stand, ist bei längerer Nutzung einer VR-Anwendung von einem Computer mit deutlich älterem Prozessor (i7 960) und mit nur 6GB Arbeitsspeicher aufgefallen, dass Prozessor und Arbeitsspeicher nicht ausreichend dimensioniert waren. Um einen störungsfreien Betrieb zu

gewährleisten, wurde besonders bei diesen Bauteilen die Mindestanforderung deutlich überboten.

4.4 WebVR

WebVR ist eine Spezifikation, die es ermöglicht in virtuelle Realitäten einzutauchen, mit dem Ziel die VR-Erfahrung zu vereinfachen, unabhängig davon, welches Ausgabegerät verwendet wird. So kann eine WebVR-Anwendung nicht nur mit einer VR-Brille, sondern zum Beispiel auch mit dem Smartphone benutzt werden. Ein Nachteil von WebVR ist, dass diese Anwendungen nur mit einer Internetverbindung benutzt werden können. Bei der heutigen, nahezu lückenlosen Mobilfunkabdeckung, ist dieser Nachteil jedoch nicht so gravierend.

4.5 A-Frame

A-Frame ist ein Web-Framework zum Erstellen von VR-Anwendungen. Die Entwicklung dazu kam von Mozilla, mit der Möglichkeit VR-Anwendungen einfacher zu erstellen. Da A-Frame ein Open Source-Projekt ist, hat es sich zu einer der größten VR-Gemeinschaften entwickelt. A-Frame basiert auf Hypertext Markup Language (HTML) und Javascript, bietet allerdings durch das eingebundene three.js eine strukturierte und erweiterbare Basis. Mittlerweile unterstützt A-Frame die Anbindung der am häufigsten genutzten HMDs und zugehörigen Controllern wie Vive, Rift, Daydream, GearVR und Cardboard. Zudem können mit A-Frame auch AR-Anwendungen geschrieben und genutzt werden. Der größte Vorteil von A-Frame ist, dass Anwendungen unabhängig vom verwendeten Gerät genutzt werden können, wodurch eine einfache und schnelle Verbreitung bei den Anwendern erfolgt.

5 Entwurf

5.1 Gestaltung

5.1.1 Infofenster

Das Infofenster soll Informationen zu der jeweiligen Leuchte bieten, um den Nutzer darauf aufmerksam zu machen, welche Eigenschaften die Leuchte erfüllt. Ausgewählte technischen Informationen stehen in jedem Infofenster, um eine Vergleichbarkeit zwischen den Leuchten zu ermöglichen. Zusätzliche Informationen werden pro Leuchte angezeigt, um die speziellen Eigenschaften der jeweiligen Leuchten hervorzuheben. Die Infofenster liegen als Bilder vor und werden vorher entsprechend gestaltet. Jedes Infofenster gibt es in deutscher und englischer Sprache

5.1.2 Verweis

Wenn eine Leuchte angeschaut wird und es mehrere Leuchten des gleichen Typs in derselben Szene gibt, werden diese rot umrandet beziehungsweise rot hinterlegt, damit sofort erkennbar ist, wo die selbe Leuchte und wo andere platziert worden sind.

Für die rote Hinterlegung wird eine Fläche genutzt, die im Bild über der Lampe liegt und normalerweise unsichtbar ist. Diese Fläche dient auch dazu, zu erkennen, wenn auf eine Leuchte fokussiert wird. Dabei muss er nur auf den roten Knopf unten im Bild schauen, um das Auswahlmenü zu öffnen.

5.1.3 Auswahlmenü

Um eine möglichst hohe Interaktivität für den Nutzer zu ermöglichen kann der Nutzer selbst steuern, welche Szene er betrachten möchte.

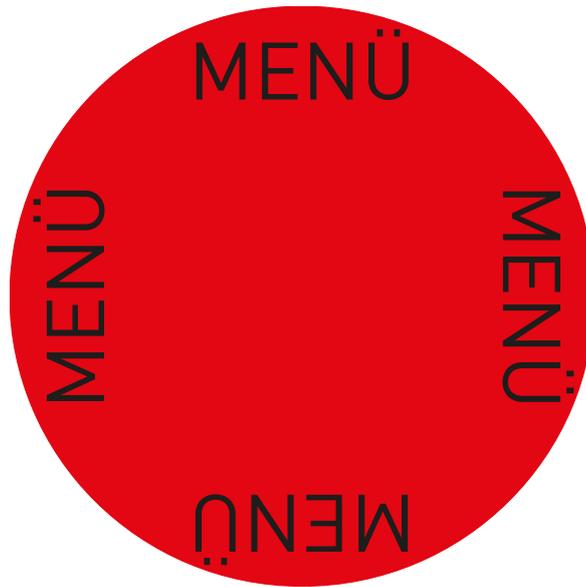


Abbildung 5.1: Knopf zum Öffnen des Auswahlmenüs

In dem Auswahlmenü besteht die Möglichkeit verschiedene Anwendungsgebiete auszuwählen. Die Anzahl der dargestellten Auswahlmöglichkeiten im Auswahlmenü soll der jeweiligen Anzahl der Videos in der VR-Anwendung entsprechen, ohne Anpassungen im Quelltext vornehmen zu müssen. Außerdem gibt es im oberen Teil des Auswahlmenüs noch eine Sprachauswahl, hier werden Deutsch und Englisch angeboten.

Wird ein Anwendungsgebiet ausgewählt, so wird dieses rot hinterlegt, damit der Benutzer sofort erkennen kann, welches der Anwendungsgebiete ausgewählt wurde. Zudem werden zwei zusätzliche Schaltflächen angezeigt, mit denen ausgewählt werden kann, ob die Lichtplanung oder das realisierte Projekt als Video gezeigt werden soll.



Abbildung 5.2: Auswahlmenü mit ausgewähltem Punkt

5.2 Klangwahrnehmung

Damit der Benutzer einen räumlichen Klang hören kann, muss der Klang über mehrere virtuelle Lautsprecher ausgesendet werden. Dadurch wird gewährleistet, dass sich bei Drehung des Kopfes der Ton der Blickrichtung anpasst.

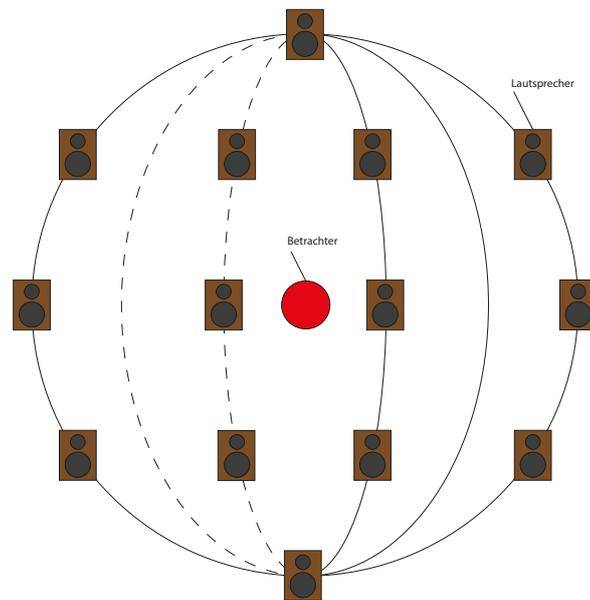


Abbildung 5.3: Anordnungsbeispiel für virtuelle Lautsprecher

Wie in Abbildung 5.3 dargestellt, werden die virtuellen Lautsprecher in Kugelform um den Anwender herum platziert. In diesem Beispiel wurden insgesamt 14 Lautsprecher benutzt. Es können auch weniger oder mehr Lautsprecher benutzt werden, wobei sich der Programieraufwand und das Datenvolumen bei mehr Lautsprechern deutlich erhöhen. Bei weniger Lautsprechern kann es je nach Anwendungsfall dazu führen, dass der räumliche Klang nicht als räumlich wahrgenommen wird.

Bei der Anwendung in der oberen Abbildung ist zu erkennen, dass die Gewichtung des räumlichen Klangs auf eine Rollbewegung des Kopfes gelegt ist.

5.3 Visuelle Wahrnehmung

Um dem Betrachter der VR-Szene die größtmögliche Illusion zu geben, sich an einem anderen Ort zu befinden, muss das Bildmaterial so hochauflösend wie möglich sein. Wie in Kapitel 3.6 beschrieben, bietet das Modell HTC Vive eine Auflösung von 1080 mal 1200 Pixel pro Auge, wobei sich das Bild auf dem linken Auge nur leicht durch eine kleine horizontale Verschiebung von dem auf dem rechten Auge unterscheidet.

Da auf dem Bildschirm nur ein kleinerer Ausschnitt und nicht das gesamte Bildmaterial gezeigt wird, sollte es eine höhere Auflösung haben als das eingesetzte HMD. Dafür wurde

festgestellt, welchen Bereich der Betrachter bei Nutzung einer VR-Brille ungefähr sieht, wozu ein Bild mit einer Auflösung von 12000 mal 6000 Pixeln herangezogen wurde. Davon sieht der Betrachter aber jeweils nur einen Ausschnitt von ungefähr 5000 mal 2800 Pixeln.

$$12000 \cdot 6000 = 72000000 \quad \text{Pixel des Beispielbilds} \quad (5.1)$$

$$5000 \cdot 2800 = 14000000 \quad \text{gesehene Pixel des Beispielbilds} \quad (5.2)$$

$$\frac{72000000}{14000000} = 5,14285714 \quad \text{Faktor} \quad (5.3)$$

$$3840 \cdot 2160 = 8294400 \quad \text{Pixel eines 4K Materials} \quad (5.4)$$

$$\frac{8294400}{5,14285714} = 1612800 \quad \text{gesehene Pixel des 4K Materials} \quad (5.5)$$

$$\sqrt{\frac{1612800}{\frac{16}{9}}} = 952,470472 \quad \text{Höhe des gesehen 4K Materials} \quad (5.6)$$

$$\sqrt{\frac{1612800}{\frac{9}{16}}} = 1693,28084 \quad \text{Breite des gesehen 4K Materials} \quad (5.7)$$

Somit würde der Betrachter bei einem 4K Bildmaterial eine Auflösung von ungefähr 1693 mal 952 Pixeln in der VR-Brille sehen, was minimal unschärfer als Full High Definition (HD) ist. Damit bietet sich 4K Bildmaterial an, da die Abweichung zu der Auflösung der Displays klein ist und das Bild auf dem Wiedergabebildschirm trotzdem scharf dargestellt wird.

8K Bildmaterial würde zwar ein schärferes Bild zeigen, weist aber noch die folgenden Nachteile auf. Erstens ist es aufwendiger und teurer 8K Bildmaterial als 4K Bildmaterial zu erzeugen. Zweitens unterstützen die aktuellen Browser nur 4K Bildmaterial und noch kein 8K Bildmaterial, wodurch die Auflösung auf maximal 4096 mal 2160 Pixel begrenzt ist.

In Zukunft wird eine höhere Auflösung für VR-Anwendungen unumgänglich sein, wenn die Immersion erhöht werden soll.

6 Realisierung

Bei der Realisierung wurde so weit wie möglich darauf geachtet, dass das Projekt zu einem späteren Zeitpunkt von anderen Personen erweitert werden kann, ohne dass große Programmierkenntnisse vorhanden sind. Zudem wurde versucht, den Quellcode so zu schreiben, dass die VR-Anwendung mit unterschiedlichen Anzahlen von Videos (Realisierungen) und Lichtplanungen funktioniert, ohne dass dieser groß geändert werden muss.

Die im Rahmen der Projektarbeit erstellte VR-Anwendung ist auf der Seite <https://vr.norka.de/Bachelorarbeit/anhang/Quellcode/HTML/> zu erreichen.

6.1 Positionsbestimmung einer Leuchte

Die Positionsbestimmung einer Leuchte muss manuell durchgeführt werden und erfordert, dass das Video bereits in den Code eingebunden wurde. Der Benutzer, der die Position einer Leuchte bestimmen möchte, ruft die Website mit einem Browser seiner Wahl auf und wählt anschließend das Video oder die Lichtplanung aus, in denen er eine Leuchtenposition definieren möchte. Anschließend werden die Entwicklertools mit der Taste „F12“ geöffnet und der Mauscursor wird im Video, beziehungsweise der Lichtplanung gedrückt. Nun muss der Benutzer die vier Eckpunkte der Leuchte mit dem Cursor der VR-Anwendung (roter Kreis) ansteuern und mit der Taste „L“ auf der Tastatur jeden Eckpunkt einzeln bestätigen. Hierbei ist die Reihenfolge: unten links, unten rechts, oben rechts und oben links zu beachten. Durch das Drücken der Taste „L“ wird die aktuelle Position des Cursors der VR-Anwendung mittels der Javascript-Datei „Kamera-Position.js“ ausgegeben. Die ausgegebene Cursorposition hat zwei Werte, von denen der erste den Winkel der Nickbewegung des Kopfes darstellt. Dieser Winkel kann von $+90^\circ$, wenn man nach oben guckt, bis -90° , wenn man nach unten guckt, variieren. Der zweite Wert gibt die Gierbewegung des Kopfes als Winkel aus. Wird nach rechts geschaut, wird der Winkel kleiner und schaut man nach links wird er größer. Im Gegensatz zum ersten Wert gibt es beim zweiten Wert kein Minimum und kein Maximum. Danach muss einmal die Enter-Taste gedrückt werden, damit die aktuelle Szene in die Browser-Konsole geschrieben wird. Anschließend sind die letzten 5 Zeilen aus der Konsole der Entwicklertools zu kopieren und in die Excel-Datei „Koordinaten_berechnung_makro.xlsm“ einzufügen, in dem auf den Knopf „Reset und Einfügen“ geklickt wird. Die Excel-Datei berechnet dann die Position, die Höhe, die Breite

und die Rotation des Verweises auf eine Leuchte. Im Anschluss muss in der Excel-Datei in Zelle B7 (grün hinterlegt) noch der Name der Leuchte angegeben werden. Zuletzt muss der gelb hinterlegte Bereich markiert, kopiert und in die HTML-Datei eingefügt werden.

Dieses Verfahren könnte auch in eine Javascript-Datei geschrieben werden, die die HTML-Datei ändert. Dann könnte sich jedoch jeder Benutzer der Seite, der sich mit der Programmierung der Seite beschäftigt, eine eigene Website mit den gleichen Funktionen erstellen. Um das zu verhindern, wurde die Positionsbestimmung in die vorher genannte externe Excel-Datei ausgelagert, die nur von autorisierten Personen geöffnet werden kann.

6.2 Leuchteninformation und Verweis

6.2.1 Informationen über Leuchten

Die Informationen über die angeschaute Leuchte werden unter dieser angezeigt, um die Leuchte nicht zu verdecken. Zudem erfolgt eine Fixierung auf die Leuchte, indem zwei rote Linien von den oberen Ecken der Information zum Mittelpunkt der Leuchte zeigen. Durch die Verwendung der Linien ist es eindeutig, auf welche Leuchte die angezeigten Informationen zutreffen (siehe dazu Abbildung 6.1). Der grau dargestellte Bereich der Abbildung ist in der VR-Anwendung transparent. Die Position der Informationen wird von der Javascript-Datei „Leuchteninfo.js“ gesteuert. Zudem wird die jeweilige Bilddatei mit den Informationen über die Leuchte eingeblendet indem die Transparenz über einen Zeitraum von 1,5 Sekunden erhöht wird. Für die Animation zum Ein- beziehungsweise Ausblenden der Informationen wird die Javascript-Datei „Leuchteninfo_effekt.js“ genutzt.

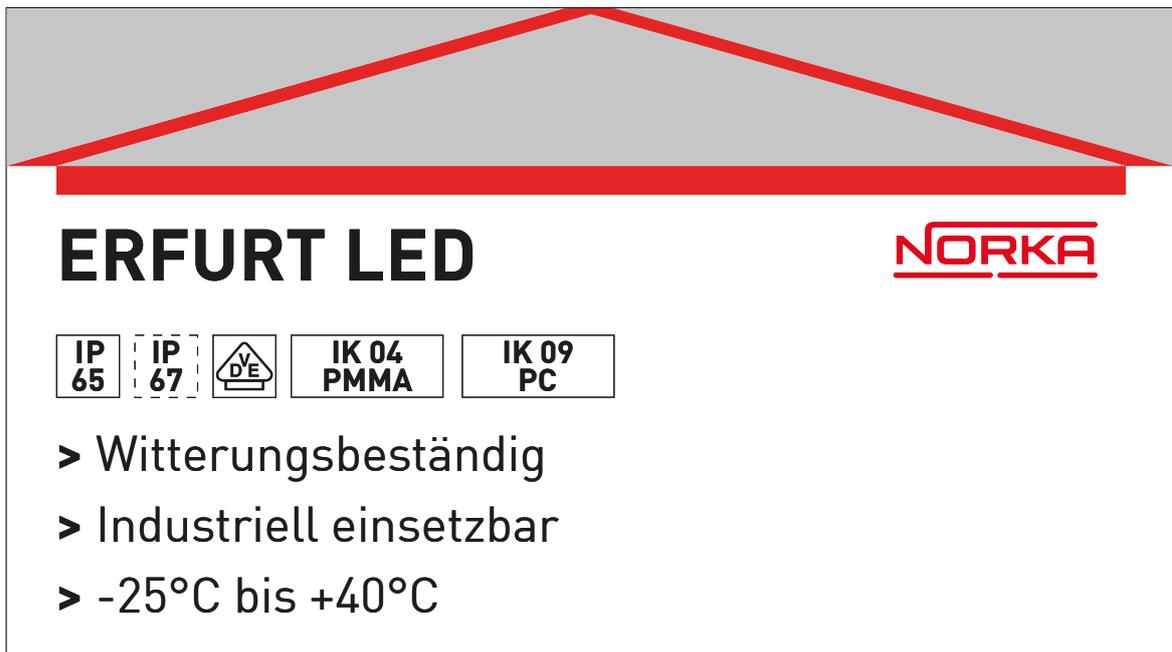


Abbildung 6.1: Informationen mit Verweis auf Leuchte

6.2.2 Verweis auf Leuchten des gleichen Typs

Wird eine Leuchte angeguckt, wird der Benutzer der VR-Anwendung auf die Leuchten des gleichen Typs, die in der aktuellen Szene verbaut sind, hingewiesen. Diese Hinweise liegen als Bild vor und bestehen dabei aus einem Kreis, der nur im äußeren Bereich rot gefüllt ist. Da der Kreis im Inneren durchsichtig ist, verdeckt er die anderen Leuchten nicht.

Die Hinweise auf andere Leuchten werden nacheinander eingeblendet und anschließend in der gleichen Reihenfolge wieder ausgeblendet. Dieser Vorgang wiederholt sich immer wieder und dauert pro Umlauf zwei Sekunden.

Realisiert wurde dieser Vorgang mit der Funktion „outlineZeigen“ (siehe Listing 6.1) in der Javascript-Datei „Leuchteninfo.js“.

```
1 function outlineZeigen(box, outline, zaehler, image, positionInfo,  
    rotationInfo)  
2 {  
3     if(aktuelleLeuchte>=outline.length)  
4     {
```

```
5   clearInterval (andereLeuchtenAnzeigen);
6   aktuelleLeuchte=-1;
7   var interval=dauerOutlineKette/outline.length;
8   andereLeuchtenVerstecken=window.setInterval (
      outlineWegnehmen, interval, box, outline, zaehler, image,
      positionInfo, rotationInfo);
9   }
10  else if (aktuelleLeuchte!=zaehler)
11  {
12    outline[aktuelleLeuchte].setAttribute('visible', "true");
13  }
14  aktuelleLeuchte+=1;
15 }
```

Listing 6.1: Funktion die auf Leuchten des gleichen Modells verweist

6.3 Erstellung eines 360° Panoramas mit Relux

Relux ist eine 3D Software, mit der eine professionelle Lichtplanung für Außen- und Innenbereiche durchgeführt werden kann. Dazu lassen sich die Räume fast beliebig gestalten und 3D-Modelle von den Leuchten der Hersteller einfügen. Anschließend kann eine Falschfarbenansicht erstellt werden, mit der zu erkennen ist, in welchen Bereichen eine gute beziehungsweise schlechte Ausleuchtung gegeben ist. Diese Falschfarbenansicht kann auf dem Desktop durch drehen der Maus in 3D angezeigt werden, jedoch gibt es keine Möglichkeit, dieses 3D-Modell zu exportieren. Deshalb soll für das Projekt eine Lösung zu finden, wie ein Kugelpanorama aus der Relux-Software erstellt werden kann.

Dafür wurden von jeder Szene insgesamt 36 Screenshots gemacht, die sich in 3 horizontale Reihen aufteilen. Dadurch ergeben sich 12 Bilder pro Reihe, was einem Winkel von 30 Grad pro Bild entspricht, womit die Bilder genug überschneidende Fläche besitzen. Bei den Screenshots wurde nur der Aufnahmewinkel und nicht die Aufnahmeposition verändert, sodass ein statisches Bild entsteht. Mit der Software „Adobe Photoshop“ wurden dann zuerst die Bilder der jeweiligen Reihen und danach die fertigen Reihen zu einem gesamten 360 Grad Bild zusammengefügt. Dieser Prozess musste so gewählt werden, da bei dem Versuch alle Bilder auf einmal durch „Adobe Photoshop“ zusammen zu fügen kein verwendbares Ergebnis heraus kam.

Eine bessere Möglichkeit wäre es ein 3D-Modell direkt aus Relux zu exportieren, weil es dadurch möglich wäre, sich in dem Raum zu bewegen und die Lichtplanung aus verschiedenen Winkeln betrachten zu können. Mit Prof. Dr. Greule, dem Mitbegründer von Relux wurde abgestimmt, dass weitere Verbesserungen vorgesehen sind.

6.4 Koordinatenumrechnung

Da die Blickrichtung der Kamera nur zwei relevante Achsen besitzt und die Position in drei Achsen angegeben wird, muss eine Umrechnung erstellt werden, die in diesem Abschnitt beschrieben wird. Als erstes wurde der Radius der Kugel, in der sich der Betrachter befindet auf 900 festgelegt. Man könnte diesen Wert auch kleiner wählen und die eingefügten Elemente kleiner machen, wobei sich dann aber bei einer seitlichen Bewegung des Kopfes das Bild mit bewegen würde. Beliebig groß darf der Wert allerdings auch nicht sein, da der Cursor ab einem Radius der größer als 1000 ist, nicht mehr mit den eingefügten Objekten interagieren kann. Somit gibt es zwischen dem gewählten und dem maximalen Radius eine Sicherheitsreserve, welche einen störungsfreien Ablauf garantiert.

Für die Umrechnung des zweiachsigen Koordinatensystems in das dreiachsige XYZ Koordinatensystem wurde eine Javascript-Datei erstellt. Zuvor wurde dafür ein Formelzusammenhang erstellt, mit dem diese Umrechnung programmiert werden kann.

$$X_{3D} = \sin(Y_{2D}) * 900 \quad (6.1)$$

$$Y_{3D} = \sin(X_{2D}) * \sqrt{900^2 - X_{3D}^2} \quad (6.2)$$

$$Z_{3D} = \sqrt{|900^2 - X_{3D}^2 - Y_{3D}^2|} \quad (6.3)$$

6.5 Darstellung des Auswahlmenüs

Die Anzahl der Schaltflächen zur Auswahl eines Anwendungsgebietes passt sich automatisch an die verwendete Anzahl der Videos an. Dazu wird in der Javascript-Datei „menu.js“ eine If-Else Abfrage verwendet, welche auf die Anzahl der vorhandenen Videos zugreift. Zudem wird bei der Abfrage direkt die Position der einzelnen Anwendungsgebiete gesetzt (siehe Abbildung 6.2).

Dadurch wird gewährleistet, dass die VR-Anwendung auch mit weniger als vier Videos funktioniert, ohne dafür Anpassungen des zuständigen Quellcodes vornehmen zu müssen. Werden beispielsweise nur drei Videos benutzt, passt sich die Position der Auswahl Schaltflächen automatisch an.

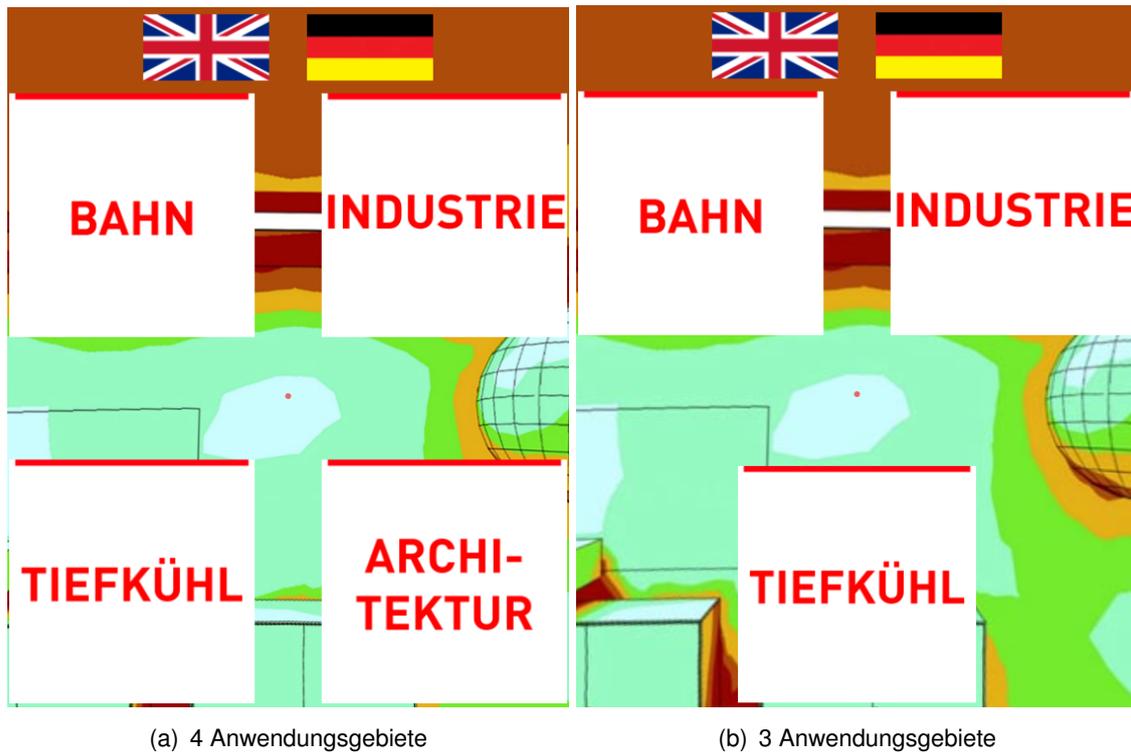


Abbildung 6.2: Darstellung des Auswahlmenüs mit verschiedener Anzahl von Anwendungsgebieten

Die Darstellung des Auswahlmenüs ist in maximal vier Anwendungsgebiete eingeteilt, wobei die Beschriftung der einzelnen Menüpunkte durch Bilder vorgegeben wird. So kann im Nachhinein ein anderer Titel für einen Menüpunkt vergeben werden, ohne den Quelltext anzupassen. Wenn ein Anwendungsgebiet ausgewählt wird, wird dieses rot hinterlegt und die anderen Anwendungsgebiete werden ausgeblendet (siehe Abbildung 6.3). Danach erscheinen zwei neue Schaltflächen, die die Auswahl der zugehörigen Lichtplanung beziehungsweise des zugehörigen Videos ermöglichen. Nach der Auswahl, ob Lichtplanung oder Video angezeigt werden soll, schließt sich das Auswahlmenü vollständig. Die vier Videos und Lichtplanungen, die für die Messe benutzt werden sind zum Abgabzeitpunkt der Bachelorarbeit nicht vollständig. Es sind trotzdem vier Anwendungsgebiete zu sehen, um die Funktionen zu veranschaulichen.



Abbildung 6.3: Realisierung des Auswahlmenüs mit ausgewähltem Anwendungsgebiet

6.6 Klangberechnung

In dem Projekt wurden vier virtuelle Lautsprecher eingesetzt, die sich alle auf der selben Höhe befinden und jeweils einer Vorne, Hinten, Links und Rechts platziert sind. Es wurden vier Lautsprecher gewählt, da die Videoaufnahmen mit 4 Audio-Kanälen aufgezeichnet worden sind. Durch die gewählte Anordnung ist die Veränderung des Tones wahrzunehmen, wenn der Kopf eine Gierbewegung macht. Bei einer Nickbewegung verändert sich der Ton

bewusst nicht, da die Position der Ohren bei einer Nickbewegung gleich bleibt und sich nicht wie bei einer Gierbewegung verändert. Bei vier Lautsprechern wird keine Rücksicht auf Rollbewegungen des Kopfes genommen, dafür wären mindestens noch 2 weitere Lautsprecher, jeweils oben und unten erforderlich. Mehr als vier Lautsprecher erhöhen zudem den Aufwandaufwand da für jeden Lautsprecher ein gesondertes Mikrofon bei den Videoaufnahmen erforderlich ist. Zudem reichen vier verwendete virtuelle Lautsprecher, um den gewünschten räumlichen Klang zu simulieren.

Sollte hingegen eine künstliche Umgebung dargestellt werden, können beliebig viele virtuelle Lautsprecher eingesetzt werden, da keine aufwendigen Aufnahme mit Mikrofonen dafür erforderlich sind.

Für jeden Lautsprecher gibt es zwei Audiodateien, jeweils eine für das linke und das rechte Ohr. Daraus ergeben sich für vier Lautsprecher insgesamt acht Audiodateien für eine Videodatei. Die Lautstärke der einzelnen Audiodateien wird dabei von der Javascript-Datei „Audio-Position.js“ gesteuert. Dabei ist es z.B. möglich, dass für das linke Ohr eine Audio-Datei eine Lautstärke von 100% hat und zwei andere von 50% und die vierte 0% hat oder zwei von 75% und zwei von 25% haben. Dies ist der Fall, wenn das linke Ohr exakt in die Richtung eines virtuellen Lautsprechers zeigt.

6.7 Mobile Geräte

Für mobile Geräte wurden diverse Anpassungen getätigt, um die Performance und die Benutzbarkeit zu verbessern. Hierfür wurde die Javascript-Datei „browsererkennung.js“ erstellt, die erkennt, mit welchem Browser und mit welchem Endgerät die VR-Anwendung aufgerufen wird. Dazu wurden die gängigsten Browser und Betriebssysteme im Quellcode gelistet. Die Funktion „BrowserDetect“ vergleicht den aktuellen Browser mit der Liste. Ist ein Browser nicht in der vorgenannten Javascript-Datei gelistet, wird er als „unbekannt“ ausgegeben. Dies erfolgt auch für das Betriebssystem und die Versionsnummer.

Stehen diese nicht in der Liste, oder der Browser bietet keine oder nicht die volle VR-Unterstützung, wird das Laden der VR-Anwendung abgebrochen und der Nutzer auf eine Website weitergeleitet, die entsprechende Informationen zu Browsern, Versionsnummern und Betriebssystemen enthält, mit denen VR-Inhalte abgerufen werden können (siehe: https://vr.norka.de/Bachelorarbeit/anhang/Quellcode/HTML/browser_inkompatibel.html). Durch diese Weiterleitung erhält der Nutzer zeitnah die Information, dass sein Browser oder Betriebssystem inkompatibel ist und wie das Problem behoben werden kann. Hiermit wurde die Benutzerfreundlichkeit deutlich verbessert.

Zur Verbesserung der Performance wurde die Auflösung der Videos in der Qualität reduziert, um die Größe der Dateien deutlich zu verringern, was auch im mobilen Netz zu

schnellen Ladezeiten führt. Dabei wurde das Original-Video beibehalten und die reduzierte Kopie dem Ordner hinzugefügt. Es ist möglich, dass der Betrachter einen geringen Unterschied bemerkt. Dieser fällt allerdings nicht deutlich auf, da der Bildschirm von einem mobilen Gerät deutlich kleiner als ein Computermonitor ist. Das gleiche wurde für die 360-Grad Bilder gemacht. Insgesamt werden so nur noch 61,8 MB statt 316 MB für die Videos und 1,95 MB statt 9,02 MB für die Bilder, also insgesamt nur ein Fünftel des Datenvolumens heruntergeladen. Die kleineren Dateien werden standardmäßig beim Start der VR-Anwendung geladen, wodurch der Datenverbrauch des mobilen Nutzers gesenkt wird. Wird erkannt, dass kein mobiler Browser genutzt wird, werden die Video-Dateien mit der hohen Auflösung geladen.

Außerdem wurde der Cursor für mobile Endgeräte vergrößert, damit er auf diesen eindeutig sichtbar ist. Auf Grund der kleineren Displays wäre er sonst nur sehr klein und schwer zu erkennen.

Zudem wurden die Töne für mobile Geräte ausgeschaltet, da es bei iOS Geräten seit Version iOS 10 eine Sperre für Autoplay-Inhalte mit Audio-Dateien gibt, weil kein Schreibzugriff vorliegt (Wenz, 2014, S. 399) und die Lautstärke von Audio-Dateien unter iOS nicht im Programm geändert werden kann. Dadurch kann nicht die volle VR-Funktionalität mit den mobilen Geräten ausgenutzt werden. Wer diese nutzen möchte, sollte auf eine an den Computer angeschlossene VR-Brille zurückgreifen, da diese ein immersiveres VR-Erlebnis garantiert.

6.8 Messe Version der WebVR Anwendung

Die Messe „Light + Building“ ist die weltweit wichtigste Messe für Licht- und Gebäudetechnik. Sie wird alle zwei Jahre in Frankfurt am Main ausgerichtet. Bei der letzten „Light + Building“ 2016 gab es insgesamt 2626 Aussteller auf einer Fläche von 249100 Quadratmetern. (Messe-Frankfurt-Exhibition-GmbH, 2018).

Für die „Light + Building“ vom 18. bis 23.03.2018 und für zukünftige Messen wurde eine leicht abgewandelte VR-Anwendung erstellt, welche aber in ihrer Funktionsweise nahezu identisch mit der normale WebVR Anwendung ist. Sie kann unter folgenden Adresse: <https://vr.norka.de/Bachelorarbeit/anhang/Quellcode/HTML/messe.html> abgerufen werden.

6.8.1 Position der Seite

Die Aufgabenstellung war, dass man mit einem Computer, an den das HMD angeschlossen ist, über einen Monitor erkennen kann, wo das aktuelle Sichtfeld des HMD liegt. Dabei war

zu beachten, dass es sich um einen Breitbildmonitor handelt, auf dem keine Ränder oder Scrollbars mehr sichtbar sein sollen. Dafür musste die Position der Website und anschließend der Zoom geändert werden, damit nur noch die Sicht eines einzelnen Auges dargestellt wird. Wenn bei der normalen VR-Anwendung nicht gezoomt und die Position nicht verschoben wird, wird die Ansicht von beiden Augen im Browser nebeneinander im Hochkantformat dargestellt (siehe Abbildung 6.4), wobei man nach oben und unten gesehen jeweils mehr sieht als der Betrachter mit der VR-Brille. Deswegen ist der obere und der untere Rand nicht relevant und wurde abgeschnitten. Die Einstellung der Verschiebung ist dabei abhängig vom verwendeten Browser und den Anzeigeeinstellungen des verwendeten Betriebssystems.

```
1 | <body style="margin:-252px -72px; padding:0">
```

Listing 6.2: Verschiebung des Body für die Messe

Um die VR-Anwendung zur richtigen Ansicht zu bringen, müssen folgende Schritte in der richtigen Reihenfolge durchgeführt werden:

- 1. Website laden
- 2. Zoom auf 90%
- 3. VR-Modus starten (entweder mit Klicken auf das VR-Symbol unten rechts, oder drücken der Taste „F“)
- 4. Vollbildmodus aktivieren mit der Taste „F11“
- 5. „Strg“ und „+“ drücken, bis Zoom auf 195% eingestellt ist (dafür wird bei Firefox das Addon „Zoom Page WE“ verwendet)



Abbildung 6.4: Ansicht beider Augen



Abbildung 6.5: Umsetzung des Wiedergabebildschirms

6.8.2 Erkennung des Browsers

Die Erkennung, welcher Browser genutzt wird, wird auf der Messe nicht benötigt, da die Anwendung nur auf einem stationären Computer mit einem definierten Browser laufen wird. Dabei ist es nicht nötig die vorgenannte Javascript-Datei zu laden, da sie nicht genutzt wird. Da in der normalen Version standardmäßig die Videos in Standard Definition (SD) benutzt werden, muss hier bei jedem Video der Pfad des Videos mit guter Qualität angegeben werden.

6.8.3 Tablet

Die Steuerung mittels Tablet (siehe Kapitel 6.9.1) soll nur auf der Messe erfolgen, da für diese Verbindung ein Server im gleichen Netzwerk benötigt wird. Deshalb wird die Javascript-Datei „tablet.js“, die die Kommunikation mit dem Tablet ermöglicht, nur für die Messe benutzt.

6.9 Externe Steuerung

6.9.1 Tablet-Steuerung

Für die Messe soll die Website von außen steuerbar sein, damit Menschen die keine VR-Erfahrung haben, die verschiedenen Szenen der Lichtplanung und der realen Beleuchtung erkunden können. Die Steuerung soll ohne Kabelverbindung erfolgen, weswegen eine Socketverbindung zu einem „Raspberry-Pi“ (kleiner Einplatinen Computer) genutzt wird. Damit kann die VR-Anwendung extern vom Tablet gesteuert werden, wenn sich der Computer und der „Raspberry-Pi“ im gleichen Netzwerk befinden.

Die Steuerung wird über die Website <https://messe.norka.de/#/vr> aufgerufen.

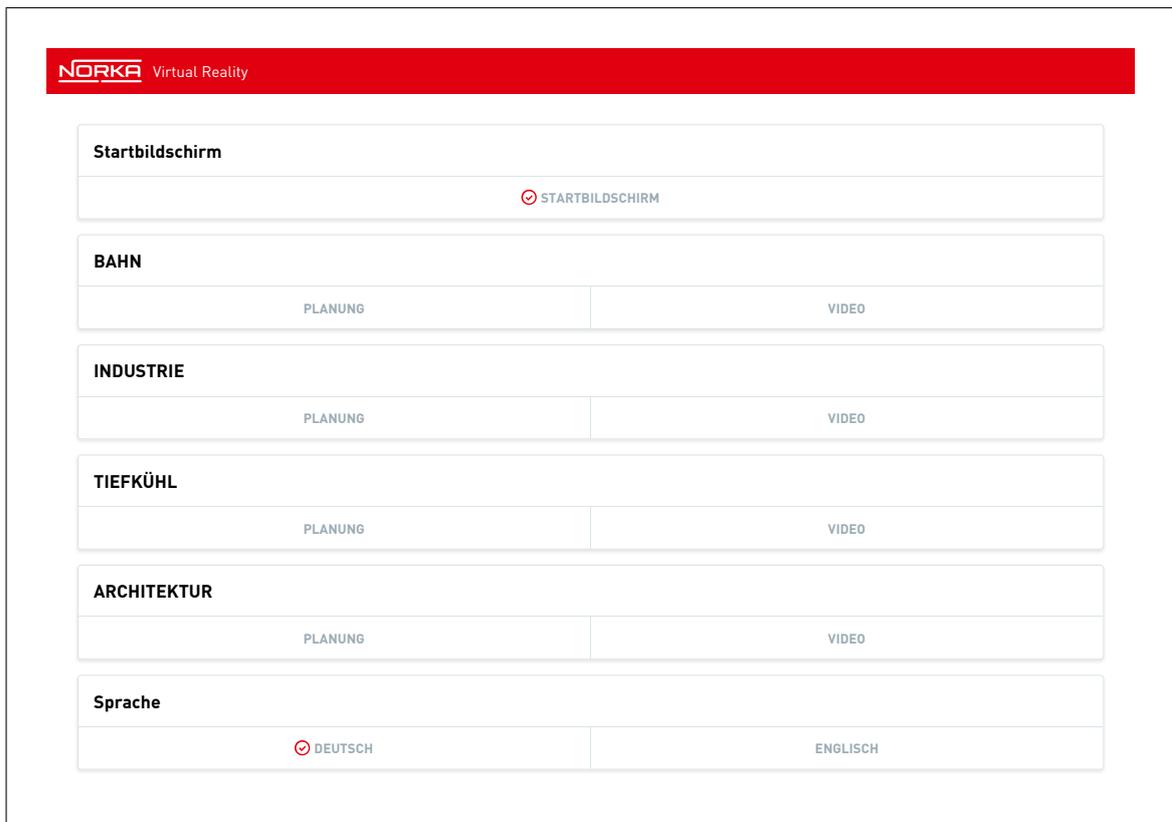


Abbildung 6.6: Ansicht der externen Website zum Steuern

Die Person, die die Steuerung am Tablet vornimmt, sieht auf der Website auf dem Tablet, welche Sprache und Szene ausgewählt ist und kann diese durch Tippen auf dem Tablet ändern. Dabei wird der Haken vor dem ausgewählten Bereich erst gesetzt, wenn die VR-Anwendung den Befehl zum Ändern der Szene gibt. Dadurch wird sichergestellt, dass in der VR-Brille das selbe gesehen wird, wie auf dem Tablet der externen Steuerung. Zudem werden die Änderungen, die der aktuelle Benutzer im Menü tätigt, an die Steuerungs-Website gesendet, um auch hier eine Information zu haben, was gerade ausgewählt ist.

Für die Benutzung der vorgenannten Socketverbindung müssen im Quelltext der VR-Anwendung zwei Zeilen im Head ergänzt (siehe Listing 6.3) und eine Javascript-Datei geschrieben werden, die die Befehle ausgibt, bzw. vom Raspberry-Pi entgegennimmt.

```
1 | <head>
```

```
2 <script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io
  /2.0.4/socket.io.js"></script> <!-- Skript zum Empfangen
  von externen Eingabe-Geräten -->
3 <script src="https://cdnjs.cloudflare.com/ajax/libs/axios
  /0.17.1/axios.min.js"></script> <!-- Skript zum Empfangen
  von externen Eingabe-Geräten -->
4 </head>
```

Listing 6.3: Socket-Verbindungsherstellung HTML

In der Javascript-Datei „tablet.js“ wird die Verbindung unter Vorgabe einer Adresse und einer IP-Adresse hergestellt (siehe Listing 6.4).

```
1 var socket = null ;
2 socket=io('https://norka.lighting:9999/vr').on('vr-control',
  handleRemoteCommand)
```

Listing 6.4: Socket-Verbindungsherstellung Javascript

6.9.2 Tastatur-Steuerung

Als Alternative zur Steuerung via Tablet gibt es eine Steuerungsmöglichkeit über die Tastatur des Computers. Somit kann die Seite auch ohne ein Zweitgerät (Tablet) bedient werden. Die Tastenbelegung ist Tabelle 6.1 zu entnehmen.

Die Tasteneingabe wird von der Javascript-Datei „tastatur.js“ übernommen. Wird eine Taste gedrückt, wird das entsprechende Ereignis ausgelöst. Dadurch reagiert die Anwendung schneller, als wenn die Taste losgelassen werden müsste. Jede Taste hat intern einen bestimmten Wert, anhand dessen die Javascript-Datei mittels If-Abfrage entscheidet, welches Ereignis auszulösen ist (siehe Listing 6.5).

```
1 addEventListener("keydown", tastaturSteuerung, false);
2 function tastaturSteuerung (e) { //Wenn Taste gedrückt wird
3   var tastaturEingabe = e.keyCode;
4   if (tastaturEingabe==77) // m: öffnet das Menü
5     {
6       einblendenMenu();
7     }
8   .
9   .
10  .
11 }
```

Listing 6.5: Erkennung eines Tastendrucks am Beispiel der Taste „M“

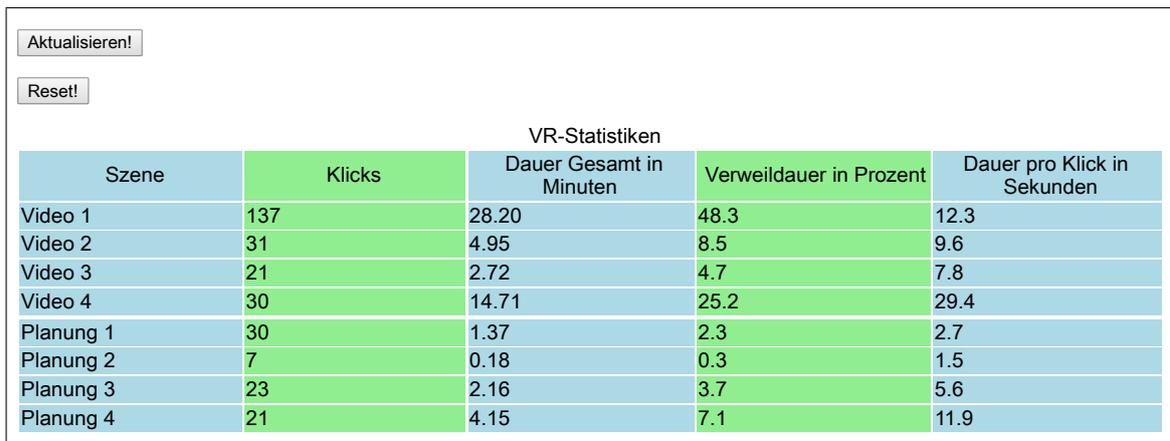
Tabelle 6.1: Tastenbelegung für Tastatursteuerung

Taste	Ereignis
1	Zeigt Planung 1 an
2	Zeigt Video 1 an
3	Zeigt Planung 2 an
4	Zeigt Video 2 an
5	Zeigt Planung 3 an
6	Zeigt Video 3 an
7	Zeigt Planung 4 an
8	Zeigt Video 4 an
W	Zeigt den Wartebildschirm an
M	Öffnet das Menü
C	Schließt das Menü
E	Wechselt die Sprache zu Englisch
D	Wechselt die Sprache zu Deutsch

6.10 Statistische Auswertung der Auswahl der Benutzer

Für die Messe „Light + Building“ in Frankfurt und für zukünftige Messen wurde eine zusätzliche Website erstellt, mit der man feststellen kann, welche Szene wie häufig angeguckt und wie lange dort insgesamt und im Durchschnitt verweilt wurde (siehe <https://vr.norka.de/Bachelorarbeit/anhang/Quellcode/HTML/statistik.html>).

Diese Website greift dabei auf den localstorage des jeweiligen Browsers zu, also kann man diese Website nur zum Auswerten benutzen, wenn man sie von dem Computer abrufen, der auf der Messe steht. Die Statistik-Seite in Abbildung 6.7 ist beispielhaft zusehen. Die gezeigten Daten basieren auf dem Entwickeln und Testen während der Erstellung des Projektes. Wenn auf Aktualisieren geklickt wird, aktualisiert sich der Inhalt der Seite, ohne dass die Seite komplett neu geladen werden muss. Bei Betätigung von Reset werden die gespeicherten Werte alle auf den Wert 0 gesetzt, wobei noch eine zusätzliche Bestätigung erfolgen muss, um auszuschließen, dass der Knopf nicht fälschlicherweise betätigt wurde.



VR-Statistiken				
Szene	Klicks	Dauer Gesamt in Minuten	Verweildauer in Prozent	Dauer pro Klick in Sekunden
Video 1	137	28.20	48.3	12.3
Video 2	31	4.95	8.5	9.6
Video 3	21	2.72	4.7	7.8
Video 4	30	14.71	25.2	29.4
Planung 1	30	1.37	2.3	2.7
Planung 2	7	0.18	0.3	1.5
Planung 3	23	2.16	3.7	5.6
Planung 4	21	4.15	7.1	11.9

Abbildung 6.7: Statistik Website für Messen

Die anonymisierten Daten, die damit gesammelt werden, werden ausschließlich unternehmensintern genutzt und dienen der Erkennung der für die Messebesucher interessantesten Anwendungsgebiete. Aus der Statistik kann abgeleitet werden, womit auf zukünftigen Messen eine größere Anzahl an Interessenten angesprochen werden sollte. Zudem lässt sich aus der Statistik erkennen, welche lichttechnischen Projekte sich nicht eignen, mit VR vorgestellt zu werden.

7 Tests

Während der gesamten Projektphase wurden neu entwickelte Funktionen sofort getestet und vorhandene Fehler Schritt für Schritt eliminiert. Erst wenn eine Funktion fehlerfrei genutzt werden konnte, wurden neue Funktionen entwickelt.

7.1 Testumgebung

Es wurde versucht im Rahmen der Möglichkeiten die gängigsten Betriebssysteme und Browser zu testen, um möglichst eine großflächige Benutzung zu gewährleisten. In Tabelle 7.1 werden die getesteten Systeme aufgeführt. Dabei wurde getestet ob alle Funktionen der VR-Anwendung auf dem jeweiligen Gerät vollständig funktionieren. Wenn das der Fall war, ist der Test erfolgreich gewesen.

Tabelle 7.1: Testsysteme

Betriebssystem	Browser	Browserversion	Test erfolgreich?
iOS 9.3.5	Safari Mobile	9	Nein
iOS 10.3.3	Safari Mobile	10	Ja
iOS 11.2.5	Safari Mobile	11	Ja
Android 6.0.1 (Marshmallow)	Google Chrome	63.0	Ja
Android 7 (Nougat)	Google Chrome	64.0	Ja
Android 8 (Oreo)	Google Chrome	64.0	Ja
macOS 10.13.3	Safari	11.0.3	Ja
Windows 10 Mobile	Edge Mobile	40.15063	Nein
Windows 7 Professional SP 1	Firefox Nightly	60.0	Ja
	Firefox	52.4.0	Ja
	Google Chrome	62.0.3202	Ja
	Internet-Explorer	11.0.9600	Nein
Windows 10 Professional	Edge	17.17	Ja
	Firefox	58.0.1	Ja
	Google Chrome	64.0	Ja

Zudem wurde eine Vielzahl weiterer Geräte virtuell mit den Firefox-Entwicklertools getestet, um eine flächendeckende Ausbreitung zu ermöglichen. Mit diesen Tools können die Bildschirmgrößen und einige Eigenschaften simuliert werden. Dies funktioniert jedoch nur in einem bedingten Rahmen, weshalb nicht für alle Geräte eine Funktionsgarantie gegeben werden kann.

7.2 Funktionstests und Fehlerbehandlungen

Funktionstest überprüfen die korrekte Programmierung der Software. Diese werden grundsätzlich beim Programmieren ausgeführt. Beispiele für die Testfälle:

- Verhalten der Webseite beim Verlust der Internetverbindung, 10 Sekunden nachdem diese geladen wurde.
- Beim Angucken der Leuchte wird der Kopf nicht gerade sondern schräg gehalten.

- VR-Anwendung läuft über längeren Zeitraum mit dem Modell „HTC Vive“ zusammen
- Positionsbestimmung, wenn eine Basisstation blockiert ist
- VR-Anwendung mit langsamen Internet laden
- Szene wird gleichzeitig mit Tablet und Tastatur geändert

Bei den Tests ist aufgefallen, dass bei älteren iPhones, die alle VR-Funktionen unterstützen, die Performance der Seite sehr stark leidet. Dies kann aber durch eine Drosselung der iPhone-Prozessorgeschwindigkeit zur Schonung der Komponenten bedingt sein. Mit neueren Smartphones läuft die VR-Anwendung dagegen sehr flüssig.

Zudem wurde bei den Tests eine Performance Statistik im Browser angezeigt, in der man in Echtzeit die Anzahl der FPS und die Latenz sieht. Die Latenz sollte so gering wie möglich gehalten werden. Wenn die Latenz deutlich ansteigt, ist während des Bedienens der VR-Anwendung zu merken, dass diese deutlich träger wird. Außerdem wurde bei den Tests am Computer mit dem Taskmanager die Auslastung des Prozessors, des Arbeitsspeichers und der Grafikkarte überwacht. Dabei ist aufgefallen, dass bei längerer Nutzung viel Arbeitsspeicher vom System benötigt wird.

Bei einem zweistündigen Dauertest, in Verbindung mit einer sich bewegenden VR-Brille, kam heraus, dass der Prozessor zu maximal 25%, die Grafikkarte mit 95% und der Arbeitsspeicher mit 7 GB belastet wurde. Daraus lässt sich ableiten, dass der Computer einen Dauereinsatz auf Messen problemlos absolvieren wird.

7.3 Anwendertest

Um die Benutzerfreundlichkeit der VR-Anwendung zu gewährleisten wurde ein Anwendertest mit zehn Testpersonen unterschiedlicher Altersstufen durchgeführt. Dieser Test wurde so gestaltet, als wenn die Testpersonen Messebesucher wären, die die VR-Anwendung noch nie benutzt haben. Die Personen mussten folgende Ziele erreichen:

- 1. Das Menü öffnen
- 2. Ein Anwendungsgebiet auswählen
- 3. Eine Lichtplanung oder ein Video öffnen
- 4. Leuchten angucken und dabei die Informationen zu den Leuchten öffnen

Dabei wurde den Testpersonen nur erzählt, was sie machen sollen und sie mussten selbstständig herausfinden, wie sie das erreichen können. Zur Auswertung wurden die Zeiten der einzelnen Ziele gestoppt. Die Auswertung ist in Tabelle 7.2 zu finden.

Tabelle 7.2: Auswertung des Anwendertests in Sekunden

Ziel	Minimum	Maximum	Durchschnitt
1.	5,00	24,00	12,8
2.	3,00	14,00	8,3
3.	2,00	9,00	5,4
4.	4,00	22,00	13,7

Die Testdaten zeigen, dass die verschiedenen Benutzer unterschiedlich lange brauchen um die jeweiligen Ziele zu erreichen. Würde man dabei alle maximalen Zeiten zusammen zählen, käme man auf eine Gesamtzeit von 1 Minute und 9 Sekunden. Allerdings hat sich jeder Nutzer der VR-Anwendung nach dem Testablauf noch länger umgeschaut um die anderen Szenen zu erkunden.

8 Evaluation

8.1 Vergleich mit dem Entwurf

Die Realisierung unterscheidet sich in einigen Punkten vom ursprünglichen Entwurf. Im folgenden werden die Unterschiede aufgezeigt und erklärt, warum Abweichungen zum Entwurf aufgetreten sind.

Als erstes zu erwähnen sind die Verweise auf Leuchten des gleichen Typs. In der Realisierung wurde nicht die ursprünglich angedachte rote Umrandung genommen, weil diese nur mittels A-Frames Entität „line“ sinnvoll hätte umgesetzt werden können. Die Entität besitzt jedoch kein Merkmal, mit der man die Dicke einer Linie verändern könnte. Dadurch sind die Linien zu dünn, um im peripheren Blickfeld des Betrachters wahrgenommen zu werden. Die andere Idee um auf gleiche Leuchten zu verweisen, kommt der Realisierung schon deutlich näher, allerdings gab es ein paar Anpassungen. So wurde beispielsweise die rechteckige Form zu einem Kreis, was in den Tests als deutlich angenehmer empfunden wurde. Zudem wurde nur der äußere Kreisbogen eingefärbt, damit man die Leuchten nicht verdeckt. Dafür wurde die Deckkraft des Verweises erhöht, sodass die runde Umrandung deutlicher zu erkennen ist.

Bei der Realisierung des Auswahlmenüs verschwinden die nicht ausgewählten Anwendungsgebiete, sobald ein Anwendungsgebiet anvisiert wurde. Im Entwurf dagegen war vorgesehen, dass die Anwendungsgebiete geöffnet bleiben. Die Entscheidung für die realisierte Variante ist darin begründet, dass die noch gezeigten Anwendungsgebiete zu wenig Platz bieten und der Nutzer somit sonst leicht nach getätigter Wahl eines Anwendungsgebietes aus Versehen ein anderes auswählt. Das würde die Benutzerfreundlichkeit senken. Damit die Möglichkeit besteht, ein anderes Anwendungsgebiet auszuwählen muss das Menü einfach erneut geöffnet werden. Zudem schließt sich das Menü nach der Auswahl eines Videos oder einer Lichtplanung automatisch, sodass der Anwender direkt in das VR-Erlebnis eintauchen kann.

8.2 WebVR fähige Geräte

Neben Computern können auch Smartphones genutzt werden, um virtuelle Welten zu erkunden. Allerdings werden nicht alle Geräte unterstützt und es kommt auf das Betriebssystem und den verwendeten Browser an. So wird WebVR auf iOS ab Version 10 im vollen Funktionsumfang unterstützt und bei Android ab Version 7.0.

Damit sind aktuell ungefähr 50% der Smartphones WebVR fähig (siehe Tabelle 8.1) (Computerbild, 2018)(Statista, 2017a)(Statista, 2017b), allerdings mit steigendem Trend, da die neuen Android-Versionen an Zuwachs gewinnen .

Tabelle 8.1: Anteil Smartphones

Betriebssystem	Marktanteil	WebVR fähiger Anteil des Browsers			
	November 2017	21.09.2017	11.12.2017	08.02.2018	
Apple	32,38%	94,0%	k.A.	k.A.	30,4%
Android	65,89%	k.A.	23,8%	29,6%	19,5%
WindowsPhone	0,98%	k.A.	k.A.	k.A.	0,0%
andere	0,75%	k.A.	k.A.	k.A.	0,0%
Sume	100,00%				49,9%

Bei dem Betriebssystem Android dauert es erfahrungsgemäß länger bis neue Versionen des Betriebssystems eine hohe Verbreitung erreicht haben. Das liegt daran, dass Android von verschiedenen Herstellern eingesetzt wird und jeder Hersteller individuelle Anpassungen vornimmt. Allerdings findet nach Markteinführung eines neuen Android-Betriebssystems zeitversetzt ein fast linear ansteigender Zuwachs der neuen Version statt (siehe Abbildung 8.1 Reihe Nougat). Aufgrund der vorliegenden Daten ist davon auszugehen, dass sich ein ähnlicher Trend auch für die neueste Version (Oreo) ergeben wird. Somit sollte in naher Zukunft der Anteil der WebVR-unterstützten mobilen Geräte deutlich an Zuwachs gewinnen.

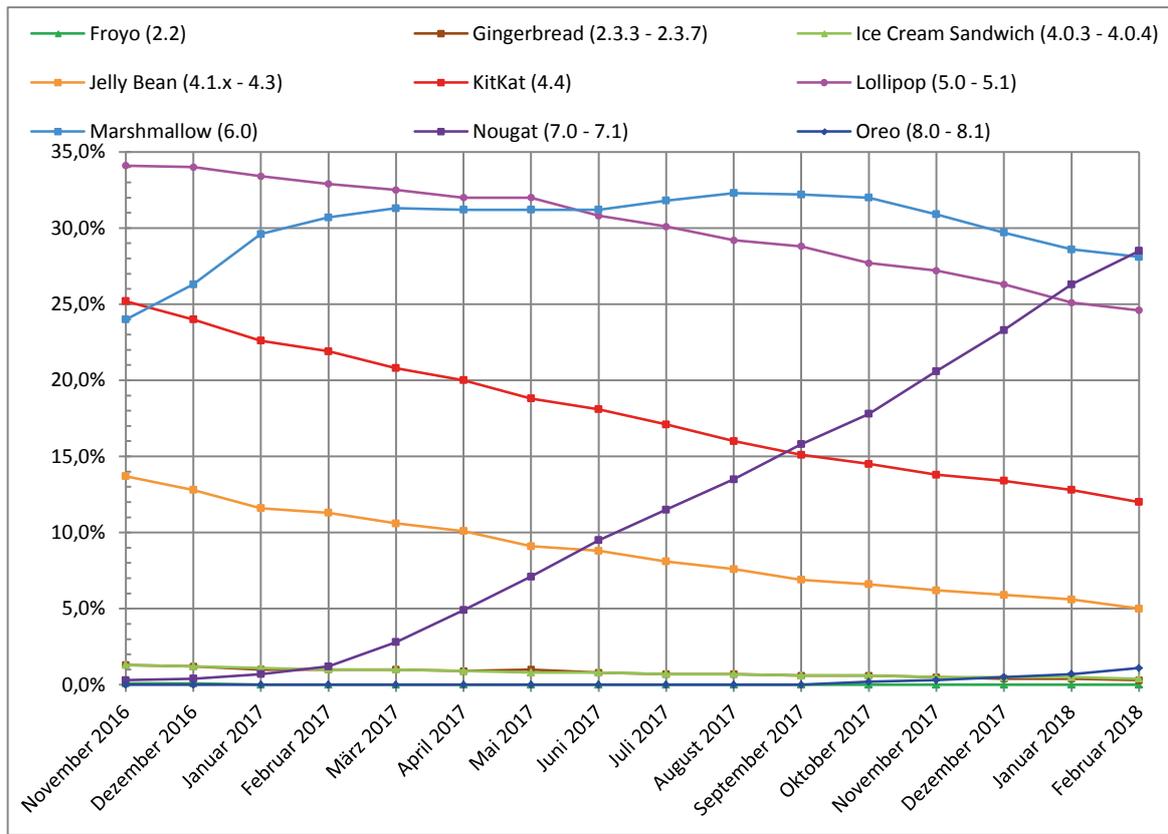


Abbildung 8.1: Verlauf der Android Versionen (Computerbild, 2018)

Die Nutzung einer WebVR-Anwendung im Browser ohne angeschlossene VR-Brille ist entweder mit Chrome ab Version 57, Safari ab Version 10, Firefox ab Version 55 oder Edge ab Version 16 möglich. Für Safari kann keine Aussage gemacht werden, da kein Testgerät zur Verfügung stand.

Mit einem Blick in die Statistiken der Browsernutzung und der Nutzung der verschiedenen Browserversionen lässt sich berechnen, wie viele Computer einen VR fähigen Browser benutzen, das sind ungefähr 74% (siehe Tabelle 8.2)(Papenbrock, 2018).

Tabelle 8.2: Anteil Rechner

Browser	Marktanteil Januar 2018	WebVR fähiger Anteil	
Chrome	36,3%	95,0%	34,5%
Safari	21,2%	92,0%	19,5%
Firefox	21,5%	84,0%	18,1%
Internet Explorer	7,3%	0,0%	0,0%
Edge	3,5%	60,0%	2,1%
Andere	10,2%	k.A.	0,0%
Summe	100,0%		74,1%

8.3 Mögliche Erweiterungen

Das Projekt hat verschiedene Möglichkeiten zur Funktionserweiterung. So wäre es zum Beispiel denkbar, dass die abgespielten VR-Inhalte in stereoskopischer Darstellung gezeigt werden. Das würde bedeuten, dass bei der Benutzung eines HMD unterschiedliche Bilder für das rechte und linke Auge benutzt werden. Dieses Bildmaterial müsste mit einer speziellen Kamera aufgenommen werden, die pro Richtung zwei Linsen besitzt. Durch den Abstand dieser Linsen zueinander entstehen zwei perspektivische Inhalte, die es ermöglichen natürliche Tiefeninformationen durch das HMD wahrzunehmen.

Zudem könnte in einer zukünftigen Version eine Funktion eingebunden sein, die es erlaubt, in einer realisierten oder geplanten Lichtumgebung weitere Leuchten mittels Controllern beliebig zu platzieren. Gleiches wäre für die Software Relux möglich, wenn Relux einen Fokus auf VR setzen würde.

Sobald Relux es ermöglicht, 3D-Modelle zu Exportieren, bestünde die Möglichkeit diese in die vorhandene VR-Anwendung einzubinden. Das würde dem Benutzer ermöglichen sich in der geschaffenen Leuchtenumgebung frei zu bewegen, wodurch er die Lichtsituation besser wahrnehmen könnte.

Eine weitere mögliche Ergänzung ist die Nutzung der Hand-Tracking Hardware von „Leap“. Diese ermöglicht es dem Nutzer VR-Anwendungen mit den Händen, aber ohne Controller zu Steuern. Die Hardware wird vorne auf das HMD montiert und durch ein USB-Kabel mit dem HMD verbunden.

9 Fazit

Kern dieser Bachelorarbeit war die Entwicklung einer intuitiven WebVR-Anwendung welche es ermöglicht, lichttechnische Planungen und die jeweils umgesetzten Realisierungen zu zeigen. Als Entwicklungsmethode wurde Extreme Programming gewählt, da die agile Vorgehensweise Flexibilität bei sich ändernden Anforderungen bietet.

Im Zuge der Umsetzung wurde als erstes eine VR-Anwendung erstellt, in der der Anwender verschiedene lichttechnische Szenen auswählen kann. Anschließend wurden kontinuierlich die gewünschten Funktionen wie zum Beispiel die Positionserkennung von Leuchten und das Einblenden von Informationen zu Leuchten der VR-Anwendung hinzugefügt.

Eine Herausforderung bot die Anpassung an unterschiedliche Geräte, da die verschiedenen Betriebssysteme und Browser unterschiedliche Einstellungen für Video- und Audiodateien haben. Um eine möglichst große Anzahl an Geräten zu unterstützen, waren viele Anpassungen notwendig. Eine weitere Hürde war die Umsetzung des Wiedergabebildschirms auf einem Monitor im Breitbildformat.

Diese Bachelorarbeit zeigt, dass das für die Entwicklung eingesetzte „A-Frame“ ein gutes Werkzeug zur Erstellung von WebVR-Anwendungen ist. Simple Anwendungen, in denen nur ein einzelnes 360 Grad Bild gezeigt wird, lassen sich mit wenig Aufwand erzeugen. Je mehr Funktionen in einer mit „A-Frame“ erstellten VR-Anwendung vorhanden sind, desto aufwändiger ist jedoch die Programmierung. Bei strukturierter Herangehensweise können allerdings selbst HTML und Javascript Neulinge mit vertretbarem Zeitaufwand umfangreichere VR-Anwendungen erstellen.

Im Rahmen des Projekts wurden aber auch die Grenzen von WebVR deutlich. Dazu gehört zum Beispiel die maximale Auflösung von 3840 x 2160 Pixeln sowie die Tatsache, dass zum jetzigen Zeitpunkt noch viel Rechenleistung benötigt wird, um eine hohe Immersion zu erzeugen. Das Potenzial der noch verhältnismäßig jungen Technologie VR ist aktuell noch nicht vollständig ausgeschöpft. Eine mögliche Entwicklung könnte darin bestehen, aus bestehenden 3D-Simulationen auf Knopfdruck eine VR-Umgebung zu erzeugen.

Literaturverzeichnis

- [AUDI 2017] AUDI: *AUDI startet Virtual Reality im Autohaus.* 2017. – URL <https://www.audi-mediacyenter.com/de/pressemitteilungen/audi-startet-virtual-reality-im-autohaus-9270/>. – Zugriffsdatum: 18.02.2018
- [Behrends und Alumni 2016] BEHREND, Jan C. ; ALUMNI: *Duale Reihe Physiologie.* Thieme, 2016. – ISBN 978-3-13138-413-3
- [Beran 2017] BERAN, Robert: *JavaScript im Web 2.0.* Karl Beran, 2017. – ISBN 978-3-7427-0060-5
- [Computerbild 2018] COMPUTERBILD: *Nougat, Oreo: Android-Verteilung im Überblick.* 2018. – URL <http://www.computerbild.de/artikel/cb-News-Handy-Android-Verteilung-Update-Statistik-18821373.html>. – Zugriffsdatum: 18.02.2018
- [Davis 2016] DAVIS, Adam L.: *Modern Programming Made Easy.* Apress, 2016. – ISBN 978-1-4842-2489-2
- [Digi-Capital 2016] DIGI-CAPITAL: *Augmented/Virtual Reality revenue forecast.* 2016. – URL <https://www.digi-capital.com/news/2016/01/augmentedvirtual-reality-revenue-forecast-revised-to-hit-120-billion-by-2020/#.Wldhwn8iFhH>. – Zugriffsdatum: 18.02.2018
- [JP-Performance 2017] JP-PERFORMANCE: *YouTube.* 2017. – URL <https://youtu.be/Need4v1c4u4?t=992>. – Zugriffsdatum: 18.02.2018
- [MDS 2017] MDS: *Virtual Reality - Mit Story-telling raus aus der Nische.* 2017. – URL <https://www.mds.eu/blog/virtual-reality-und-storytelling>. – Zugriffsdatum: 18.02.2018
- [Mehler-Bicher 2014] MEHLER-BICHER, Anett: *Augmented Reality: Theorie und Praxis.* De Gruyter Oldenbourg, 2014. – ISBN 978-3-11-035384-6

- [Messe-Frankfurt-Exhibition-GmbH 2018] MESSE-FRANKFURT-EXHIBITION-GMBH: *Zahlen und Fakten*. 2018. – URL <https://light-building.messefrankfurt.com/frankfurt/de/zahlen-fakten.html>. – Zugriffsdatum: 18.02.2018
- [Papenbrock 2018] PAPENBROCK, Jan: *Browser Statistiken*. 2018. – URL <https://www.browser-statistik.de/>. – Zugriffsdatum: 18.02.2018
- [Sherman und Craig 2003] SHERMAN, William R. ; CRAIG, Alan B.: *Understanding Virtual Reality*. Morgan Kaufmann Publishers, 2003. – ISBN 978-1-55860-353-0
- [Siemens 2018] SIEMENS, Sönke: *Die Zukunft von Virtual Reality: Wo steht VR-Gaming in ein bis zwei Jahren?* 2018. – URL <http://www.pcgames.de/Virtual-Reality-Hardware-258542/Specials/Zukunft-VR-Gaming-in-zwei-Jahren-1248320/>. – Zugriffsdatum: 18.02.2018
- [Statista 2017a] STATISTA: *Marktanteile der mobilen Betriebssysteme in Deutschland*. 2017. – URL <https://de.statista.com/statistik/daten/studie/184332/umfrage/marktanteil-der-mobilen-betriebssysteme-in-deutschland-seit-2009/>. – Zugriffsdatum: 18.02.2018
- [Statista 2017b] STATISTA: *Stolperstart für iOS 11?* 2017. – URL <https://de.statista.com/infografik/11186/verteilung-von-ios-versionen-auf-mobilen-apple-geraeten/>. – Zugriffsdatum: 18.02.2018
- [Statista 2018] STATISTA: *Statistiken zum Thema Virtual Reality*. 2018. – URL <https://de.statista.com/themen/2534/virtual-reality/>. – Zugriffsdatum: 18.02.2018
- [Vive 2017] VIVE: *Vive Deutschland | Erste Schritte mit Vive*. 2017. – URL <https://www.vive.com/de/setup/>. – Zugriffsdatum: 18.02.2018
- [VR-Nerds 2017] VR-NERDS: *Die Geschichte der virtuellen Realität*. 2017. – URL <https://www.vrnerds.de/die-geschichte-der-virtuellen-realitaet/>. – Zugriffsdatum: 18.02.2018
- [WeCon 2018] WECON: *Lichtplanung in Virtueller Realität*. 2018. – URL <http://www.wecon-solution.com/news/lichtplanungvr/>. – Zugriffsdatum: 18.02.2018
- [Wenz 2014] WENZ, Christian: *JavaScript: Grundlagen, Programmierung, Praxis*. Galileo Press, 2014. – ISBN 978-3-8362-1979-2

Anhang

Der Anhang hat folgende Ordnerstruktur:

- Quellcode
 - HTML
 - Javascript
- Dokumente
 - 360Grad_Videos
 - 360Grad_Bilder
 - Audios
 - Leuchteninfos
 - Menue_Auswahl
- Website-Quellen

Die CD mit Anhang ist bei Prof. Dr. Heike Neumann einsehbar.

Abkürzungsverzeichnis

AR Augmented Reality	12
CAVE Cave Automatic Virtual Environment	16
FOV Field of View	11
FPS Frames per Second.....	11
HD High Definition.....	27
HDMI High Definition Multimedia Interface	21
HMD Head-Mounted-Display	11
HTML Hypertext Markup Language.....	22
RAM Random Access Memory	21
SD Standard Definition	39
VR Virtual Reality.....	10

Index

A-Frame, 22

Android, 49

HTC Vive, 19, 26, 46

iOS, 49

Motion-Sickness, 14

NORKA, 10

Relux, 31

Tests, 44

WebVR, 22

Wirtschaftlichkeit, 16

Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 20. Februar 2018

Ort, Datum

Unterschrift