



UNIVERSITÄT  
KOBLENZ · LANDAU

Fachbereich 4: Informatik

# Konzeption und Entwicklung einer AR-Ausmessungsapp

## Bachelorarbeit

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

vorgelegt von  
Valentina Sicilia

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

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

Koblenz, im März 2020

## Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja    Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.       

Koblenz, 11.03.2020

.....  
(Ort, Datum)

.....  
(Unterschrift)

## Zusammenfassung

In dieser Arbeit wird die Konzeption, Implementierung und Evaluierung einer *Augmented Reality*-App beschrieben. Diese wurde mit dem Ziel entwickelt, Objekte im realen Raum mit virtuellen Hilfsmitteln auszumessen, sodass diese Anwendung einen Holzgliedermaßstab ersetzen kann. Hinzu kommt die praktische Speicherung der Messwerte. Angefertigt wurde die App mit der *Unity Engine* und programmiert in C#. Schwerpunkte dieser Arbeit sind die Benutzerfreundlichkeit der App, sowie die Eignung von *AR Foundation* für das Ausmessungstool. Die Anwendung wird auf die genannten Kriterien im Rahmen eines Nutzertests in einer abschließenden Evaluation bewertet. Als Ergebnis ließ sich festhalten, dass sich die AR-App noch im Prototyp-Stadium befindet, aber im Allgemeinen schon als benutzerfreundlich gilt. Kleinere Änderungen sollen und müssen noch vorgenommen werden, um auch den Umgang mit dem AR-Tool zu vereinfachen.

## **Abstract**

This thesis describes the conception, implementation and evaluation of an augmented reality app. This app was developed with the goal to measure objects in real space with virtual tools, so that this application can even replace a folding ruler. In addition, the practical storage of the measured values is included. The app was built with the *Unity Engine* and programmed in C#. The focus of this thesis is the usability of the app, as well as the suitability of *AR Foundation* for the measurement tool. The application will be evaluated against the mentioned criteria in a user test in a final evaluation. As a result, it can be said that the AR app is still in the prototype stage, but in general it is already considered user-friendly. Minor changes should and must still be made in order to simplify the use of the AR tool.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Stand der Technik</b>	<b>2</b>
2.1	Grundlagen . . . . .	2
2.1.1	Augmented Reality . . . . .	2
2.1.2	Verwandte Arbeiten . . . . .	4
2.2	Unity Engine . . . . .	5
2.2.1	Arbeiten mit Unity . . . . .	6
2.2.2	AR Foundation - Eignung und Alternativen . . . . .	7
<b>3</b>	<b>Konzeption der App</b>	<b>10</b>
3.1	Idee . . . . .	10
3.2	Aufbau der App . . . . .	11
<b>4</b>	<b>Umsetzung und Implementierung</b>	<b>14</b>
4.1	C# Programmierung . . . . .	14
4.2	Measure-Tool . . . . .	15
4.3	PlayerPrefs . . . . .	18
4.4	Softwareergonomie der App . . . . .	19
<b>5</b>	<b>Evaluation</b>	<b>20</b>
5.1	Evaluationstest . . . . .	20
5.2	Auswertung und Diskussion . . . . .	22
5.3	Impressionen und Ergebnisbilder . . . . .	29
<b>6</b>	<b>Fazit und Ausblick</b>	<b>31</b>

<b>A Anhang</b>	<b>33</b>
A.1 Fragebogen . . . . .	33
A.2 Testwerte . . . . .	36

# 1 Einleitung

Wie oft kommt es vor, dass man etwas auf die Schnelle ausmessen möchte, aber leider keinen Holzgliedermaßstab in der Nähe hat? Sei es ein Gegenstand, oder sogar eine Räumlichkeit. Heutzutage haben wir mit *Augmented Reality* (AR) die Möglichkeit Virtuelles im realen Raum zu platzieren. Mit diesen virtuellen Objekten, mit denen wir interagieren und unsere Welt erweitern, können wir auch praktische Funktionen für den Alltag schaffen.

Im Rahmen dieser Arbeit wird eine Applikation entwickelt, welche im AR Objekte ausmessen soll. Dies wird ermöglicht, indem der Nutzer durch das Schwenken der Smartphone-Kamera mehrere Objektflächen erfassen kann. In diesen Flächen wird der Nutzer im Stande sein, Maße einer Strecke zu nehmen und diese anschließend nach Belieben abzuspeichern.

Die Motivation hinter dieser Arbeit besteht darin, etwas zu kreieren, was man selbst jederzeit benutzen könnte. Eine App, die im Alltag Gebrauch findet, hat sich diesbezüglich als geeignet erwiesen. So wird die Konzeption und die Implementierung als Hauptschwierigkeit dieser Arbeit eingestuft. Ziel dabei ist es herauszufinden, ob die AR-Anwendung unter dem ausgewählten *Software Development Kit* (SDK) funktionstüchtig ist und ob die App als benutzerfreundlich erachtet werden kann. Dies wird anhand von Nutzertests evaluiert und schließlich im Hinblick auf zukünftige Möglichkeiten beurteilt.

Diese Arbeit lässt sich in sechs Kapitel gliedern. In Kapitel 2 wird der Stand der Technik in Bezug auf bisherige Arbeiten wiedergegeben. Darunter wird erstmal *Augmented Reality* im Kontext eingeordnet und weitere Begriffe erläutert, die eng damit verwandt sind. Exemplarisch für AR werden drei verwandte Arbeiten kurz aufgegriffen. Des Weiteren wird die Eignung der *Unity Engine* bezüglich der Appentwicklung betrachtet. Die Verwendung des ausgewählten SDK's wird anschließend thematisiert und erklärt. Die Konzeption der App wird in Kapitel 3 erläutert, sowie die Idee dahinter. Darüber hinaus wird auf die Umsetzung und Implementierung in Kapitel 4 eingegangen, wobei die Hauptaspekte einzeln vorgeführt werden. Hier werden die Schwerpunkte der App veranschaulichend erklärt. Als nächstes wird die Anwendung mit einem Nutzertest evaluiert (s. Kapitel 5). Die Evaluation zielt darauf ab zu untersuchen, ob die entwickelte AR-App den Ansprüchen der Genauigkeit und Benutzerfreundlichkeit gerecht wird. Abschließend wird sich im Fazit (s. Kapitel 6) unter anderem zu den Perspektiven dieser Anwendung geäußert. Im Anhang befinden sich die Evaluationsbogen-Vorlage, sowie die genauen Messwerte während der Nutzerstudie und die Antworten der Probanden im Rahmen der gestellten offenen Fragen.

## 2 Stand der Technik

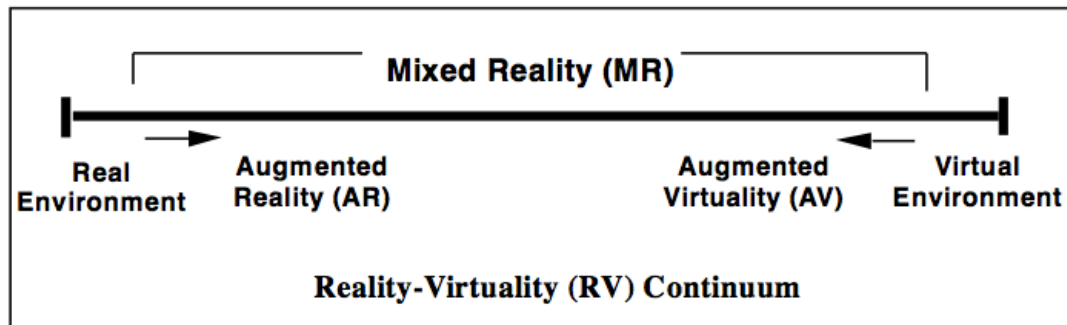
In Folgenden wird der technische Stand vermittelt, indem die benötigten Grundlagen zum Verstehen dieser Arbeit und die Engine, mit der die Anwendung konzipiert wurde, erläutert werden.

### 2.1 Grundlagen

In diesem Kapitel werden die Grundlagen zum technischen Verständnis aufgefasst und erklärt. Der Fokus richtet sich hierbei auf die Definition, Funktion und Entwicklung von *Augmented Reality*, sowie auf verwandte Arbeiten.

#### 2.1.1 Augmented Reality

Der Begriff der *Augmented Reality* steht selten allein im Raum. Meistens bringt man unter Anderem auch Begriffe wie *Virtual Reality* (VR) und *Mixed Reality* (MR) ins Spiel. Diese Begrifflichkeiten erklärte Milgram [PHAF95] im Jahr 1994 anhand seines Realitäts-Virtualitäts-Kontinuums (s. Abb. 1).



**Abbildung 1:** Realitäts-Virtualitäts-Kontinuum (nach Milgram[PHAF95])

Wie auf dem Bild veranschaulicht, befinden sich auf den gegenüberliegenden Seiten einerseits die reale Realität (Real Environment), andererseits die vollkommene Virtualität (Virtual Environment). Der Raum, der sich zwischen der Realität und Virtualität befindet, wird als MR[SH16] bezeichnet. Dazu gehören AR und VR, die lückenlos ineinander übergehen. Bei der *Augmented Reality* werden virtuelle Objekte im realen Raum dargestellt, während *Virtual Reality* mit einer generierten Welt die reale Umgebung ersetzt. Laut Schart und Tschanz[ST18] ist *Mixed Reality* auch als hybride Realität zu verstehen. Diese wird also als Erweiterung der *Augmented Reality* betrachtet, welche optisch reale Objekte zur Interaktion im realen Raum generieren. Die genaue Definition von AR beinhaltet nach Azuma[Azu97] drei Charakteristika:



Charakteristik 1: Die reale Welt wird mit virtuellen Elementen kombiniert.

Charakteristik 2: Die Interaktion geschieht in Echtzeit.

Charakteristik 3: Die Erkennung der Nutzerposition läuft über drei Dimensionen, sodass die generierten Objekte sich im gleichen Koordinatensystem, wie das der realen Welt, befinden.

Es geht also nicht nur darum digitale Elemente in der Welt zu platzieren, sondern auch mit diesen in der erweiterten Wahrnehmung ständig zu interagieren und daraus Daten zu erfassen. Da letzteres nicht ganz auf das AR-Spiel *PokémonGo* trifft, wird die App von den Experten eher als "ortsbasiertes Entertainment" bezeichnet [Mer16]. Trotzdem hat *Augmented Reality* seit dem Release des Spiels, im Juni 2016, eine stetige Entwicklung erfahren. Durch dieses Begeisterung erregende Spiel - was zur Anfangszeit von hunderten von Millionen Menschen heruntergeladen wurde und die Download Charts zum Explodieren brachte [Rau17] - wurde AR weit verbreitet. *Augmented Reality* ist seitdem nicht länger etwas, womit sich ausschließlich Großunternehmen, Forscher und Gelehrte beschäftigen. AR ist nun etwas, was jede Person kostenlos mit sich führen und eigenständig erfahren kann, selbst wenn *PokémonGo* eine reduzierte Variante der tatsächlichen AR ist.



**Abbildung 2:** *PokémonGo* (eigene Abbildung)  
Hier wird ein *Pokémon* im realen Raum angezeigt.

Dies hatte zur Folge, dass viele Entwickler und Unternehmen damit anfangen viel mehr Zeit und Geld in AR zu investieren, da die Gesellschaft nun für diese

Technik bereit zu sein schien. So wurden, um diese Technik nutzbar zu machen, immer mehr Hardware-Geräte produziert, wie z.B. die *HoloLens* (s. Abb. 3), *Samsung Gear VR* [SAM19] und *MagicLeap* [Spa20], welche allesamt *Mixed Reality*-Brillen sind.



**Abbildung 3:** *HoloLens* (von *Microsoft* [Mic20])  
Hier sieht man eine beispielhafte Anwendung einer MR Brille.

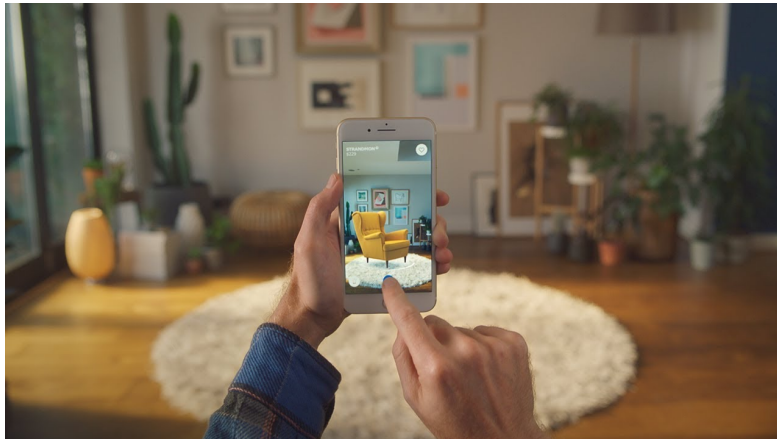
Auch *Apple* setzte sich über einem längeren Zeitraum mit der AR/VR-Technologie auseinander. So wurde schon in 2017 das *ARKit* (s. Kapitel 2.2.2) präsentiert[ST18].

### 2.1.2 Verwandte Arbeiten

Smartphone Applikationen gibt es heutzutage unzählig viele. Fokussieren wir uns also im Folgenden auf AR-Anwendungen. Exemplarisch werden AR-Apps von drei Großkonzernen vorgestellt.

**IKEA Place by Apple** Als das oben genannte *ARKit* präsentiert wurde, war dies in Kollaboration mit *IKEA*. Hierbei hat der Nutzer die Möglichkeit *IKEA*-Möbelstücke im realen Raum zu platzieren [IKE20]. Positioniert man also ein Möbelstück in einem Wohnraum, so kann man beispielsweise schauen, ob die Farbe des Möbelstücks mit der Farbe der anderen Möbel harmoniert. Auch die Größe des Gegenstandes kann man hier im Verhältnis zum Raum setzen(s. Abb. 4). Dieser Schritt sicherte dem schwedischen Einrichtungskonzern das Image eines innovativen Unternehmens.

**AR Filter by FACEBOOK** Auch *Mark Zuckerberg*, der Gründer von *Facebook* und Besitzer von *Instagram* und *Whatsapp*, setzt auf die AR Technik, indem er im Jahr 2017 die *Camera Effects Developer Platform* präsentierte[ST18]. Damit können *Facebook*- und



**Abbildung 4:** *IKEA Place* Applikation (*IKEA*-Werbespot [IKE20])

*Instagram*-Nutzer sowohl AR-Filter einfacher entwickeln, als auch nutzen. Das AR-Studio von der *Facebook*-Gruppe ist auf eine genaue Positionserkennung spezialisiert, sowie auf das Objekt-*Tracking* und die Tiefeninformationen. Das Innovative daran ist, dass z.B. normale *Instagram*-Nutzer ohne jegliche Informatik-Kenntnisse ihre eigenen AR-Filter erstellen und nachdem diese von *Instagram* geprüft und genehmigt worden sind, ihrer *Community* kostenlos zur Verfügung stellen können. Dadurch erreichten die *Social Media* Seiten, die eine sehr große Reichweite haben, wieder einen Meilenstein für die Verbreitung von AR im Alltag. Diese Filter können entweder durch die Gesichtserkennung etwas abbilden oder einen Filter auf die ganze Umgebung setzen. Die Beliebtheit dieser selbst-erzeugten AR-Masken zeigt sich besonders seit 2019. Seitdem nutzten über eine Milliarde Menschen diese AR-Anwendung[Bog19].

**Measure by GOOGLE** *Google* veröffentlichte 2018 eine Maßband-App, die auf der *ARCore*-Plattform [Goo18] basiert. *Google* wirbt damit, mit *Measure* auf die Nutzung eines Maßbands verzichten zu können. Da die Nutzer der App, wie sie selbst in den Bewertungen im *Playstore* schreiben, eher unzufrieden sind, weil das Versprechen der Genauigkeit nicht eingehalten werden kann, spiegelt sich dies in der Bewertungsskala wider. Die App wurde bis heute über 1 Mio. Mal heruntergeladen und über 6.700 Mal bewertet. Die Durchschnittsbewertung beträgt aktuell 3.7 von 5 Sternen [App20b]. Insgesamt fand die App bis Dato nicht genug Anklang.

## 2.2 Unity Engine

Als Entwicklungs-Engine wurde die Spiele-Engine *Unity* benutzt, welche im Folgenden näher erläutert wird. Daraufhin werden die verschiedenen AR Möglichkeiten beleuchtet.

## 2.2.1 Arbeiten mit Unity

*Unity* ist als Spiele-Engine aufgrund der vielfältigen Einsatzmöglichkeiten bekannt und beliebt. Das *Unity Framework* erlaubt dem Entwickler Anwendungen und Spiele für diverse Plattformen zu entwerfen. Nicht nur für Spielkonsolen, wie *Nintendo Switch* oder *PlayStation 4*, sondern auch für alle PC-Betriebssysteme, sowie mobile Betriebssysteme von *IOS* bis hin zu *Android*. Auch bestimmte Webbrowser-Anwendungen lassen sich über *Unity* programmieren. Der *Unity Editor* bietet neben den ganzen Endgerätplattformen auch wichtige technische Eigenschaften. Die Grafik-Engine beispielsweise basiert auf OpenGL und verwendet das *Deferred Shading*-Verfahren [Tec20a]. Dazu werden auch diverse Beleuchtungseffekte unterstützt, die durch eigene *Shader* erweitert werden können. Zudem ist das Einbauen von Animationen in *Unity* auch möglich. Möchte man z.B. physikalische Kräfte einbauen, so kann man dies über die sogenannte *PhysX-Engine* [NVI20] vornehmen. Auch Hintergrundmusik und Geräusche sind integrierbar, um auditiv die richtige Stimmung zu erzeugen. *Unity* stellt *GameObjects* und visuelle Komponenten bereit, mit denen man interagieren kann. Diese sind durch einfaches Binden selbstgeschriebener Skripte modellierbar. Außerdem gibt es im *Asset Store* unzählige fertige Pakete zu den verschiedensten Spiele- und App-Themen, vor allem im Bereich Design. Diese Pakete lassen sich mit zwei Klicks in das aktuelle Projekt importieren. Was aber wichtig für diese Arbeit ist, ist die Verknüpfung zum AR. Auch hier bietet *Unity* verschiedene Möglichkeiten an, welche im nächsten Kapitelabschnitt 2.2.2 genauer erläutert werden.

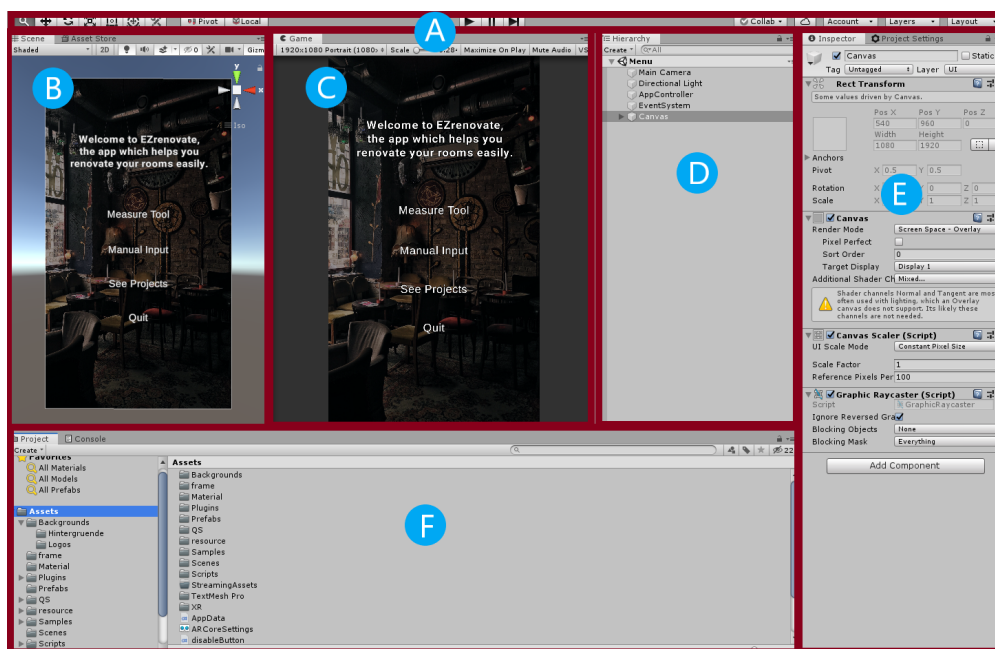


Abbildung 5: *Unity* Arbeitsfläche (eigene Abbildung)

Die Arbeitsfläche des *Unity Interfaces* (s. Abb. 5) lässt sich in sechs Abschnitte gliedern. Diese sind beliebig verschiebbar, sodass die Anordnung individuell angepasst werden kann [Tec20c].

**A:** Die *Toolbar* bzw. Symbolleiste - Hier befinden sich die wichtigsten Arbeitsfunktionen. Auf der linken Seite findet man wesentliche Funktionen zur Skalierung und Drehung der Szene oder des gewählten Objekts. Mittig findet man die Abspiel- und Pause-Taste, welche das Testen der Anwendung ermöglichen.

**B:** Die *Scene View* - Die Szenenansicht ermöglicht dem Entwickler die visuelle Navigation und Bearbeitung der Szene. Als Alternativfenster steht daneben der *Asset Store* bereit.

**C:** Die *Game View* - Die Spielansicht ermöglicht dem Entwickler die Simulation des Spiels bzw. der Anwendung durch die Szenenkamera zu sehen.

**D:** Die *Hierarchy* - Das Hierarchiefenster ist die strukturierte Textdarstellung aller *GameObjects* in der Szene.

**E:** Der *Inspector* - Das Inspektor-Fenster ermöglicht dem Entwickler die Einsicht in allen Eigenschaften des ausgewählten *GameObjects*.

**F:** Das *Project Window* - Das Projektfenster ist die Bibliothek aller importierten und erstellten *Assets*.

Insgesamt lassen sich durch all die genannten Möglichkeiten eine Vielzahl an Objekten in *Unity* erstellen, welche sich durch selbstgeschriebene Skripte ergänzen und erweitern lassen. Dem kreativen Fluss sind also kaum Grenzen gesetzt.

### 2.2.2 AR Foundation - Eignung und Alternativen

**AR Foundation** ist ein *AR Framework*, welches alle wesentlichen Funktionen aller Plattformen enthält. Somit ist das einmalige Entwickeln und das einfache Konvertieren für andere Endgeräte sicher gestellt. *AR Foundation* [Tec20e] beinhaltet Funktionen aus den folgenden vier Paketen (s. Abb. 6): *ARKit* [App20a], *ARCore* [Goo19], *Magic Leap* [Spa20] und *HoloLens* [Mic20]. So muss man beispielsweise eine *Android*-App für *IOS* nicht neu erstellen, sondern es reicht die Pakete einfach zu aktualisieren und etwas anzupassen, womit die Integration schon gewährleistet wird.

### Unity's AR Foundation Supported Features

● Supported  
● Pending

Functionality	ARKit	ARCore	Magic Leap	HoloLens
Pass-through video	●	●		
Device tracking	●	●	●	●
Raycast	●	●	●	●
Plane tracking	●	●	●	●
Reference points	●	●	●	●
Point cloud detection	●	●	●	
Gestures			●	●
Face tracking	●	●		
2D image tracking	●	●	●	
3D object tracking	●			
Environment probes	●	●	●	
Meshing			●	●
2D & 3D body tracking	●			
Human segmentation and occlusion	●			
Collaborative participants	●			

**Abbildung 6:** Schnittstellen der *AR Foundation* (Unity Seite [Tec20e])  
*ARKit* und *ARCore* haben vieles gemeinsam, wobei *ARKit* erweiterte Funktionen hat.

**ARKit:** *Apple* brachte in 2017 ein SDK heraus, was Entwicklern erlaubte AR-Anwendungen für *iPhones* und *iPads* zu erstellen.

**ARCore:** 2018 zog *Google* nach und veröffentlichte *ARCore*, um besagte AR-Anwendungen auch für *Android* Endgeräte zu ermöglichen.

**MR-Brillen:** Die *Magic Leap* und die *HoloLens* bieten im *AR Package* zusätzliche Anpassungen für die entsprechende *Mixed Reality*-Brille.

*AR Foundation* stellt somit diverse Programmierschnittstellen zur Verfügung. Dazu zählen unter Anderem das *World Tracking* und die *Plane Detection*. Durch das *World Tracking* wird die Position und die Orientierung des Geräts im physischen Raum verfolgt. Die *Plane Detection* erkennt horizontale und vertikale Ebenen. Diese Aspekte waren letzten Endes fundamental für die Wahl der Benutzung von *AR Foundation* für die angehende AR-Ausmessungsapp.

**ARCore** Der Fokus dieser Arbeit richtet sich auf das Zwischenspiel mit *ARCore*, welches für AR-Anwendungen auf *Android* Geräten geeignet ist. Um die Arbeitsweise dieses SDK's zu verstehen, wird auf die verschiedenen Arbeitsaspekte[Lan18] von *ARCore*

eingegangen. Diese sind das Bewegungstracking, die Umgebungserkennung und die Lichtabschätzung.

Beim **Bewegungstracking** ist es essenziell wichtig die aktuelle Position und die Bewegung zu erkennen. Hierfür werden visuelle Merkmalspunkte mithilfe der mobilen Kamera identifiziert und im Rahmen der aktuellen Nutzung gespeichert, sodass das gleiche Objekt nicht immer wieder erneut gescannt werden muss. Dies passiert durch die Verfolgung der Merkmalspunkte in Echtzeit, wie in Abbildung 7 veranschaulicht.



**Abbildung 7:** Merkmalspunktverfolgung in *ARCore* (nach Micheal Lanham[Lan18])  
Hier werden bestimmte Punkte der Couch als Merkmalspunkte identifiziert.

Des Weiteren muss *ARCore* auch die **Umgebung** an sich verstehen, um einzelne Ebenen diverser Objekte identifizieren zu können. Dies geschieht durch das sogenannte *Meshing*. Das heißt, dass aus der dreidimensionalen Wahrnehmung automatisch die Oberfläche durch Primitive vereinfacht wiedergegeben wird, ohne die vorherige Rekonstruktion dieser durchzuführen.

Die **Abschätzung der Lichtverhältnisse** ergibt sich vor allem aus der Kombination der zwei vorherigen Punkte. Durch das Tracken der Nutzerposition und durch die Umgebungserkennung übermittelte Objektposition kann das ambientale Licht erkannt und zugeordnet werden. So hat *ARCore* die Möglichkeit virtuelle Objekte mit einem real wirkenden Schatten oder Lichtpunkt zu versehen.

Insgesamt arbeitet *ARCore* ohne Sensoren oder weitere Kameras. Es genügt das handliche Smartphone, was die Arbeit mit diesem SDK sehr attraktiv macht.

**Alternativen** *Unity* selbst kann man optional auch als Bibliothek [Tec20h] verwenden. Das heißt, dass man bereits entwickelte Apps um die *Unity* AR-Funktionalitäten ergänzen kann. Da es aber in dieser Arbeit um die eigenständige Konzeption und Entwicklung einer App geht, scheidet diese Option aus.

Der wohl größte Konkurrent ist die *Vuforia AR Engine* [PTC20b], welche als die meist verbreitete Plattform für AR Entwicklungen gilt. *Vuforia* arbeitet mit Markern, wor-

unter auch das *Object Target* und das *Ground Plane* fallen [PTC20a]. Dadurch besitzt das Object Target einen äußerst großen praktischen Nutzen. Mit einer externen Anwendung wird das Objekt von allen Seiten gescannt, wodurch eine sehr annähernde Form nachgebildet werden kann. Diese wird dann für das *Tracking* des Objekts genutzt. Das *Ground Plane* von *Vuforia* kann allerdings genau die Hälfte von der *Plane Detection* der *AR Foundation*, denn nur die horizontalen Oberflächen werden erkannt. Letzteres führte zum Ausschluss der *Vuforia Engine*.

Im Laufe des Jahres werden zwei weitere Alternativen hinzu kommen: Zum einen wird **MARS** [Tec20f] demnächst erscheinen, was für *Mixed and Augmented Reality Studio* steht. Dieses Studio soll als Erweiterung durch Begleit-Apps ein dynamisches Erstellen von Anwendungen ermöglichen und dies mit nur wenig bis gar keiner spezifischen Programmierung. Zum anderen wird **Instant-AR** bald veröffentlicht. *Unity* verspricht damit schnelle Klein-Anwendungen und -Spiele zu erstellen. Mehr Informationen wurden bisher noch nicht preisgegeben. Man kann sich lediglich für die Beta-Version registrieren [Tec20g].

### 3 Konzeption der App

Die Konzeption der App fing umzugsbedingt mit einer Idee an. Die Struktur musste nach und nach mit dem Heranreifen der Anwendung angepasst werden. Der endliche Aufbau wird im Folgenden erklärt.

#### 3.1 Idee

Die Idee hinter der entwickelten App ist es, den Renovierungsvorgang eines Raumes oder Hauses zu vereinfachen. Dazu soll man anhand eines AR-Tools jegliche Art von Objekten oder Ebenen ausmessen und sinnvoll speichern können, sodass man beispielsweise beim Einkauf in einem Baumarkt darauf zugreifen kann, um die benötigten Werte einzusehen. Ursprünglich war sogar geplant anhand der Werte Räumlichkeiten als Simulation nachzubilden und mit gewählten Möbelobjekten diverse Kompositionen generieren zu lassen. Auf letzteres musste aber im Rahmen des Bachelorarbeitszeitplans vorerst verzichtet werden.

So verlagerten sich die Schwerpunkte auf das AR-Ausmessungstool, sowie auf die Benutzerfreundlichkeit und Software-Ergonomie der Applikation. Die App trägt den Namen *EZrenovate* (gesprochen: easy renovate), welcher nicht nur dem Nutzer zeigt, wie einfach man die gebrauchten Werte entnehmen kann, sondern auch die ständige Abrufbarkeit betont, was diese App alltagstauglich macht.

Ziel dieser Arbeit ist es zu überprüfen, ob die genannten Punkte für den Nutzer ansprechend gestaltet sind. Des Weiteren wird auch *AR Foundation* auf die Probe gestellt.



Letztendlich soll festgestellt werden, ob die Ansprüche hinreichend erfüllt werden können, sodass der Nutzer selbst die App privat gebrauchen würde.

### 3.2 Aufbau der App

Beim Öffnen der App erscheint über zwei Sekunden das Splash-Image (s. Abb. 8) mit dem Logo von *Unity* und dem Logo der App.

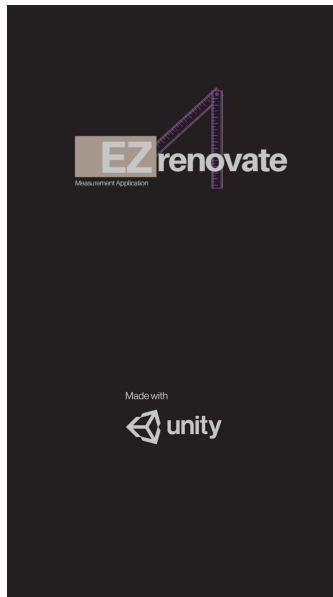


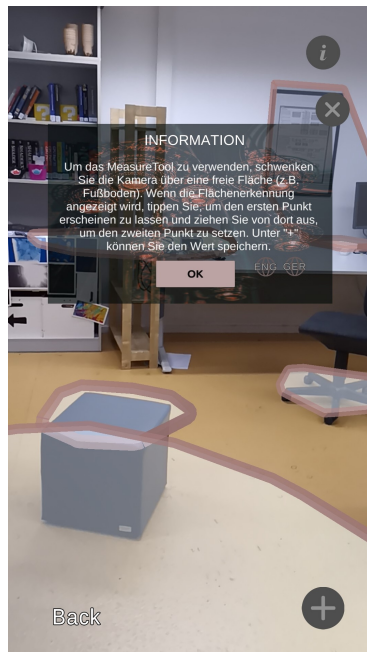
Abbildung 8: Splash Image, *EZrenovate* (eigene Abbildung)

Hier befindet man sich nun im Hauptmenü der App (in Abbildung 9), welches vier mögliche *Buttons* beinhaltet: *Measure Tool*, *Manual Input*, *See Projects* und *Quit*.

Beim Betätigen des **Measure Tools** öffnet sich die Kamera des mobilen Endgeräts. Dabei gibt es in den Ecken des Kamerafensters auch weitere *Buttons*. Der *Info Button* öffnet ein Pop-Up-Fenster (s. Abb.10), welches den Nutzer darüber informiert, die Kamera über einer Ebene leicht zu schwenken, sodass die Fläche erkannt wird. Folgt man der Anweisung, fängt die Flächenerkennung an sichtbar zu werden. Je mehr man mit der Kamera aufnimmt, desto genauer und spezifischer werden die Flächen.



**Abbildung 9:** Main Menu, *EZrenovate* (eigene Abbildung)  
Startbildschirm mit den grundlegenden Funktionen



**Abbildung 10:** Infobox im *Measure Tool*, *EZrenovate* (eigene Abbildung)  
Die Flächenerkennung läuft parallel, während der Nutzer die Information liest. Die Anweisung wurde auf Deutsch umgestellt.

Innerhalb dieser erkannten Flächen ist der Nutzer in der Lage einen Punkt zu setzen. Zieht der User diesen von dort aus zu einer anderen Stelle, so erscheint während des Prozesses sowohl der Endpunkt, eine verbindende Linie und ein Schriftzug, der sich bei jeder Bewegung neu anpasst (s. Abb. 22). Letzteres gibt die aktuell ermittelte Maße aus. In der unteren Ecke auf der rechten Seite befindet sich ein Plus-Symbol (s. Abb.11). Dieses weist den Nutzer darauf hin, die genommenen Daten über den nun erscheinenden *Save-Button* speichern zu können.



**Abbildung 11:** *Savebox* im *Measure Tool*, *EZrenovate* (eigene Abbildung)  
Dem Nutzer wird der Speichervorgang erklärt.

Wenn er dies betätigt, so wird er wieder zum Hauptmenü weitergeleitet und die Werte werden zwischengespeichert. Der Nutzer kann nun - je nach dem - seine Daten vervollständigen und anschließend an einer bestimmten Stelle speichern oder letzteres direkt tun.

Öffnet man das **Manual Input** (s. Abb. 19), so hat man die Möglichkeit Daten sowohl manuell einzugeben, als auch die gespeicherten AR-Werte nun zu importieren. Das zwischengespeicherte Objekt, welches Höhe, Breite und Tiefe beinhaltet, darf auch benannt werden. Durch den *Back To AR Button* soll das Wechseln zwischen AR und manueller Eingabe dynamisch gehalten werden. Ist man mit seinen Angaben zufrieden, kann man diese durch die Add-Funktion einem Projekt hinzufügen. Hierfür wird man zur letzten erreichbaren Seite des Hauptmenüs weitergeleitet: **See Projects** (s. Abb. 20). Hier sind zehn Projekte aufgelistet, die durch eine *Scroll View* einsehbar sind. Nun kann der Nutzer eines der besagten Projekte wählen, um beispielsweise bereits im *Manual Input* eingege-

bene Werte zu importieren. Ist man also **auf einer bestimmten Projektseite** (s. Abb. 21), so bieten sich dem Nutzer folgende Möglichkeiten. Jedes Projekt beinhaltet fünf Objekte. Wählt man eines, so öffnet sich das dazugehörige Aufklappfenster, worin sich die gleichen Eingabefelder (Name, Höhe, Breite, Tiefe) wie im *Manual Input* wiederfinden. Rechts im Fenster gibt es drei *Buttons*: eines zum Schließen des Aufklappfensters, eines zum Importieren der Daten aus dem *Manual Input* in das aktuelle Objekt und eines zum Speichern der neuen Werte. Abgesehen von den einzelnen Objekten, kann man auch das Projekt, in dem sich die Objekte befinden, umbenennen. Die Speicherung dieser Werte erfolgt in der gleichen Leiste mit einem Häkchen. Benötigt man das Projekt nicht mehr, kann man im unteren Bereich der Seite auf den *Delete Button* klicken. Ebenso findet man auf der Seite mit den aufgelisteten Projekten ebenfalls einen *DeleteAll Button*, welcher dementsprechend alle Projekte samt Objekte leert, sowie das *Manual Input*.

Grundsätzlich befindet sich auf jeder Seite, die durch das Hauptmenü erreichbar ist, eine kurze Information, die sich von Englisch auf Deutsch übersetzen lässt und umgekehrt. Ebenso befindet sich unten links immer der *Back Button*, womit die vorherige Seite aufgerufen wird.

## 4 Umsetzung und Implementierung

Bei der Entwicklung der App wurde die meiste Zeit in das *Measure Tool* und die anschließende Speicherung von Daten investiert. Allgegenwärtig sollte dabei die Softwareergonomie der Anwendung sein.

### 4.1 C# Programmierung

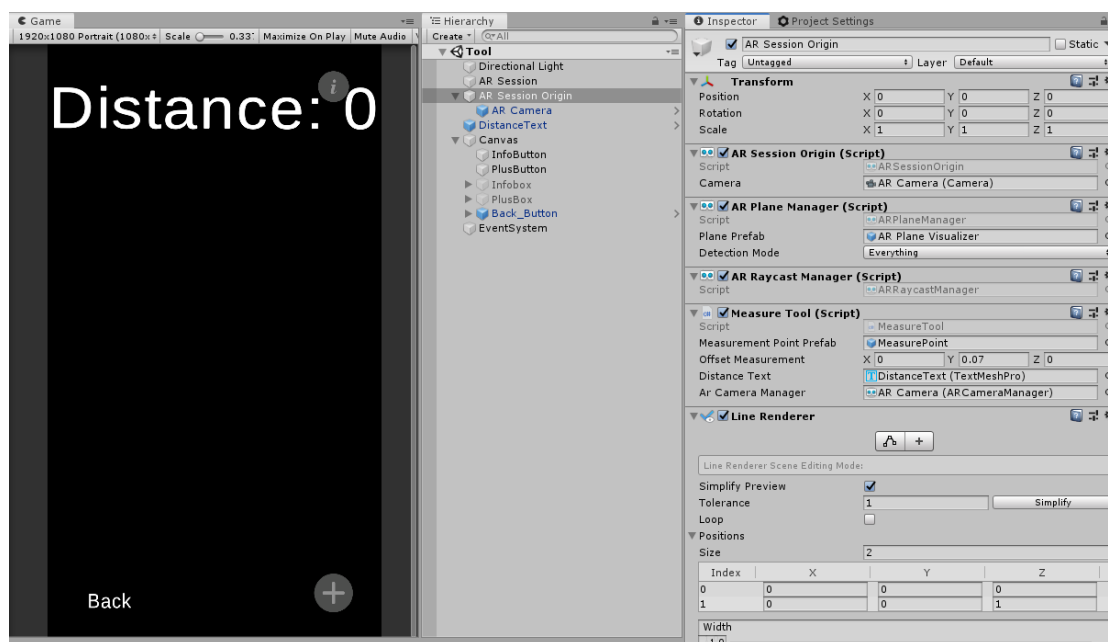
Das *Scripting* in *Unity* erfolgt über die Programmiersprache C#. Da *Unity* wesentliche Anteile selbst übernimmt, wie zum Beispiel das Ausführen der Anwendung, kann man sich auf das *Scripting* des sogenannten *Gameplays* konzentrieren. Als *Gameplay* bezeichnet man das Zusammenspiel von *Scripts* und Komponenten, die mit den *Game-Objects* verbunden werden, sodass man mit diesen ggf. interagieren kann [Uni20a]. Als Entwicklungsumgebung wurde Visual Studio gewählt. Hauptschwierigkeit hierbei war zu verstehen, wie man ein Ausmessungstool im AR programmiert. Des Weiteren wurde sich auch mit der Speicherung und Abrufung von Daten beschäftigt. Auf diese Punkte wird im Folgenden detaillierter eingegangen.

## 4.2 Measure-Tool

**Einrichtung in Unity** Die Idee des *Measure-Tools* bestand darin, eine bestimmte Strecke im Raum messen zu können. Um dies umzusetzen, wurde das *Measure Tool* in *Unity* als *Scene* angelegt. Über den *Package Manager* hat man die Möglichkeit die benötigten Pakete herunterzuladen. In diesem Fall wurde die *AR Foundation Library* und das *ARCore XR Plugin* heruntergeladen. Wenn man eine *IOS*-App entwickeln möchte, muss man anstelle des *ARCore* das *ARKit XR Plugin* laden. *AR Foundation* macht es möglich die App für beide mobile Betriebssysteme fertigzustellen. Hierfür kann man dementsprechend beide AR Plugin-Pakete herunterladen. Wichtig zu wissen wäre hierbei, dass man, um eine *IOS*-fähige App zu entwickeln, gewisse Bedingungen erfüllen muss. Dazu gehört die Programmierung auf einem *Mac* Gerät in *XCode*, wobei man einen über *iTunes* genehmigten *Developer Account* besitzen muss [Kra17].

In der Hierachy lassen sich nun neue Sachen hinzufügen. Als erstes fügt man das *AR Session* Objekt hinzu. Mit diesem Objekt wird nicht direkt interagiert, aber es bietet einige Setup-Optionen an, die in AR-Szenen benötigt werden, damit diese ordnungsgemäß laufen können. Als nächstes Objekt wird noch das *AR Session Origin* hinzugefügt, welches essenziell ist, um mit der Welt um uns zu interagieren. Da dieses auch eine *AR Camera* enthält, darf die *Main Camera* gelöscht werden.

Nun ist es möglich mit den eigentlichen *GameObjects* zu beginnen. Hierzu brauchen wir eine Ausgabe, die als *TextMesh Pro* [Tec20b] *Distance: 0* definiert wurde. *TextMesh Pro* ist eine neuere Textart von *Unity*, welche mehr Optionen für die Textdarstellung bietet, z.B. Textschrift farbig umranden. Die Punkte, die später gesetzt werden sollen, legen wir allerdings nicht in der *Hierarchy* an, da wir wollen, dass diese erst bei Touch-Berührungen des Nutzers auftauchen. Wie genau diese angelegt werden, wird im Nachgang erläutert.



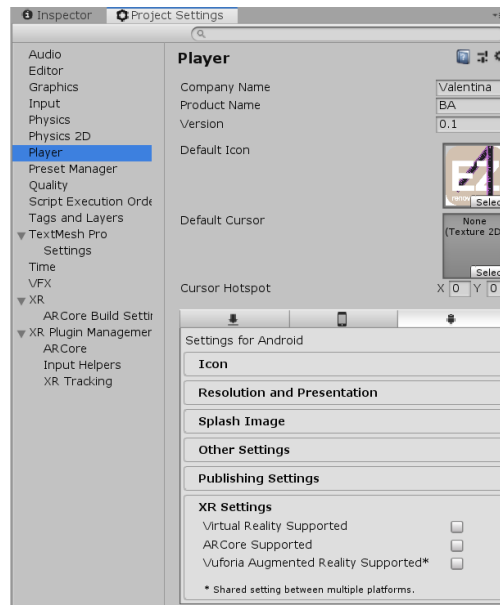
**Abbildung 12:** *AR Session Origin* Komponenten (eigene Abbildung)

Die *Game View* (links) ist schwarz hinterlegt, weil auf einem mobilen Endgerät die Kamera aktiviert werden würde. In der *Hierarchy* (mittig) sind alle verwendeten *GameObjects* aufgelistet. Im *Inspector* (rechts) sieht man immer die Komponenten des aktuell angewählten Objekts.

Vorerst muss darauf geachtet werden, dass alle wichtigen Einstellungen (s. Abb. 13) und Komponenten im *ARSessionOrigin* enthalten sind (s. Abb. 12). Dabei muss ein sogenanntes *Prefab* erstellt werden, welches hier *AR Plane Visualizer* genannt wurde. *Prefabs* sind *GameObjects*, welche als Templates genutzt werden können. Die vorgenommenen Einstellungen bleiben über Szenen hinweg erhalten. Diesem *Prefab* müssen folgende Komponenten hinzugefügt werden: *AR Plane*, *AR Plane Mesh Visualizer*, *Mesh Collider*, *Mesh Filter*, *Mesh Renderer* und *Line Renderer*. Diese Einstellungen sind notwendig, damit man eine Ebene erkennen und erzeugen kann. Beim *Line Renderer* kann man dementsprechend die Schatten und das *World Space* deaktivieren, da sie nicht benötigt werden. Material und Breite der Flächenerkennungskontur kann ebenfalls hier angepasst werden. Durch *Drag-and-Drop* in den *Asset*-Bereich kann man aus seinem *GameObject* ein *Prefab* machen. Nun kann man das *GameObject* in der *Hierarchy* sogar löschen und das *Prefab* so mit dem *AR Plane Manager* verknüpfen. Auch den Punkt, den wir später zum Setzen und Ziehen nutzen, können wir schon als *Prefab* anlegen und erstmal ungebraucht lassen. Als Materialfarbe wurde ein metallisches rot gewählt, um sicherzustellen, dass die Punkte gut sichtbar sind.

Die gleichen Einstellungen vom *Line Renderer* des vorhin angelegten *Prefabs* können wir auch hier im *AR Session Origin* vornehmen. Diese Linie ist diesmal unser

Maßband, welches die Distanz von Punkt A nach Punkt B misst. Eine andere Farbe ist aber empfehlenswert.



**Abbildung 13:** *Player, XR Settings* in *Project Settings* (eigene Abbildung)

Die Häkchen dürfen an dieser Stelle für eine reibungslose Wiedergabe nicht gesetzt werden.

Das *User Interface* der App-Struktur wurde als eigene Szene angelegt, separiert von der *Measure Tool* Szene. So wurde beim AR Tool lediglich ein *Canvas* als fixes Fenster hinzugefügt, um die Interaktionen mit den *Buttons* einheitlich zu halten.

**Scripting** Um dem *Measure Tool* auch gewisse Funktionalitäten zuteilen zu können, muss ein *Script* hinzugefügt werden, welches direkt als Komponente angelegt werden kann. Praktisch hierbei ist die Nutzung von *SerializeFields* [Tec20d], welche Objekte oder Variablen jederzeit neu anpassbar machen. Diese erscheinen in *Unity* als interaktive Felder im Inspector. So kann die Kugel, die Textausrichtung, der Text selbst und die *AR Camera* in Einem integriert werden (s. Abb. 12).

Zur Implementierung des Ganzen brauchen wir im Grunde zwei Funktionen.

Die *void Awake()*-Funktion soll durch den *AR Raycast Manager* bloß die Flächenerkennung instanziiieren. Die zwei Punkte, sowie der Text sind also noch deaktiviert. In der *void Update()*-Funktion passiert dann alles Andere. Durch diverse *if*-Bedingungen wird geprüft, wann der Nutzer den Bildschirm berührt. Beim Tippen soll der *startpoint* aktiviert werden. Beim Streichen bzw. Ziehen soll dann zusätzlich zum Erscheinen des *endpoints* auch das *measureLine* (*Line Renderer*) erscheinen. Dadurch wird das virtuelle Maßband sichtbar. In einer letzten *if*-Bedingung wird dem *Distance Text* der ermittelte Abstandswert übergeben (s. Codeausschnitt 1).

**Listing 1:** DistanceText - if-Bedingung

```
1 if (startpoint.activeSelf && endpoint.activeSelf)
2     {
3         DistanceText.transform.position =
4             endpoint.transform.position
5             + offsetMeasurement;
6         DistanceText.transform.rotation =
7             endpoint.transform.rotation;
8         measureLine.SetPosition(0,
9             startpoint.transform.position);
10        measureLine.SetPosition(1,
11            endpoint.transform.position);
12
13        DistanceText.text = $"Distance:{
14 (Vector3.Distance(startpoint.transform.position,
15 endpoint.transform.position)).ToString("F2")} m";
16    }
```

Die *if*-Bedingung trifft ein, wenn sowohl der Startpunkt als auch der Endpunkt gesetzt wurde. Der *Distance Text* wird über *Transform* an der aktuellen Stelle des *endpoints* positioniert. Mit dem angelegten *offsetMeasurement* kann man zwischen dem Punkt und der Schrift einen Abstand zum Adjustieren der Position anwenden. Der Text soll auch je nach gezogener Linie mitrotieren. Auch das wird über *Transform* der Endpunkt-Rotation angepasst. Des Weiteren wird der *measureLine* durch *SetPosition* gesagt, dass sie beim ausgewählten Startpunkt startet, wobei diese erst erscheint, wenn auch ein Endpunkt gesetzt wurde.

Als Ausgabe des Textes wird neben dem fixen *Distance* definiert, dass die Distanz aus *Vector3* [Tec20j] (Positionen - und Richtungsübergabe) zwischen Start- und Endpunkt ermittelt wird und anschließend als String ausgegeben wird. Der ermittelte Abstandswert wird mit F2 auf zwei Nachkommastellen limitiert, da es sich hierbei um Meterangaben handelt.

### 4.3 PlayerPrefs

Die Speicherung von Werten wurde über die *PlayPrefs*-Methode [Tec20i] abgewickelt. Diese speichert und greift auf Spielereinstellungen zu. Die Einfachheit des Zugriffs macht diese Methode besonders attraktiv. Denn die Methoden sind von *Unity* schon vordefiniert, was heißt, dass man nur die Funktionen aufrufen muss. Allerdings sind sie auch auf bestimmte Datentypen beschränkt: *int*, *float*, *string*. Die Funktionen, die man über den *PlayerPref*-Befehl aufrufen kann, lassen sich in fünf Gruppen gliedern.



**SetFloat** zum Beispiel - alternativ **SetInt** oder **SetString** - legt den Wert durch den identifizierten Präferenzschlüssel fest. Das ist im Grunde der Schritt, der die Speicherung des Typen bestimmt. **GetFloat** jedoch - alternativ **GetInt** oder **GetString** - nimmt die Daten, die mit dem entsprechenden Schlüssel gespeichert worden sind und integriert diese in der Anwendung. Mit **HasKey** kann man abfragen, ob ein Schlüsselname bereits vergeben ist. Des Weiteren gibt es zwei **Delete**-Befehle: bei dem *DeleteKey* werden dem Schlüssel zugewiesene Werte gelöscht und bei *DeleteAll* werden alle zuweisbaren *PlayerPrefs* entleert. Das bereitgestellte **Save** ist nicht zwingend notwendig, da es nur nochmals die modifizierten Einstellungen speichert. Sinnvoll wäre es allerdings bei *Checkpoints* von Spielanwendungen. Zusammenfassend lässt sich erschließen, dass man zu jeder Aktion eine Referenz bzw. einen Schlüssel benötigt. Dabei ist der genannte Schlüssel als String zu nutzen. *PlayerPrefs* wurden im Rahmen der Speicherung und Lagerung der App-Daten als geeignet empfunden. Während der Programmierung wurde festgestellt, dass es allerdings ziemlich aufwändig sein kann über zweihundert (zehn Projekte a fünf Objekte mit je vier Werte) Schlüssel anzulegen und anzupassen. Jedoch erreicht man dadurch eine flexible Speicherung von Daten, die jederzeit abrufbar und überschreibbar sind, was im Sinne dieser Arbeit ist.

#### 4.4 Softwareergonomie der App

Bei der Benutzerfreundlichkeit wurden drei Hauptaspekte betrachtet:

1. **Effektivität** - Wie kann man die App genau und vollständig gestalten, sodass der Nutzer damit das Ziel erreichen kann, die App über den traditionellen Holzgliedermaßstab zu nutzen?

Um das Ersetzen abzuwägen, muss die App mindestens genauso zuverlässig sein, wie ein Holzgliedermaßstab selbst. Das heißt, dass die Genauigkeit des AR Tools unerlässlich ist. Zusätzlich sollen die entnommenen Werte so gespeichert werden, dass der Bediener die Werte einsehen und bei Bedarf anpassen kann.

2. **Effizienz** - Wird der benötigte Aufwand im Verhältnis zur Genauigkeit und Vollständigkeit minimal gehalten?

Ist die App vollständig und liefert genaue Werte, so muss das Anwenden einfach gehalten werden und schnell arbeiten, damit von Effizienz gesprochen werden kann. Nicht nur die Flächen müssen präzise und schnell erkannt werden, sondern auch die Bedienbarkeit muss rapide ablaufen können.

3. **Zufriedenheit** - Kann Beeinträchtigungsfreiheit und Akzeptanz der Nutzung erzielt werden?

Diesbezüglich ist Grundvoraussetzung, dass die App selbsterklärend ist und die Bedienung intuitiv erscheint. Ein ansprechendes Design schafft die Verbindung zum

Anwender, sowie eine angenehme Freude der Nutzung dieser Anwendung. Des Weiteren sollte die Zufriedenheit des Nutzers nicht durch eine fehlerhafte Anwendung eingeschränkt werden.

Da es sich hierbei um einen Prototypen handelt, kann letzteres nicht garantiert werden. Die Funktionalitäten sind zwar alle gegeben, aber gewisse Daten erscheinen erst beim zweiten Anlauf, dann aber fehlerfrei.

Allgemein wurde als Bildschirmmaß das 1920x1080 Portrait in *Unity* gewählt, was dem Standard für Smartphones entspricht. Beim Programmieren und Einrichten klassischer Funktionen, wie beispielsweise diverse Seitenwechseleoptionen, wurde darauf Wert gelegt, dass dies durch gängige Wort-*Buttons*, wie *Back* geschieht. Andere *Buttons*, die konventionelle Symbole enthalten, sollen ebenso für eine schnelle Erkennung der Funktionalität sorgen. Die entsprechenden Icons wurden meistens aus einem *Asset Store* Packet namens *simple icon pastel tone* [Uni20b] benutzt, einige wurden auch selbst erstellt oder angepasst (*Sprach-Buttons*, *AR Import-Button*). Des Weiteren wurden natürliche Effekte hinzugefügt, die der App Leben einhauchen. So wurde zum Beispiel dafür gesorgt, dass beim Betätigen eines *Buttons*, dieser eine andere Farbe annimmt. Ebenso wurde eine *Scrollbar* bei der Projektauflistung hinzugefügt, die das Bedienen erleichtern soll. Die Optik der App sollte im eleganten, aber schlichten Stil erscheinen, um teils einen seriösen, ansprechenden Eindruck zu hinterlassen und teils eine Verbindung zum Mobilium herzustellen. Das Logo von *EZrenovate* wurde anhand des *Canva Design-Gestalters* [Can20] selbst kreiert, um der App Ausdruck zu verleihen. Für das Hintergrundbild der App wurde ein dunkles lizenzfreies Möbelbild [Pex20] benutzt, welches mit der weißen Schrift und den Altosé-Akzenten harmoniert. Das Bedienungskonzept wurde einfach gehalten. Der Nutzer legt ein Objekt an, sucht sich einen Speicherort in einem Projekt aus und importiert die Objektdaten. Das gilt sowohl für manuelle Eingaben, als auch für die Erfassung der AR-Daten. Bei dem *Measure Tool* wurde die Funktionsweise durch eine Infobox erklärt. Das genauere Bedienungskonzept ist in Kapitel 3 näher erläutert.

## 5 Evaluation

In diesem Kapitel wird die Studie zur AR-App behandelt. Hier wird nochmal die Zielsetzung aufgegriffen, die Durchführung der Studie erläutert und zum Abschluss werden die Ergebnisse und Bilder des Projekts vorgestellt.

### 5.1 Evaluationstest

**Intention** Um die App zu testen, wurde ein Evaluationsstudie vorbereitet. Diese beinhaltet zum einen das Untersuchen der App auf Benutzerfreundlichkeit und zum anderen

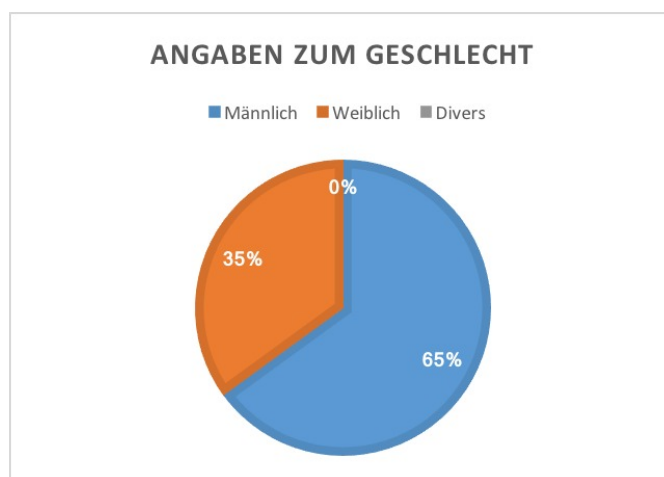
überprüft sie, ob das mit *AR Foundation* erstellte Tool den Genauigkeitsansprüchen gerecht wird. Hierbei wurde darauf geachtet Probanden aus verschiedenen Altersklassen zu nehmen und die App an unterschiedlichen Objekten zu unterschiedlichen Tageszeiten zu testen. Dies hatte zur Absicht mehrere Faktoren, wie Belichtung oder Form des Objekts, beim Vergleich miteinzubeziehen.

**Fragebogen** Der Nutzer muss zuerst sein Geschlecht und Alter angeben. Zudem muss er ankreuzen, ob er den täglichen Umgang mit diversen Apps pflegt und ob er bereits Erfahrung mit AR-Apps gemacht hat. Dann werden dem Nutzer acht kleine Aufgaben präsentiert, sowie das Versuchsobjekt. Der User kriegt ein *Android* Smartphone - das *Huawei Mate 20* - zur Verfügung gestellt, worauf die App bereits installiert und geöffnet ist. Dass die Anwendung über *Android* läuft, ist wegen dem identischen Erscheinungsbild auf *IOS* irrelevant, weshalb das im Nutzertest nicht weiter untersucht wird.

Die erste Aufgabe besteht darin eine manuelle Eingabe durchzuführen und das Objekt mit Namen, Höhen-, Breiten- und Tiefenwert an einem beliebigen Ort abzuspeichern. Diese Aufgabe soll den Nutzer mit der Logik der App vertraut machen und die Basisfunktionen demonstrieren. Nachdem dies verstanden wurde, darf die Versuchsperson dem Projekt einen Namen geben. Nach jedem Schritt wird der Nutzer gebeten die Änderungen zu speichern. Als dritte Aufgabe soll der User die gleichen Werte in einem bestimmten Objekt eines anderen Projekts importieren. Das betroffene Projekt soll als nächstes den Namen des Testraums annehmen. Die Werte des Objekts, die vorhin importiert wurden, sollen nun mit den AR-Werten überschrieben werden. Dazu muss der User das *Measure-Tool* anwenden und das anfangs vorgestellte Testobjekt abmessen, importieren und nach der Umbenennung speichern. Die sechste Aufgabe gibt der Testperson die Möglichkeit die mit dem AR-Tool genommenen Werte mit einem Holzgliedermaßstab abzugleichen. Als siebte Aufgabe sollte der User zur Auflockerung die App neu starten und überprüfen, ob die Werte, die er eingespeichert hat, noch da sind. Zum Abschluss darf der Nutzer dann alle Projekte löschen. Während der Versuchsreihe wird das laute Denken der Testperson beobachtet und protokolliert. Im Anschluss daran absolvieren die Testpersonen einen Fragebogen zur Benutzerfreundlichkeit der App und zur Anwendung des AR-Tools. Der Fragebogen umfasst insgesamt 20 Fragen. Die ersten 17 sind geschlossene Fragen in Form von Aussagen, die mit einer Likert-Skala [Sta20] zu bewerten sind. Die Probanden haben dabei jeweils vier Möglichkeiten, um ihre Meinung zu den Hypothesen zu äußern. Diese waren zwischen völliger Ablehnung (- -) bis zu völliger Zustimmung (++). Auf die neutrale Mitte wurde bewusst verzichtet, um verwertbare Daten zu erzielen. Am Ende gibt es drei offene Fragen, die der Proband frei beantworten kann, sodass auch weiterführende Aspekte festgehalten werden können. Zum Schluss soll der Proband noch ankreuzen, ob er die App selbst nutzen würde und ob dieser die App weiterempfehlen würde.

## 5.2 Auswertung und Diskussion

Die Anwendung wurde insgesamt von 20 Probanden getestet, wovon die Mehrheit männlich war (s. Abb. 14). Das Durchschnittsalter der Versuchspersonen betrug 26,8 Jahre, wobei die Altersspanne von 15 Jahren bis 48 Jahren aufgezo-



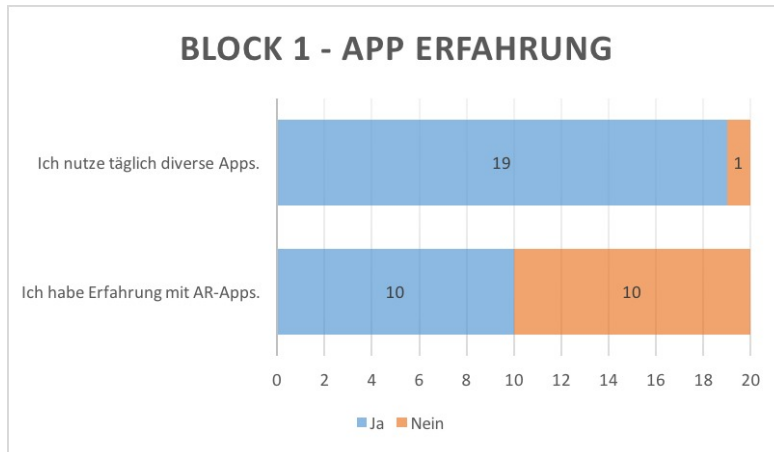
**Abbildung 14:** Geschlecht der Probanden, Kreisdiagramm (eigenes Diagramm)

Die Testzeiten variierten je nach experimentellen Willen zwischen zehn Minuten und 30 Minuten. Die Testpersonen waren freigestellt die App zu nutzen, solange sie es wollten. Über die Hälfte der Probanden wollte mit der AR-Anwendung experimentieren und nicht nur die vorgeschriebenen Aufgaben absolvieren. Sie wollten eigenständig andere Objekte mit der Flächenerkennung untersuchen. Festzustellen war hierbei, dass die Nutzer die App nicht nur bewusster anwendeten, sondern auch die Messwerte genauer wurden. Dies hat den Hintergrund, dass viele Nutzer anfangs nicht viel Wert auf Genauigkeit gelegt hatten, weil sie mit der Form der erkannten Fläche bereits unzufrieden waren, was laut geäußert wurde. Sobald ihnen erklärt wurde, dass die Flächenerkennung von *ARCore* sich dem Halten der Kamera anpasst, änderte sich die Herangehensweise der Nutzer, was auch positiven Einfluss auf die neuen Maße hatte.

Die Auswertungen werden blockweise anhand gestapelter Balkendiagramme veranschaulicht.

**Block 1** Bezüglich der App-Erfahrung gaben 19 von 20 Personen an, zwar täglich den Gebrauch verschiedener Apps zu machen, lediglich die Hälfte der Probanden hatte bereits mal den Umgang mit AR-Apps gepflegt (s. Abb. 15). Dies machte sich besonders im Umgang mit dem AR-Tool bemerkbar, wodurch die Meinungen stark variieren. Dass so gut wie jeder vertraut mit verschiedenen Apps ist, zeigt sich auch in der Benutzer-

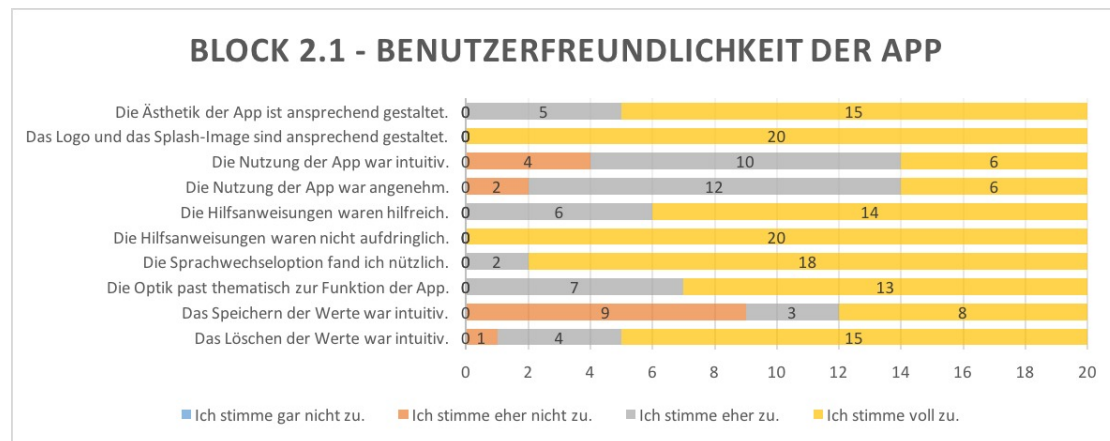
freundlichkeitsauswertung. Die Nutzer haben vieles als selbstverständlich erachtet und waren schnell verwirrt, wenn etwas nicht der Norm entsprach. Dies wird in den nächsten Blöcken an Beispielen gezeigt.



**Abbildung 15:** Bisherige App Erfahrung, Balkendiagramm (eigenes Diagramm)

**Block 2.1** Durch die oben genannte Vertrautheit zu unterschiedlichen Apps konnte man beobachten, wie intuitiv gewisse Funktionen behandelt worden sind. So war beispielsweise die *Button*-Bedienung schnell mit einer bestimmten Selbstverständlichkeit erfasst worden. Der einzige *Button*, der vermehrt Schwierigkeiten gemacht hatte, war der unkonventionelle Speicher-*Button* im Aufklappfenster des jeweiligen Objekte (s. Abb. 21). Korrekterweise merkten fast alle Nutzer an, dass der *Refresh-Button* nicht als Speicherung verstanden wurde. Hierbei sei die Konvention eine Speicherdiskette, welche hier stark erwünscht war. Dementsprechend floss diese Tatsache negativ in die Bewertung hinein - mit neun eher nicht zustimmenden Meinungen bezüglich der intuitiven Speicherung (s. Abb. 16). Die Personen, die allerdings dem eher oder sogar voll zustimmten, waren auch die aufmerksamen Leser, die die Hilfsanweisungen verinnerlicht hatten und somit wussten, wie sie etwas im Zweifelsfall speichern können. Die Hilfestellungen, die im Übrigen als positiv bewertet wurden, sind den Probanden teilweise mündlich vorgetragen worden. Interessant war folgende Beobachtung: Nutzer, die in der englischen Sprache nicht sicher genug waren, fanden schnell die Sprachwechsoption, während Nutzer, die die englischen Anweisungen direkt verstanden hatten, die Option teilweise nicht einmal wahrnahmen. Diese wurde nach Aufklärung dennoch als sehr nützlich eingestuft. Herausstechend sind die besonders ansprechende Gestaltung des Logos und des Splash-Images, sowie die unaufdringliche Positionierung der Hilfsanweisungen. Dem stimmten alle Probanden ausnahmslos voll zu (s. Abb. 16). Die generelle Nutzung der Anwendung wurde in den meisten Fällen (18 von 20 Probanden) als angenehm empfunden. Allerdings war man sich bei der Intuitivität der Nutzung nicht ganz einig. Vier Nutzer aus der älteren Altersklasse klagten, dass sie die Logik der App nicht verstanden hätten. Zehn Nutzer

betonten, dass die App an sich gut sei, aber kleinere Feinheiten das Ganze intuitiver machen könnten. Dazu wurden berechnete Punkte genannt, wie ein Erklärungsfenster für die unterschiedlichen Symbol-Buttons, sowie eine vereinfachte Speicheroption. Sechs Nutzer kamen problemlos mit der Anwendung zurecht und stuften sie als sehr intuitiv ein. Das Löschen der Werte wurde 15 Mal mit der völligen Zustimmung als intuitiv bewertet, allerdings gab es auch eine Person, die der englischen Sprache nicht mächtig war und es eigenständig nicht schaffte, die bearbeiteten Projekte zu löschen. Dazu wurde angemerkt, dass die Sprachwechseleoption auch die Buttons übersetzen sollte und man dies auch seitenübergreifend einstellen sollte. Die restlichen vier Probanden fanden es etwas störend, dass der Löschvorgang erst nach dem Neustarten der App ersichtlich wurde, weshalb sie nur eher zugestimmt haben. Die Optik vermittelte insgesamt eine elegante und adäquate Atmosphäre. Ein paar wenige Probanden äußerten allerdings, dass das Hintergrundbild eher chaotisch wirkt und das mit dem Ausmessungstool nicht harmoniere, obwohl die Farbgebung sonst sehr gut zusammenpassen würde. Außerdem wurde auch festgestellt, dass der Scroll Button keine gute Auflösung aufweist, weshalb dem ansprechendem Ästhetik-Aspekt nicht immer voll zugestimmt wurde.

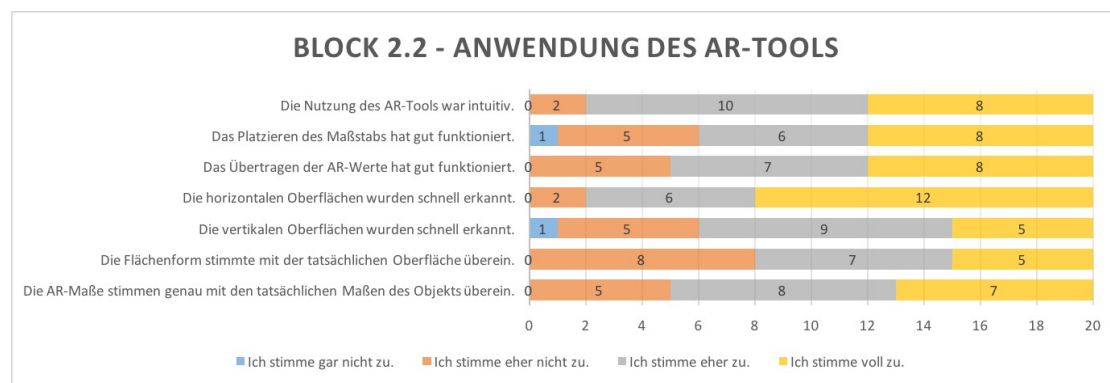


**Abbildung 16:** Benutzerfreundlichkeit der App, Balkendiagramm (eigenes Diagramm)

Insgesamt lässt sich feststellen, dass kein Proband einem Aspekt der Benutzerfreundlichkeit der App gar nicht zustimmen konnte. Im Balkendiagramm (s. Abb. 16) dominiert die volle Zustimmung, was sehr für eine Akzeptanz der Oberfläche dieser Anwendung spricht. Dennoch wurde deutlich, dass die App noch als Prototyp zu betrachten ist, der nach kleineren Änderungen die völlige Akzeptanz erreichen könnte.

**Block 2.2** Bei der Nutzung des AR-Tools ließen sich bei fast jedem Probanden unterschiedliche Ergebnisse feststellen (s. Abb. 17). Die Auswahl an Objekten wurde auf zwei Gegenstände begrenzt. Dies hatte den Vorteil, dass man die genommenen Werte besser miteinander vergleichen konnte und dennoch unterschiedliche Gegenstände untersucht

hat. Die genommenen Maße wurden protokolliert und liegen dem Anhang bei. Bemerkenswert war die Erkenntnis, dass bei hohen Objekten die horizontale Flächenerkennung erschwert war, während bei kleineren Objekten die vertikale Flächenerkennung erst später auftauchte. Die Nutzer wurden in beiden Fällen durch die Infobox darauf hingewiesen die Kamera über eine größere Ebene, wie dem Boden, zu schwenken. Dort wurde auch erklärt, wie der Nutzer in der Lage sei, die gewünschten Maße zu entnehmen. Die Nutzung des AR-Tools wurde überwiegend als intuitiv bewertet, wobei das Platzieren des virtuellen Maßstabs meistens gut geklappt hat. Acht Personen waren damit voll zufrieden und sechs Personen konnten dem eher zustimmen. Fünf Probanden stimmten dem eher nicht zu und eine Person sogar gar nicht. Dieses Ergebnis zeigt, dass zwölf von zwanzig Befragten nicht gänzlich von der Maßstabplatzierung überzeugt waren. Dies hat oft den Hintergrund gehabt, dass die Kugeln, die zum Platzieren von Start- und Endpunkt verwendet wurden, von dem führenden Finger verdeckt wurden oder die runde Form öfters als ungünstig deklariert wurde. Ein weiterer Grund könnte auch die nicht perfekte Flächenerkennung sein. Wenn beispielsweise die Fläche nicht ganz erkannt oder fehlerhaft angezeigt wurde, konnte man keine brauchbaren Werte entnehmen. Je länger und forscher der Nutzer jedoch das Tool anwandte, umso genauer wurde die Fläche erkannt und umso genauer waren die entnommenen Werte.



**Abbildung 17:** Anwendung des AR-Tools, Balkendiagramm (eigenes Diagramm)

Beim Ausmessen eines Schrankes stimmten die Werte meistens genau überein. Jedoch ließ sich feststellen, dass die Tiefe des Schrankes aufgrund der wahrscheinlich schlechteren Beleuchtung nicht als Ebene angezeigt wurde. Die Maße konnten dennoch entnommen werden, indem die erkannte Vorderseite und die Wand als Differenz genutzt werden konnten. Die Fähigkeit des Maßstabs zwischen unterschiedlichen Ebenen zu springen, wurde ebenfalls als positiv erachtet. Als Gegenstandsalternative wurde auch eine Sitzbox ausgemessen. Je nach Haltung des Smartphones wurde erst die Bodenebene erkannt und beim Nähern zur Sitzbox auch die obere Fläche. Hier entsprach die Flächenform der tatsächlichen Gegenstandsoberfläche ziemlich genau. Im Gegensatz zum Schrank war die Sitzbox eintönig und hatte auch keine anderen Gegenstände darauf liegen. Da der Schrank mit Bildern geschmückt war, die leicht nach außen geformt waren, wurden

manchmal auch mehrere Ebenen aufeinander gezogen. Bei der Sitzbox kam es in genau sechs Fällen vor, dass die vertikale Ebene nicht angezeigt wurde. Fünf dieser Probanden stellten aber fest, dass die vertikale Wand erkannt wurde, aber die Sitzbox wahrscheinlich zu kleinflächig war, um im Nachgang die Vertikale noch zu erkennen. Ein Proband testete die Erkennung der vertikalen Ebenen nicht weiter und stimmte somit dem gar nicht zu. Bei fünf Personen erschien die vertikale Ebene an der Sitzbox direkt und bei neun weiteren nach mehreren Versuchen die Kamera anders zu halten. Während der Versuchsreihe wurden insgesamt die horizontalen Oberflächen eher erkannt. Dies lässt sich damit begründen, dass sich Objekte auf horizontaler Ebene befinden und diese somit dominiert. Darüber hinaus ist markant, dass zwölf Personen die Sitzbox ausmessen wollten, was genau den völlig zustimmenden Meinungen entspricht (s. Abb. 17). Die Genauigkeit des AR-Tools wurde von den Probanden gemischt bewertet. Fünf würden der Genauigkeit eher nicht zustimmen, acht Personen jedoch schon eher und sieben Probanden waren davon überzeugt, dass das AR-Tool genau ist. Wodurch können solche Meinungsverschiedenheiten auftreten? Zum einen muss der Nutzer erkennen, ob er die richtige Ebene zum Ausmessen angewählt hat. Geht er zum Beispiel mit dem Endpunkt über die gemeinte Fläche hinaus, so kann man beobachten, wie sowohl Punkt als auch Schrift die Größe verändern. Das ist ein Hinweis dafür, dass die Ebene gewechselt wurde. Somit berechnet man also eine Diagonale über zwei Ebenen, was zu deutlich größeren Werten führen kann. Weichen die Werte nur wenige Zentimeter von den tatsächlichen Werten ab, so lässt sich das auf Messungenauigkeit des Nutzers zurückführen. Dies kann auch daher kommen, dass den Nutzern nicht immer deutlich wurde, dass das Zentrum der zu setzenden Kugel der Anfangs- bzw. Endwert ist. Es ist auch vorgekommen, dass die Sitzbox auf einer gewissen Distanz nicht erkannt wurde und somit nur die Bodenebene darüber gezogen wurde. Misst man aber nun die Höhe, ist der Wert wieder um einiges größer. Hierbei handelt es sich nämlich um eine Strecke, die nach hinten zeigt, anstatt nach oben. Der Nutzer muss also durch die Konturen der Flächenerkennung schlussfolgern, ob besagte Wunschfläche erkannt wurde, oder ob er zum Beispiel um das Objekt herum gehen sollte, um der Kamera eine andere Ansicht zu schenken. Diese Art der Schwierigkeit war ausschlaggebend für die fünf schlechteren Bewertungen. Die Übertragung der AR-Werte hat meistens gut funktioniert. Ein Problem war allerdings, wenn der User die Werte durch den *Plus Button* hinzufügen wollte und dieser *Button* sich in der Flächenerkennung befand. Die AR-Fläche hatte die größere Priorität, weshalb durch das Antippen auf den *Plus Button*, sich die gesetzten Punkte versetzten, wenn die Erkennung sich dahinter befand. Um das zu meiden, musste der Nutzer darauf achten beim Hinzufügen zum Beispiel kurz nach oben zu schauen. Acht Probanden hatten gar kein Problem damit, sieben fanden es noch in Ordnung und fünf Personen fanden das schon eher störend. Insgesamt hing die Bewertung des AR-Tools stark von den Probanden selbst ab. Deutlich wurde aber, dass die Platzierung des Maßstabs optimiert werden sollte, sowie das AR-Tool selbsterklärender gestaltet werden könnte. Schaut man sich die letzten vier Fragen des Blocks an, lässt sich schließen, dass *ARCore's* Flächendetektion insgesamt mit über 73% als mindestens eher gut bewertet wurde. Vorstellbar ist, dieses Ergebnis durch genauere Anweisungen zur Messung zu optimieren.



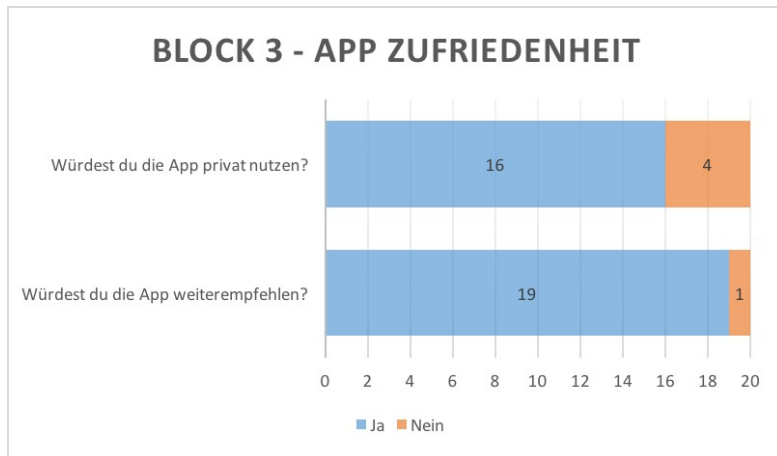
**Block 3** Hier wurden drei offene Fragen gestellt und im Anschluss zwei allgemeine Ankreuzfragen. Die offenen Fragen sollten dem Nutzer die Möglichkeiten geben, auch andere Aspekte zu äußern.

Zur ersten Frage, was besonders gefallen hätte, gab es viele ähnliche Antworten. Die häufigsten werden hier aufgeführt. Alle Antworten sind im Anhang einsehbar, wobei auch hier die häufigsten nur einmal vorkommen. Viele Nutzer fanden die Gestaltung, die Farbgebung, das moderne Design der App samt Logo sehr elegant und atmosphärisch seriös. Diese ansprechende Harmonie sorgte für eine Verbindung zwischen App und User. Auch die Hilfsanweisungen und die Sprachwechsoption wurde oft als nützlich und sinnvoll erachtet, was dem Nutzer oft zu Gute kam. Die Probanden hatten auch besonders beim praktischen Teil des Tests Spaß und fanden die Idee des innovativen Ausmessens sehr gut. Einige der Testpersonen haben die Ordnerstruktur als übersichtlich und gut empfunden. Auch die Dynamik zwischen AR-Daten importieren oder auch manuell Werte eingeben zu können, kam im Allgemeinen sehr gut an. Oft wurde auch die einfache und intuitive Nutzung betont, was auf eine hohe Benutzerfreundlichkeit der Anwendung hinweist. Die Probanden, die besonders erfolgreich mit dem AR-Tool umgehen konnten, äußerten auch auf positiver Weise Erstaunen über die Genauigkeit des virtuellen Maßbands.

Nicht so gut kamen dagegen die kleineren Bugs an, wie die Speicherung nach dem zweiten Anlauf oder dass beim Hinzufügen eines bestimmten AR-Wertes darauf geachtet werden musste, dass dies nicht im detektierten Bereich passiert. Oft wurde auch allgemein der Speichervorgang mit dem Zwischenspeicher meist als umständlich oder zu indirekt empfunden. Auch die überlappenden Ebenen erschwerten das korrekte Ausmessen.

Erfreulich waren konstruktive Verbesserungsvorschläge, bei denen genaue Vorstellungen geteilt wurden. Kleinere Feinheiten, wie die Meterangaben im *Measure Tool* sollen noch ergänzt werden. So wurde sich auch oft die Speicherdiskette als symbolhafte Speicherung gewünscht, was konventioneller sei als das benutzte Refresh-Symbol. Außerdem wurde bedauert, dass die Konturen der Flächenerkennung so breit seien, dass es bei feineren Objekten die genaue Erfassung erschwere. Weiterhin wurde auch der Wunsch geäußert eine direktere Möglichkeit der AR-Werte-Speicherung zu haben. Dies könnte man lösen, indem man nach dem Betätigen des *Plus Buttons* anstelle eines Textfensters eine direkte Zuweisung durchführen könnte, sodass kein Zwischenspeicher benötigt wird. Als besonders gutes Feedback wird die Idee bewertet anstelle der Kugeln kreuzförmige Start- und Endpunkte zu nutzen. Dadurch könnte eine bessere Messgenauigkeit erzielt werden. Des Weiteren wurde sich gewünscht beim Öffnen des AR-Tools als Hinweis verdeutlicht zu bekommen, dass sich der Nutzer durchaus durch den gesamten Raum bewegen kann und dabei mehrere Objekte mit der Flächenerkennung erfassen kann. Diese Funktion wurde nämlich durch die einfache Nutzung nicht direkt deutlich und werfe die App in ein ganz anderes Licht. Alternativ wurde auch vorgeschlagen am Anfang ein symbolisiertes, also möglichst textfreies, Anfangstutorial darzustellen, bei der beispielsweise eine

animierte Hand samt Smartphone über eine Ebene schwenkt. So würde der Nutzer ohne viel zu lesen schon wissen, was er zu tun habe.



**Abbildung 18:** App-Zufriedenheit, Balkendiagramm (eigenes Diagramm)

Zum Abschluss sollten die Probanden angeben, ob sie die App - wenn sie nicht mehr nur ein Prototyp wäre - nutzen oder diese weiterempfehlen würden. Insgesamt würden 16 von 20 Personen die App selbst nutzen. Vier Probanden verneinten dies, wovon drei Personen angaben kein Bedarf zu haben Dinge mit einer App ausmessen zu müssen. Einer der Probanden war von der Anwendung wegen der misslungenen Werte nicht überzeugt, weshalb er sie nicht nutzen und nicht weiterempfehlen würde. Die restlichen 19 Nutzer denken, dass diese App durchaus Verwendung im Alltag finden kann, wenn sie fertig ist.

### 5.3 Impressionen und Ergebnisbilder

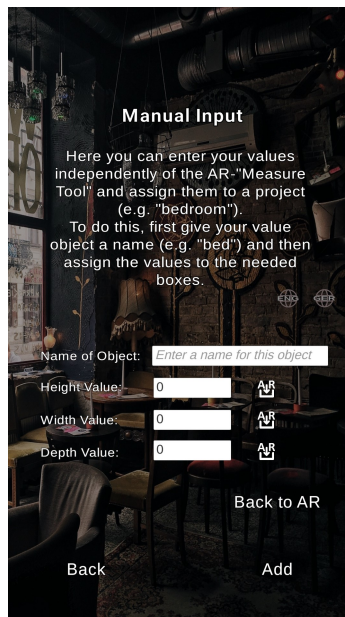


Abbildung 19: *Manual Input*, *EZrenovate* (eigene Abbildung)

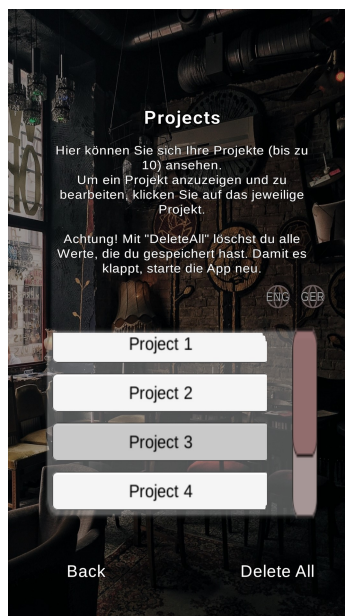
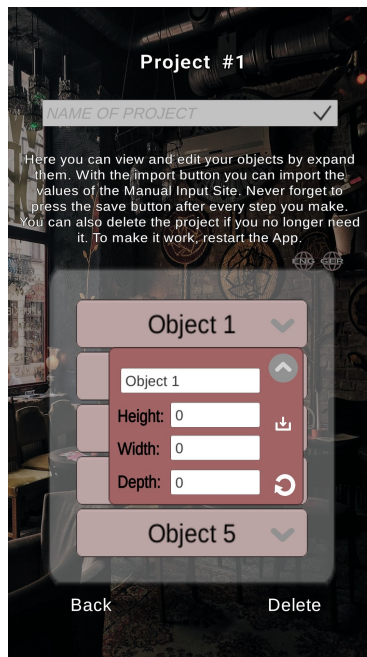


Abbildung 20: Projektübersicht, *EZrenovate* (eigene Abbildung)  
*Project3-Button* wird gerade betätigt. Hilfsanweisung wurde auf deutsch umgestellt.



**Abbildung 21:** Bestimmte Projektseite mit offenem Aufklappfenster, *EZrenovate* (eigene Abbildung)



**Abbildung 22:** *Measure Tool* - Höhenwert an einem Schrank, *EZrenovate* (eigene Abbildung)



**Abbildung 23:** *Measure Tool* - Tiefenwert an einer Sitzbox, *EZrenovate* (eigene Abbildung)

## 6 Fazit und Ausblick

Mit dem Ziel durch *AR Foundation* eine AR-App zu entwickeln, wurde erstmal der technische Stand untersucht, indem Grundlagen erforscht und sich mit der *Unity Engine* vertraut gemacht wurde. Daraufhin wurde die Konzeption der Anwendung entwickelt und diese dann umgesetzt und implementiert. Um die AR-App zu bewerten, wurde eine Nutzerstudie durchgeführt, welche Folgerungen über die Benutzerfreundlichkeit der Anwendung und der Genauigkeit des von *ARCore* zur Verfügung gestellten SDKs schließen ließ. So wurden auch mögliche Verbesserungsansätze erörtert und herauskristallisiert. Die Testergebnisse haben gezeigt, dass es sich bei der App um einen Prototypen handelt, welcher noch ausgearbeitet werden muss. Die AR-Anwendung unter *ARCore* hat sich als sehr genau erwiesen. Jedoch führt die Flächendetektion manchmal zu Überlappungen, die für den Nutzer teilweise fatal sein können und somit zu Verfälschungen der Ergebnisse führen. Im Großen und Ganzen fand die Anwendung aber viel Anklang. Kleinere Feinheiten im Bereich der Benutzerfreundlichkeit der Oberfläche können schnell gelöst werden. Zu optimieren ist hauptsächlich das Speicherkonzept, welches noch nutzerfreundlicher gestaltet werden muss, sowie darauf bezogene Bugs behoben werden müssen.

Das Interesse gegenüber *Augmented Reality* steigt im technischen Bereich stetig an. Daher ist es durchaus realistisch, dass immer mehr AR-Anwendungen entwickelt werden, die für den normalen Smartphone-User interessant sein können. *AR Foundation*

bietet aber neben *ARCore* auch das *ARKit* von *Apple* an, was durch wenige Einstellungen auch für *IOS* Endgeräte kompiliert werden kann. Durch die Einschränkung aktuell kein *Mac* Gerät und kein *Apple Developer Account* zu besitzen, konnte die App vorerst nur für *Android* Endgeräte hergestellt werden. Es ist aber denkbar, aus dem bisherigen Prototypen durch die angegebenen Korrekturen *EZrenovate* zu finalisieren. Wenn die App nach weiteren Evaluationen überzeugt, wäre eine Veröffentlichung denkbar.



Abbildung 24: App-Icon und Logo (Kürzel), *EZrenovate* (eigene Abbildung)



Abbildung 25: Logo (Langform), *EZrenovate* (eigene Abbildung)

# A Anhang

## A.1 Fragebogen

### Fragebogen zur AR-Ausmessungsapp EZrenovate

#### Teil 1

##### Angaben zur Person

Männlich       Weiblich       Divers

Alter: \_\_\_\_\_

Ich nutze täglich diverse Apps.                       Ja                       Nein

Ich habe Erfahrung mit AR-Apps.                       Ja                       Nein

##### Aufgaben

1. Gib beliebige Werte (Höhe, Breite, Tiefe und Name) manuell ein und füge sie dann einem beliebigen Projekt hinzu. Speichere dein Objekt.
2. Benenne das Projekt nach Belieben und speichere den Namen ab, bevor Du dann die Seite verlässt.
3. Importiere die gleichen Werte auch in Projekt 8, Objekt 2 und speichere dann.
4. Projekt 8 benennst Du nun um zu „CG-Labor“.
5. Überschreibe nun die Werte im gleichen Objekt (8.2) mit den AR-Daten des Dir vorgestellten Möbelstücks. Nachdem Du diesem Objekt auch den passenden Namen gegeben hast, speichere das nochmal ab.
6. Schau Dir an, inwiefern die Werte des AR-Tools den Werten des Holzgliedermaßstabs entsprechen.
7. Schließe nun die App. Starte sie dann erneut, um zu überprüfen, ob die gespeicherten Objekte unverändert geblieben sind.
8. Lösche zum Abschluss all deine Projekte.

## Teil 2

### Anmerkung:

Bitte ließ die Fragen bzw. Aussagen aufmerksam durch und versuch diese vollständig und wahrheitsgemäß zu beantworten. Das anfängliche Blockweise-Lesen wird empfohlen.

### Bewertungsskala:

Ich stimme voll zu.	++
Ich stimme eher zu.	+
Ich stimme eher nicht zu.	-
Ich stimme gar nicht zu.	--

### 1. Benutzerfreundlichkeit der App

--   -   +   ++

Die Ästhetik der App ist ansprechend gestaltet.				
Das Logo und das SplashImage sind ansprechend gestaltet.				
Die Nutzung der App war intuitiv.				
Die Nutzung der App war angenehm.				
Die Hilfsanweisungen waren hilfreich.				
Die Hilfsanweisungen waren nicht aufdringlich.				
Die Sprachwechseleoption fand ich nützlich.				
Die Optik passt thematisch zur Funktion der App.				
Das Speichern der Werte war intuitiv.				
Das Löschen der Werte war intuitiv.				

### 2. Anwendung des AR-Tools

--   -   +   ++

Die Nutzung des AR-Tools war intuitiv.				
Das Platzieren des Maßbands hat gut funktioniert.				
Das Übertragen der AR-Werte hat gut funktioniert.				
Die horizontalen Oberflächen wurden schnell erkannt.				
Die vertikalen Oberflächen wurden schnell erkannt.				
Die Flächenform stimmte mit der tatsächlichen Oberfläche überein.				
Die AR-Maße stimmten genau mit den tatsächlichen Maßen des Objekts überein.				



### 3. Allgemeines

a. Das hat mir besonders gut gefallen:

---

---

---

b. Das hat mir nicht so gut gefallen:

---

---

---

c. Verbesserungsvorschläge:

---

---

---

Würdest Du die App privat nutzen?  Ja  Nein

Würdest du die App weiterempfehlen?  Ja  Nein



Vielen Dank für die Teilnahme!

## A.2 Testwerte

### Alter der Testpersonen:

- Summiert: 27+48+17+47+15+20+27+31+22+27+22+23+23+30+25+24+23+24+28+32
- Durchschnittsalter: 26,75 Jahre

### Schrankwerte während der Nutzertests:

<u>Schrank</u>	Ist-Wert	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	$\bar{x}$	Differenz
Höhe	1,9	1,86	1,78	1,89	2,05	1,71	1,9	1,76	1,92	1,86	0,04
Breite	1,2	1,13	1,32	1,2	1,1	1,53	1,22	1,16	1,2	1,23	0,03
Tiefe	0,45	0,45	0,46	0,45	0,5	0,52	0,44	0,46	0,45	0,47	0,02

### Sitzboxwerte während der Nutzertests:

<u>Sitzbox</u>	Ist-Wert	Test 1	Test 2	Test 3	Test 4
Höhe	0,4	0,42	0,41	0,37	0,41
Breite	0,38	0,39	0,42	0,37	0,38
Tiefe	0,38	0,41	0,38	0,37	0,42

Test 5	Test 6	Test 7	Test 8
0,51	0,33	1,16	0,42
0,54	0,38	0,42	0,38
0,43	0,39	0,61	0,38

Test 9	Test 10	Test 11	Test 12	$\bar{x}$	Differenz
0,31	0,4	0,39	0,4	0,46	0,06
0,39	0,38	0,37	0,38	0,40	0,02
0,51	0,38	0,38	0,37	0,42	0,04

Die farblich hinterlegten Felder entsprechen den tatsächlichen Werten des jeweiligen Möbelstücks.

### **Block 3 – Auswertung**

Die Antworten sind im Folgenden akkumuliert aufgelistet worden. Die Reihenfolge misst sich nach der Häufigkeit der Antworten von oben nach unten.

#### **1. Das hat den Probanden besonders gut gefallen:**

- a. Gestaltung/ Farbgebung/ modernes Design der App, erzeugte Atmosphäre (Logo, Splash-Image)
- b. Gute Hilfsanweisungen
- c. Sprachwechsoption
- d. Idee, Konzeption, praktisches Arbeiten
- e. Genauigkeit des Tools
- f. gute Flächenerkennung (meistens horizontal)
- g. Einfache Nutzung
- h. Die Möglichkeit, sowohl AR-Daten zu importieren als auch manuell Werte eingeben zu können
- i. Ordnerstruktur sehr nützlich und übersichtlich
- j. Schnelles Speichern
- k. individuelle Namensgebung für alle Objekte und Projekte
- l. Intuitive Bedienung des AR-Tools

#### **2. Das hat den Probanden nicht so gut gefallen:**

- a. Speichern eher umständlich, zu indirekt, nicht intuitiv
- b. Bug mit dem Hinzufügen der AR-Werte, nur auf einer nicht detektierten Fläche
- c. Bug mit dem Übertragen der AR-Werte, erst beim zweiten Durchgang möglich
- d. Keine genaue Übereinstimmung vom AR-Flächenerkennungssystem mit den Oberflächen
- e. Flächenerkennung meist vertikal verspätet
- f. AR-Punkte setzen nicht ist nicht intuitiv
- g. Fehlerhafte Flächenerkennung mit überlappenden Ebene

#### **3. Verbesserungsvorschläge der Probanden:**

- a. Speicherbutton als Speicherdiskette (oder einheitlich für die ganze Projektseite mit einem Häkchen/einer Diskette), Icon-Konvention folgen
- b. Direkteres Zuweisen der genommenen Maße (ideal: im gleichen AR-Screen)
- c. Einfacheres Platzieren des Maßbandes -> anstelle der Punkte, wäre ein x-förmiger Start und Ende als Fixpunkte besser
- d. Schmalere Konturen bei der Flächenerkennung im eleganten Stil
- e. Hinweis, dass man durch den ganzen Raum gehen und mehrere Objekte mit der Flächenerkennung erfassen kann / Anfangstutorial anstelle der Infobox
- f. Meter-Einheit ergänzen bei der manuellen Eingabe
- g. Automatische Wertspeicherung im Projektordner
- h. Selbstständiges Anlegen von Projekten, anstelle der vorgefertigten
- i. Hintergrundbild schlichterer Möbel
- j. Ein Reset-Button im AR, wenn die Fläche falsch berechnet wurde, anstatt „Back“

## Literatur

- [App20a] Apple Developer Documentation. Arkit, 5.3.2020.
- [App20b] Apps bei Google Play. Maßband, 2.3.2020.
- [Azu97] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.
- [Bog19] Alessandro Bogliari. Instagram stories ar filters: What are they and how are brands using them? *Influencer Marketing Hub*, 25.6.2019.
- [Can20] Canva. Canva: Die zusammenarbeit und das erstellen von tollen grafikdesigns ist kostenlos, 9.3.2020.
- [Goo18] GoogleWatchBlog. Praktisch & beeindruckend: Google veröffentlicht virtuelles maßband für alle arcove-smartphones, 2018.
- [Goo19] Google Developers. Arcove, 20.11.2019.
- [IKE20] IKEA Unternehmensblog. Neue ar-app ikea place, 2.3.2020.
- [Kra17] Marco Kratzenberg. ios-app programmieren: Voraussetzungen und tipps. *GI-GA.de*, 29.5.2017.
- [Lan18] Michael Lanham. *Learn ARCore - Fundamentals of Google ARCore: Learn to build augmented reality apps for Android, Unity, and the web with Google ARCore 1.0*. 2018.
- [Mer16] Tim Merel. Pokémon go changes everything (and nothing) for ar/vr. *TechCrunch*, 12.08.2016.
- [Mic20] Microsoft. Microsoft hololens: Mixed reality-technologie für unternehmen, 9.3.2020.
- [NVI20] NVIDIA Developer. Physx sdk, 5.3.2020.
- [Pex20] Pexels. Kostenlose creative commons zero bilder, 6.3.2020.
- [PHAF95] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: a class of displays on the reality-virtuality continuum. pages 282–292. International Society for Optics and Photonics, 1995.
- [PTC20a] PTC inc. Getting started with vuforia engine in unity, 3.3.2020.
- [PTC20b] PTC inc. Vuforia engine: Developer library - overview, 4.3.2020.
- [Rau17] Rauschnabel, P.A., Rossmann, A. & Tom Dieck, M.C. An adoption framework for mobile augmented reality games: The case of pokémon go: Pokémon go, computers in human behavior. 2017.

- [SAM19] SAMSUNG. Samsung gear vr: Virtual reality brille, 2019.
- [SH16] D. Schmalstieg and Tobias Hollerer. *Augmented reality: Principles and practice*. Addison-Wesley, Boston, 2016.
- [Spa20] Spatial Computing for Enterprise. Magic leap, 5.3.2020.
- [ST18] Dirk Schart and Nathaly Tschanz. *Augmented und Mixed Reality: Für Marketing, Medien und Public Relations*. UVK Verlagsgesellschaft, Konstanz and München, 2., überarbeitete und erweiterte auflage edition, 2018.
- [Sta20] Statista. Likert-skala, 7.3.2020.
- [Tec20a] Unity Technologies. Deferred shading rendering path: Unity - manual, 2.3.2020.
- [Tec20b] Unity Technologies. Textmesh pro: Unity - manual, 2.3.2020.
- [Tec20c] Unity Technologies. Unity's interface: Unity - manual, 2.3.2020.
- [Tec20d] Unity Technologies. Serializefield: Unity - scripting api, 27.1.2020.
- [Tec20e] Unity Technologies. Das ar foundation-framework von unity, 5.3.2020.
- [Tec20f] Unity Technologies. Das mars-projekt von unity: Mixed and augmented reality studio, 5.3.2020.
- [Tec20g] Unity Technologies. Unity for instant games, 5.3.2020.
- [Tec20h] Unity Technologies. Use unity as a library, 5.3.2020.
- [Tec20i] Unity Technologies. PlayerPrefs: Unity - scripting api, 6.3.2020.
- [Tec20j] Unity Technologies. Vector3: Unity - scripting api, 6.3.2020.
- [Uni20a] Unity. Learning c# and coding in unity for beginners, 9.3.2020.
- [Uni20b] Unity Asset Store. simple icon pastel tone: 2d icons, 6.3.2020.