



Masterarbeit im Studiengang Game Engineering und Visual Computing

# Entwicklung einer HoloLens-Anwendung zur interaktiven Simulation des mittelal- terlichen Burglebens

Jonas Erkert

## Kurzbeschreibung

Durch die Entwicklung einer Augmented Reality Anwendung für die HoloLens 2 wird der Frage nachgegangen, welche Merkmale eine interaktive und immersive Anwendung auszeichnen und mit welchen aktuell zur Verfügung stehenden Mitteln sich diese erstellen lässt. Dabei wird auch auf die verwendete Hardware und Software eingegangen. Des Weiteren wird aufgezeigt, in welchen Bereichen Augmented Reality eingesetzt werden kann und welche Alternativen zu dieser Technologie zur Verfügung stehen.

Aufgabensteller/Prüfer: Prof. Dr. René Bühling

Arbeit vorgelegt am: 15.12.2021

Durchgeführt an der: Fakultät für Informatik

Anschrift des Verfassers: -

Matrikel-Nr.: -

E-Mail: -

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	<b>I</b>
<b>1 Einleitung.....</b>	<b>1</b>
1.1 Zielsetzung.....	1
1.2 Mixed Reality.....	2
1.3 Augmented Reality .....	2
<b>2 Methoden .....</b>	<b>4</b>
2.1 Software .....	4
2.2 Hardware .....	5
<b>3 Microsoft HoloLens.....</b>	<b>7</b>
3.1 Ein Vergleich von HoloLens 1 und HoloLens 2 .....	7
3.2 Alternative Augmented Reality HMD Geräte .....	10
3.2.1 Magic Leap .....	11
3.2.2 Third Eye Gen X2 MR Glasses.....	11
3.3 Design Leitlinien .....	13
3.3.1 Komfort .....	13
3.3.2 Audio .....	18
3.3.3 Spatial Mapping .....	19
3.4 Interaktionsmöglichkeiten und Multimodalität .....	21
3.4.1 Hände .....	21
3.4.2 Spracheingaben.....	25
3.4.3 Blickrichtung.....	25
3.5 User Experience Elemente .....	26
3.5.1 Farbe, Beleuchtung und Materialien .....	26
3.5.2 Skalierung.....	27
3.5.3 Sound .....	28
3.5.4 Interaktive Elemente und Funktionen .....	30

---

<b>4</b>	<b>Entwicklung einer eigenen Hololens Anwendung .....</b>	<b>35</b>
4.1	Designüberlegungen.....	35
4.1.1	Komfort .....	35
4.1.2	Audio .....	36
4.1.3	Spatial Mapping .....	36
4.1.4	Hände .....	37
4.1.5	Spracheingaben.....	37
4.1.6	Blickrichtung.....	38
4.1.7	Farbe, Beleuchtung und Materialien .....	38
4.1.8	Optimierung .....	40
4.1.9	Hand Menu.....	40
4.2	Technische Umsetzung.....	41
4.2.1	3D-Modelle und Texturen .....	41
4.2.2	Gameplay .....	45
<b>5</b>	<b>Augmented Reality in der Lehre .....</b>	<b>51</b>
5.1	Anwendung in der Lehre und E-Learning.....	51
5.1.1	Vor- und Nachteile .....	52
5.1.2	Zukunft im Lernbereich .....	55
5.2	Anwendungsbereiche.....	55
5.2.1	Medizin und Pflege.....	56
5.2.2	Ingenieurwesen und Industrie .....	56
5.2.3	Visualisierung von Geschichte und Kultur.....	57
<b>6</b>	<b>Alternativen zu Augmented Reality HMDs.....</b>	<b>59</b>
6.1	Virtual Reality .....	59
6.1.1	Anwendungsbereiche .....	59
6.1.2	Vor- und Nachteile .....	60
6.2	Handheld Augmented Reality .....	62
6.2.1	Anwendungsbereiche .....	63
6.2.2	Vor- und Nachteile .....	64

---

<b>7</b>	<b>Zusammenfassung</b> .....	<b>65</b>
<b>8</b>	<b>Diskussion</b> .....	<b>67</b>
<b>9</b>	<b>Abkürzungsverzeichnis</b> .....	<b>70</b>
<b>10</b>	<b>Glossar</b> .....	<b>71</b>
<b>11</b>	<b>Literaturverzeichnis</b> .....	<b>74</b>
	<b>Selbstständigkeitserklärung</b> .....	<b>79</b>

# 1 Einleitung

Die Grundlage für die Entwicklung von Augmented Reality (AR) und Virtual Reality (VR) legte Ivan Sutherland [1] bereits im Jahr 1968. Durch ein *Head-Mounted Display (HMD)* gelang es ihm, ein stereoskopisches Bild auf Monitore zu projizieren. Aus der Bewegung des Kopfes wurde dabei die Perspektive des Nutzers berechnet. Dadurch konnten einfache dreidimensionale Objekte, wie zum Beispiel ein Würfel, dargestellt werden.

Mit der Präsentation der von Microsoft entwickelten HoloLens im Jahr 2015 [2], 47 Jahre nach der Forschung von Sutherland, wurde eine Augmented Reality Brille vorgestellt, welche unter anderem eine hohe Auflösung, ein geringes Gewicht sowie eine kabellose Funktionalität vorweisen kann. Augmented Reality Anwendungen durch *HMDs* konnten damit einer größeren Anzahl an Personen zur Verfügung gestellt werden. Hierfür wurden Entwicklungswerkzeuge erstellt, womit die Entwicklung von Augmented Reality Anwendungen möglichst einfach durchgeführt werden kann und viele Menschen von dieser Technologie profitieren können.

Mit der Veröffentlichung der HoloLens 2, welche weitere Fortschritte bezüglich der technischen Eigenschaften besitzt, stellt sich daher die Frage, in welchen Bereichen aktuelle Augmented Reality Geräte vorteilhaft eingesetzt werden können. Wie fortgeschritten ist die Hardware, aber auch die Software, mit der Anwendungen erstellt werden können? Auf welche Grundlagen muss bei der Entwicklung geachtet werden?

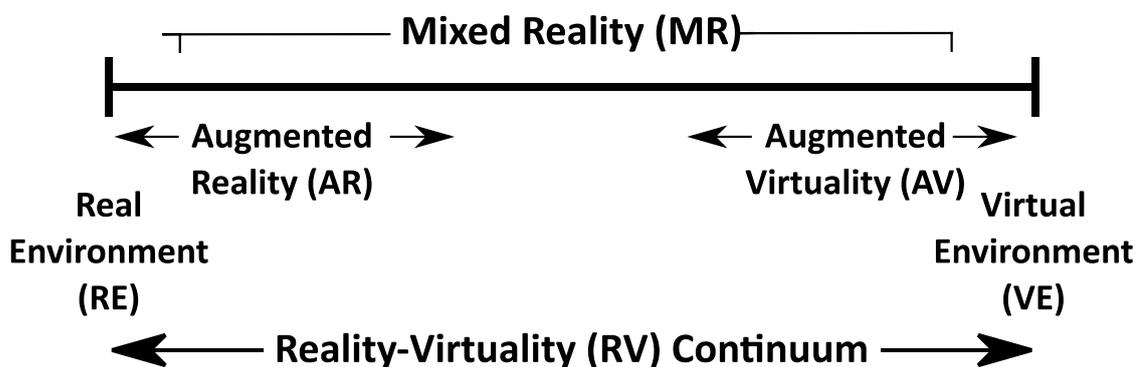
## 1.1 Zielsetzung

Das Ziel dieser Masterarbeit ist die Entwicklung einer Augmented Reality Anwendung, welche basierend auf den aktuell zur Verfügung stehenden Leitlinien, bezüglich des Komforts, der Interaktionsmöglichkeiten und dem *User Experience (UX) Design*, eine möglichst interaktive sowie immersive Nutzererfahrung bietet. Die entwickelte Anwendung soll eine beispielhafte Umsetzung dieser Leitlinien demonstrieren und den Stand der zum aktuellen Zeitpunkt zur Verfügung stehenden Hardware und Software für Augmented Reality verwenden. Da ein historisches Setting als Basis der Anwendung gewählt wurde, soll außerdem der Einsatz von Augmented Reality in der Lehre bewertet werden. Da, neben am Kopf befestigten Augmented Reality Geräten, auch ähnliche Technologien zur Verfügung stehen, wird auf diese im Rahmen dieser Masterarbeit ebenfalls näher eingegangen.

Als Augmented Reality Hardware wurde die von Microsoft entwickelte HoloLens 2 verwendet. Für die Entwicklung der Software wurde die Unreal Engine mit unterschiedlichen Plugins für diese genutzt, welche die Arbeit mit der Spiele-Engine und der HoloLens 2 ermöglichen und erleichtern.

## 1.2 Mixed Reality

Das Konzept von Mixed Reality wurde von Ivan Sutherland geprägt, welcher 1968 erstmals ein am Kopf befestigtes Display entwickelte, welches die reale Welt überlagernde, virtuelle 3D-Objekte darstellen konnte [3]. Für die Einordnung von Mixed Reality erstellten Milgram et al. das Reality-Virtuality Continuum, welches in Abbildung 1 dargestellt wird. Der Übergang zwischen realen Umgebungen, welche mit virtuellen Objekten erweitert werden, und einer virtuellen Umgebung, welche mit realen Daten erweitert wird, ist fließend und schwierig einzuordnen. Anwendungsfälle werden daher in die Bereiche Augmented Reality und Augmented Virtuality unterteilt. Die Definition von Mixed Reality schließt diese beiden Bereiche, jedoch nicht das Real Environment und das Virtual Environment, ein [4].



**Abbildung 1:** Das von Milgram et al. zur Definition von Mixed Reality erstellte Reality-Virtuality Continuum [4] (verändert).

Die Begriffe Augmented Reality und Mixed Reality werden heute oft als Synonym verwendet. Augmented Reality bezieht sich jedoch auf die Überlagerung von Daten über die reale Welt, während Mixed Reality Technologien virtuelle Objekte über einen realen Hintergrund legen. Die HoloLens von Microsoft ist ein Beispiel für ein Augmented Reality Gerät, während das Handyspiel „Pokémon Go“ ein Beispiel für eine Augmented Reality Anwendung darstellt [5].

## 1.3 Augmented Reality

Augmented Reality integriert digitale Informationen und Objekte in die reale Welt, wobei diese in Echtzeit verarbeitet und generiert werden. Dadurch erhöht sich die Informationsmenge, welche ein Mensch der Umgebung entnehmen kann. Die Eigenschaften eines Augmented Reality Systems werden von Azuma et al. [5] definiert. Dabei wird beschrieben, dass Augmented Reality reale und virtuelle Objekte in einer realen Umgebung kombiniert. Weitere Eigenschaften von Augmented Reality Anwendungen sind die Möglichkeiten zur Interaktion und Integration der virtuellen Objekte im dreidimensionalen Raum. Alle Systeme, welche diese drei genannten Eigenschaften besitzen, werden auch als Augmented Reality Systeme bezeichnet. Andere Autoren haben die Definition von Augmented Reality erweitert. Beispielsweise benötigt ein Augmented

Reality System nicht zwangsweise ein am Kopf angebrachtes Display, um die erzeugten virtuellen Objekte oder Informationen zu betrachten. Das Augmented Reality System muss die reale Welt interpretieren und alle zur Verfügung stehenden Informationen in der Umgebung anbringen. Während die Umgebung in Virtual Reality komplett vordefiniert ist, kann sich die reale Umgebung bei einer Augmented Reality Anwendung ständig verändern und ist damit unberechenbar [5]. Milgram et al. dagegen beschreiben zwei Definitionsbereiche für Augmented Reality. Der erste Definitionsbereich bezieht sich auf Displays, welche am Kopf befestigt sind. Diese Art von Displays werden *Head-Mounted-Displays (HMDs)* genannt. Informationen werden durch das *Heads-Up-Display (HUD)* dargestellt. Im Fall von Augmented Reality *HMDs* betrachtet der Nutzer die reale Welt über ein spezielles Display. Dies ist entweder ein transparentes Display oder ein Bildschirm auf dem ein in Echtzeit erzeugtes Video, auf welches virtuelle Objekte überlagert werden, abgespielt wird. Der zweite Definitionsbereich beschreibt dagegen, dass Augmented Reality auch ohne *HMDs* auskommt und deckt alle Fälle ab, in denen eine reale Umgebung durch virtuelle Objekte erweitert wird. Dadurch können auch Monitor-basierte Displays, wie zum Beispiel Smartphones, als Augmented Reality Geräte gewertet werden [4].

Augmented Reality wird genutzt, um die Wahrnehmung des Nutzers in der realen Welt zu erweitern. „Pokémon Go“ ist ein Augmented Reality Spiel, welches auf Smartphones gespielt werden kann. Das Spiel wurde 2016 veröffentlicht und innerhalb von zwei Monaten weltweit über 500 Millionen Mal heruntergeladen. Obwohl „Pokémon Go“ nicht das erste Augmented Reality Spiel auf dem Markt war, erlangte bisher keine andere Anwendung in diesem Bereich eine solche weite Verbreitung. Dies lässt sich durch die stetig wachsende Verfügbarkeit an Smartphones erklären. Smartphones besitzen die notwendigen Sensoren, wie beispielsweise ein Gyroskop, ein Beschleunigungssensor und eine Kamera, wodurch eine virtuelle Wahrnehmung der realen Welt ermöglicht wird. Diese Informationen können vom Smartphone genutzt werden, um die reale Welt durch zusätzliche virtuelle Objekte oder Informationen zu erweitern [6].

## 2 Methoden

Vor dem Beginn der Anwendungsentwicklung wurde zunächst ein Game Design Dokument erstellt, in welchem die von Microsoft dokumentierten Design-Leitlinien aufgelistet sind. Zu diesen Leitlinien gehört der Komfort, die *UX* und die unterschiedlichen Interaktionsmöglichkeiten, welche bei der Entwicklung von Augmented Reality Anwendungen beachtet werden sollten. Zusätzlich werden mögliche Funktionen der HoloLens 2, sowie technische Spezifikationen und Einschränkungen im Game Design Dokument aufgelistet. Dadurch konnte noch vor der Entwicklung auf diese Leitlinien geachtet und Funktionen entsprechend implementiert werden. Des Weiteren konnte geplant werden, welche Interaktionsmöglichkeiten bei welcher Funktionalität eingesetzt werden können, um eine immersive und möglichst interaktive Augmented Reality Anwendung zu erstellen.

Anschließend wurde das Gameplay, der Spielablauf und mögliche Gebäude im Spiel definiert. Um in der Anwendung platzierbare Gebäude und den Stil dieser zu definieren, wurde nach verschiedenen historischen Burgen geforscht. Als Inspiration für Gebäude dienten unter anderem der Tower of London, die Kaiserburg Nürnberg, die Burg Hohenzollern, die Burg Burghausen und das Hohe Schloss Füssen. Durch Grundrisse dieser Burgen wurde eine Liste möglicher Gebäude erstellt. Die gesammelten Gebäude dieser Liste wurden anschließend priorisiert, damit die Funktionalität wichtiger Gebäude implementiert und 3D-Modelle für diese vorrangig erstellt werden konnten. Für die Gebäude wurde anschließend eine Anzahl und die Aufgaben der zugehörigen Künstliche Intelligenz (*KI*)-Charaktere definiert, wie zum Beispiel welche anderen Gebäude diese *KI*-Charaktere besuchen können. Zusätzlich wurde nach dem historischen Nutzen der Gebäude geforscht, um dem Spieler zusätzliche Informationen zu vermitteln und damit eine Lernerfahrung zu erzeugen. Diese gesammelten Kriterien wirkten sich auf das Aussehen der 3D-Modelle der Gebäude aus.

Einige Elemente des Game Design Dokuments, beispielsweise der Spielablauf, wurden während der Entwicklung an die Gegebenheiten angepasst. Weiterhin wurden Interaktionsmöglichkeiten, beispielsweise der Einsatz von *Eye-Tracking*, verändert, da diese in der Praxis andere Auswirkungen als zunächst geplant hatten.

### 2.1 Software

Die Implementierung der Anwendung wurde in der von Epic Games entwickelten Unreal Engine 4.26 umgesetzt. Diese Spiele-Engine wurde ausgewählt, da der Unreal Editor mit zahlreichen Funktionen ausgestattet ist, welche eine effiziente Erstellung einer Augmented Reality Anwendung erlauben. Dazu zählen die *Behavior-Tree* Funktionalität, welche zur Erstellung von *KI* genutzt werden kann, ein Graph-basiertes Materialsystem und das *Blueprint Visual Scripting* (*Blueprints*, *BP*) System, welches eine schnelle Iteration von Gameplay-Logik erlaubt. Durch Nutzung der Unreal Engine lässt sich die Logik einer Anwendung neben *BP* auch mit C++ umsetzen. Im Rahmen der für diese Masterarbeit erstellte Anwendung wurde das *BP* System gewählt, da die

benötigte Gameplay-Logik damit, in den meisten Fällen, schneller umgesetzt werden kann. Für komplexe mathematische Berechnungen wird das Programmieren in C++ empfohlen. Da jedoch keine komplexen Berechnungen für das Gameplay in der Anwendung nötig waren, wurde ausschließlich *BP* genutzt.

Die Entwicklung von HoloLens 2 Anwendungen in der Unreal Engine wird seit der Version 4.23 unterstützt [7]. Um Anwendungen für Augmented Reality mit der Unreal Engine zu entwickeln, stehen zwei Plugins zur Verfügung. Einerseits das Windows Mixed Reality, andererseits das vor kurzem veröffentlichte OpenXR Plugin. Während beide Plugins eine ähnliche Funktionalität bieten, unterstützt das OpenXR Plugin viele unterschiedliche Mixed Reality Geräte. Beide Plugins wurden für die Anwendung in dieser Masterarbeit getestet. Da jedoch der Einsatz des OpenXR Plugins bei der Generierung einer *spatial Map* Probleme verursachte, wird in dieser hier vorliegenden Anwendung das Windows Mixed Reality Plugin verwendet.

Für die in der Entwicklung von Augmented Reality Anwendungen häufig verwendeten Funktionen und Grafik-Effekte wurde von Microsoft das *Mixed Reality Toolkit (MRTK)* entwickelt und auf GitHub veröffentlicht. Das *MRTK* beinhaltet die Plugins *UX Tools (UXT)* [8] und *Graphics Tools (GT)* [9]. Die für die Augmented Reality Entwicklung wichtigen Elemente, wie Input Simulation, Hand Interaktionen, interaktive Buttons, Hand und Near Menus, sind unter anderem im *UXT* Plugin enthalten, während das *GT* Plugin beispielsweise Performance-optimierte Shader und einen visuellen Performance-Profiler mit *Tag-Along* Verhalten bereitstellt.

Obwohl die Gameplay-Logik der vorliegenden Anwendung ohne C++ Code entwickelt wurde, benötigt das Projekt, durch die Verwendung der *MRTK* Plugins sowie für das Packaging zu einer HoloLens 2 Anwendung, dennoch eine Entwicklungsumgebung (*IDE*) und den MSVC Compiler. Aus diesem Grund wurde als *IDE* Visual Studio 2019 genutzt.

Für die Erstellung von 3D-Modellen wurde Blender genutzt, während Texturen mit Hilfe des Programms Substance Designer generiert wurden.

Um die Implementierung und die Arbeitsschritte der Anwendung klar zu definieren, wurde der *Issue-Tracker* YouTrack von JetBrains genutzt. Als Versionsverwaltungssystem (*VCS*) wurde Git verwendet und als Serverdienst wurde dabei GitLab genutzt. Die erstellte Verbindung von GitLab und YouTrack erlaubt eine effiziente Bearbeitung von *Issues* und *Commits*. Dabei kann beispielsweise über *Commit*-Nachrichten ein *Issue* aktualisiert oder geschlossen werden, während in YouTrack spezifische *Commits* verlinkt werden können. Dadurch kann eine Übersicht über den aktuellen Aufgabenstand gewonnen werden.

## 2.2 Hardware

Ein Großteil der in dieser Anwendung implementierten Funktionen konnte mit Hilfe des *UXT* Plugins im Unreal Engine Editor durch Input Simulation überprüft werden. Für das Testen von nicht simulierbaren Funktionen, wie zum Beispiel der Performance auf dem Gerät selbst oder der Generierung einer *spatial Map*, wurde jedoch die HoloLens 2 verwendet. Dabei wurde eine

---

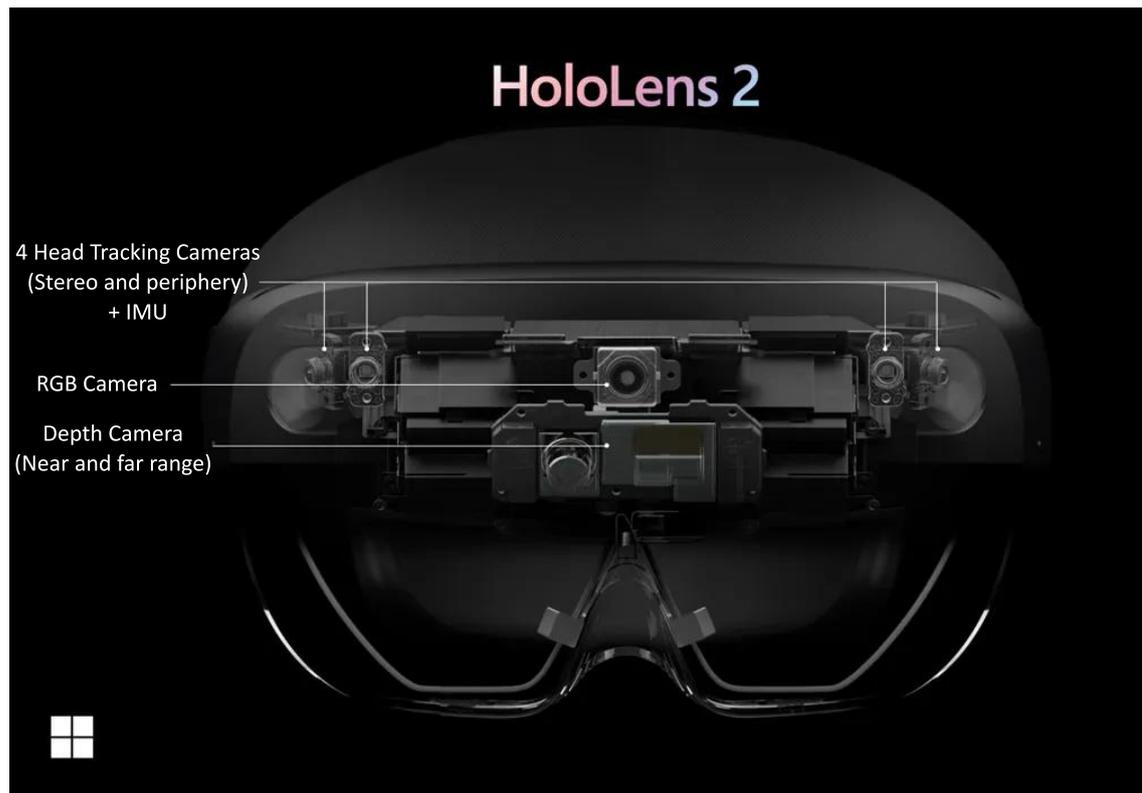
Mischung aus Streaming, also die kabellose Verbindung von einem PC zur HoloLens 2 via WLAN, und eine lokale Installation der Anwendung auf dem Gerät verwendet. Technische Spezifikationen der HoloLens 2 werden im Abschnitt 3.1 aufgelistet. Zur Entwicklung der Anwendung in der Unreal Engine auf dem PC wurde der Prozessor Intel Core i9-9900k und als Grafikkarte eine Nvidia RTX 2080Ti verwendet, wodurch ein effizientes Arbeiten in den unterschiedlichen Programmen, wie beispielsweise im Unreal Editor, in Blender und in Substance Designer, ermöglicht wurde.

## 3 Microsoft HoloLens

Für den praktischen Teil dieser Arbeit wurde die HoloLens 2 verwendet, welche zu diesem Zeitpunkt das neueste Modell der beiden von Microsoft entwickelten Augmented Reality Brillen ist. Da die HoloLens 2 im Vergleich zu ihrem Vorgänger, der HoloLens 1, einige Neuerungen erhalten hat, wurden die Möglichkeiten, eine immersive Augmented Reality Erfahrung zu erstellen, deutlich verbessert. Auf diese Neuerungen und die daraus resultierenden Möglichkeiten wird im folgenden Abschnitt eingegangen. Alternativen zur HoloLens 2 und der technische Stand dieser Konkurrenzprodukte werden in Abschnitt 3.2 vorgestellt.

### 3.1 Ein Vergleich von HoloLens 1 und HoloLens 2

Die HoloLens der ersten Generation wurde im Januar 2015 von Microsoft vorgestellt [2]. Käuflich wurde diese Augmented Reality Brille 2016. Drei Jahre nach dem Marktstart der ersten HoloLens stellte Microsoft auf dem Mobile World Congress die HoloLens 2 vor. Diese wurde um einige neue Funktionen erweitert. Unter anderem wurde der Tragekomfort erhöht [10]. Neue Funktionen wie *Eye-Tracking* und die Erkennung beider Hände des Nutzers ermöglichen neue Interaktionen, welche auf der HoloLens 1 nicht durchführbar sind. Beide HoloLens-Versionen besitzen eine *Holographic Processing Unit (HPU)*, welche für die Generierung von 3D-Modellen der Umgebung zuständig ist. Dabei benötigt die *HPU* Daten, welche von der *Inertial Measurement Unit (IMU)*, vier *Head-Tracking* Kameras und einer Tiefenkamera generiert werden. Um die Bewegungen des Nutzers in der realen Welt wahrzunehmen, besitzt die HoloLens eine Reihe von Sensoren. In Abbildung 2 wird die Positionierung der verschiedenen Kameras und der *IMU* dargestellt. Die HoloLens 2 nutzt die von den Sensoren und den Kameras erzeugten Informationen, um Bilder und damit Hologramme auf dem Display zu rendern. Mit den virtuellen Objekten kann der Nutzer dadurch interagieren und diese aus unterschiedlichen Blickwinkeln betrachten [11].



**Abbildung 2:** Frontansicht der HoloLens 2 sowie der Positionierung der Kameras und der *IMU*, welche die nötigen Informationen für die *HPU* erzeugen [12].

Die HoloLens 2 verbessert fast jeden Aspekt der ersten HoloLens. Die Auflösung des Displays wurde von 1280 x 720 Pixel auf 2048 x 1080 Pixel erhöht, wodurch höher aufgelöste Hologramme gerendert werden können. Mit einem *Field of View (FOV)* von 52 Grad wurde dieses in Bezug auf die HoloLens 1 nahezu verdoppelt, wodurch der Nutzer mit Hologrammen einfacher interagieren kann [13]. Die HoloLens 2 verfügt außerdem über eine zusätzliche Infrarotkamera, welche für das *Eye-Tracking* eingesetzt wird. Durch die Einführung von *Eye-Tracking* auf der HoloLens 2 kann der Nutzer nun mit Hologrammen über Blickkontakt interagieren. Die Verbesserung von *Hand-Tracking* erlaubt der HoloLens 2 außerdem neue Gesten- und Interaktionsmöglichkeiten. Die neuen Gesten sind so konzipiert, dass diese sich so intuitiv und natürlich wie möglich für den Nutzer anfühlen. Tabelle 1 stellt die unterschiedlichen Funktionen und Eigenschaften der HoloLens 1 und der HoloLens 2 gegenüber [14].

**Tabelle 1:** Ein Vergleich der Display-Eigenschaften und anderer Fähigkeiten der HoloLens 1 und der HoloLens 2 [12–16].

	HoloLens 1	HoloLens 2
<b>Optik</b>	Transparente holographische Linsen (Wellenleiter)	Transparente holographische Linsen (Wellenleiter)
<b>Auflösung</b>	1280 × 720 Pixel 16:9 Lichtgenerator (pro Auge)	2048 × 1080 Pixel 3:2 Lichtgenerator (pro Auge)
<b>Holographische Dichte</b>	>2500 Radiant (Lichtpunkte pro Bogenmaß)	>2500 Radiant (Lichtpunkte pro Bogenmaß)
<b>Augenbasierte Darstellung</b>	Automatische Kalibrierung des Pupillenabstands	Display-Optimierung für 3D-Augenposition
<b>Field of View</b>	34°	52°
<b>Head-Tracking</b>	Ja	4 Kameras für sichtbares Licht
<b>Eye-Tracking</b>	Nicht vorhanden	2 Infrarotkameras
<b>Hand-Tracking</b>	Gesten-Erkennung	Beidhändig voll bewegliches Modell, direkte Manipulation
<b>Spracheingabe</b>	Ja	Befehle und Steuerung auf dem Gerät, natürliche Sprache mit Internetverbindung
<b>Tiefenwahrnehmung</b>	Tiefenkamera	1-MP-Time-of-Flight-(ToF)-Tiefensensor
<b>IMU</b>	Ja	Beschleunigungsmesser, Gyroskop, Magnetometer
<b>Kamera</b>	2MP Foto, 1080p Video	8MP Foto, 1080p30 Video
<b>Über Brille tragbar</b>	Ja	Ja
<b>Hochklappbares Display</b>	Nein	Ja

Die Rechenleistung der HoloLens 2 wurde in Bezug auf ihr Vorgängermodell drastisch verbessert. In Tabelle 2 werden die technischen Spezifikationen der HoloLens 1 und HoloLens 2 gegenübergestellt. Die integrierte *HPU*-Technologie übernimmt die räumliche Berechnung, wodurch die Belastung der CPU drastisch verringert wird. Eine verbesserte *HPU* bedeutet, dass eine leistungsfähigere Rechenleistung, in Bezug auf Grafik und Berechnungen, bereitgestellt werden kann. Die HoloLens 2 vergrößert den Arbeitsspeicher von 2 GB auf 4 GB. Durch die Erhöhung des Arbeitsspeichers kann die HoloLens 2 komplexere Anwendungen verarbeiten. Bereits die erste HoloLens wurde mit Bluetooth- und WLAN-Funktionalität ausgestattet. Beide Technologien wurden bei der HoloLens 2 auf den neusten Stand aktualisiert, wodurch eine effizientere Verbindung mit anderen Geräten ermöglicht wird. Weitere Verbesserungen haben den Tragekomfort der HoloLens 2 erhöht. Beispielsweise wird durch die Verschiebung der Batterie an die Rückseite die Balance des Geräts verbessert. Außerdem wurde das Gewicht reduziert, wodurch ein Nutzer die HoloLens 2 über einen längeren Zeitraum komfortabel tragen kann [14].

**Tabelle 2:** Ein Vergleich der technischen Daten der HoloLens 1 und der HoloLens 2 [12,14–16].

	<b>HoloLens 1</b>	<b>HoloLens 2</b>
<b>Prozessor</b>	Intel Atom x5-Z8100P, 4-Kern CPU	Qualcomm Snapdragon 850, 8-Kern CPU
<b>HPU</b>	Spezielle von Microsoft entwickelte Verarbeitungseinheit ( <i>HPU</i> 1.0)	Spezielle von Microsoft entwickelte Verarbeitungseinheit ( <i>HPU</i> 2.0)
<b>Arbeitsspeicher</b>	2 GB	4 GB
<b>Speicher</b>	64 GB	64 GB
<b>Mikrofon-Array</b>	4 Kanäle	5 Kanäle
<b>Lautsprecher</b>	Spatial Audio	Spatial Audio
<b>WLAN</b>	Wi-Fi 802.11ac	Wi-Fi 5 (802.11ac 2x2)
<b>Bluetooth</b>	Bluetooth 4.1 LE	Bluetooth 5
<b>USB</b>	Micro USB 2.0	USB Typ-C
<b>Gewicht</b>	579 g	566 g
<b>Akkulaufzeit</b>	2-3 Stunden aktive Nutzung	2-3 Stunden aktive Nutzung

### 3.2 Alternative Augmented Reality HMD Geräte

Neben der HoloLens 1 und der HoloLens 2 von Microsoft bieten auch andere Unternehmen am Kopf befestigte Augmented Reality Systeme an. In diesem Abschnitt wird daher auf zwei alternative Geräte zur HoloLens und den technischen Spezifikationen dieser Konkurrenzmodelle eingegangen.

### 3.2.1 Magic Leap

Die Augmented Reality Brille Magic Leap 1 vom gleichnamigen Hersteller wirbt mit speziellen Sensoren für Umgebungserkennung, einer leistungsstarken 3D-Visualisierung und einem *multi-modalen* Input-System, um unterschiedliche Interaktionen zu ermöglichen. Die Brille und der dazugehörige Controller werden in Abbildung 3 dargestellt. Der Controller erlaubt eine präzise Manipulation mit einem haptischen Feedback. Die Brille verfügt zusätzlich über *Hand-* und *Eye-Tracking*. Anwendungen für die Magic Leap 1 können außerdem die Spracherkennung des Geräts nutzen.



**Abbildung 3:** Die Magic Leap 1 Augmented Reality Brille mit dem dazugehörigen Controller [17].

Eine Auflösung von 1280 x 960 Pixel pro Auge soll eine qualitative Darstellung von virtuellen Objekten garantieren. Die Brille kann optional mit einem Computer durch Bluetooth oder WLAN verbunden werden. Ein Gewicht von 316 Gramm und eine Akkulaufzeit von 3,5 Stunden sorgt für ein komfortables Tragen des Augmented Reality Geräts über einen längeren Zeitraum. Als Betriebssystem verwendet die Brille Lumin OS [17]. Die in Abschnitt 3.2.2 erstellte Tabelle 3 bietet einen Überblick über die technischen Spezifikationen der Magic Leap 1 und vergleicht diese mit weiteren, auf dem Markt verfügbaren, Augmented Reality Geräten.

### 3.2.2 Third Eye Gen X2 MR Glasses

Ein weiterer Hersteller eines Augmented Reality Systems ist das Unternehmen Third Eye Gen. Mit dem Gerät X2 MR Glasses, welches in Abbildung 4 dargestellt wird, verspricht der Hersteller eine Mixed Reality Brille mit wenig Gewicht, einem großen *FOV* und leistungsstarken Sensoren [18].



**Abbildung 4:** Die Augmented Reality Brille mit dem Namen X2 MR Glasses wird vom Unternehmen Third Eye Gen entwickelt [19].

Das X2 MR Glasses ist in der Lage, Gesten und die Blickrichtung des Nutzers zu erkennen. Alternativ kann über einen kabellosen Controller interagiert werden. Des Weiteren kann über Lautsprecher Audio wiedergegeben werden. Die Brille verfügt über *Inside-Out 6DOF Tracking* und kann in Innenräumen sowie im Außenbereich eingesetzt werden. Das Betriebssystem dieser Augmented Reality Brille basiert auf Android [20]. Andere technische Spezifikationen werden in Tabelle 3 aufgelistet.

**Tabelle 3:** Spezifikationen der Magic Leap 1 [17] und der X2 MR Glasses [20] im Vergleich zur Microsoft HoloLens 2 [12–16].

	<b>Magic Leap 1</b>	<b>X2 MR Glasses</b>	<b>HoloLens 2</b>
<b>Auflösung</b>	1280 x 960 Pixel (pro Auge)	1280 x 720 Pixel (pro Auge)	2048 x 1080 Pixel (pro Auge)
<b>Field of View</b>	50°	42°	52°
<b>Sensoren</b>	Beschleunigungsmesser, Magnetometer	Unter anderem Wärmebildsensor	Beschleunigungs- messer, Gyroskop, Magnetometer
<b>Kamera</b>	2 Megapixel, 1080p30 Video	13 Megapixel	8 Megapixel, 1080p30 Video
<b>Arbeitsspeicher</b>	8 GB	8 GB	4 GB
<b>Speicher</b>	128 GB	64 GB	64 GB

<b>Lautsprecher</b>	Spatial Audio, Eingebaute Stereo-Lautsprecher	Keine Angaben	Spatial Audio
<b>WLAN</b>	IEEE 802.11a/g/b/n/ac, dual-band	Vorhanden, keine näheren Angaben	Wi-Fi 5 (802.11ac 2x2)
<b>Bluetooth</b>	Bluetooth 4.2	Vorhanden, keine näheren Angaben	Bluetooth 5
<b>USB</b>	USB Typ-C	USB Typ-C	USB Typ-C
<b>Gewicht</b>	316 g	170 g	566 g
<b>Akkulaufzeit</b>	3,5 Stunden aktive Nutzung	Keine Angaben	2-3 Stunden aktive Nutzung

### 3.3 Design Leitlinien

Um eine effiziente, komfortable und immersive Augmented Reality Anwendung zu entwickeln, werden von Microsoft unterschiedliche Leitlinien empfohlen. Auf einige dieser Leitlinien wird in diesem Abschnitt eingegangen.

#### 3.3.1 Komfort

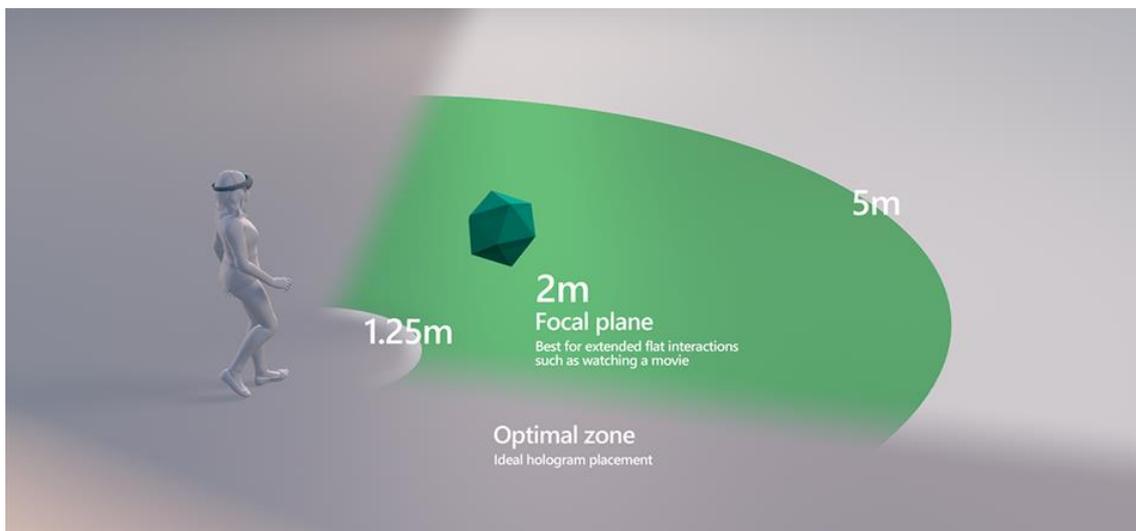
Das menschliche Gehirn verlässt sich beim Sehen auf mehrere Informationsquellen, um 3D Objekte und die relative Position dieser Objekte zueinander zu deuten. Einige dieser Hinweise basieren auf dem Input eines einzigen Auges und werden monokulare Hinweise genannt. Einer dieser Hinweise ist die lineare Perspektive. Bei dieser Art der Hinweisverarbeitung werden vertraute Größen, Verdeckung von Objekten, Tiefunschärfe und *Akkommodation* als Informationsquellen genutzt. Andere Hinweise benötigen jedoch beide Augen und werden binokulare Hinweise genannt. Dazu zählen die *Konvergenz* und das *stereoskopische Sehen*.

Um den bestmöglichen Komfort von *HMDs* zu gewährleisten, sollte der Inhalt einer Augmented Reality-Anwendung so erstellt und präsentiert werden, dass dieser die zuvor genannten, vom Menschen benötigten Hinweise, nachahmt. Neben Designüberlegungen, bezogen auf die Kompatibilität des *HMDs* mit der Sicht des Menschen, sollten weiterhin physische Aspekte beachtet werden, welche beispielsweise ermüdende Bewegungen in den Armen oder im Nacken vermeiden.

Um Objekte deutlich zu sehen, müssen Menschen den Fokus ihrer Augen an die Distanz des Objekts anpassen. Gleichzeitig muss die Rotation beider Augen zur Distanz des Objekts konvergieren, damit Bilder nicht doppelt gesehen werden. Bei den meisten *HMDs* passen sich die Augen des Nutzers an die Brennweite des Displays an, um ein scharfes Bild zu erlangen. Dabei konvergieren die Augen zur Entfernung des Objektes, um ein einzelnes Bild zu sehen. *Akkommodieren* und konvergieren die Augen der Nutzer jedoch auf eine unterschiedliche Distanz, wird die

natürliche Verbindung dieser beiden Hinweise gebrochen, was zu Unwohlsein oder Müdigkeit führen kann.

Für den höchsten Komfort sollte ein Hologramm im optimalen Bereich platziert werden, welcher zwischen 1,25 Metern und 5 Metern liegt. Die optimale Darstellung von Hologrammen liegt bei der HoloLens 2 bei einer Distanz von etwa 2 Metern. Dieser Bereich wird *Focal Plane* genannt. Abbildung 5 zeigt den optimalen Bereich und die *Focal Plane*. Die *Focal Plane* eignet sich für andauernde, einfache Interaktionen mit Hologrammen. Für manche Anwendungen müssen Objekte jedoch in unter 1 Meter Distanz vom Nutzer platziert werden. Dabei sollten die Objekte aber nie näher als 40 Zentimeter vom Nutzer entfernt dargestellt werden. Um das Rendering von Objekten mit einer geringeren Distanz zu vermeiden, kann die *Near Clipping Plane* auf diese Distanz konfiguriert werden. Die Augen der Nutzer müssen sich an die Distanz der Hologramme *akkommodieren*, um ein klares Bild zu erhalten. Entwickler können dabei den Punkt, an dem die Augen konvergieren, beeinflussen, indem die Hologramme in unterschiedlichen Tiefen platziert werden. Das Unwohlsein, welches durch den Konflikt zwischen der *Akkommodation* und der *Konvergenz* entsteht kann vermieden, beziehungsweise verringert werden, indem interessante Inhalte so nahe wie möglich in 2 Meter Distanz platziert werden. Werden Inhalte nicht im optimalen Bereich platziert, so ist das Unwohlsein des Nutzers am größten, wenn der Blick häufig zwischen den großen unterschiedlichen Distanzen wechselt.

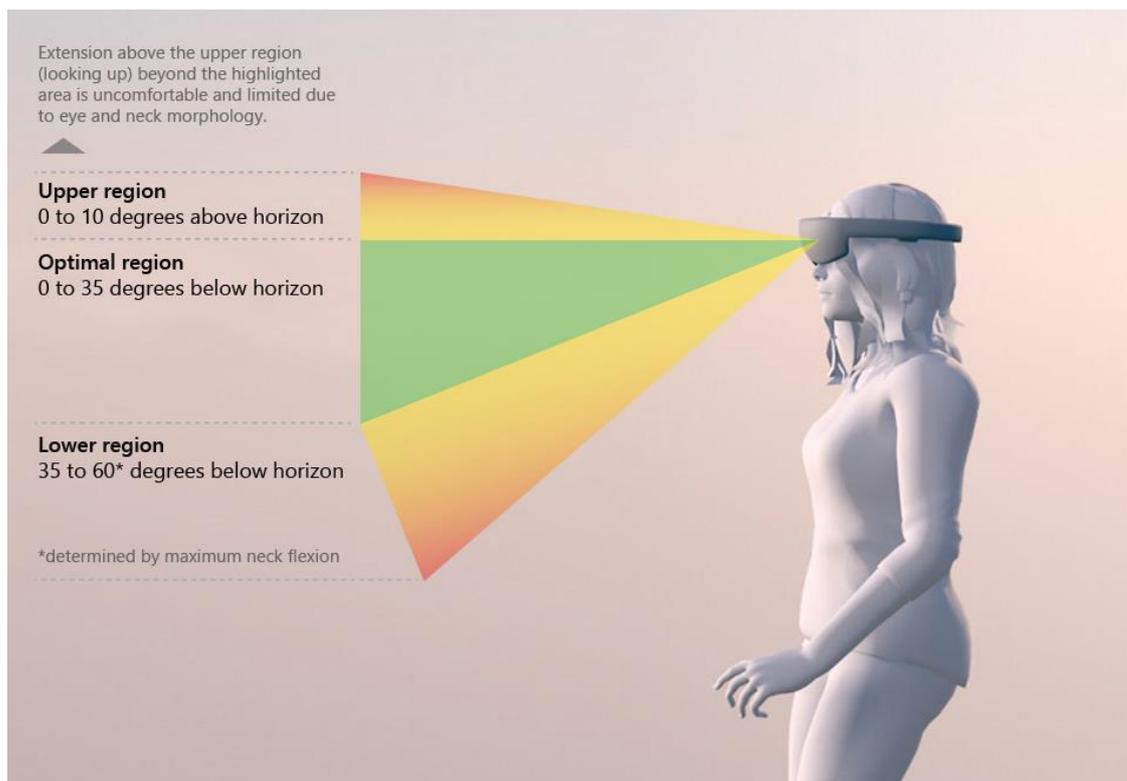


**Abbildung 5:** Darstellung des optimalen Bereichs, in dem Hologramme gerendert werden sollten, um einen hohen Komfort beim Umgang mit Hologrammen zu ermöglichen [21].

Werden Anwendungen durch ein *HMD* betrachtet, so ist die Position der Augen des jeweiligen Nutzers, basierend auf der Display-Position, entscheidend. Sowohl der Pupillenabstand als auch der vertikale Abstand, sind maßgeblich für ein komfortables Nutzen von *HMDs*. Der Pupillenabstand bezieht sich auf den Abstand zwischen den Pupillen einer Person, während sich der vertikale Abstand auf den möglichen vertikalen Abstand des digitalen Inhalts, welcher jedem Auge relativ zur horizontalen Achse der Augen des Betrachters angezeigt wird, bezieht. Wird einer

oder beide dieser Abstände nicht auf den Nutzer einer Anwendung abgestimmt, so kann dies die Auswirkungen des Unwohlseins, welche durch den *Konvergenz-Akkommodations*-Konflikt entstehen, verstärken. Auch für den Fall, dass der Konflikt minimiert werden kann, beispielsweise wenn Inhalte in der optimalen Entfernung von 2 Metern dargestellt werden, können diese Faktoren Unwohlsein verursachen. Durch die HoloLens 2 wird der Pupillenabstand eines Nutzers geschätzt und bei der Kalibrierung des Geräts eingerichtet. Für neue Nutzer der HoloLens 2 ist daher eine Kalibrierung dringend empfohlen, damit der Pupillenabstand korrekt eingestellt werden kann. Der vertikale Abstand wird dabei von der HoloLens 2 automatisch passend zum Nutzer eingerichtet.

Bei großen oder vielen Objekten in einer Mixed Reality Anwendung sollte berücksichtigt werden, wie stark der Nutzer den Kopf und Nacken bewegen muss, um mit diesen Objekten zu interagieren. Dabei lässt sich die Art der Kopfbewegung in drei Kategorien unterteilen. Erstens vertikale Bewegungen, bei welchen der Kopf von oben nach unten bewegt wird, zweitens horizontale Bewegungen, bei denen der Nutzer von links nach rechts schaut und drittens immersive Bewegungen, welche die gleichzeitige horizontale und vertikale Kopfbewegung des Nutzers beschreiben, um den Inhalt einer Anwendung zu erfassen. Microsoft empfiehlt die meisten Interaktionen entweder auf die horizontale oder vertikale Kategorie zu beschränken. Optimalerweise finden die meisten Interaktionen dabei im Zentrum des *holografischen Frames* statt, während sich der Kopf des Nutzers in einer neutralen Position befindet. Inhalte einer Anwendung sollten so gestaltet werden, dass eine übermäßige Belastung von Augen und Nacken vermieden wird. Abbildung 6 zeigt den optimalen Bereich, in welchem Inhalte einer Anwendung gerendert werden sollten. Dieser Bereich liegt zwischen 0 und 35 Grad unterhalb des Horizonts, wobei der Blickwinkel im besten Fall zwischen 10 und 20 Grad liegt. In diesem Bereich ist der Kopf leicht nach unten geneigt und befindet sich in einer Ruheposition. Werden Inhalte bei einem Blickwinkel von mehr als 10 Grad über oder 60 Grad unter dem Horizont dargestellt, so kann dies zu häufigen Kopf- und Nackenbewegungen und dadurch zu Unwohlsein führen. Eine Rotation des Nackens von mehr als 45 Grad entlang der horizontalen Achse sollte ebenfalls vermieden werden. Zusammenfassend sollten daher Interaktionen, bei denen der Nutzer ständig seine Ansicht in eine unnatürliche Kopfposition bewegen muss, vermieden werden.



**Abbildung 6:** Im *holografischen Frame* befinden sich unterschiedliche Sichtbereiche, in welche die Inhalte einer Anwendung gerendert werden sollten [21].

Häufige Interaktionen sollten vor allem durch horizontale Kopfbewegungen stattfinden, während vertikale Bewegungen dagegen spärlich und vorwiegend für ungewöhnliche Ereignisse genutzt werden sollten. Generell soll der Nutzer jedoch dazu verleitet werden, den kompletten Körper zu bewegen, anstatt Interaktionen lediglich auf Kopf- und Nackenbewegungen zu beschränken. Dies kann erreicht werden, indem virtuelle Objekte um den Nutzer im Raum verteilt werden. Anwendungen mit sich bewegenden oder großen Objekten sollten Kopfbewegungen besondere Beachtung schenken, insbesondere wenn sie häufige Bewegungen entlang der horizontalen und vertikalen Achse erfordern.

Mixed Reality Anwendungen heben sich von Desktop Anwendungen ab, da sich der Nutzer frei in der Welt bewegen und mit virtuellen Inhalten interagieren kann, als würden diese reale Objekte darstellen. Eine stabile Verankerung der Hologramme in der Umgebung sowie eine reibungslose Animation dieser vermittelt dabei den Eindruck realer Objekte. Wird die Anwendung mit einer Framerate von mindestens 60 *Frames per Second (FPS)* gerendert, kann dies dabei helfen, Objekte stabil und reibungslos in die Welt einzubetten. Damit Hologramme den Eindruck vermitteln stabil in der Welt verankert zu sein, müssen die Anwendungen Bilder von der Position, beziehungsweise Sicht des Nutzers, rendern. Da die Berechnung eines Bildes Zeit in Anspruch nimmt, muss die HoloLens 2 vorhersagen können, wo sich der Kopf des Nutzers als nächstes befinden wird, wenn das Bild auf dem Display gerendert wird. Der Algorithmus, welcher

diese Vorhersage berechnet, beschreibt eine Annäherung an die tatsächliche Position. Die Hardware und der Algorithmus der HoloLens 2 passen das gerenderte Bild zwischen der vorhergesagten und tatsächlichen Kopfposition an, um den Unterschied der beiden Positionen zu korrigieren. Dieser Vorgang stellt das angezeigte Bild so dar, als wäre dieses von der korrekten Position gerendert, wodurch die Hologramme für den Nutzer stabil erscheinen. Die Vorhersage der Kopfposition des Nutzers funktioniert am besten für kleine Veränderungen der Kopfposition. Die Korrekturen können jedoch keine Unterschiede im gerenderten Bild, welche durch Bewegungsparallaxe entstehen, ausgleichen. Wird mit einer minimalen Framerate von 60 FPS gerendert, so wird das Hologramm auf zwei Arten stabilisiert. Durch die hohe Framerate wird die Vibration von Hologrammen reduziert, welche im Zusammenhang mit ungleichmäßigen Bewegungen und doppelten Bildern auftritt. Eine niedrige Framerate und schnelle Hologrammbewegungen verursachen dagegen eine stärker ausgeprägte Vibration des Hologramms. Daher sollte stets eine Framerate von mindestens 60 FPS angestrebt werden. Durch die Reduzierung der Latenz wird ebenfalls die Stabilität eines Hologramms verbessert. Bei einer geringeren Latenz kann die Abweichung, welche unter Umständen bei der Berechnung der vorhergesagten Kopfposition entsteht, verringert werden.

Desktop Spiele nutzen *HUDs*, um dem Nutzer Informationen, wie beispielsweise die Lebenspunkte der Spielfigur, eine Minikarte der Spielwelt oder das Inventar, anzuzeigen. *HUDs* können in diesem Fall den Spieler auf dem aktuellen Stand des Spielgeschehens halten, ohne dabei das Spielerlebnis zu beeinträchtigen. In Mixed Reality Anwendungen können *HUDs* dagegen zu Unwohlsein führen und müssen an den immersiveren Kontext angepasst werden. *HUDs*, welche statisch an die Kopfausrichtung des Nutzers angebracht werden, führen dabei insbesondere zu Unwohlsein. In Mixed Reality Anwendungen wird daher ein *HUD* verwendet, welches sich an der Ausrichtung des Körpers und des Kopfes des Nutzers orientiert, anstatt statisch an der Kopfrotation verankert zu sein. Diese spezielle Form eines *HUDs* wird *Tag-Along* genannt und wird verschoben, wenn sich der Nutzer im Raum bewegt. Sie wird jedoch erst rotiert, wenn eine bestimmte Schwelle der Kopfrotation erreicht ist. Sobald diese überschritten wird, kann sich das *HUD* so ausrichten, dass Informationen weiterhin im Sichtfeld des Nutzers angezeigt werden können.

Insbesondere bei Anwendungen, in denen der Nutzer Text lesen muss, ist eine gute Lesbarkeit von Vorteil. Dies kann dem Nutzer dabei helfen, die Belastung der Augen zu reduzieren und den Komfort aufrechtzuerhalten. Die Lesbarkeit hängt dabei von unterschiedlichen Faktoren ab. Dazu zählen die Eigenschaften des Displays, beispielsweise die Pixeldichte, sowie die Helligkeit und der Kontrast. Ein weiterer Faktor sind Linseneffekte, wie beispielsweise *chromatische Aberration*, welche für einen unklaren Umriss des Textes sorgen können. Schriftart-Eigenschaften, wie Schriftstärke, Schriftabstand, Serife, sowie Vorder- und Hintergrundfarben besitzen ebenfalls Auswirkungen auf die Lesbarkeit von Text. Für eine gute Lesbarkeit sollte der Text deshalb so groß wie möglich dargestellt werden.

Eine Anwendung, welche vom Nutzer erwartet, dass die Hand über eine längere Zeit angehoben werden soll, kann zu einer Ermüdung der Muskeln führen. Benötigt eine Anwendung über einen Zeitraum beispielsweise viele *Air-Tap* Gesten, so kann dies ebenfalls zu einer Ermüdung führen. Um eine Ermüdung der Muskeln zu vermeiden, können kurze Pausen in die Anwendung integriert werden. Außerdem können bestimmte Gesten durch Spracheingaben ersetzt werden [21].

### 3.3.2 Audio

Die Einbindung von Audio ist ein essenzieller Bestandteil des Designs und der Effektivität in einer Mixed Reality Anwendung. Audiofeedback kann das Vertrauen des Nutzers in Sprach- und Gesteninteraktionen erhöhen. Das Abspielen von Sound kann den Nutzer zu den nächsten Schritten in der Anwendung leiten. Die Verwendung von Audio kann außerdem dabei helfen, virtuelle Objekte in die reale Welt einzubetten und damit die Mixed Reality Anwendung immersiver zu gestalten. Das *Head-Tracking* erlaubt, unter anderem durch die geringe Latenz der HoloLens 2, eine hochwertige *Head Related Transfer Function (HRTF) spatialization*. Diese Technologie ermöglicht die räumliche Darstellung von Audio in der Anwendung. Dadurch kann der Nutzer auf visuelle Elemente aufmerksam gemacht werden. Außerdem hilft die Technologie dem Nutzer dabei, sich in der realen Welt zurecht zu finden. Audio bindet Hologramme tiefer in die Mixed Reality Welt ein und liefert dem Nutzer Hinweise auf die reale und virtuelle Umgebung von Objekten. Mixed Reality Anwendungen erfordern bei der Verwendung von Audio einen anderen Ansatz als Desktop-Anwendungen. Welche Sounds räumlich dargestellt werden und welche nicht, gehört zu den wichtigsten Entscheidungen beim Sounddesign. Die Implementierung von Sound wirkt sich stark auf die Lernkurve, die Effektivität und das Vertrauen des Nutzers im Umgang mit der Anwendung aus.

Bei *spatial Audio* gibt die *spatialization* an, aus welcher virtuellen Position und Richtung ein Sound stammt. Die Nutzung von *HRTF* in den eingebauten Kopfhörern verbessert die akkurate und komfortable Wiedergabe von Audio. Die *HRTF* Technologie wird von Windows unterstützt und ist auf der HoloLens 2 hardwarebeschleunigt. *HRTFs* manipulieren Pegel- und Phasenunterschiede zwischen den Ohren über das gesamte Frequenzspektrum. Diese Unterschiede basieren auf physikalischen Modellen und Messungen des menschlichen Kopfes, dem Torso und der Ohrform. Das Gehirn reagiert auf die Unterschiede, indem dieses den wahrgenommenen Sound einer Richtung in der Umgebung zuordnet. Da jeder Mensch eine einzigartige Kopfgröße, Ohrposition und Ohrform besitzt, nutzt die HoloLens 2 den Abstand der Augen des Nutzers, um die *HRTF* individuell an die Kopfgröße anzupassen und somit die Genauigkeit der *spatialization* zu verbessern.

*Spatial Audio* schließt nicht nur die Richtung ein, aus der ein Sound kommen kann, sondern auch andere Effekte, wie Okklusion und Hall. Diese Effekte können auch durch den Begriff Akustik beschrieben werden. Ohne Akustik fehlt der *spatialization* die vom Nutzer wahrgenommene Distanz. Deshalb wird Hall beispielsweise von jeder Audio-Engine unterstützt. Die Veränderung der Akustik kann einfach oder komplex sein. Komplexe Effekte, welche andere Objekte in einer

3D Szene in die Wiedergabe des Sounds miteinbeziehen, können eine interessantere und immersivere Akustik erzeugen [22].

### 3.3.3 Spatial Mapping

Durch *spatial Mapping* können virtuelle Objekte auf realen Oberflächen platziert werden. Eine *spatial Map* hilft dabei, die Objekte in der Umgebung des Nutzers zu verankern. Die nötigen Informationen dafür können aus Tiefenhinweisen, welche von den Tiefenkameras der HoloLens 2 stammen, erzeugt werden. Hologramme, die durch reale Objekte oder auch andere Hologramme verdeckt werden, können den Nutzer davon überzeugen, dass sich diese tatsächlich in der realen Welt befinden. In der Luft schwebende Hologramme, oder virtuelle Objekte, welche sich mit dem Nutzer bewegen, können die Immersion verringern. Werden Hologramme bewegt oder platziert, dann sollten die Flächen in der Umgebung des Nutzers visualisiert werden. Dies hilft dem Nutzer dabei zu entscheiden, wo ein Hologramm platziert werden kann und ob für den ausgewählten Bereich bereits eine *spatial Map* generiert wurde. Eine *spatial Map* ist ein Mesh, bestehend aus Dreiecken, und kann je nach Anwendungsfall beständig aktualisiert werden, um die Umgebung abzubilden. Die HoloLens 2 sammelt durchgängig neue Daten und sobald Änderungen in der Umgebung auftreten, werden neue Flächen erzeugt oder entfernt. Die HoloLens 2 kann zwei verschiedene Arten von *spatial Maps* erzeugen. Eine *spatial Map*, bestehend aus Polygonen, welche die Umgebung möglichst genau abbildet oder ein *Scene-Understanding Mesh*, welches versucht, Objekte in der realen Welt zu erkennen und einzuordnen. Beispielsweise kann das *Scene-Understanding Mesh* Wände, Tische und den Boden abbilden.

Wird die Platzierung von Hologrammen auf Oberflächen limitiert, bietet dies einen natürlichen Übergang vom 3D- auf den 2D-Raum. Dadurch wird die Informationsmenge, welche vom Nutzer benötigt wird, reduziert. Die Interaktionen werden dadurch beschleunigt, sowie vereinfacht und gleichzeitig wird die Genauigkeit beim Platzieren von virtuellen Objekten erhöht. Normalerweise kommunizieren Menschen einen Abstand im Raum nicht auf physikalische Weise mit dem Computer. Da stattdessen eine Richtung angegeben wird, kann diese Methode dabei helfen, ein Hologramm ohne großen Aufwand am vom Nutzer vorgesehenen Ort zu platzieren. Anwendungen können außerdem die Form und die Ausrichtung eines Objektes nutzen, um die Platzierung und Rotation eines Hologramms zu bestimmen. Neben der Hilfestellung, Oberflächen von realen Objekten als einen Ort zur Platzierung zu nutzen, können diese Objekte durch die Erstellung einer *spatial Map* verhindern, dass sich Hologramme und reale Objekte überlagern. Visuelles Feedback, während der Platzierung virtueller Objekte, ist für den Nutzer von großer Bedeutung. Das Feedback gibt Hinweise darauf, wie viel Abstand ein Hologramm zu einer Oberfläche besitzt oder ob das Objekt nicht weiter in eine bestimmte Richtung verschoben werden kann, da dieses ansonsten mit einer Oberfläche eines realen Objekts kollidiert. Hat die Anwendung für die Fläche, an dem ein Hologramm platziert werden soll, keine *spatial Map* erzeugt, muss dies dem Nutzer mitgeteilt werden.

Sobald eine *spatial Map* generiert wurde, kann die Anwendung eine Physiksimulation durchführen, um durch das generierte 3D-Objekt Kollisionen zu berechnen. Die Verwendung von Physik bei virtuellen Objekten verstärkt den Eindruck, dass sich diese tatsächlich in der realen Welt befinden. Eine Physiksimulation bietet auch die Möglichkeit, für den Nutzer bekannte Interaktionsmöglichkeiten in der Anwendung zu integrieren. Ein Beispiel dafür ist ein Button, der vom Nutzer gedrückt und nach hinten geschoben werden kann, um eine bestimmte Aktion in der Anwendung zu starten. Bei der Verwendung einer *spatial Map* muss berücksichtigt werden, wie der Scan- und Updatevorgang die Physiksimulation beeinflusst. Beispielsweise muss die Entscheidung getroffen werden, ob Änderungen in der Umgebung einen Einfluss auf das Geschehen in der Anwendung haben. Je nach Anwendung muss auf Änderungen schnell reagiert werden, während in anderen Fällen unbeabsichtigte Veränderung vermieden werden sollten, indem die *spatial Map* nicht weiter aktualisiert wird.

Virtuellen Charakteren kann durch die Verwendung einer *spatial Map* die Fähigkeit verliehen werden, sich selbstständig, wie reale Personen, durch die reale Welt zu bewegen. Dies kann die Präsenz und den Realismus von holographischen Charakteren verbessern, da sich diese an den gleichen Einschränkungen und der Umgebung des Nutzers orientieren müssen.

Eine der hauptsächlichen Anwendungen des *spatial Mappings* ist die Okklusion von Hologrammen. Diese einfache Funktion hat einen großen Einfluss auf den vom Nutzer wahrgenommenen Realismus von Hologrammen. Durch die Okklusion werden virtuelle Objekte in einem immersiveren Umfang in die reale Welt eingebettet. Wird ein Hologramm okkludiert, so kann der Nutzer dadurch automatisch die räumliche Position des Objekts in der Umgebung bestimmen. Die Okklusion von Hologrammen hinter Wänden kann die visuelle Unordnung reduzieren. Andererseits kann die Verdeckung virtueller Inhalte nützliche Informationen verbergen. Um das Hologramm zu verbergen oder zu enthüllen, muss der Nutzer lediglich seinen Kopf bewegen. In bestimmten Fällen ist die Verdeckung von Hologrammen nicht erwünscht. Muss der Nutzer zum Beispiel in einer Anwendung mit dem Hologramm interagieren, muss ermöglicht werden, dass dies vom Nutzer gesehen werden kann. Das gilt auch, wenn ein Hologramm von einer Fläche aus der realen Welt, beispielsweise von einer Wand, verborgen wird. In diesem Fall sollte dem Nutzer durch eine visuelle Veränderung des Hologramms der Hinweis gegeben werden, dass dieses okkludiert wird. Dadurch kann der Nutzer das Hologramm weiterhin lokalisieren, ist sich jedoch dabei bewusst, dass dieses sich hinter einer Fläche befindet.

Die *spatial Map* der abgebildeten Oberflächen sollte in einer Anwendung, in den meisten Fällen, nicht sichtbar sein, um somit visuelle Unordnung zu verringern. Im Gegensatz dazu kann die Visualisierung einer *spatial Map* dennoch von Vorteil sein, obwohl die realen Flächen weiterhin sichtbar sind. Platziert ein Nutzer beispielsweise ein Hologramm auf einer Oberfläche, so kann eine Erdung dieses Objekts, durch das Werfen eines Schattens des Objekts auf die darunterliegende Oberfläche, meist hilfreich sein. Der Schattenwurf gibt dem Nutzer ein klareres Empfinden für die genaue Position zwischen dem Hologramm und der Oberfläche. Durch die Visualisie-

rung der *spatial Map* kann die Anwendung dem Nutzer ihr Verständnis der Umgebung vermitteln. Das Polygon-Mesh, welches durch *spatial Mapping* generiert wird, kann unter Umständen Löcher und andere Artefakte enthalten. Daher sollte die *spatial Map* angemessen visualisiert werden. Traditionelle Beleuchtungsmethoden können diese Fehler optisch hervorheben, was vom Nutzer als störend wahrgenommen werden kann. Visuell klare Texturen, beispielsweise ein Schachbrettmuster, welches auf die Oberfläche der *spatial Map* projiziert wird, kann zu einem geordneten Erscheinungsbild beitragen [23].

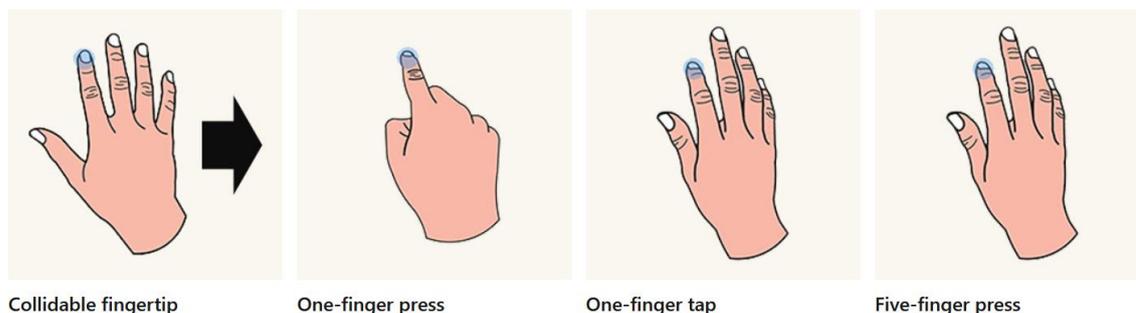
### 3.4 Interaktionsmöglichkeiten und Multimodalität

Um mit virtuellen Objekten einer Augmented Reality Anwendung zu interagieren, können unterschiedliche Sinne genutzt werden. Der Nutzer kann dabei aktiv seine Hände, die Sprache oder den Blick dafür verwenden. Da somit mehrere Sinne gleichzeitig zur Kommunikation zwischen der Anwendung und dem Nutzer genutzt werden können, sind Augmented Reality Anwendungen *multimodal*.

#### 3.4.1 Hände

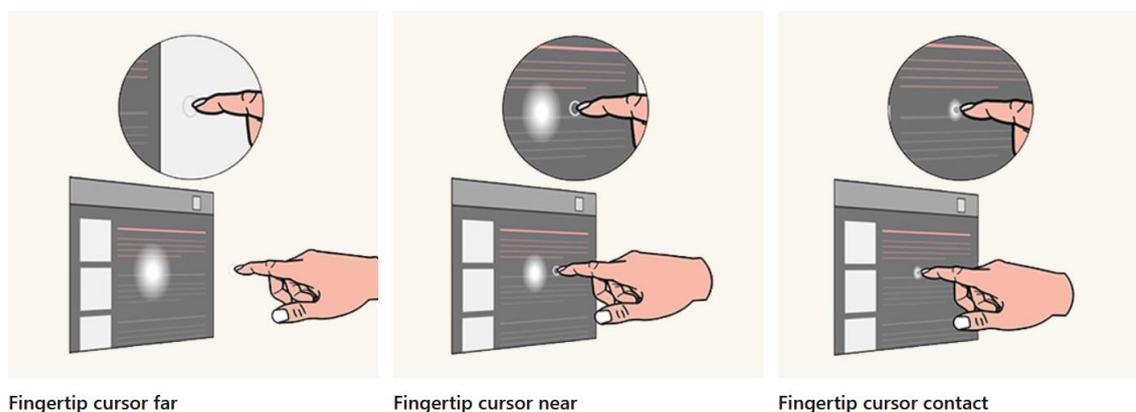
Die direkte Manipulation ist ein Eingabemodell, bei dem Hologramme von den Händen berührt und manipuliert werden. Objekte sollen sich bei diesem Modell wie in der realen Welt verhalten. Buttons können durch einfaches Drücken aktiviert und Objekte aufgenommen werden, indem sie angefasst werden. 2D Inhalte dagegen besitzen das Verhalten eines virtuellen Touchscreens. Alle Interaktionen bei der direkten Manipulation basieren auf einem visuellen Element, welches berührt oder aufgenommen werden kann. Dabei müssen dem Nutzer, bevor das Objekt manipuliert werden kann, keine symbolischen Gesten beigebracht werden. Da dieses Modell am besten für Interaktionen ausgelegt ist, welche sich in der Reichweite des Nutzers befinden, wird das Modell als nahes Eingabemodell bezeichnet. Die direkte Manipulation von Objekten ist das primäre Eingabemodell der HoloLens 2. Dabei wird das neue *Hand-Tracking* System verwendet, welches auch einzelne Fingerbewegungen nachverfolgen kann.

Die Hände des Nutzers werden von der HoloLens 2 erkannt und entweder als linke oder rechte Hand interpretiert. Für beide Hände wird dann ein virtuelles Skelett erstellt, damit das Mesh der holographischen Hände in der Anwendung verformt werden kann. Um ein Hologramm mit der kompletten Hand zu manipulieren, müssen an jede Fingerspitze *Collider* angebracht werden. Das Fehlen von taktilem Feedback bei der Interaktion mit virtuellen Objekten kann bei insgesamt 10 *Collidern* zu unerwarteten und unvorhersehbaren Kollisionen mit den Hologrammen führen. Daher wird lediglich ein *Collider* an den Zeigefingern jeder Hand angebracht. Diese beiden *Collider* reichen aus, um als Berührungspunkt zu dienen und damit verschiedene Handgesten zu ermöglichen. Abbildung 7 stellt drei auf der HoloLens 2 mögliche Gesten dar. Zu diesen zählen der one-finger press, der one-finger tap und der five-finger press.



**Abbildung 7:** Die Position eines *Colliders* am Zeigefinger und die bei der HoloLens 2 möglichen Gesten einer Hand [24].

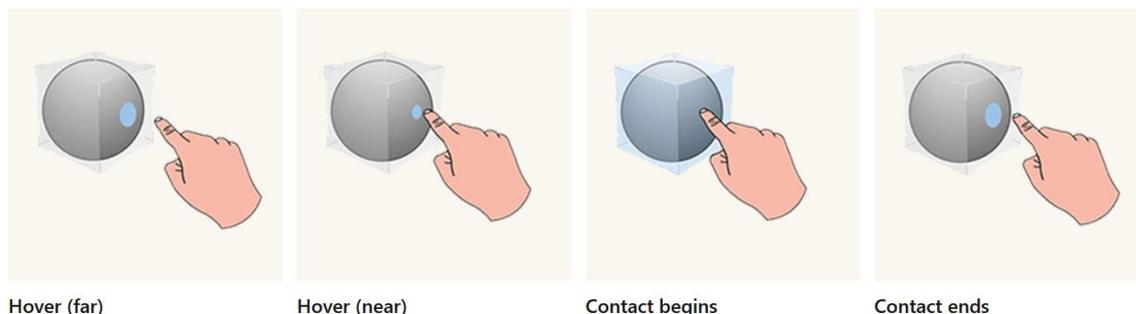
Um die Interaktivität des Zeigefingers der virtuellen Hand zu visualisieren, wird in vielen Anwendungen ein zusätzlicher Fingerspitzen-Cursor gerendert. Damit soll eine bessere Nutzererfahrung bei der Interaktion von naheliegenden Objekten erzeugt werden. Der ringförmige Cursor wird vor dem Zeigefinger platziert. Abhängig von der Entfernung der Hologramme verändert dieser dynamisch seine Position und Rotation. Bewegt sich der Zeigefinger in die Richtung eines Hologramms, so ist der Cursor parallel zur Oberfläche des Objekts ausgerichtet und verkleinert sich, je näher sich der Finger am Hologramm befindet. Sobald der Finger eine virtuelle Oberfläche mit Kollision berührt, verkleinert sich der Cursor zu einem Punkt und löst einen Berührungseffekt aus. Das Verhalten des Cursors wird in Abbildung 8 dargestellt. Durch dieses interaktive Feedback kann der Nutzer präzise Aufgaben in der näheren Umgebung ausführen.



**Abbildung 8:** Verhalten des Zeigerfinger-Cursors bei nahen und fernen Hologrammen [24].

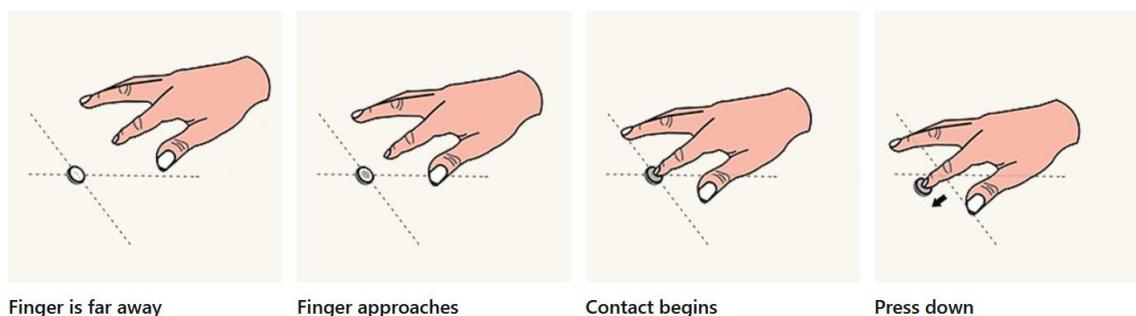
Neben visuellen Hilfestellungen für die Hand benötigt auch das Hologramm selbst die Fähigkeit visuelles und akustisches Feedback zur Verfügung zu stellen, um das Fehlen von taktilem Feedback auszugleichen. Dazu wird für das zu interagierende Hologramm eine *Bounding-Box* erstellt, welche einen *Proximity-Shader* besitzt. Die *Bounding-Box* umschließt das virtuelle Objekt, während der *Proximity-Shader* ein visuelles Feedback für die Entfernung des Zeigefingers liefert. Wie in Abbildung 9 dargestellt, wird bei einer großen Entfernung des Zeigefingers zur *Bounding-Box*

ein großer Punkt auf der Oberfläche gerendert. Bewegt sich der Zeigefinger weiter in die Richtung des Objekts, so wird dieser Punkt kleiner. Berührt der Nutzer das Hologramm, so ändert sich die Farbe der *Bounding-Box* oder anderweitige visuelle Effekte werden ausgelöst. Wird der Kontakt beendet, kann neben visuellen Effekten auch auditives Feedback übermittelt werden, um die Berührung der *Bounding-Box* zu untermalen.



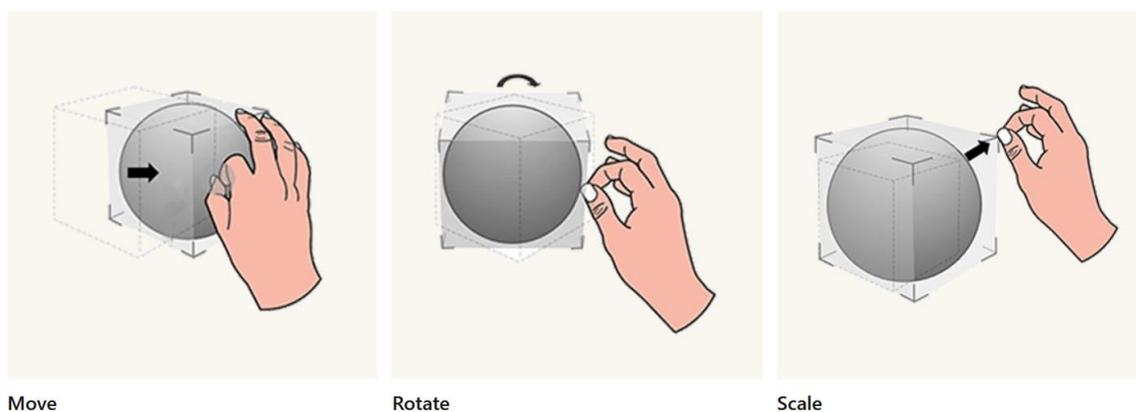
**Abbildung 9:** Der *Proximity-Shader* rendert unterschiedlich große Punkte auf die *Bounding-Box*, basierend auf der Entfernung des Zeigefingers [24].

Da die Fingerspitze einen *Collider* besitzt, kann der Nutzer durch das Drücken eines Buttons mit einer der holographischen Komponenten interagieren. Beispielsweise ist ein holographischer Button auf ein direktes Drücken des Fingers ausgelegt. Da auch bei dieser Interaktionsmöglichkeit taktiler Feedback fehlt, bestehen verschiedene Möglichkeiten dieses auszugleichen. So kann beispielsweise die bei interaktiven Objekten eingesetzte *Bounding-Box* mit *Proximity-Shader* bei diesen Buttons ebenfalls eingesetzt werden, um dem Nutzer ein Gefühl des Abstands zwischen dem Finger und dem Objekt zu vermitteln. Eine andere Methode ist das Verschieben des Buttons in die Richtung, in welche der Button gedrückt wird. Dies erzeugt beim Nutzer bei der Berührung des Buttons das Gefühl, dass dieser taktiler Feedback besitzt. Durch die Berührung des *Colliders* an der Fingerspitze und des Buttons bewegt sich das Objekt nahe des Fingers entlang der gedrückten Richtung. Wie in Abbildung 10 dargestellt, kann der Button ausgelöst und eine Aktion gestartet werden, wenn der Button einen bestimmten Abstand zurückgelegt hat oder diesen Abstand wieder verlässt, nachdem dieser überschritten wurde. Soundeffekte sollten bei der Auslösung des Buttons ebenfalls zum Einsatz kommen, um das Feedback zu verbessern.



**Abbildung 10:** Die Verwendung eines Buttons durch den *Collider* auf der Spitze des Zeigefingers [24].

Die HoloLens 2 ermöglicht dem Nutzer 3D Objekte mit den Händen zu manipulieren und zu steuern, indem um die Hologramme eine *Bounding-Box* erstellt wird. Um mit den *Bounding-Boxen* und damit mit den virtuellen Objekten zu interagieren, stellt die HoloLens 2 zwei unterschiedliche Methoden zur Verfügung. Bei der ersten Methode werden an den Ecken und Kanten der *Bounding-Box* Hilfsobjekte, auch *affordances* genannt, hinzugefügt, welche dem Nutzer erlauben, diese zu greifen und damit das Objekt zu transformieren. Abbildung 11 zeigt die Verwendung einer *Bounding-Box* und deren *affordances*. Die *Bounding-Box* und die *affordances* eines Objekts werden angezeigt, sobald sich die Hand des Nutzers in dessen Nähe befindet. Um ein Objekt zu bewegen, kann die *Bounding-Box* gegriffen und bewegt werden. Durch das Greifen der *affordances* an den Kanten der *Bounding-Box* kann das Hologramm entweder um die horizontale oder die vertikale Achse rotiert werden. Die an den Ecken platzierten *affordances* erlauben eine uniforme Skalierung des Hologramms.



**Abbildung 11:** Durch Hilfsobjekte, sogenannte *affordances*, an den Ecken der *Bounding-Box* kann ein Hologramm transformiert werden [24].

Eine andere Methode, welche den Nutzer die Transformation von Hologrammen manipulieren lässt, basiert nicht auf der Nutzung von *affordances*. Stattdessen kann die *Bounding-Box* für eine direkte Manipulation genutzt werden. Wird diese mit der Hand aufgenommen, so ist die Translation und Rotation des Hologramms mit der Orientierung der Hand verbunden. Dreht der Nutzer seine Hand, so wird das Objekt um die gleiche Rotationsachse gedreht. Diese Methode ist auf einfache, schnelle Interaktionen ausgelegt. Für eine präzise Manipulation eines Hologramms sollte daher die auf *affordances* basierte Manipulation genutzt werden, da diese ein hohes Maß an Genauigkeit bietet. Bei einer flexiblen Manipulation von Hologrammen kann dagegen auf *affordances* verzichtet werden. Der Einsatz direkter Manipulation kann dazu führen, dass der Nutzer unbeabsichtigt ein Hologramm mit seiner Hand berührt und dadurch eine unerwartete Manipulation startet. Das Tracking der Augen des Nutzers kann bei diesem Problem helfen, indem die Manipulation nur bei solchen Objekten aktiviert wird, welche vom Nutzer angesehen werden [24].

### 3.4.2 Spracheingaben

Spracheingaben können bei der HoloLens 2 genutzt werden, um Aktionen in einer Anwendung auszulösen, ohne dass dafür Handgesten notwendig sind. Ein natürlicher Weg für den Nutzer, seine Absicht zu kommunizieren, sind Sprachbefehle. Bei komplexen Menüs können Spracheingaben dabei helfen, schnell durch das *User Interface (UI)* zu navigieren, da verschachtelte Menüs durch einen Sprachbefehl umgangen werden können. Die HoloLens 2 bietet unabhängig von der Anwendung verschiedene Sprachbefehle, um mit Menüs oder anderen Funktionen zu interagieren. So kann beispielsweise mit einem Sprachbefehl zum Hauptmenü navigiert werden, Anwendungen gestartet, der System-Sound aktiviert sowie deaktiviert oder die Helligkeit des Displays verändert werden. Bei der Verwendung von Sprachbefehlen in einer Anwendung sollte auf die empfohlene Vorgehensweise geachtet werden. Das bedeutet, Sprachbefehle sollten aus Schlüsselwörtern bestehen und zwei oder mehr Silben besitzen, da diese von der Anwendung besser differenziert werden können, falls der Nutzer mit einem Akzent spricht. Außerdem sollte einfaches Vokabular, anstatt Fachwörter, verwendet werden. Um die Auslösung unerwarteter Aktionen zu vermeiden, sollte weiterhin darauf geachtet werden, keine ähnlich klingenden Sprachbefehle zu verwenden. Auf den Einsatz von Sprachbefehlen, welche nativ für die Bedienung der HoloLens 2 verwendet werden, sollte verzichtet werden. Der Einsatz von Spracheingaben kann den Nutzer bei der Verwendung einer Anwendung Interaktionen effizienter ausführen lassen, den Aufwand verringern und damit die Aufgaben flüssiger und müheloser gestalten. Während Spracheingaben viele Vorteile bieten, kann die Nutzung jedoch auch Probleme verursachen. Obwohl die Erkennung von Sprachbefehlen ständig verbessert wird, besteht die Möglichkeit, dass bestimmte Wörter falsch erkannt oder falsch interpretiert werden. Ein weiteres Problem resultiert daraus, dass Spracheingaben bei bestimmten Interaktionen nicht ohne großen Aufwand präzise formuliert werden können. Befindet sich der Nutzer nicht allein in einem Raum besteht die Möglichkeit, dass die Spracherkennung von anderen Menschen beeinträchtigt wird. Andererseits kann eine Störung anderer Menschen im Umfeld unerwünscht sein. Ein weiteres Hindernis kann die Vermittlung aller zur Verfügung stehenden Sprachbefehle darstellen, da dies den Nutzer überfordern kann. Einige dieser Probleme können gelöst werden, indem dem Nutzer Feedback gegeben wird. Beispielsweise kann das von der Anwendung erkannte Wort angezeigt werden, damit der Nutzer potentielle Probleme erkennen kann [25].

### 3.4.3 Blickrichtung

Die Augen oder der Kopf des Nutzers können von einer Anwendung genutzt werden, um zu bestimmen, was sich der Nutzer ansieht. Die Kopfrichtung, der *Head-Gaze*, stellt die Richtung dar, in welche der Kopf des Nutzers gerichtet ist. *Head-Gaze* kann als die Position und Ausrichtung des Geräts selbst angesehen werden, wobei die Position der Mittelpunkt des Displays ist. Die Blickrichtung, der *Eye-Gaze*, repräsentiert die Richtung, in welche die Augen des Nutzers gerichtet sind. Der Ursprung der Richtung befindet sich zwischen den Augen des Nutzers [26].

Durch die Einführung von *Eye-Tracking* auf der HoloLens 2 kann dem Nutzer die Verwendung vereinfacht werden, da damit auf die Information zugegriffen werden kann, was dieser sich momentan ansieht. Damit *Eye-Tracking* korrekt funktioniert, muss der Nutzer diese Funktionalität vor der Nutzung kalibrieren. Dabei muss eine Reihe von Hologrammen angesehen werden, um die nötigen Informationen zur Kalibrierung zu ermitteln. Dies erlaubt der HoloLens 2 sich an den Nutzer anzupassen und ein exaktes *Eye-Tracking* zu gewährleisten. Da diese Funktion vom Nutzer ausgeschaltet werden kann oder das *Eye-Tracking* in manchen Fällen aus anderen Gründen nicht verwendet wird, sollte für diese Fälle ein Ersatz gefunden werden, damit die Anwendung weiterhin korrekt funktioniert [27]. Die Nutzung von *Eye-Tracking* bietet verschiedene Vorteile, wie beispielsweise die Schnelligkeit, mit welcher ein Nutzer mit Objekten interagieren kann. Der Augenmuskel ist einer der am schnellsten reagierenden Muskel des menschlichen Körpers. Um ein Objekt anzusehen sind kaum körperliche Bewegungen notwendig. Die Anwendung kann durch die Augenbewegung vorhersagen, mit welchem Ziel der Nutzer voraussichtlich interagieren möchte. Da der Nutzer an Hand-Augen Koordination gewöhnt ist, bietet *Eye-Tracking* eine gute Ergänzung zu Handgesten-Interaktionen. Ein weiterer Vorteil ist die Möglichkeit zu erkennen, auf welches Objekt sich ein Nutzer im Moment fokussiert. Andererseits stellt die Anwendung des *Eye-Trackings* die Entwickler auch vor Herausforderungen. Sieht sich der Nutzer beispielsweise die Details eines Hologramms genauer an, können Interaktionen unbeabsichtigt ausgelöst werden, da das *Eye-Tracking* durchgehend aktiviert ist. Dies kann zu einem frustrierenden Erlebnis führen. Daher sollte *Eye-Tracking* in Verbindung mit Sprachbefehlen, Handgesten oder einem Tastendruck genutzt werden, um eine Interaktion zu aktivieren. Alternativ kann ein Zeitraum definiert werden, in dem der Nutzer ein Objekt im Blick behalten muss, bevor eine Interaktion aktiviert wird. Dies erlaubt dem Nutzer sich in der Umgebung umzusehen, ohne von unbeabsichtigt ausgelösten Ereignissen überwältigt zu werden. Das gleiche Problem sollte außerdem beachtet werden, wenn visuelle und auditive Effekte eingesetzt werden. Häufige Sounds oder visuelle Effekte bei Blickkontakt sollten daher vermieden werden [28].

### 3.5 User Experience Elemente

Durch die Einführung einer virtuellen Hand mit einem *Collider* an der Spitze des Zeigefingers ermöglicht die HoloLens 2 viele verschiedene Interaktionsmöglichkeiten. Diese sollen in diesem Abschnitt beschrieben und deren optimaler Einsatz aufgezeigt werden. Außerdem sollte bei der Erstellung von Inhalten für die Anwendung auf bestimmte Eigenschaften geachtet werden, um visuell ansprechende Hologramme darzustellen und um für eine gute Kommunikation zwischen dem Nutzer und der Anwendung zu sorgen.

#### 3.5.1 Farbe, Beleuchtung und Materialien

Das Design von Inhalten in einer Mixed Reality Anwendung erfordert eine sorgfältige Planung der Beleuchtungskonfiguration, Materialeigenschaften und Einfärbung der virtuellen Objekte. Für ästhetische Zwecke kann die Verwendung einer bestimmten Beleuchtung oder bestimmter

Materialien die Stimmung einer Umgebung immersiver gestalten, während für funktionale Zwecke auffallende Farben gewählt werden können, um Objekte hervorzuheben und so den Nutzer auf eine bevorstehende Aktion hinzuweisen. Holographische Displays, wie jenes der HoloLens 2, nutzen additive, transparente RGB-Displays, um Hologramme zu rendern. Eine Anwendung sollte nicht nur auf einem gewöhnlichen Bildschirm, sondern ebenfalls auf dem Gerät selbst getestet werden, da sich die Darstellung auf den Displays der HoloLens 2 von Monitoren unterscheidet. Die Beleuchtung eines Hologramms sollte von jeder Seite überprüft werden, damit sichergestellt werden kann, dass das gewünschte Ergebnis erreicht wird. Da unterschiedliche Beleuchtungsverhältnisse bei der Nutzung einer Anwendung herrschen können, sollte die Sichtbarkeit von Hologrammen für unterschiedliche Helligkeitsstufen getestet werden. Holographische Geräte verwenden additive Displays. Virtuelle Objekte können auf diesem Display gerendert werden, indem das Licht der Hologramme das Licht der realen Welt überlagert. Weiß erscheint hell auf dem Display, Schwarz wird transparent gerendert. Der Einfluss der Farbe variiert mit der Umgebung des Nutzers, da im Raum unterschiedliche Lichtverhältnisse herrschen können. Damit Hologramme in diesen unterschiedlichen Lichtverhältnissen klar erkennbar bleiben, sollte die virtuelle Szene über ausreichend Kontrast verfügen. Hologramme, welche eine uniforme Beleuchtung besitzen, werden am effizientesten gerendert. Der Einsatz dynamischer Beleuchtung kann die Rechenleistung mobiler Geräte, wie der HoloLens 2, überfordern. Durch die Funktionsweise der additiven Farbmischung können bestimmte Farben auf holographischen Displays unterschiedlich erscheinen. Bei spezifischen Beleuchtungsumgebungen stechen einige Farben hervor, während andere unauffälliger erscheinen. Kühle Farben treten in den Hinter-, warme Farben in den Vordergrund. Daher sollte auf unterschiedliche Faktoren bei der Auswahl von Farben geachtet werden. Helle Farben erscheinen auf dem Display grell und sollten daher sparsam verwendet werden. Große, weiße, helle Flächen sollten vermieden werden, da diese zu Unwohlsein führen können. In den meisten Fällen sollte beim Einsatz heller Farben ein RGB-Wert von maximal 235 in jedem der Farbkanäle gewählt werden. Durch die Verwendung additiver Displays erscheinen dunkle Farben transparent. Ein komplett schwarzes Objekt kann daher nicht vom Nutzer wahrgenommen werden. Um den Eindruck einer schwarzen Oberfläche zu erwecken kann ein Farbwert von 16 in jedem der RGB Farbkanäle verwendet werden. Die HoloLens 2 besitzt einen weiten Farbraum, wodurch einige Farben unterschiedliche Qualitäten und Darstellungen aufweisen. Für die Erzeugung eines realistischen Hologramms sind Materialien ein wichtiger Faktor. Durch die Bereitstellung der korrekten visuellen Eigenschaften kann ein virtuelles Objekt immersiv in die reale Umgebung eingebettet werden. Materialien sind weiterhin wichtig, um dem Nutzer visuelles Feedback für die unterschiedlichen Interaktionen zu vermitteln [29].

### 3.5.2 Skalierung

Um realistische Hologramme zu erzeugen, sollten die visuellen Eigenschaften der realen Welt so genau wie möglich nachgeahmt werden. Damit der Nutzer die Position, die Größe und die Be-

schaffenheit der virtuellen Objekte einschätzen kann, müssen visuelle Hinweise von der Anwendung bereitgestellt werden. Mixed Reality ermöglicht die Betrachtung von Objekten in Originalgröße und unterscheidet sich damit von Bildschirm-Visualisierungen. Mehrere Möglichkeiten können genutzt werden, um dem Nutzer die Größe eines Objekts zu vermitteln. Einige dieser Möglichkeiten wirken sich jedoch auf andere Wahrnehmungsfaktoren aus. Objekte sollen in einer realistisch wirkenden Größe angezeigt werden und diese Größe beibehalten, sobald sich der Nutzer bewegt. Die Größe der Hologramme verändert sich je nach Entfernung des Nutzers und verhalten sich damit wie reale Objekte.

Nahe gelegene Objekte erfordern, dass der Nutzer seinen Kopf und Körper bewegt, um das gesamte Objekt zu erfassen. Wird ein Objekt weiter entfernt platziert, kann sich der Nutzer ein Gefühl für den Maßstab schaffen. Für eine genaue Inspektion kann dieser sich dann näher an das Hologramm heranbewegen. Ein Hologramm kann außerdem dazu verwendet werden, um die reale Umgebung zu verändern. Wände oder Decken können ausgetauscht oder mit Löchern, beziehungsweise Fenstern, versehen werden. Dadurch kann der Effekt erzeugt werden, dass große Objekte scheinbar aus dem physischen Raum herausragen. Soll beispielsweise ein großer Baum visualisiert werden, kann die Decke des Raums durch einen virtuellen Himmel ausgetauscht werden, um diesen im korrekten Maßstab darzustellen. Diese Methode erlaubt dem Nutzer um den Stamm des virtuellen Baums zu laufen und damit ein Gefühl für die Größe und das Erscheinungsbild zu erlangen. Der Nutzer kann nach oben schauen, um zu sehen, dass sich der Baum über den physischen Raum hinaus erstreckt [30].

### 3.5.3 Sound

Mixed Reality kann Sound dazu nutzen, um den Nutzer über den aktuellen Zustand, in der sich die Anwendung befindet, zu informieren oder die Eindrücke der aktuellen Situation zu verstärken. *Spatialization* kann in angemessenen Situationen genutzt werden, um Sound in die virtuelle Welt einzubetten. Werden die auditiven und visuellen Komponenten verbunden, können Interaktionen intuitiver wirken und damit das Vertrauen des Nutzers beim Umgang mit der Anwendung verstärken. Durch das Fehlen von taktilem Feedback haben Mixed Reality Anwendungen oft einen höheren Bedarf an Sound als 2D Anwendungen. Bei Ereignissen, welche nicht vom Nutzer gestartet werden, kann Sound die Information vermitteln, dass eine Änderung stattgefunden hat. Interaktionen können mehrere Phasen enthalten, wobei der Einsatz von Sound die Übergänge verdeutlichen kann. Die Verwendung von Sound sollte den Nutzer jedoch nicht überfordern, da dieser nicht die Fähigkeit besitzt, beliebig viele Audio-Informationen aufzunehmen. Daher sollte Sound lediglich spezifische, wichtige Informationen vermitteln. Wird von der Anwendung ein Sound abgespielt, um den Nutzer zu informieren, wird empfohlen, dass die Lautstärke anderer Audioquellen reduziert wird. Eine Anwendung sollte auch dann funktionsfähig bleiben, wenn der Sound deaktiviert wurde. Wird ein Sound ausgelöst, wenn sich die Hand des Nutzers in der Nähe eines Buttons befindet, kann durch eine Zeitverzögerung beim Abspielen des Sounds sichergestellt werden, dass der Sound sich nicht überlagert und störend wirkt. Wenn

der Nutzer das Ende eines Tastendrucks erreicht, ist Sound hilfreich, da der Nutzer keinen anderen Hinweis auf diese Limitierung erhält. Der Sound kann dabei subtil sein. Ähnlich wie von Gesten gesteuerte Interaktionen sollte ein Tastendruck einen kurzen, taktilen Sound abspielen. Wird der Button losgelassen, kann ein ähnlicher Sound mit einer größeren Tonhöhe abgespielt werden. Wird ein Objekt gegriffen oder losgelassen, so verdeckt die Hand des Nutzers in vielen Fällen die Effekte des Objekts, weshalb dieses häufig nur visuell unvollständig dargestellt werden kann. Greift der Nutzer ein Hologramm, ist ein gedämpfter, taktiler Sound angemessen, der das Gefühl vermittelt, dass sich die Hände des Nutzers um ein Objekt schließen. Sound kann diese unterschiedlichen Interaktionen erfolgreich auditiv bestätigen.

Bei Sprachbefehlen werden, wie in Abschnitt 3.4.2 beschrieben, visuelle Effekte eingesetzt, um den Nutzer auf eine erfolgreiche Erkennung hinzuweisen. Neben visuellen Hinweisen können hier auch auditive Hinweise zum Einsatz kommen. Um einen erfolgreichen Sprachbefehl zu bestätigen, können kurze, positiv klingende Sounds genutzt werden. Um den Nutzer auf einen Fehler bei der Spracherkennung hinzuweisen, werden kurze, weniger positiv klingende Sounds verwendet. Anstatt negativ klingende Sounds zu verwenden, sollten neutrale Sounds eingesetzt werden, damit dem Nutzer mitgeteilt wird, dass die Anwendung die Interaktion beendet. Ist eine Anwendung so konfiguriert, dass diese auf Sprachbefehle wartet, kann für diesen Zweck ein subtiler, sich wiederholender Sound genutzt werden.

Ein Sound sollte *spatialized* werden, wenn dieser mit einem Ereignis an einem bestimmten Ort im Raum verbunden ist. *UI*, Stimmen virtueller Charaktere und visuelle Indikatoren sind Beispiele für ein solches Ereignis. Interaktionen, bei denen Objekte durch Berühren oder Greifen manipuliert werden, vermitteln ein natürlicheres Feedback, wenn das Audiofeedback *spatialized* wurde.

Auf *spatialization* sollte dagegen verzichtet werden, wenn Hologramme oder andere Elemente in der Anwendung keine klar definierte Position im Raum besitzen. *Spatialization*, welche keinem visuellen Element zugeordnet werden kann, wirkt auf den Nutzer störend, da dieser nach der Quelle des Sounds sucht und diese nicht finden kann. Der Einsatz von *spatialization* ist mit einer hohen Rechenleistung der CPU verbunden. Da viele Anwendungen höchstens zwei Sounds gleichzeitig abspielen, ist dieser Rechenaufwand jedoch vernachlässigbar. Trotzdem sollte dieser Einfluss auf die Framerate des Spiels beachtet werden.

In der realen Welt werden Sounds, welche weiter von einer Person entfernt sind, leiser. Die Audio-Engine kann die Dämpfung des Sounds, basierend auf der Distanz der Audio-Quelle, berechnen. Entfernungsbasierte Sound-Dämpfung kann eingesetzt werden, wenn sich visuelle Indikatoren und andere Ereignisse außerhalb des Sichtfeldes des Nutzers befinden. Den Abstand zu visuellen Indikatoren, animierten Hologrammen und anderen informativen Sounds einschätzen zu können, hilft dem Nutzer bei der Anwendung [31].

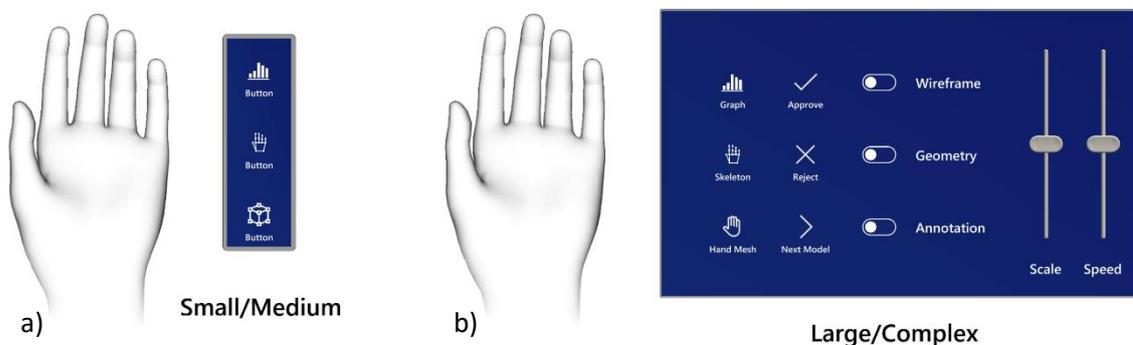
### 3.5.4 Interaktive Elemente und Funktionen

Die HoloLens 2 bietet verschiedene Interaktionsmöglichkeiten, um mit der Anwendung und mit Hologrammen zu interagieren. Einige dieser unterschiedlichen Interaktionen werden in diesem Abschnitt näher beschrieben.

#### 3.5.4.1 Hand Menu

Ein wichtiges Werkzeug, zur Interaktion mit der Anwendung selbst, ist das Hand Menu. Dieses erlaubt dem Nutzer ein *UI* anzuzeigen, welches an der Hand verankert wird. Das Hand Menu eignet sich für schnelle Aktionen, da dieses jederzeit geöffnet und leicht ein- und ausgeblendet werden kann. Um ein optimales Erlebnis des Nutzers mit dem Hand Menu zu garantieren, sollte bei der Entwicklung auf verschiedene Faktoren geachtet werden.

Beispielsweise sollte die Anzahl an Buttons auf einem niedrigen Niveau gehalten werden, da das Hand Menu unweit von den Augen des Nutzers entfernt ist und sich der Nutzer lediglich auf einen kleinen Bereich fokussieren kann. Für einfache Menüs kann ein Layout mit drei Buttons in einer Spalte erstellt werden, da dabei das komplette Hand Menu im Sichtfeld bleibt, selbst wenn der Nutzer seine Hand in die Mitte des Sichtfelds bewegt. Ein Hand Menu kann geöffnet werden, indem sich die Hand des Nutzers im Sichtfeld der HoloLens 2 befindet und dieser seine Hand öffnet. Die Innenseite ist dabei in Richtung des Nutzers gerichtet. Benötigt eine Anwendung ein großes Menü, sollte dieses die Möglichkeit bieten, an die Umgebung oder an den Körper des Nutzers angeheftet werden zu können. Abbildung 12 zeigt das Layout der Buttons und die Größe eines einfachen und komplexen Hand Menus.



**Abbildung 12:** a) Einfaches Hand Menu mit drei Buttons. b) Komplexes Hand Menu mit vielen Buttons und zwei Schiebereglern [32] (verändert).

Der Übergang von der Verankerung an der Hand zur Umgebung kann ausgelöst werden, indem der Nutzer die Hand schließt und diese aus dem Sichtbereich der HoloLens 2 bewegt. Mit Hilfe einer anderen Methode kann das Menü selbst greifbar gemacht werden, um dieses an die Umgebung anzuheften. Ist das Menü nicht mehr an der Hand, sondern in der Umgebung verankert, so kann der Nutzer über einige Zeit komfortabel mit dem Menü interagieren, ohne, dass der Arm dabei ermüdet. Wird das Menü jedoch an die Umgebung angeheftet, muss die Anwendung

die Möglichkeit bereitstellen, dass das Menü durch die Welt bewegt und auch wieder geschlossen werden kann, wenn dieses nicht mehr benötigt wird. Das Menü kann von der Umgebung erneut an die Hand des Nutzers angeheftet werden, wenn dieser die Hand noch einmal im Sichtbereich der HoloLens 2 öffnet.

Das Anheben und Beibehalten der Handposition kann zu einer Ermüdung des Arms führen. Bei einem einfachen Hand Menu, welches schnelle Interaktionen erlaubt, kann dieses an die Hand des Nutzers angebracht werden. Komplexe Menüs, bei denen der Nutzer lange Interaktionen durchführen muss, sollten an die Welt oder an den Körper angeheftet werden, um ein Ermüden des Arms zu verhindern.

Menüs sollten zur Bedienhand entgegengesetzten Schulter und zur Mitte des Kopfes hin ausgerichtet werden. Dadurch kann der Nutzer mit der freien Hand natürliche Handbewegungen zur Interaktion ausführen und vermeidet unangenehme Handpositionen, sollte eine Interaktion mit den Buttons des Hand Menus erforderlich sein.

Da der Nutzer eventuell nicht beide Hände nutzen kann, sollte auf diesen Umstand beim Design von Hand Menus geachtet werden. Um einhändige Menüs zu ermöglichen, kann das Menü, wie bereits beschrieben, von der Hand gelöst und in der Welt oder am Körper des Nutzers angebracht werden, sobald sich die Hand schließt. Außerdem können Sprachbefehle genutzt werden, um das Hand Menu zu öffnen.

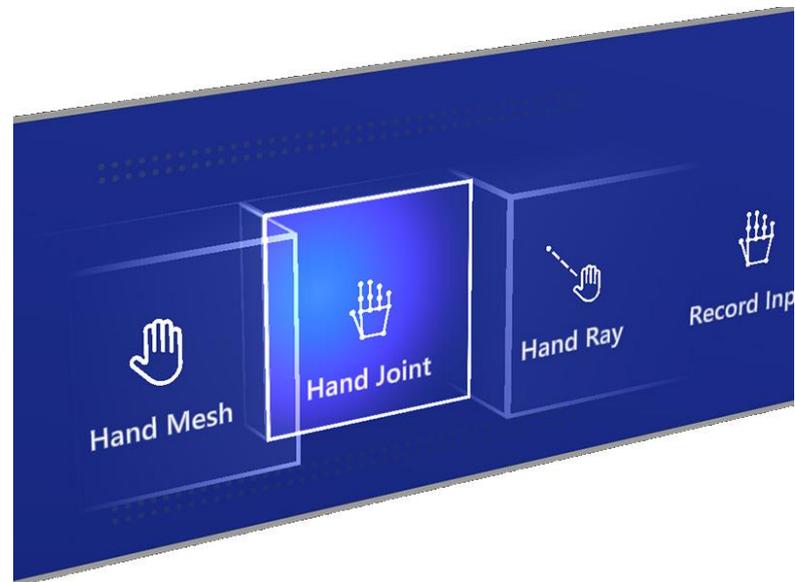
Durch zu nahes platzieren des Hand Menus am Handgelenk, kann ein unbeabsichtigtes Berühren des Home Buttons durch den Nutzer erfolgen, welche die Anwendung pausiert und das Hauptmenü der HoloLens 2 öffnet.

Wird lediglich das Öffnen der Hand als Indikator zur Öffnung des Hand Menus angesehen, kann dies dazu führen, dass der Nutzer unbeabsichtigt das Menü öffnet. Um ein unbeabsichtigtes Öffnen zu vermeiden, kann ein weiteres Ereignis als Indikator hinzugefügt werden. Einer dieser Indikatoren ist eine komplett geöffnete Hand. Anstatt das Hand Menu bei einer halb geöffneten Hand zu aktivieren, kann die Anwendung eine vollständig geöffnete Hand vom Nutzer verlangen. Dadurch kann vermieden werden, dass Gesten erkannt werden, mit welchen der Nutzer ursprünglich mit Objekten interagieren wollte. Da die HoloLens 2 *Eye-Tracking* ermöglicht, kann der Blick des Nutzers verfolgt werden. Sieht ein Nutzer auf seine Hand, während er diese öffnet, ist dies ein weiterer Indikator dafür, dass der Nutzer die Absicht hat, das Hand Menu zu öffnen [32].

#### **3.5.4.2 Button**

Ein Button ist einer der grundlegenden *UI*-Elemente in Mixed Reality. Er lässt den Nutzer sofortige Aktionen auslösen. Da in Mixed Reality kein physisches Feedback erfolgt, sollte die Anwendung ausreichend visuelles und akustisches Feedback liefern. Vom *MRTK* werden daher unterschiedliche Effekte bereitgestellt, welche die Tiefenwahrnehmung des Nutzers unterstützen und anderweitig bei der Interaktion mit Buttons helfen. Als Hilfestellung zur Position des Finger-

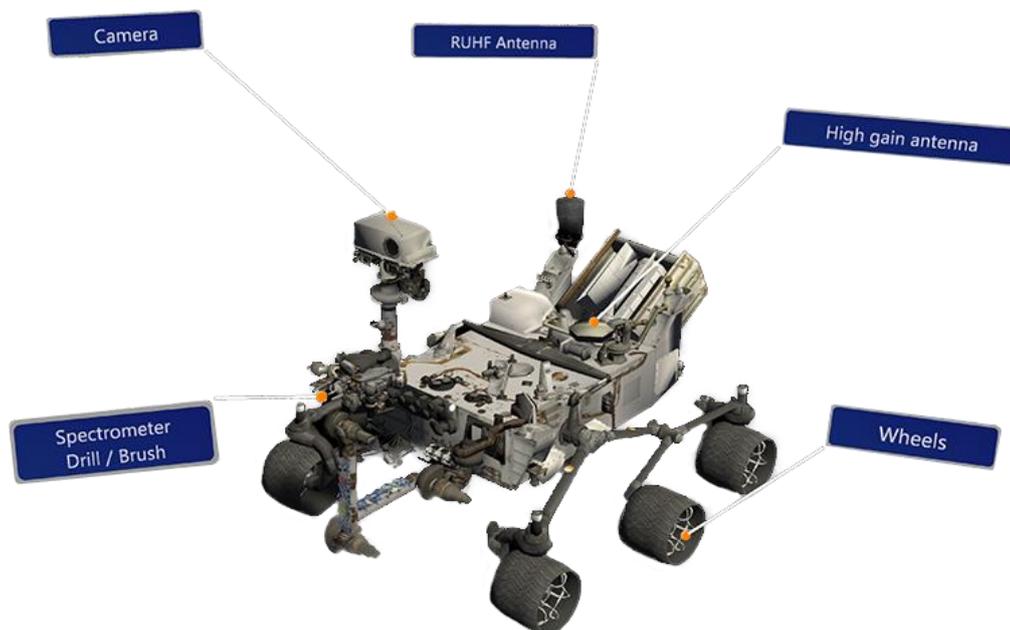
Cursors kann ein *Proximity-Shader* genutzt werden. Wird ein Button gedrückt, so wird diese Aktion visualisiert, indem dieser Button dem Fingerdruck folgt und verschoben wird. Die Verschiebung und die Beleuchtung der UI durch den *Proximity-Shader* wird in Abbildung 13 dargestellt [33].



**Abbildung 13:** Wird ein Button gedrückt, so verschiebt sich dieser, um dem Nutzer ein visuelles Feedback zu geben [33] (verändert).

### 3.5.4.3 Tooltip

Ein Tooltip ist eine kurze Beschreibung, welche an ein Hologramm angeheftet werden kann. Tooltips werden verwendet, um einen Hinweis oder zusätzliche Informationen zu vermitteln. Sie helfen dem Nutzer unbekannte Objekte oder Details eines Hologramms, welche nicht direkt beschrieben werden, zu erkennen. Wie in Abbildung 14 dargestellt wird, zeigen Tooltips meist auf wichtige Komponenten eines Hologramms [34].



**Abbildung 14:** Tooltips beschreiben beispielsweise die relevanten Komponenten eines Fahrzeugs [34] (verändert).

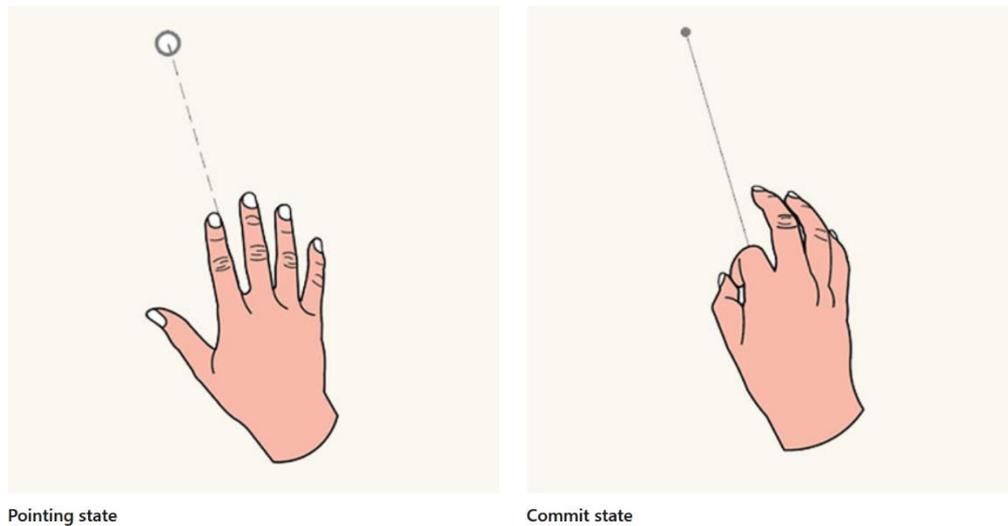
#### 3.5.4.4 Near Menu

Ein Near Menu ist eine Sammlung aus Buttons auf einem Panel, welches dem Nutzer mit einem *Tag-Along* Verhalten folgt. Das Near Menu wird häufig in Anwendungen verwendet, wenn das Hand Menu von der Hand entfernt und in der Welt platziert wurde. Das Panel bewahrt eine bestimmte Reichweite und Orientierung zum Nutzer, wodurch eine komfortable Nutzung dieses Menüs ermöglicht wird. Durch einen Pin-Button kann das Near Menu an die Umgebung angeheftet werden. Greift und platziert der Nutzer das Menü, so wird dieses ebenfalls in der Welt angeheftet. Dadurch kann der Nutzer sich auf die Buttons und Funktionen des Near Menüs fokussieren. Durch erneutes Drücken des Pin-Buttons kann das *Tag-Along* Verhalten wieder aktiviert werden [35].

#### 3.5.4.5 Hand Ray

Ein Nutzer kann in Mixed Reality auch mit 2D und 3D Objekten interagieren, welche sich außerhalb der Reichweite der Hände befinden. Die Einführung von *Hand-Tracking* auf der HoloLens 2 ermöglicht ein neues Eingabemodell, welches point and commit with hands genannt wird. Dabei wird ein Strahl, der *Hand Ray*, aus der Mitte der Handfläche des Nutzers geschossen. Dieser Strahl wird als eine Verlängerung der Hand angesehen. Am Ende des Strahls ist ein Cursor, welcher anzeigt, an welcher Position sich der Strahl mit einem Hologramm überschneidet. Das Objekt, welches von dem Cursor getroffen wird, kann auf dieselbe Weise manipuliert werden wie ein Objekt, welches vom Nutzer direkt mit den Händen berührt wird. Zeigt der Nutzer mit dem *Hand Ray* auf ein Hologramm, ohne damit zu interagieren, wird der *Hand Ray* als gestrichelte Linie

dargestellt. Der Cursor hat dabei eine Ringform. Sobald der Nutzer mit dem Hologramm interagiert, schrumpft der Cursor zu einem Punkt und der Strahl besteht aus einer durchgezogenen Linie. Diese visuelle Veränderung gibt dem Nutzer Feedback für diese Interaktion. Die visuelle Veränderung des Cursors und des *Hand Rays* wird in Abbildung 15 dargestellt. Dieses Interaktionsmodell ist nur in Mixed Reality möglich, da Menschen in der realen Welt nicht in dieser Weise mit Objekten interagieren können [36].



**Abbildung 15:** Der *Hand Ray*, welcher, je nach Zustand der Interaktion, unterschiedlich visualisiert wird [36].

## 4 Entwicklung einer eigenen HoloLens Anwendung

In diesem Abschnitt wird auf die Entwicklung, den Spielablauf und auf die technischen Elemente der im Rahmen dieser Masterarbeit entwickelten HoloLens 2 Anwendung eingegangen. Des Weiteren wird erläutert, inwieweit sich die Beachtung der von Microsoft dokumentierten Design-Leitlinien noch vor Beginn der Entwicklung auf die unterschiedlichen implementierten Elemente in der Anwendung ausgewirkt hat.

### 4.1 Designüberlegungen

Bei der Implementierung der im Rahmen dieser Masterarbeit erstellten Anwendung wurde auf die in den Abschnitten 3.3 bis 3.5 vorgestellten Eigenschaften geachtet, welche zu einer bestmöglichen Erfahrung bei der Nutzung von Augmented Reality führen sollen. Auf die konkret umgesetzten Einzelheiten wird in diesem Abschnitt eingegangen.

#### 4.1.1 Komfort

Mögliche Interaktionen der eigenen Anwendung benötigen größtenteils horizontale Kopfbewegungen. Die Building Bar, auf welcher die unterschiedlichen Gebäude platziert sind, ist horizontal ausgerichtet. Auch die Burg, welche der Nutzer durch das Spielen der Anwendung errichtet, erfolgt beispielsweise auf einem Tisch und damit auf der horizontalen Ebene. Der Präsentations-Modus, bei welchem unter anderem ein Bild des historischen Gebäudes und ein Informationstext eingeblendet werden, sind ebenfalls auf der horizontalen Ebene. Lediglich zwei hohe Gebäude, der Bergfried und der Glockenturm erfordern eine vertikale Kopfbewegung. Dadurch erfolgen häufige horizontale, jedoch wenig vertikale Kopfbewegungen. Dieses Verhalten wird von Microsoft empfohlen. Da die Building Bar auf Höhe des Kopfes in angemessener Entfernung eingeblendet wird, kann der Nutzer zur Auswahl der Gebäude eine neutrale Kopfposition, also eine Ruheposition, einnehmen. Auch der Platzierungs-Modus transformiert das Gebäude auf eine für den Nutzer angemessene Position. Je nach Höhe der Oberfläche, auf welcher die Burg platziert wird, nimmt der Nutzer eine andere Kopfposition ein. Diese liegt je nach Oberfläche eventuell nicht mehr im holographischen Frame, also nicht im optimalen Bereich. Dadurch, dass der Nutzer dazu aufgefordert wird, Gebäude zu platzieren, wird dieser ebenfalls dazu animiert, sich in einem bestimmten Rahmen durch die Umgebung zu bewegen, um die Objekte zu transformieren und zu betrachten.

Da sich der Nutzer in seiner Umgebung bewegen kann, um Gebäude auf eine Oberfläche zu stellen wird unter Umständen nicht die optimale Distanz eingehalten, welche durch den optimalen Bereich definiert wird. Damit Objekte nicht zu nahe vor dem Nutzer gerendert werden und Unwohlsein verursachen, wurde in den Unreal Engine Projekteinstellungen die *Near Clipping Plane* auf 40 Zentimeter festgelegt. Da Objekte durch das Gameplay automatisch auf der Oberfläche verteilt werden, hilft dies dem Nutzer bei der *Konvergenz* der Augen. Da die Anwendung keine Objekte enthält, welche sich über große Entfernungen durch die Umgebung bewegen und der

Blick dadurch nicht oft zwischen unterschiedlichen Distanzen wechselt, wird kein Unwohlsein verursacht.

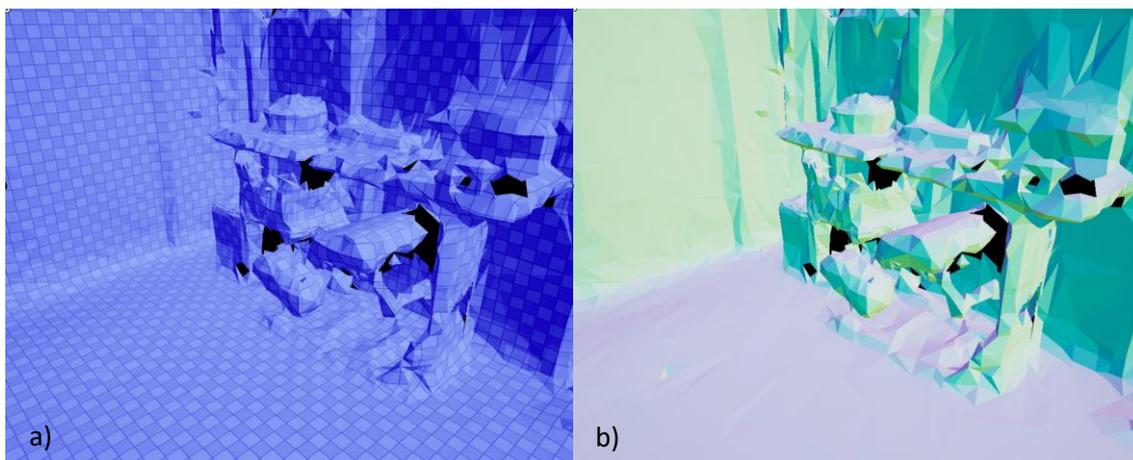
### 4.1.2 Audio

Da Audio ein wichtiger Bestandteil beim Design einer Augmented Reality Anwendung ist, wurden unterschiedliche Interaktionen und Ereignisse durch Sound untermalt, um dem Nutzer weiteres Feedback zur Verfügung zu stellen. Damit das Audio immersiver wirkt, wird dieser *spatialized*. Mögliche Okklusion, Hall oder Dämpfung von Audio wird von der Unreal Engine und der HoloLens 2 berechnet. Damit *spatialization* funktioniert, muss eine Mono-Tonspur als Audioquelle bereitgestellt werden. Da die genutzten Soundquellen zuerst lediglich in Stereo verfügbar waren, wurde das Programm Audacity benutzt, um diese in eine Mono-Tonspur umzuwandeln. In der hier vorliegenden Anwendung werden Interaktionen mit Objekten, die Nutzung von Buttons und die Erkennung von Sprachbefehlen mit Sound verbessert. Buttons besitzen einen unterschiedlichen Sound für das Hineindrücken und das Loslassen. Dadurch wird dem Nutzer die Information vermittelt, zu welchem Zeitpunkt der Button vollständig gedrückt oder losgelassen wurde und somit verdeutlicht, dass eine Aktion ausgelöst wurde. Die Platzierung von Gebäuden wurde neben visuellem auch durch auditives Feedback verdeutlicht. Wird ein Gebäude platziert, so wird ein Partikelsystem ausgelöst, welches eine Staubwolke bildet. Gleichzeitig wird ein Sound abgespielt, welcher die Platzierung auditiv untermalt. Durch *spatialized* Sound kann der Nutzer besser einschätzen, an welcher Position ein Button ausgelöst oder ein Gebäude platziert wurde. Da das Feedback für die Erkennung von Sprachbefehlen keine räumliche Position besitzt, wurde in diesem Fall auf *spatialization* verzichtet.

### 4.1.3 Spatial Mapping

Da in der Anwendung eine wichtige Komponente der Bau der Burg ist, wurde *spatial Mapping* verwendet, um die Umgebung des Nutzers im virtuellen Raum abzubilden. Da lediglich eine einfache Oberfläche für die Generierung eines *Nav-Meshs*, welches für die Wegfindung der KI-Charaktere genutzt wird, ausreicht, wird kein *Scene-Understanding Mesh* benötigt. In den aktuellen Versionen der *MRTK* Plugins wird die Generierung eines *Scene-Understanding Meshs* nicht unterstützt, weshalb selbst bei einem sinnvollen Einsatz in der eigenen Anwendung diese Funktionalität nicht genutzt werden könnte. Um dem Nutzer ein visuelles Feedback vom Zustand der generierten *spatial Map* zu liefern, wird diese im Platzierungs-Modus visualisiert. Das *GT* Plugin stellt unterschiedliche Materialien für *spatial Maps* bereit. Für visuelle Klarheit eignet sich ein Schachbrettmuster Material, welches deshalb in dieser Anwendung verwendet wurde. Dieses Schachbrettmuster und ein auf den Normalen der Geometrie basierendes Material wird in Abbildung 16 dargestellt. Die *spatial Map* sollte ursprünglich als Grundlage zur Generierung des *Nav-Meshs* verwendet werden. Obwohl die Unreal Engine ein *Nav-Mesh* zur Laufzeit generieren kann, konnte die Engine das *spatial Mesh* nicht erkennen und dadurch kein *Nav-Mesh* erzeugen. Um dieses Problem zu umgehen, ist eine große Kollisions-Box unterhalb des Bergfrieds angebracht. Der Bergfried ist zentral zu den nachfolgend platzierten Gebäuden und wird stets als

erstes Gebäude platziert. Dadurch kann das *Nav-Mesh* eine für die *KI* begehbare Fläche um den Bergfried erzeugen. Da das Design der Anwendung vorsieht, Gebäude in der Nähe des Bergfrieds auf einer ebenen Oberfläche zu platzieren, kann dadurch der Eindruck erzeugt werden, dass die *KI*, basierend auf der *spatial Map*, das *Nav-Mesh* generiert. Um die Entfernung zwischen dem *spatial Mesh* und dem zu platzierenden Gebäude besser zu visualisieren, kann ein *Proximity-Shader* eingesetzt werden, welcher einen vereinfachten künstlichen Schatten vom Gebäude auf die *spatial Map* wirft.



**Abbildung 16:** a) *Spatial Map* mit einem Material, welches durch ein Schachbrettmuster texturiert wird. b) *Spatial Map* mit einem Material, welches basierend auf den Normalen der Geometrie texturiert wird (Screenshot, *spatial Map* Mesh und Materialien sind Teil des *GT* Plugins).

#### 4.1.4 Hände

Das *UXT* Plugin enthält mehrere Komponenten, um eine direkte Manipulation von Objekten zu unterstützen. Durch einen Simulations-Modus im Unreal Engine Editor können Handinteraktionen simuliert werden, wodurch die Entwicklung erleichtert und beschleunigt wird, da nicht nur auf der HoloLens 2 Funktionen getestet werden können. Handinteraktionen mit Objekten werden in der Anwendung durch einen *Proximity-Shader* und eine *Bounding-Box* visualisiert, damit fehlendes taktiles Feedback ausgeglichen werden kann. Das Aufnehmen eines Objekts oder das Manipulieren und Transformieren einer *Bounding-Box* durch *affordances* wird außerdem durch subtile Sounds unterstützt, welche vom *UXT* Plugin bereitgestellt werden. Gebäude können nach dem Präsentations-Modus durch *affordances* der *Bounding-Box* rotiert und transformiert, der Bergfried zusätzlich skaliert werden. Um eine horizontale Rotation der Gebäude zu verhindern, wurden die *affordances* für diese Interaktion entfernt. Die Skalierung aller Gebäude, außer dem Bergfried wird verhindert, indem die *affordances*, welche an den Kanten platziert sind, entfernt werden. Dadurch kann die *Bounding-Box* nicht mehr vom Nutzer skaliert werden.

#### 4.1.5 Spracheingaben

Um die Bedienung der Anwendung zu erleichtern, wurden verschiedene Sprachbefehle in dieser implementiert. Diese können wichtige Funktionalitäten aktivieren oder deaktivieren. Die

Sprachbefehle können zum Beispiel Funktionen des Hand- beziehungsweise Near Menus auslösen. Die Building Bar einer Gebäudekategorie kann demnach durch einen Sprachbefehl geöffnet werden, anstatt den entsprechenden Button des Hand Menus zu drücken. Wird ein Sprachbefehl erkannt, wird dieser als 3D-Widget dem Nutzer für kurze Zeit angezeigt, um dem Nutzer dadurch Feedback zu geben.

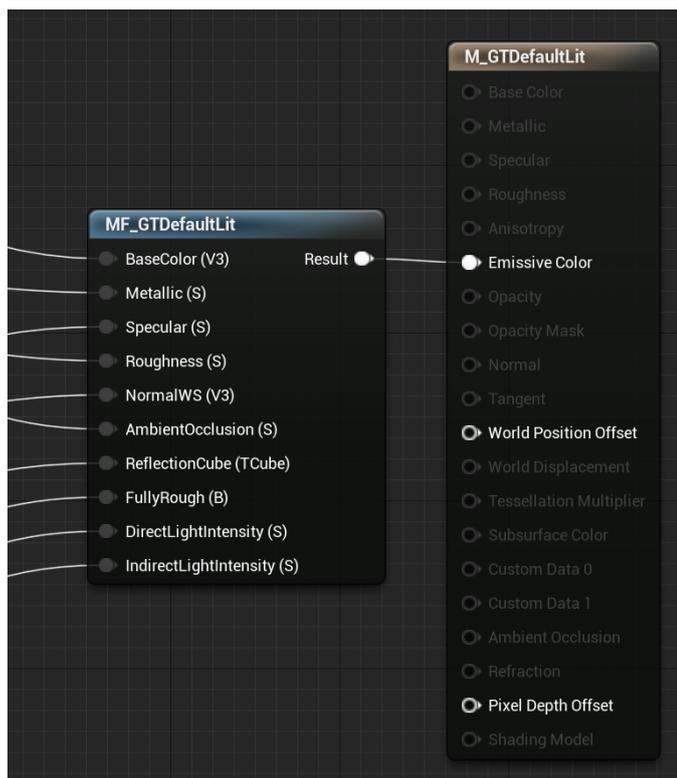
#### 4.1.6 Blickrichtung

Die *Eye-Tracking* Funktionalität der HoloLens 2 wurde sich in der Anwendung zu Nutze gemacht, indem das Öffnen des Hand Menus, neben einer vollständig geöffneten Hand, den Blick in die richtige Richtung des Nutzers erfordert. Dadurch kann vermieden werden, dass sich das Hand Menu unbeabsichtigt öffnet und nur dann aufgerufen wird, wenn der Nutzer dies beabsichtigt. Ein anderer Anwendungsfall für *Eye-Tracking* war zunächst die Inspektion von platzierten Gebäuden durch den Nutzer. Hat der Nutzer ein Gebäude angesehen, wurde ein Zeit-Indikator der Blickinteraktion angezeigt. Wurde ein Gebäude länger als drei Sekunden angesehen, wurde der obere Teil entfernt, um einen Blick in das Innere des Gebäudes zu erlauben. Die Anwendung der *Eye-Tracking* Funktionalität in diesem Zusammenhang erwies sich jedoch für den Nutzer als störend, da der Indikator auch dann geöffnet wurde, wenn der Blick nur kurz über ein Gebäude schweifte. Diese Funktionalität erlaubte dem Nutzer außerdem nicht, die Details der Gebäude über einen längeren Zeitraum zu betrachten. Daher wurde das *Eye-Tracking* für diese Interaktion entfernt und stattdessen durch eine Handinteraktion ersetzt. Hält der Nutzer eine Hand in das Gebäude, wird der Zeit-Indikator angezeigt und nach drei Sekunden wird der obere Teil des Gebäudes ausgeblendet. Diese Interaktion ist weniger störend, da der Nutzer seine Hand bewusst in das jeweilige Gebäude halten muss und somit die Burg und die Gebäude in Ruhe betrachten kann.

#### 4.1.7 Farbe, Beleuchtung und Materialien

Für die Beleuchtung der Szene wurde zunächst die vom *GT* Plugin bereitgestellte, direktionale Lichtquelle genutzt. Diese spezielle Lichtquelle sorgt für die Beleuchtung der ebenfalls vom *GT* Plugin bereitgestellten Shader. Die Shader, welche für eine hohe Performance auf Mobilgeräten optimiert sind, sorgen für eine uniforme Beleuchtung der Oberflächen. Außerdem erlauben diese bereitgestellten Shader spezielle Effekte, wie beispielsweise *Clipping-Primitive* oder künstliche Lichtquellen. Die künstlichen Lichtquellen berechnen Lichtfall nicht durch traditionelle Methoden, sondern basieren ausschließlich auf Distanz und werfen außerdem keine Schatten. Die Performance-optimierten Shader erlauben, neben der *Base Color Map*, die Nutzung verschiedener anderer Texture Maps, wie zum Beispiel einer *Roughness*-, *Metallic*- und *Normal Map*. Da der Shader aus Performancegründen, wie in Abbildung 17 dargestellt, schlussendlich jedoch lediglich den Emission-Output des Material Graphen für das Rendering nutzt, kann dieser nur begrenzt die Auswirkung von *Normal*- und *Roughness Maps* darstellen. Damit die Texturen der Gebäude realistisch wirken und die verschiedenen Texture Maps zur Geltung kommen, wurde

der Performance-optimierte Shader durch einen Standard-Shader ersetzt, bei welchem die Beleuchtung basierend auf *Physically Based Rendering (PBR)* und einer direktionalen Lichtquelle berechnet wird. In diesem Fall wurde die Anwendung mit der HoloLens 2 überprüft, da ein Testen im Editor mit einem PC kein akkurates Ergebnis der Performance mit dem eigentlichen Augmented Reality Gerät widerspiegeln kann. Da die Gebäude dynamisch platziert werden, ist außerdem eine dynamische Beleuchtung erforderlich. Obwohl der rechenintensivere Shader und dynamische Beleuchtung verwendet wird, konnte beim Testen mit der HoloLens 2 eine Frame-rate von 60 FPS erreicht werden.



**Abbildung 17:** Der Performance-optimierte Shader MF\_GTDefaultLit, der unterschiedliche Texture Maps akzeptiert, zum Rendering jedoch nur Emission nutzt (Screenshot des Shaders aus dem GT Plugin).

Große weiße Flächen, welche Unwohlsein verursachen, können bei der vorliegenden Anwendung teilweise auftreten, da einige Gebäude mit einem weißen Putz versehen sind. Diese werden jedoch durch Fenster, Pflanzen, Texturvariationen, Wandbemalungen und andere Details aufgelockert. Damit können zusammenhängende, den Nutzer blendende Flächen vermieden werden. Da durch das stilisierte Aussehen der Gebäude kontrastreiche Materialien und warme Farben verwendet werden können, treten die Gebäude in den Vordergrund. Um die Sichtbarkeit der Hologramme bei unterschiedlichen Lichtzuständen im Raum des Nutzers zu prüfen, wurde die Anwendung in unterschiedlichen Helligkeitsstufen gestartet und getestet.

Um dem Nutzer ein Gefühl für die Größe der Gebäude zu vermitteln, wird im Präsentations-Modus ein KI-Charakter vor das Gebäude gestellt. Der Charakter liefert einen für Menschen vertrauten Anhaltspunkt für den Maßstab des Gebäudes.

Während Text im Präsentations-Modus dazu verwendet wird dem Nutzer allgemeine Informationen zum Gebäude zu vermitteln, werden in der Anwendung auch Tooltips genutzt, um auf einzelne interessante Elemente der Gebäude aufmerksam zu machen.

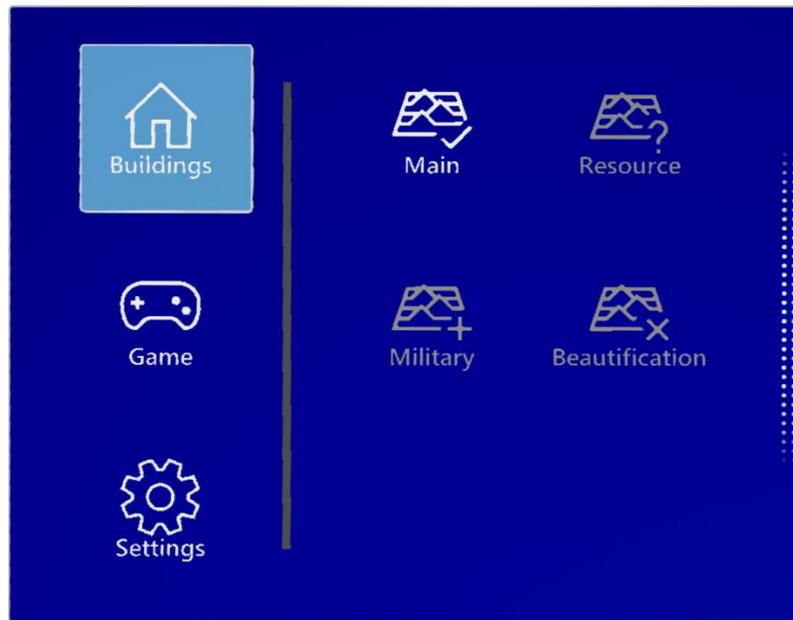
#### 4.1.8 Optimierung

Durch unterschiedliche Performance-Einstellungen wurde versucht, die Anwendung mit möglichst vielen *FPS* zu rendern. Um Unwohlsein und eine Verringerung der Immersion, durch ruckelnde oder schwimmende Hologramme, zu verhindern, sollte eine Framerate von mindestens 60 *FPS* erreicht werden. Da die Hardware der HoloLens 2 der eines Mobilgeräts entspricht, wurden Modelle mit möglichst wenigen Polygonen, Materialien und Texturen erstellt. Die Unreal Engine ermöglicht die HoloLens 2 kabellos mit einem PC zu verbinden, wodurch das volle Potential eines Computers verwendet werden kann, um Hologramme und die virtuelle Umgebung zu rendern. Die Anwendung wurde mit dem Ziel entwickelt, auf der mobilen Hardware der HoloLens 2 ausgeführt werden zu können. Bei eventuellen Frameeinbrüchen kann jedoch auch auf Streaming vom PC zur HoloLens 2 zurückgegriffen werden. Im Rahmen der Performance-Optimierung wurde eine Liste zu Unreal Engine spezifischen Projekt- und Materialeinstellungen erstellt, welche die Performance verbessern. Das *GT* Plugin besitzt dabei zu diesem Zweck einen visuellen Performance-Profiler, mit dem die Anzahl der *Draw Calls*, die Polygonanzahl der Szene, sowie die Zeit, welche die Anwendung für das Rendering eines Bildes oder der Berechnung der Gameplay-Logik in einem Frame benötigt. Der Profiler kann nicht nur im Editor verwendet werden, sondern ist auch bei der Nutzung der HoloLens 2 selbst von Vorteil, da der Profiler *Tag-Along* Verhalten besitzt und sich zum Nutzer ausrichtet.

#### 4.1.9 Hand Menu

Da das Hand Menu in der Anwendung mehrere Buttons besitzt, wurde die Funktionalität dieser, unter anderem um Übersichtlichkeit zu gewährleisten, in Kategorien unterteilt. Trotz einer Reduzierung der gleichzeitig sichtbaren Buttons im Hand Menu kann die Interaktion mit diesen zu einer Ermüdung der Arme oder des Nackens führen, da der Nutzer unter Umständen den Arm lange in einer erhobenen Position halten muss, um den gesuchten Button zu finden. Daher wurde ermöglicht das Hand Menu über einen interaktiven Bereich, welcher jeweils links und rechts vom Hand Menu platziert ist, von der Hand abzulösen und in ein Near Menu umzuwandeln. Die unterschiedlichen Kategorien des Hand Menus und ein interaktiver Bereich wird in Abbildung 18 dargestellt. Das Near Menu ist nach der Interaktion mit dem entsprechenden Bereich in der Welt verankert und kann durch einen Button oder durch ein erneutes Öffnen des Hand Menus aus der Umgebung entfernt werden. Der erste Menüpunkt des Hand Menus ist *Building*. Hier werden die vier unterschiedlichen Gebäudekategorien aufgelistet. Der zweite Menüpunkt ist *Game*. Hier kann der Nutzer spielbezogene Einstellungen, wie zum Beispiel Tooltips, welche die Namen der platzierten Gebäude anzeigen, aktivieren. Die Spieleinstellungen erlauben außerdem ein manuelles Ein- und Ausblenden der *spatial Map*. In der dritten Kategorie,

*Settings*, kann der Nutzer allgemeine Einstellungen vornehmen. Beispielsweise kann die in Textfeldern dargestellte Sprache gewechselt werden. Außerdem finden sich in dieser Kategorie Debug-Funktionen, welche beim *Profiling* der Performance und der manuellen Auslösung von Ereignissen der Anwendung helfen.



**Abbildung 18:** Das implementierte Hand Menu. Die Punkte, ein interaktiver Bereich, auf der rechten Seite weisen den Nutzer darauf hin, dass das Hand Menu in ein Near Menu umgewandelt und in der Welt verankert werden kann.

Um eine gute Lesbarkeit bei Texten zu erreichen, wurde der Info-Text im Präsentations-Modus möglichst groß dargestellt. Dadurch kann eine Belastung der Augen reduziert werden. Als Schriftart wird Roboto eingesetzt.

## 4.2 Technische Umsetzung

In diesem Abschnitt wird auf die technische Umsetzung der Anwendung, welche im Rahmen dieser Masterarbeit erstellt wurde, eingegangen. Neben der Erläuterung des Arbeitsablaufes für die Erstellung der 3D-Modelle und Texturen, wird anschließend der Ablauf des Gameplays beschrieben.

### 4.2.1 3D-Modelle und Texturen

Für die Erstellung der 3D-Modelle wurde Blender 2.92.0 verwendet. Um eine schnelle Exportierung der Gebäudemodelle in die Unreal Engine zu ermöglichen, wurde das Blender Addon Send to Unreal, welches von Epic Games entwickelt wurde, genutzt. Dadurch kann der manuelle Export- und Import von Objekten im .fbx Format beschleunigt werden, indem durch einen Button in Blender dieser Vorgang automatisiert durchgeführt werden kann.

Aus der Liste von Gebäuden, welche im Game Design Dokument definiert sind, wurden zehn Gebäude mit einer hohen Priorisierung ausgewählt. Die Gebäude wurden in die vier Kategorien Haupt-, Ressourcen-, Militär- und Ziergebäude eingeteilt. Zur Kategorie der Hauptgebäude zählen der Bergfried, die Kapelle und die Kemenate. Als Ressourcengebäude kann der Nutzer eine Küche, eine Schmiede, einen Brunnen und einen Kräutergarten platzieren. Als Militärgebäude steht die Kaserne, die Waffenkammer und Stallungen zur Verfügung. In der vierten Gebäudekategorie, den Ziergebäuden, kann ein Glockenturm platziert werden. Insgesamt mussten daher diese zehn unterschiedlichen Gebäude modelliert werden. Als Referenz für das Design und Aussehen der Gebäude wurde das Hohe Schloss Füssen ausgewählt, da dieses über kontrastreiche Farben, einfache Gebäudeformen und einfache Oberflächen verfügt.

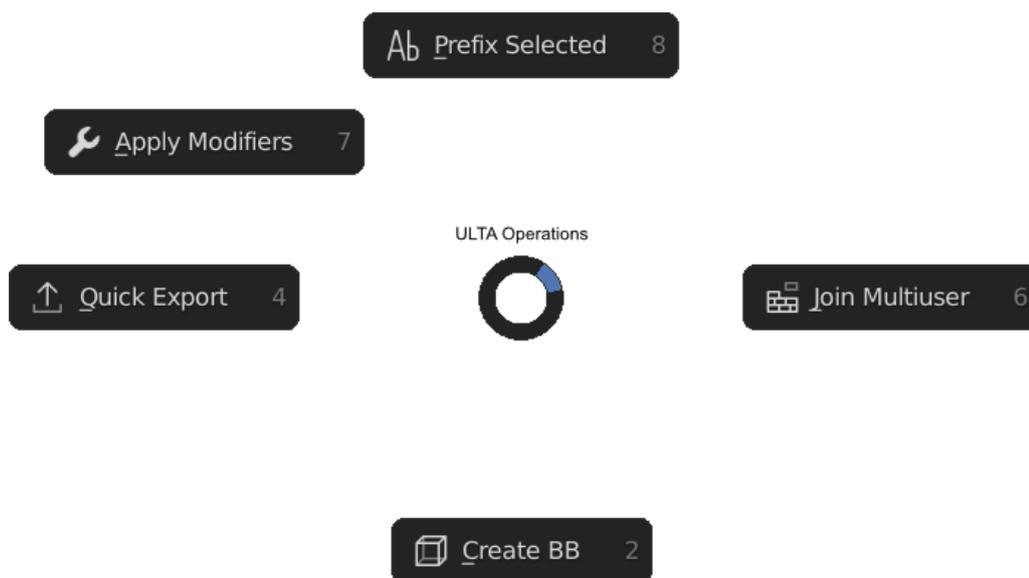
Um die Gebäude effizient zu erstellen, wurde eine Ansammlung aus modularen Gebäudeteilen, wie zum Beispiel unterschiedliche Fenster, Türen und andere Erweiterungen erstellt. Einige der verwendeten modularen Gebäudeteile werden in Abbildung 19 dargestellt. Die Gebäudeteile können dann zu den benötigten Gebäuden zusammengesetzt werden.



**Abbildung 19:** Eine Übersicht von einigen modularen Gebäudeteilen, welche zur Erstellung der unterschiedlichen Gebäude verwendet wurden.

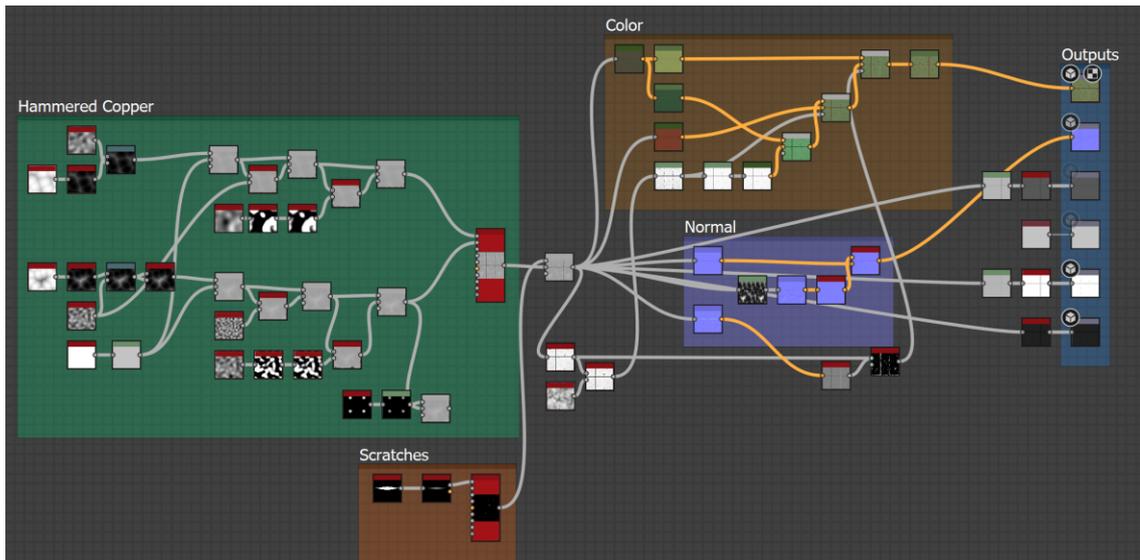
Um unterschiedliche Aktionen für den Export von Blender in die Unreal Engine zu vereinfachen, wurde ein eigenes Blender Addon mit python entwickelt. Dieses besitzt fünf Operatoren, welche über ein *Pie Menu* aufgerufen werden können. Operatoren sind eine Reihe von Anweisungen, welche mehrere Aktionen in Blender automatisieren können. Durch das Addon können in Blender ausgewählte Objekte, durch den Operator *Prefix Selected*, an den Unreal Engine Asset Name Standard angepasst werden. Ein Objekt ohne *Armature* bekommt daher das Präfix *SM\_ (Static*

*Mesh*), während ein Objekt mit *Armature* automatisch das Präfix *SK\_ (Skeletal Mesh)* zugewiesen bekommt. Da die Gebäude aus geklonten Gebäudeteilen bestehen und Blender nur einzelne Instanzen zu einem Objekt zusammenfügen kann, wurde der Operator *Join Multiuser* entwickelt. Bei diesem Operator werden mehrere Arbeitsschritte automatisiert, unter anderem das Festlegen des Pivot-Punkts und die Entfernung von überlappenden Vertices, welche durch das Zusammenfügen zu einem Objekt entstehen. Für die Generierung einer *Bounding-Box*, welche für die direkte Manipulation von Gebäuden in der Anwendung benötigt wird, sorgt der Operator *Create BB*. Da das Send to Unreal Addon für den Export von Objekten voraussetzt, dass sich diese in einem bestimmten Unterordner der Szenen-Hierarchie von Blender befinden, ist eine manuelle Verschiebung aller Objekte in diesen Ordner erforderlich. Damit die selbst definierte Ordnerstruktur beibehalten werden kann, wurde der Operator *Quick Export* entwickelt. Dieser Operator verschiebt ausgewählte Objekte in den spezifischen Unterordner, garantiert die korrekte Position des Pivot-Punkts und versetzt die exportierten Objekte anschließend in ihre ursprünglichen Ordner zurück. Da Objekte zum Teil mehrere *Modifier* verwenden, hilft der Operator *Apply Modifiers* dabei, einen kompletten *Modifier-Stack* in der richtigen Reihenfolge auf das Objekt anzuwenden. Das *Pie Menu*, welches die beschriebenen Operatoren aufrufen kann, wird in Abbildung 20 dargestellt.



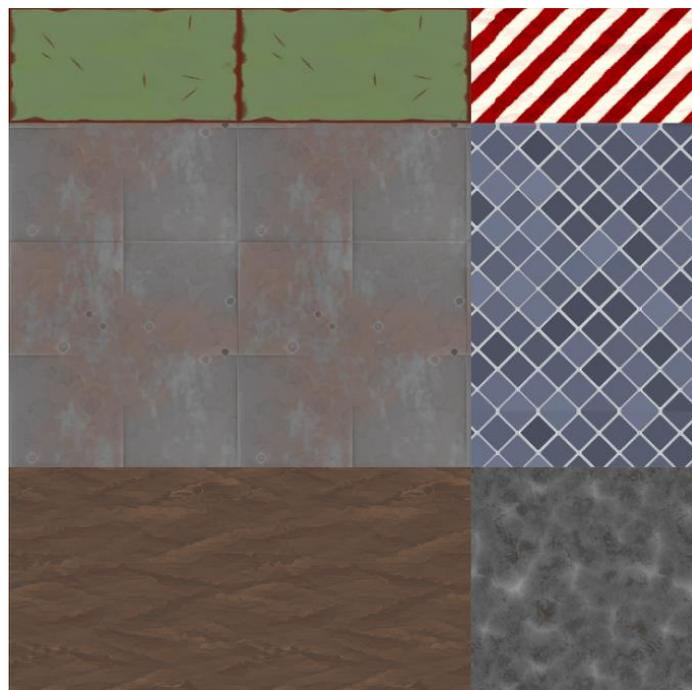
**Abbildung 20:** Der Export von Gebäudemodellen von Blender in die Unreal Engine wurde durch ein eigens entwickeltes Addon beschleunigt. Die Operatoren können durch ein *Pie Menu* aufgerufen werden.

Die stilisierten Texturen wurden mit Hilfe des Programms Substance Designer erstellt, welcher durch einen Node-basierten Graph Editor prozedurale Texturen erzeugen kann. Einzelne Merkmale, zum Beispiel die Abnutzung, Farbe und Anzahl an Kratzern auf einem Kupferdach, können dadurch parametrisiert und durch wenige Anpassungen der Parameter verändert werden. Abbildung 21 zeigt beispielhaft den Graphen, welcher für die prozedurale Kupferdach-Textur erstellt wurde. Andere Texturen des Trim Sheets besitzen eine ähnliche Graph-Komplexität.



**Abbildung 21:** Substance Designer Graph für eine prozedurale Kupferdach-Textur.

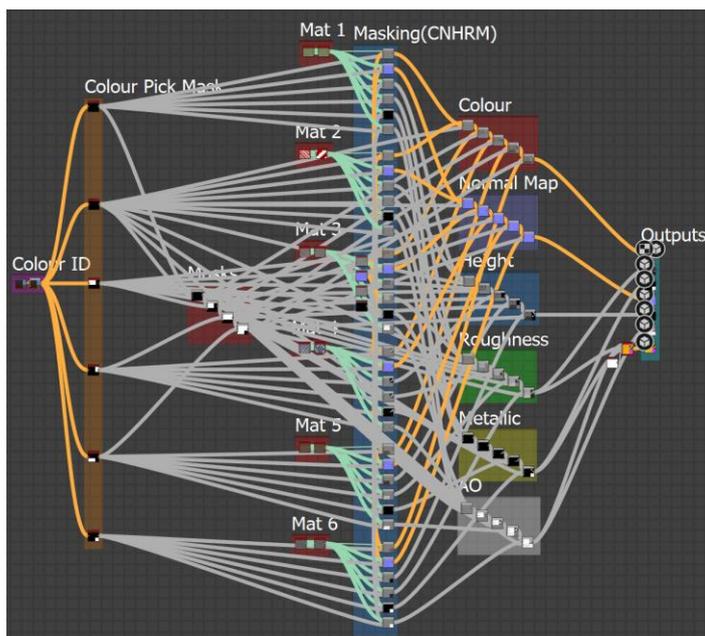
Zugehörige weitere *PBR*-Maps, wie *Normal*- oder *Roughness*-Maps, werden, basierend auf den Änderungen, aktualisiert. Abbildung 22 zeigt die *Base Color Map* eines erstellten Trim Sheets. Auf diesem wurde eine Kupferdach-, Fensterladen-, Stein-, Glas-, Holz- und Metall-Textur untergebracht. Für die Anwendung wurden zwei Trim Sheets mit insgesamt 9 Texturen erstellt.



**Abbildung 22:** Ein Trim Sheet, welches in sechs Bereiche unterteilt ist und damit sechs unterschiedliche Oberflächen aus einem bestimmten Material repräsentieren kann.

Wie in Abschnitt 4.1.8 beschrieben, wurde aus Performancegründen versucht, möglichst wenige Materialien und Texturen zu erstellen. Durch die Nutzung weniger Materialien kann die Anzahl

an *Draw Calls* reduziert werden. Des Weiteren verringert die Verwendung von möglichst wenigen Texturen die Anzahl an *Texture Samples* in Shadern und reduziert den benötigten Speicher. Die Reduzierung von Materialien und Texturen konnte durch die Verwendung von Trim Sheets erreicht werden. Bei Trim Sheets werden mehrere unterschiedliche Texturen innerhalb einer UV-Map gespeichert. Dadurch können unterschiedliche Oberflächen von einem Material und durch eine Textur erzeugt werden. Mit Hilfe von Substance Designer wurde ein Graph entwickelt, durch welchen bis zu sechs Materialien zu einem Trim Sheet angeordnet werden können. Dieser Graph wird in Abbildung 23 dargestellt.



**Abbildung 23:** Der Substance Designer Graph, welcher ermöglicht, bis zu sechs unterschiedliche Materialien in eine Textur zu verpacken. Die Positionierung der Bereiche erfolgt durch eine *Color-ID* Map.

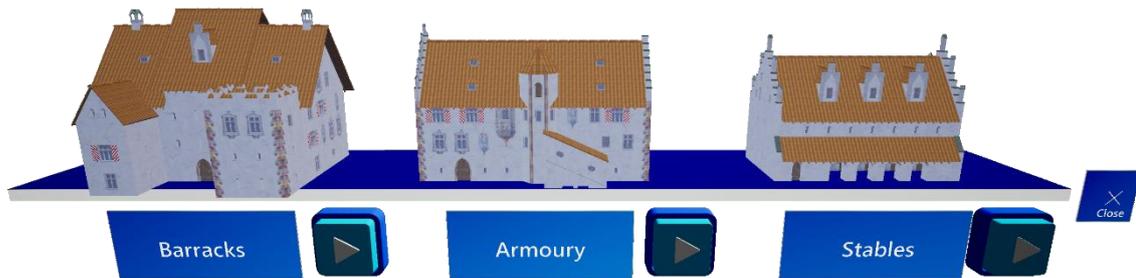
Da die Putz-Textur großflächig bei Gebäuden angewendet wird und keine sichtbaren Kanten der Textur auf den Gebäuden zu sehen sein sollten, wurde für dieses Material kein Trim Sheet verwendet. Anstatt das Material basierend auf die in Blender erstellten UV-Koordinaten zu projizieren, wurde in der Unreal Engine ein Shader erstellt, welcher prozedural, basierend auf dem kompletten Objekt, UV-Koordinaten erzeugt. Dadurch kann eine, sich nahtlos wiederholende Textur für ein beliebig großes Gebäude verwendet werden.

#### 4.2.2 Gameplay

Um ein Gebäude zu platzieren, müssen vom Nutzer einige Schritte durchgeführt werden. Dies hat die Absicht, dass der Nutzer mit den Gebäuden interagieren muss und dabei eine Lernerfahrung erhält. Durch unterschiedliche Auswahlmöglichkeiten entstehen unterschiedliche Vorgehensweisen, mit welchen der Nutzer durch die Anwendung navigieren kann. Um eine Übersicht über die Abläufe zu erhalten, wurde ein Flussdiagramm erstellt, welches am Ende dieses Abschnitts, in Abbildung 27, dargestellt wird.

Nachdem der Nutzer die Anwendung startet, wird dieser dazu aufgefordert, sich in seiner Umgebung umzusehen, damit die HoloLens 2 eine *spatial Map* erstellen kann. Die Anwendung besteht danach aus einem leeren Level mit einer *spatial Map*. Anschließend wird dem Nutzer der Hinweis gegeben, dass nun das erste Gebäude platziert werden kann. Dazu muss der Nutzer seine Hand vollständig öffnen und diese ansehen, damit sich das Hand Menu öffnet. Beim Hand Menu können drei Kategorien ausgewählt werden. Im ersten Menüpunkt, *Building*, können die vier Gebäudekategorien ausgewählt werden. Im zweiten Menüpunkt, welches den Namen *Game* trägt, können Spielspezifische Einstellungen, wie das Aktivieren und Deaktivieren von Gebäude-Tooltips, getroffen werden. Im dritten Menüpunkt, *Settings*, kann der Nutzer allgemeine Einstellungen zur Anwendung vornehmen. Beispielsweise kann die Sprache geändert werden. Hier kann zwischen Deutsch oder Englisch gewählt werden. Bei der Entwicklung der Anwendung wurde sich die von der Unreal Engine bereitgestellte Lokalisierungs-Funktion zu Nutze gemacht. Die Funktion erlaubt, die im Projekt verteilten Textkomponenten in einer Liste aufzuführen. Dadurch kann Text an einer zentralen Position übersetzt und eine Lokalisierung erstellt werden.

Wurde die Anwendung vor kurzem gestartet und noch kein Gebäude platziert, muss der Bergfried als erstes Gebäude platziert werden. Deshalb ist zu Beginn lediglich die *Main* Gebäudekategorie verfügbar. Andere Gebäudearten sind deaktiviert. Ausschließlich der Bergfried kann mittels der *affordances* einer *Bounding-Box* skaliert werden. Dadurch kann ein Maßstab für alle anderen Gebäude festgelegt und somit verhindert werden, dass der Nutzer Gebäude auf unterschiedliche Größen skaliert. Wird einer der vier Buttons der Gebäudekategorien betätigt, so öffnet sich die Building Bar. Dieser ist ein zentraler Bestandteil der Anwendung. Alle Gebäude, die sich in dieser Kategorie befinden, werden auf der Building Bar aufgereiht und präsentiert, können betrachtet und ausgewählt werden. Ein Beispiel einer Building Bar wird in Abbildung 24 dargestellt. Die Building Bar wird unmittelbar vor dem Körper des Nutzers erzeugt und befindet sich damit immer im Sichtfeld. Wird eine andere Gebäudekategorie vom Nutzer im Hand Menu ausgewählt, wird die existierende Building Bar entfernt und die neue Building Bar an der entsprechenden Position neu erzeugt. Das Konzept der Building Bar wurde erstellt, damit der Nutzer nicht mit verschachtelten Menüs oder einem großen Hand Menu überfordert wird. Die Anzahl an Gebäuden auf einer Building Bar ist nur durch den horizontalen Raum begrenzt. Damit ist eine einfache Addition weiterer Gebäude möglich. Ist das Hand Menu zu diesem Zeitpunkt noch geöffnet, kann der Nutzer dieses optional schließen, damit die Verdeckung anderer Objekte im Level vermieden wird. Die Building Bar besitzt neben den Gebäuden einen Close Button, damit die Building Bar auf Wunsch auch wieder aus dem Level entfernt werden kann. Der Nutzer kann die Building Bar selbst durch direkte Manipulation rotieren und verschieben. Will der Nutzer ein Gebäude auswählen, muss er dazu den unter dem Gebäude befindlichen Play Button drücken. Befindet sich die Hand des Nutzers in der Nähe des Buttons, wird ein gefärbter Umriss (Outline) um das zugehörige Gebäude gerendert. Damit wird dem Nutzer zusätzliches visuelles Feedback vermittelt. Beim Drücken der Taste wird der Präsentations-Modus gestartet.



**Abbildung 24:** Die Building Bar für die Gebäudekategorie *Military*. Darauf befinden sich die Kaserne, die Waffenkammer und die Stallungen. Neben den Namen der Gebäude befindet sich ein Button, welcher den Präsentations-Modus startet.

Für die Buttons wurden nicht überall die vom *UXT* Plugin bereitgestellten, flachen Buttons genutzt (siehe Buttons des Hand Menus in Abschnitt 4.1.9), sondern eigene Buttons modelliert, welche eine ausgeprägtere dreidimensionale Form besitzen. Diese lenken dadurch eher die Aufmerksamkeit des Nutzers auf sich und sorgen für ein taktileres Gefühl beim Drücken des Buttons.

Der Präsentations-Modus besteht aus mehreren Schritten. Zunächst wird das Gebäude von der Position der Building Bar ins Sichtfeld unmittelbar vor den Nutzer verschoben. Anschließend wird das Gebäude rotiert und während dieser Rotation wird der obere Teil des Gebäudes ausgeblendet, damit in das Innere des Gebäudes gesehen werden kann. Nach einer kurzen Pause werden dann zwei Info-Panels mit einer Animation eingeblendet. Ein Panel enthält dabei Text mit allgemeinen Informationen zum Gebäude, ein anderes Panel zeigt das Bild eines realen historischen Gebäudes. Gleichzeitig werden Tooltips geöffnet, welche auf einzelne Elemente des Gebäudes aufmerksam machen und diese benennen. Im Präsentations-Modus stehen dem Nutzer zwei Buttons zur Verfügung. Ein Button erlaubt den oberen Teil des Gebäudes ein- und auszublenken, der andere Button startet den Platzierungsmodus. Die unterschiedlichen Elemente des Präsentations-Modus werden in Abbildung 25 dargestellt.



**Abbildung 25:** Der Präsentations-Modus mit Info-Panels und Tooltips, nachdem alle Animationen abgespielt wurden.

Im Platzierungs-Modus kann der Nutzer das aktuell ausgewählte Gebäude transformieren. Wird ein Bergfried platziert, stehen zusätzlich *affordances* zur Skalierung zur Verfügung. Andere Gebäude werden, sobald der Platzierungs-Modus gestartet wird, auf den Maßstab des zuvor platzierten Bergfrieds skaliert. Ist der Nutzer mit der Transformation des Gebäudes zufrieden, kann durch einen Confirm Button das Gebäude platziert werden. Außerdem besteht die Möglichkeit, den kompletten Vorgang abzubrechen, indem der Cancel Button gedrückt wird. Sobald ein Gebäude platziert wurde, kann dieses nicht mehr transformiert werden. Die KI-Charaktere werden gespawnt und beginnen ihre Arbeit.

Für die Logik der KI-Charaktere wurde der in der Unreal Engine verfügbare *Behavior-Tree* Editor genutzt. Der für die Anwendung erstellte *Behavior-Tree* wird in Abbildung 26 dargestellt. Nachdem die KI gespawnt wurde, wird zunächst überprüft, welche Eigenschaften ein Charakter besitzt. Je nach Gebäude und Geschlecht sucht sich ein KI-Charakter einen bestimmten Arbeitsplatz. Da der Arbeitsplatz bei einem anderen Gebäude liegt, muss zunächst festgestellt werden, ob ein bestimmtes Gebäude bereits platziert wurde. Dafür wurde eine Hilfsklasse im Level platziert, welche vom Nutzer platzierte Gebäude in einer Liste speichert und die Zusammenhänge zu anderen Gebäuden aufbaut. Dadurch kann im *Behavior-Tree* abgefragt werden, ob bestimmte Gebäude platziert sind und ob dort ein Arbeitsplatz frei ist. Die „Arbeit“ besteht aus dem Abspielen einer Animation und einer anschließenden Pause. Wurde kein Arbeitsplatz ausfindig gemacht, dann bewegt sich die KI zu einem zufälligen Punkt im Level, um nicht tatenlos vor einem Gebäude zu stehen.

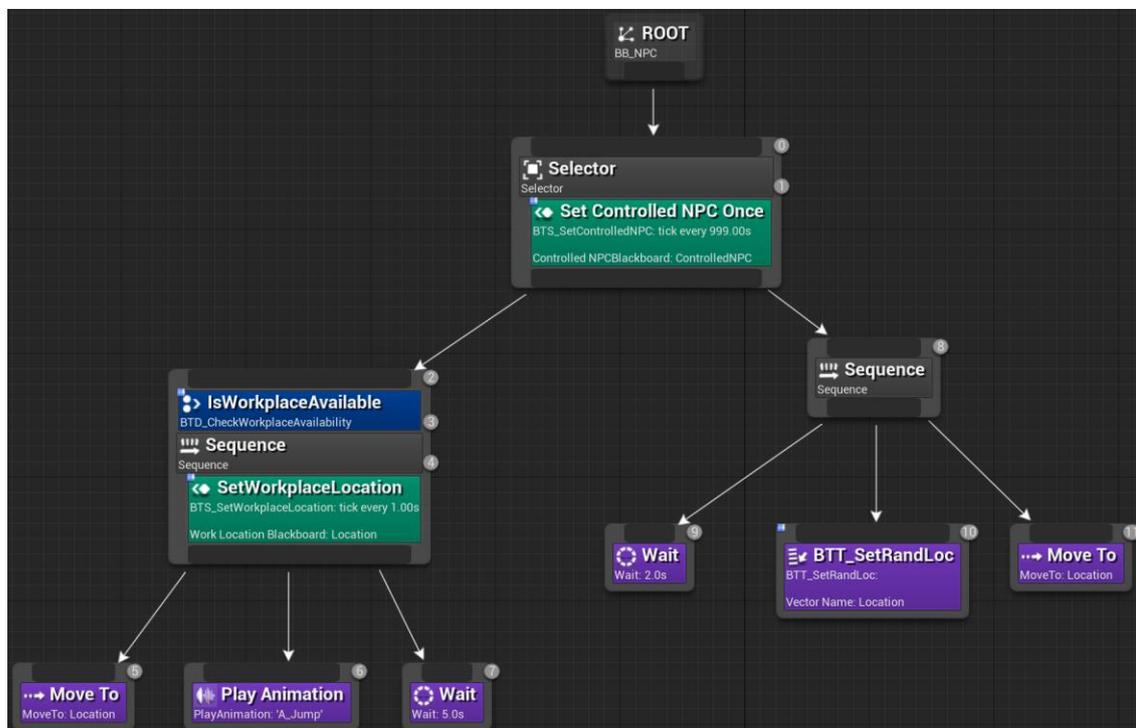
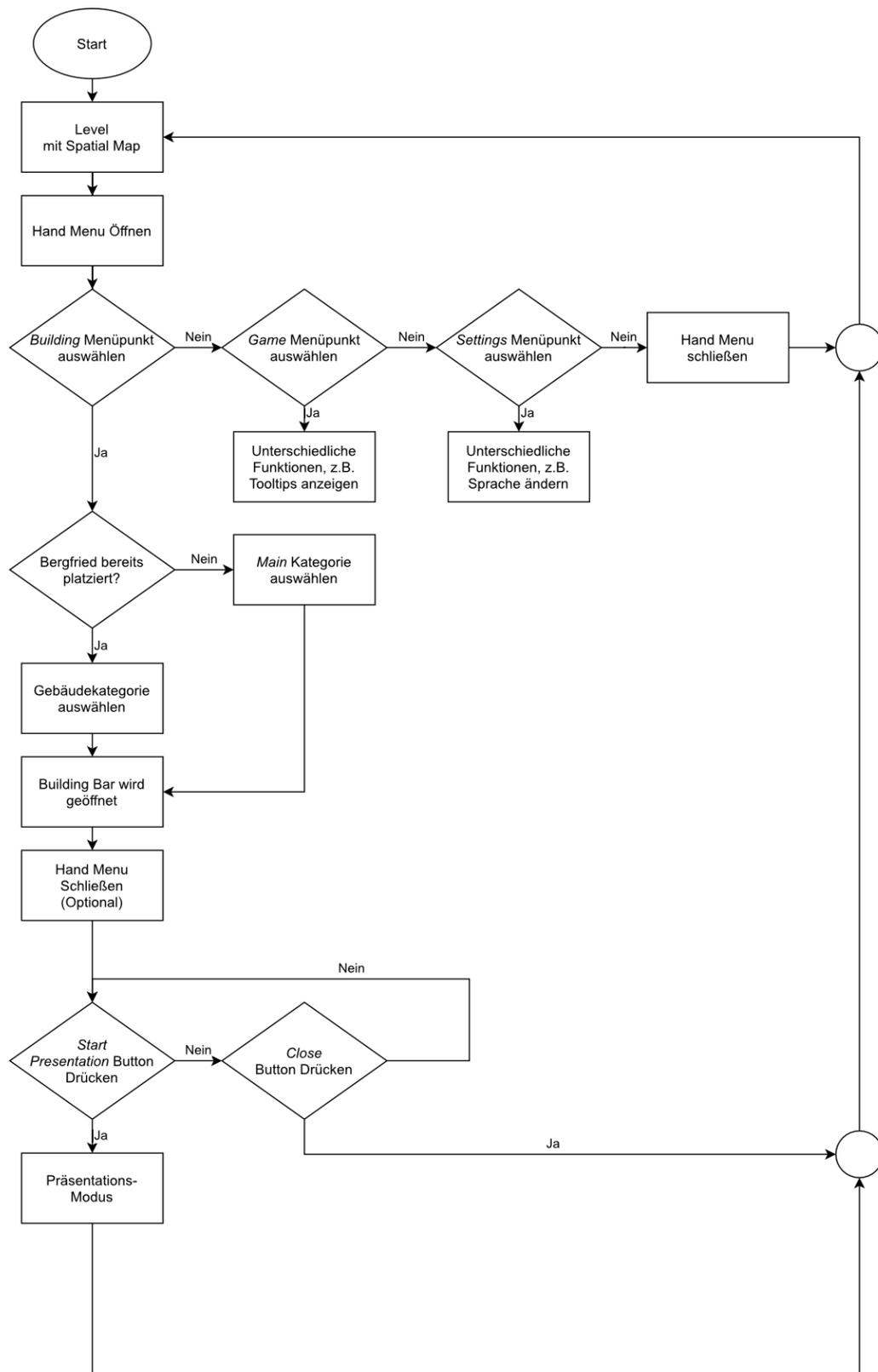
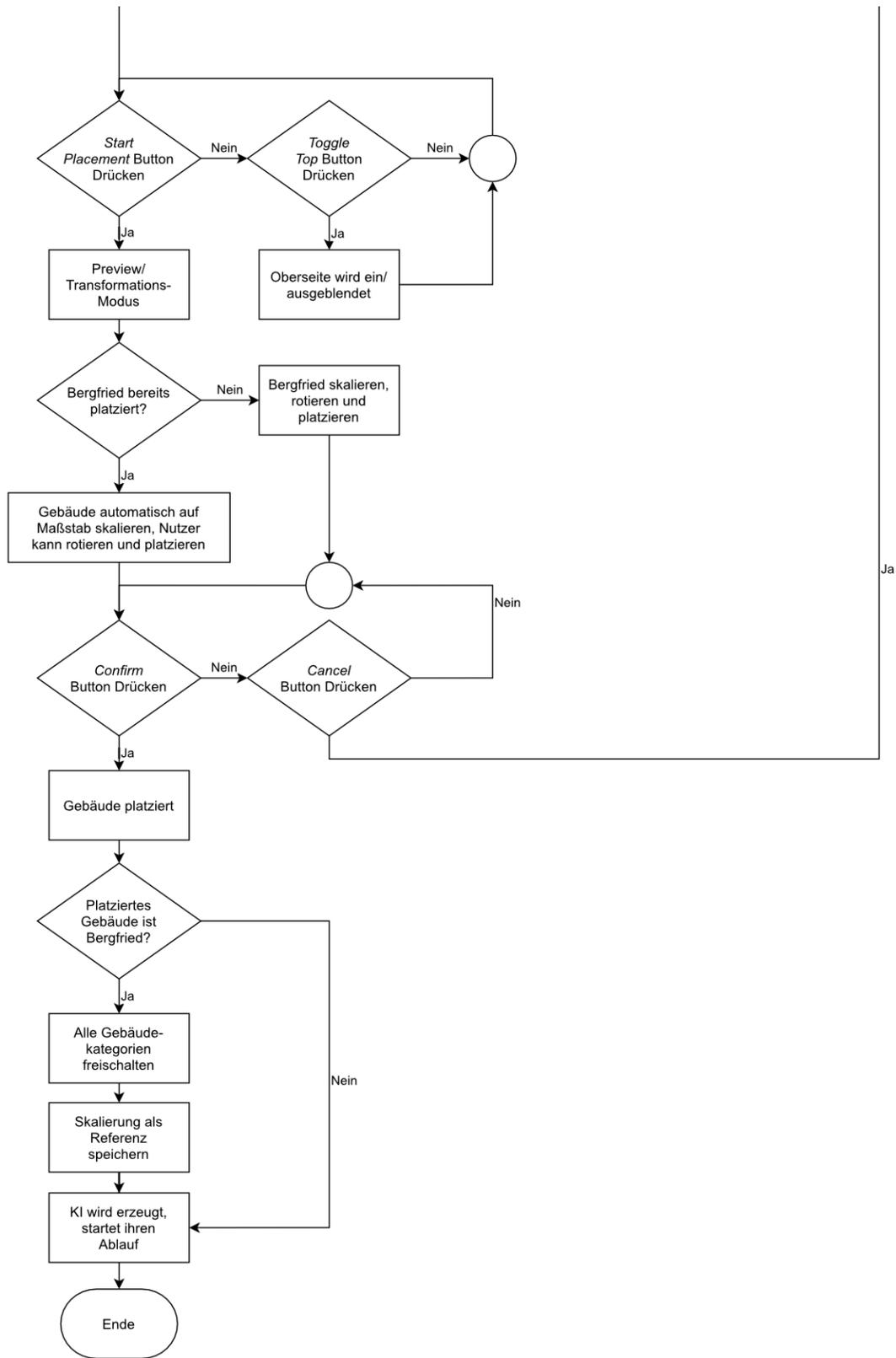


Abbildung 26: Der *Behavior-Tree*, welcher das Verhalten der KI-Charaktere steuert.





**Abbildung 27:** Flussdiagramm, welches den Gameplay-Ablauf der HoloLens 2 Anwendung darstellt.

## 5 Augmented Reality in der Lehre

Der Einsatz von Augmented Reality kann in vielen unterschiedlichen Bereichen, wie zum Beispiel in der Industrie, der Wissenschaft und der Lehre zu Verbesserungen führen. In diesem Abschnitt wird daher auf die verschiedenen Einsatzmöglichkeiten, die Vor- und Nachteile und die Zukunft dieser Technologie im Lernbereich, eingegangen.

### 5.1 Anwendung in der Lehre und E-Learning

Um herauszufinden, in welchem Ausmaß Augmented Reality den Nutzer in der Lehre unterstützen kann, wurden verschiedene Studien durchgeführt. Ein Beispiel dafür ist die Forschung von Billingham et al. [37], bei welcher Lehrbücher durch Augmented Reality ergänzt wurden. Dabei bekamen die Schüler im Alter von 10 bis 14 Jahren Schulbücher, bei welchen eine Augmented Reality Anwendung erklärende Informationen und virtuelle Objekte über die Buchseiten projizierte. Mit Hilfe von Audio-Effekten und Erzählungen des Autors wurde diese Lernerfahrung weiter ausgebaut. Ein von den Lehrern erstellter Bericht kam zu dem Schluss, dass die Schüler Spaß bei der Nutzung der Anwendung empfanden und dadurch motivierter waren. Dieselbe Studie untersuchte auch das Lernen im Physikunterricht mittels einer Augmented Reality Anwendung. Diese Anwendung visualisierte Themen, wie magnetische Felder, durch den Einsatz von 3D Modellen. Um die Wirksamkeit der Anwendung zu prüfen, wurden zwei Gruppen verglichen. Eine Kontrollgruppe nutzte das Lehrbuch, um Informationen zu magnetischen Feldern vermittelt zu bekommen, während die zweite Gruppe zusätzlich Augmented Reality für den Lernprozess einsetzte. Die Testgruppe, welche die Augmented Reality Anwendung nutzte, erreichte in einem anschließend durchgeführten Test ein besseres Ergebnis, mit einem Mittelwert von 72 Prozent richtig beantworteter Fragen, während die Kontrollgruppe einen Mittelwert von 60 Prozent richtiger Antworten erreichte. Vier Wochen nach dem ersten Test wurden beide Gruppen erneut befragt. Die Gruppe, welche durch Augmented Reality unterstützt wurde, erreichte in diesem Test ebenfalls ein besseres Ergebnis, mit einem Mittelwert von 55 Prozent, während die Kontrollgruppe bei diesem erneut durchgeführten Test lediglich einen Mittelwert von 45 Prozent erzielte. Daraus lässt sich schließen, dass jene Schüler, welche durch Augmented Reality unterstützt wurden, die vermittelten Informationen besser im Gedächtnis behalten konnten. Durch Augmented Reality unterstütztes Lernen hat sich für aktive Lernmethoden als effektiv erwiesen, da die Technologie den Nutzern die Möglichkeit bietet, Informationen im Langzeitgedächtnis zu speichern. Dies wurde in einer Studie zu Augmented Reality in der Lernumgebung von Santos et al. erklärt. In dieser Studie wurde die Hypothese aufgestellt, dass die Augmented Reality unterstützten Lernumgebungen bei der Lehre helfen können. Da die Lernumgebung vom Nutzer aktive Eingaben verlangt, prägt sich dieser die, mit den Interaktionen verbundenen sensorischen Informationen, besser ein. Die sensorischen Informationen können dabei visuelle oder auditive Effekte sein oder durch Berührung von Hologrammen entstehen. Die Information wird zunächst im Kurzzeitgedächtnis gespeichert, wobei durch die aktive Teilnahme das sensorische Gedäch-

nis ebenfalls miteinbezogen wird und die Information anschließend im Langzeitgedächtnis gespeichert werden kann. Die Studie empfiehlt außerdem drei Kriterien, welche bei der Entwicklung einer Augmented Reality Anwendung für die Lehre beachtet werden sollten, damit Informationen im Langzeitgedächtnis gespeichert werden können. Das erste Kriterium ist die Annotation der realen Welt. Objekte in der realen Welt können mit virtuellem Text, Symbolen oder 3D-Objekten versehen werden, um diese dem Nutzer näher zu beschreiben. Das zweite Kriterium zur erfolgreichen Speicherung von Informationen im Langzeitgedächtnis ist, dass reale Objekte in den Fokus der Lernumgebung gerückt werden sollen. Durch Augmented Reality erzeugte, virtuelle Elemente können dieses Objekt mit einzigartigen Interaktionsmöglichkeiten erweitern, wodurch die Lernbereitschaft verbessert werden kann. Das dritte Kriterium beruht auf der *multimedialen Lerntheorie*. Durch die Verwendung von multimedialen Inhalten können die Schüler im Unterricht unterstützt werden. Diese Inhalte können Videos, Audio oder interaktive Elemente sein, welche ohne den Einsatz von Augmented Reality ansonsten im Unterricht nicht möglich sind. Werden die drei genannten Kriterien beachtet, kann sich der Schüler mit Hilfe der Anwendung in ein Thema vertiefen, um dieses selbstständig in einer sinnvollen Art und Weise zu erlernen [38].

### 5.1.1 Vor- und Nachteile

Die Nutzung virtueller Technologien, wie Augmented Reality, wird mit einer Verbesserung der Leistung und Motivation, sozialen und kollaborativen Fähigkeiten, sowie psychomotorischer und kognitiver Fähigkeiten der Schüler verbunden [39]. Dies beruht auf der immersiveren Visualisierung stereoskopischer 3D-Effekte. Des Weiteren werden durch deren Nutzung Zeit und Kosten gespart [40]. Virtuelle Technologien ermutigen zu einer aktiven Teilnahme am Unterricht, da Augmented Reality bei der Interaktion mit virtuellen Objekten oder Umgebungen die Entscheidungsfindung der Schüler fördert. Dadurch wird ein selbstständiges Erkunden, Verstehen komplexer Konzepte, Schaffen neuer Erfahrungen und praktisches Handeln ermöglicht. Darüber hinaus erlaubt eine Echtzeit-Interaktion mit virtuellen Umgebungen eine sofortige Visualisierung der Ergebnisse. Dadurch sind Schüler in der Lage, Entscheidungen, basierend auf diesen aktualisierten Ergebnissen, zu treffen, was wiederum die Lernleistung und kognitiven Fähigkeiten verbessert. Augmented Reality ermöglicht eine kollaborative Interaktion, wobei die virtuellen Umgebungen die Interaktion und Zusammenarbeit der Schüler untereinander fördern können. Diese Vorteile stärken das Engagement der Schüler, indem immersive Erfahrungen genutzt werden, um Ablenkungen aus der realen Umgebung zu reduzieren. Durch ein besseres und schnelleres Feedback kann eine positive Einstellung gegenüber dem Lernen geschaffen werden, wodurch Lernziele einfacher erreicht werden können. Da Augmented Reality eine First-Person Erfahrung bietet, ist die Immersion in die virtuelle Umgebung eine der größten Vorteile dieser Technologie. Eine durchdachte virtuelle Umgebung entspricht daher einer direkteren und zielgerichteteren Lernerfahrung, als diese beispielsweise durch Lehrfilme geboten werden kann.

Weiterhin erhöhen virtuelle Technologien die Motivation und das Engagement von Schülern. Die Schüler erhalten beim Einsatz dieser Technologie eine immersive Erfahrung und übernehmen die Rolle des Protagonisten, während sie Informationen durch den Einsatz von 3D-Modellen vermittelt bekommen. Des Weiteren erlauben diese Technologien einen konstruktivistischen Lernansatz. Die Schüler können entweder mit virtuellen Objekten oder anderen Studenten interagieren. Dadurch können Schüler nachforschen, experimentieren und Feedback erhalten. Dies führt zu einer verbesserten Lernerfahrung.

Virtuelle Technologien, wie Handheld Augmented Reality, auf welches in Abschnitt 6.2 eingegangen wird, sind kostengünstig und leicht zugänglich. Durch technologische Fortschritte wird der Zugang zu Augmented Reality auf Smartphones und Tablets erleichtert. Komplexe Geräte sind keine zwingende Voraussetzung mehr. Ein weiterer Vorteil ist die einfache Bereitstellung virtueller Umgebungen für körperlich eingeschränkte Menschen. Diese können trotz unterschiedlicher Einschränkungen, mit Hilfe von Augmented Reality, mit virtuellen Objekten und anderen Studenten interagieren.

Die Nutzung von Augmented Reality in der Lehre ist ähnlich vorteilhaft zum Einsatz von computerunterstütztem Unterricht, beispielsweise durch computerbasierte Simulationen. Selbst, wenn Simulationen nur die Realität nachbilden, so werden bestimmte Merkmale genutzt, um eine reale Erfahrung zu verbessern. Bei der Simulation können zum Beispiel zusätzliche Informationen präsentiert werden, welche bei einer realen Situation nicht direkt sichtbar sind. Simulationen ermöglichen den Schülern, neue Bereiche zu erforschen, Vorhersagen zu treffen, Experimente zu entwerfen und Ergebnisse zu interpretieren. Der Erfolg von computerunterstütztem Unterricht basiert auf der Lehrfähigkeit des Systems, der Nutzung neuer Lehransätze und der Entwicklung kognitiver Fähigkeiten und positiven Einstellungen gegenüber dem Lehrgebiet [39].

Neben den Stärken, welche Augmented Reality für Lernumgebungen bietet, müssen jedoch auch potenzielle Probleme bewältigt werden. Während Augmented Reality neue Möglichkeiten bereitstellt, Informationen in der Lehre zu vermitteln, besteht die Sorge, dass diese den Nutzer kognitiv oder emotional überlasten. Verwendet ein Nutzer Augmented Reality zum ersten Mal, muss dieser zunächst lernen, wie das Gerät und die Anwendung verwendet werden können. Außerdem muss der Nutzer sich mit dem *UI*, Gesten und den Interaktionsmöglichkeiten vertraut machen. Dies kann problematisch sein, wenn gleichzeitig komplexe Aufgaben durchgeführt werden müssen. Ein weiteres Problem ist die Überforderung durch Gameplay-Elemente, da die Aufmerksamkeit des Nutzers unter Umständen nicht auf dem Lerninhalt, sondern auf dem Gameplay liegt. Dies kann vor allem dann vorkommen, wenn der Nutzer mit zu vielen Gameplay-Elementen gleichzeitig konfrontiert wird. Nutzer können sich dabei eventuell zu stark auf die Anwendung fokussieren, wodurch die Produktivität des Lernens verringert wird. Dabei können sich die Nutzer auch selbst in Gefahr begeben, indem diese durch die Augmented Reality Anwendung abgelenkt werden und möglichen Gefahren in ihrer Umwelt weniger Beachtung schenken.

Für die Entwickler einer Augmented Reality Anwendung bestehen zusätzliche Herausforderungen. Die Anwendung muss in der Lage sein, den Nutzer zu beschäftigen und sich mit den vermittelten Informationen auseinanderzusetzen. Hsin-Kai Wu et al. [41] ging dabei auf die Anforderungen ein, welche eine Augmented Reality Anwendung im Bereich der Lehre einbringen muss, um ein aktives Lernen zu fördern. Das erste Kriterium beschreibt, dass der Nutzer Spaß bei der Nutzung der Anwendung haben muss. Dass die Anwendung ein bestimmtes Ziel in Aussicht stellen sollte, damit der Nutzer sich dauerhaft mit der Anwendung auseinandersetzt, wird vom zweiten Kriterium definiert. Das dritte Kriterium fordert, dass die Neugier angeregt wird, damit das Thema, welches vermittelt werden soll, weiter vom Nutzer erforscht wird [38].

Augmented Reality kann als ein Werkzeug für ein spielerisches Lernen eingesetzt werden, da dies als Motivationsfaktor genutzt werden kann. Die Nutzung von Augmented Reality in der Lehre bietet dem Nutzer völlige Selbstständigkeit, da dieser die zu lösenden Aufgaben in der eigenen Geschwindigkeit bearbeiten und die interessanten Bereiche im Detail erlernen kann. Daraus ergibt sich, dass Augmented Reality den Nutzer über das übliche Maß hinaus, welche traditionelle Lernmethoden bieten, motivieren kann.

In einer Studie von Di Sario et al. wurde diese Theorie in die Praxis umgesetzt, um zu prüfen, ob Augmented Reality Schüler der Mittelstufe motivieren kann. Schülern, welche über italienische Kunst der Renaissance lernen sollten, wurden Gemälde gezeigt, welche durch eine Augmented Reality Anwendung um zusätzliche Informationen, darunter Text, Audio, Videos und 3D-Objekte, erweitert wurden. Die Schüler konnten dann selbstständig die zur Verfügung gestellte Anwendung erforschen. Da bei der Studie keine Kontroll- und Fokusgruppen eingesetzt wurden, sind die gesammelten Ergebnisse nicht vollständig zuverlässig. Insgesamt wurden 69 Studenten befragt, wobei 14 Fragebögen als ungültig bewertet wurden, da diese fehlende Informationen oder Abweichungen enthielten. Die ersten Tests fokussierten sich dabei auf die Aufmerksamkeit, die Relevanz der Informationen, das Vertrauen und die Zufriedenheit der Schüler, welche durch eine Augmented Reality Anwendung Informationen vermittelt bekamen. Für jede der Kategorien wurde dabei eine Punktzahl von eins bis fünf vergeben, wobei fünf die bestmögliche Punktzahl darstellt. In der ersten Testrunde erhielten alle der genannten Kategorien einen niedrigeren Wert als 3,5. Diese Werte wurde jedoch in einer zweiten Testrunde zu einem Wert von etwas über 3,5 erhöht. Die Punktzahl der Ergebnisse hat sich um einen Mittelwert von 0,17 zwischen den Tests erhöht. Dieser Wert deutet darauf hin, dass die Nutzung von Augmented Reality in diesem Fall das Interesse der Schüler nur mäßig erhöht hat [38].

Am Beispiel der Studien von Billingham et al. und Di Sario et al. lässt sich erkennen, dass Augmented Reality nicht in allen Bereichen des Lernens sinnvoll eingesetzt werden kann. Faktoren, wie Thema des zu vermittelten Lerninhalts, Zielgruppe der Augmented Reality Anwendung oder Umsetzung der virtuellen Technologie spielen dabei eine wichtige Rolle beim Erfolg des Lernens.

### 5.1.2 Zukunft im Lernbereich

Die Entwicklung von Virtual- und Augmented Reality wird in Zukunft auch von der Unterhaltungs-Branche vorangetrieben. Deshalb werden bereits effizientere Entwicklungs- und Verarbeitungsmethoden zur Erstellung von Augmented Reality entwickelt. Neue Hardware, welche die Interaktion mit virtuellen Objekten erlaubt und dabei die potenzielle Nutzung der menschlichen Sinne verstärkt, befindet sich bereits in der Entwicklung. Sinne, wie beispielsweise der Geschmacks- und Geruchssinn, können außerdem in Betracht gezogen und auf eine sinnvolle Nutzung überprüft werden. In Bezug auf den Bildungssektor sollte sich auf unterschiedliche Bereiche fokussiert werden, um Augmented Reality optimal nutzen zu können. Einer dieser Bereiche ist der Präsenz-Unterricht. Dieser Bereich konzentriert sich auf die Entwicklung von Simulatoren, welche die Möglichkeit bieten, Situationen aus der realen Welt in einer spezifischen Umgebung nachzubilden. Hierbei sollten den Schülern möglichst realistische Bedingungen geboten werden. Die Zukunft liegt in der Erarbeitung von Entwicklungs-Plattformen, welche erlauben, diese Umgebungen und Bedingungen schnell zu erschaffen und zu konfigurieren. Wird eine neue Umgebung benötigt, sollte diese nicht von Grund auf neu entwickelt werden müssen. Dadurch können Umgebungen aktualisiert und erneuert werden, was in niedrigeren Kosten für die Bildungseinrichtungen und einer besseren Lernerfahrung für die Schüler resultiert. Ein anderer Bereich ist die Kollaboration, beispielsweise dem Austausch von virtuellen Bildern, Informationen und Objekten im Präsenz-Unterricht. Maßgeschneiderte Modelle für Bereiche wie Gesundheit, Mechanik oder Architektur können erstellt und die nötigen Informationen den Schülern vermittelt werden. Schüler können mit diesen Modellen in Echtzeit interagieren, wobei Schüler im selben Klassenzimmer dabei gemeinsam die Modelle betrachten und manipulieren können. Ein Anwendungsbeispiel dafür ist ein Modell des menschlichen Körpers, bei dem eine theoretische Erklärung eingeblendet wird und einzelne Elemente, wie zum Beispiel Organe, verschoben und betrachtet werden können.

Die Mobilität des Nutzers innerhalb der Umgebung ist ein weiterer wichtiger Punkt bei der Förderung von Augmented Reality. Eine erhöhte Mobilität lässt sich durch die Entfernung von Kabeln erreichen. Die Entwicklung von Systemen, welche es dem Nutzer ermöglichen zu rennen, zu springen und sich frei in der Umgebung zu bewegen, ohne dabei die kontrollierte physische Umgebung zu verlassen, sollte vorangetrieben werden. Die Entwicklung von Peripheriegeräten, welche besser an die Biomechanik eines Menschen angepasst sind, um die Interaktion mit virtuellen Objekten zu verbessern, wird ein weiterer wichtiger Schritt für die Zukunft von Augmented Reality Anwendungen sein [42].

## 5.2 Anwendungsbereiche

Augmented Reality erweitert und verbessert die Wahrnehmung und Interaktion mit der realen Welt. Die Informationen, welche durch Augmented Reality vermittelt werden, helfen dem Nutzer dabei, Aufgaben und Probleme in der realen Welt zu lösen. Die Fähigkeiten dieser Techno-

logie können neben der Lehre erfolgreich in verschiedenen anderen Bereichen, wie beispielsweise in der Medizin, in der Industrie, im Militär, im Marketing, im Tourismus oder in der Visualisierung eingesetzt werden. Einige dieser Anwendungsbereiche von Augmented Reality werden in den folgenden Abschnitten näher beschrieben und Fallbeispiele genannt [43].

### **5.2.1 Medizin und Pflege**

Augmented Reality Anwendungen sind in der Lage, im Medizin- und Pflegebereich eingesetzt zu werden, um Patienten eine erweiterte Behandlung bereitzustellen. Ein Beispiel dafür ist das „Autism Glass Project“ der medizinischen Fakultät der Stanford Universität. In einem Projekt wurde „Google Glass AR“, eingesetzt, um Kindern mit Autismus bei der Deutung der Emotionen anderer Menschen zu helfen. Das Ziel des Projekts ist, dass die Patienten Erfahrungen sammeln und in Zukunft auch ohne Hilfe die Emotionen anderer interpretieren und soziale Kontakte knüpfen können [40]. Eine andere medizinische Anwendung von Augmented Reality besteht darin, dass medizinische Daten und der Patient im gleichen physischen Raum visualisiert werden können. Dies erfordert eine Visualisierung der Daten in Echtzeit, welche exakt über den Körper des Patienten gelegt werden müssen. Eine weitere Anwendung von Augmented Reality in der Medizin ist die Visualisierung von Ultraschallbildern. Durch die Nutzung einer Augmented Reality Anwendung kann beispielsweise ein volumetrisches Bild des Fötus über den Bauch einer schwangeren Frau gelegt werden, wodurch auch bei unterschiedlichen Blickwinkeln der Eindruck entsteht, dass sich dieser virtuelle Fötus im Bauch der Frau befindet. Eine kooperative chirurgische Anwendung, welche durch Hand-Gesten gesteuert und durch Informationen aus dem Augmented Reality System unterstützt werden, ist ebenfalls eine sinnvolle Anwendungsmöglichkeit. Dabei können die verschiedenen Vorteile der Technologie und die Erfahrung des Chirurgen vereint werden [43].

### **5.2.2 Ingenieurwesen und Industrie**

Forschung im Bereich Augmented Reality für die Produktion von Gütern ist ein großes und wachsendes Gebiet. Die Herausforderung in diesem Bereich ist das Design und die Implementierung von Augmented Reality Systemen, welche sich nahtlos in den Produktionsprozess integrieren und die Entwicklung von neuen Produkten verbessern. Dadurch soll eine kürzere Vorlaufzeit, reduzierte Kosten und eine erhöhte Qualität erreicht werden. Augmented Reality kann die Wahrnehmung des Nutzers erweitern und bei einer komplexen Montage eines Produktes helfen. Durch zusätzliche Informationen, graphische Darstellungen und Animationen kann der Produktionsprozess unterstützt werden. Diese unterstützenden Elemente können nach Wunsch eingeblendet und virtuell auf die realen Produkte gelegt werden. Die Darstellung der Anweisungen und Animationen hängen vom Zustand der Fertigung ab und können automatisch an die aktuellen Umstände des Produkts angepasst werden. Anweisungen und graphische Hilfestellungen können in regelmäßigen Abständen mit den neusten Erkenntnissen aktualisiert werden. Durch den Einsatz von Augmented Reality können Präsenz-Schulungen reduziert und eine Überladung von Informationen beim Personal verhindert werden [43]. In einer Studie von Tang et al.

[44] wurde zur Überprüfung der Effektivität ein Vergleich von drei unterschiedlichen Medien zur Schulung von Mitarbeiter durchgeführt. Diese im Vergleich genutzten Medien waren Handbücher in Papierform, eine monitorbasierte, computergestützte Anweisung und ein am Kopf befestigtes Augmented Reality Gerät. Bei der Auswertung der Fehlerhäufigkeit bei der Montage, wurde festgestellt, dass durch die Überlagerung von Anweisungen auf die Komponenten des Produkts, durch Augmented Reality, die Fehlerquote um 82% reduziert wurde [43].

### 5.2.3 Visualisierung von Geschichte und Kultur

Die Herausforderungen, immersives Lernen durch Augmented Reality an den Geschichtsunterricht anzupassen, wurde in einer Studie von Blanco-Fernández et al. untersucht. Das Ziel der Studie war, eine immersive Lernerfahrung im Geschichtsunterricht zu erstellen. Diese erlaubt den Schülern, an einem historischen Event teilzunehmen, wobei, basierend auf historischem Kontext, Entscheidungen getroffen werden können. Die Anwendung zeigt dabei die Verzweigungen und die Konsequenzen der Entscheidungen auf. Dadurch bekommen die Schüler vermittelt, dass die Welt von einer Reihe aus kleinen und großen Ereignissen historischer Figuren und Umständen geprägt wird. Die Ergebnisse der Studie sollen, anstatt die Lernfähigkeit oder die emotionale Reaktion der Probanden, die Qualität der Erfahrung selbst bewerten. Durch den Einsatz von Fragebögen wurde festgestellt, dass die Erfahrung eine hohe Lernerfahrung für Kinder und Erwachsene darstellt, aber weniger für Jugendliche geeignet ist.

Der Einsatz von Augmented Reality im Geschichtsunterricht bietet außerdem den Vorteil, dass der Nutzer in die Umgebung eintauchen kann. Die physische Präsenz vor Ort vermittelt dem Nutzer ein Gefühl von historischem Einfühlungsvermögen, welches nicht im Klassenzimmer mit einem Textbuch vermittelt werden kann. In einer Studie zum Einsatz von Augmented Reality in Kulturerbestätten hat Chang et al. dieses Gefühl als Sense of Place beschrieben. Der Sense of Place wird von den Autoren als eine Kombination von Verbundenheit, Abhängigkeit, Sorge, Identität und Zugehörigkeit beschrieben, welche Menschen zu einem Ort entwickeln können. Die Studie experimentiert mit einer von Augmented Reality geführten Tour, welche die Nutzer zu verschiedenen historischen Orten führt. Die Teilnehmer waren Universitätsstudenten, denen unterschiedliche Versionen der Anwendung bereitgestellt wurden. Eine Gruppe erhielt die vollständige Augmented Reality Anwendung, welche Gebäude erkennen und Informationen bereitstellen kann. Die zweite Gruppe erhielt eine Tour, welche durch Audioguides geführt wurde. Die Kontrollgruppe erhielt keine zusätzlichen Informationen, beziehungsweise Medien, zur Tour. Anschließend wurden Fragebögen verteilt, um das Wissen und das emotionale Verständnis bezüglich der Tour abzufragen. Dabei erhielt die Gruppe, welche eine Augmented Reality geführte Tour absolvierte, die höchste Punktzahl, während die von Audioguides geführte Gruppe eine weniger hohe Punktzahl erzielte. Ein ähnliches Ergebnis wurde festgestellt, als die Kriterien bezüglich dem Sense of Place getestet wurden. Die Gruppe, welche die Augmented Reality Anwendung für die Tour verwendete, verzeichnete eine höhere emotionale Bindung zu den besuchten Orten. Die Ergebnisse dieser Studie sind von Relevanz, da die Gruppe, welche lediglich von Audio geführt wurde, schlechtere Ergebnisse als die Gruppe mit der Augmented Reality Anwendung

erhielt. Dies deutet darauf hin, dass eine audiobasierte Führung nicht ausreicht, um das Interesse der Studenten zu wecken und diese erfolgreich in die Umgebung und die vermittelten Inhalte eintauchen zu lassen.

In einer anderen Studie wurde der Lerneffekt von ortsbasiertem Augmented Reality untersucht. Die Studie verwendete eine Smartphone-basierte Augmented Reality Anwendung, welche die Unterschiede einer Sehenswürdigkeit, den Roddick Gates in Montreal, vom 19. Jahrhundert im Vergleich zum heutigen Zustand, visualisiert. Die Teilnehmer sollten bei der Studie die Unterschiede herausfinden. Dabei wurden die Bereiche, welche von den Teilnehmern angeschaut wurden, durch *Eye-Tracking* aufgezeichnet. Von den Teilnehmern wurden die meisten Unterschiede zwischen dem alten und neuen Zustand erfolgreich wahrgenommen. Bei der Studie wurde auch die emotionale Reaktion der Teilnehmer untersucht. Bei der Auswertung der Ergebnisse stellte sich heraus, dass ein Großteil der Teilnehmer Spaß an der Aufgabe hatte. Lediglich eine Minderheit zeigte kein Interesse am Ort, dem Thema oder dem Tour-Guide. Insgesamt zeigen die Ergebnisse der Studie, dass die durch Augmented Reality geführten Teilnehmer einer Führung ein tieferes Verständnis für die historische Sehenswürdigkeit entwickelten [38].

## 6 Alternativen zu Augmented Reality HMDs

Wie in Kapitel 5 beschrieben, besitzt der Einsatz von am Kopf befestigten Augmented Reality Geräten, unter anderem in der Lehre, unterschiedliche Vorteile. Durch den hohen Preis und die vergleichsweise geringe Verfügbarkeit von Augmented Reality Geräten, wie zum Beispiel der HoloLens 2, wird ein verbreiteter Einsatz im Unterricht erschwert. In diesem Abschnitt soll daher auf die Alternativen zu Augmented Reality Headsets, wie beispielsweise auf Virtual Reality Headsets und Handheld Augmented Reality, eingegangen werden.

### 6.1 Virtual Reality

Virtual Reality ermöglicht eine interaktive Computersimulation, welche sensorische Informationen an den Nutzer vermittelt. Diese Informationen ersetzen oder erweitern das reale Umfeld des Nutzers. Daher kann Virtual Reality als eine durch einen Computer generierte, künstliche Umgebung definiert werden, welche eine reale oder fiktive Situation simuliert. Virtual Reality Systeme können unterschiedliche Wahrnehmungssinne ansprechen. Diese Systeme bieten dem Nutzer die Möglichkeit, visuell in eine spezifische virtuelle Umgebung einzutauchen. Peripheriegeräte, wie beispielsweise spezielle Controller, ermöglichen dem Nutzer den Tastsinn zu verwenden, wohingegen durch die Verwendung von Lautsprechern, beziehungsweise Kopfhörern, der Hörsinn miteinbezogen wird. Während einige Virtual Reality Anwendungen eine hohe Rechenleistung und speziell entwickelte Hardware benötigen [42], können andere Anwendungen dagegen mit Hilfe eines Smartphones und einer Halterung für dieses verwendet werden. Beispiele dafür sind das Google Cardboard [45] oder das Samsung Gear VR [46]. Die Spiele- und Unterhaltungsindustrie ist eine der treibenden Kräfte bei der Entwicklung von Virtual Reality. Dieser Bereich besitzt eine hohe Anzahl an Nutzern, welche dazu bereit sind, Geld für ein verbessertes Spielerlebnis zu investieren. Andere Bereiche, wie z.B. Kommunikation, Marketing und wissenschaftliche Anwendungen haben das Potential von Virtual Reality erkannt und haben damit begonnen, diese Technologien in ihrem Umfeld für ihren Vorteil einzusetzen [42].

#### 6.1.1 Anwendungsbereiche

Virtual Reality Geräte haben in der Lehre einen neuen Bereich erschlossen, in welchem die Technologie ihr volles Potential entwickeln kann. Die Lehre beinhaltet unterschiedliche Elemente, welche durch den Einsatz von Virtual Reality eine positive Auswirkung erlangen können. Die Lernmethoden mit der größten Wirkung konfrontieren die Schüler mit einer realen Situation, welche diese dann mit theoretischem Wissen lösen müssen oder die Fähigkeiten der Schüler zur Problemlösung verbessern. Bisher wurden Situationen oder Aufgabenstellungen durch Text, in seltenen Fällen durch Audio oder Video, beschrieben. Durch die Verwendung von Virtual Reality können diese Situationen mit mehreren Einflussfaktoren neu erstellt oder verändert werden, damit die Schüler auf die unterschiedlichen Situationen reagieren können. Anwendungen können an jedes Fach, Wissensgebiet oder auf unterschiedliche Zielgruppen angepasst werden. Schüler, welche Schwierigkeiten haben Lernziele zu erreichen und Probleme bei der Lösung von

Aufgaben haben, können durch den Einsatz von Virtual Reality dabei unterstützt werden, die Lernziele erfolgreich zu bewältigen. Die Technologie trägt dazu bei, dass unterschiedliche abstrakte Konzepte verständlich gemacht werden können. Virtual Reality kann die Erfolgsquote bei Schülern erhöhen, welche ohne diese Technologien nicht die Fähigkeit besitzen, komplexe Probleme zu lösen.

Neben der Lehre kann Virtual Reality auch in der Industrie oder im medizinischen Bereich eingesetzt werden, wobei diese bei der Visualisierung von Objekten oder Konzepten einen Beitrag leisten kann. Ein Beispiel dafür ist die Darstellung von Architektur, bei welcher die unterschiedlichen Schichten und Elemente eines Gebäudes, wie Verkabelung, Leitungen und unterschiedliche Baumaterialien, mit Hilfe von Virtual Reality visualisiert werden können [42].

Eine Studie zum Einsatz virtueller Anwendungen bei Krankenhauspatienten ergab, dass die Nutzung von Virtual Reality zu einer hohen Patientenzufriedenheit führt und weiterhin auch sicher in deren Anwendung ist. Auch klinische Studien für naturbasierte, immersive Virtual Reality Spiele liefern vielversprechende Ergebnisse. Eine Virtual Reality Anwendung, welche die lokale Natur virtuell im Krankenzimmer abbildet, fördert ein Gefühl der Entspannung bei Krebspatienten, welche durch eine Chemotherapie behandelt werden. Weiterhin konnte bei Menschen mit Angststörung diese mit Hilfe eines Virtual Reality Geräts erheblich reduziert werden. In anderen Fällen konnten naturbasierte Virtual Reality Anwendungen die Patienten von Schmerzen, Angst oder Stress für eine kurze Zeit ablenken [47].

### **6.1.2 Vor- und Nachteile**

Der Einsatz von Virtual Reality bringt unter anderem im Bereich der Lehre einige Vorteile mit sich. Beispielsweise erlaubt eine speziell angefertigte virtuelle Umgebung den Schülern in Lehrerfahrten einzutauchen, bei denen mehrere Sinne genutzt werden können. Dies kann dem Schüler beim Verständnis unbekannter Konzepte und Themen helfen. Die räumliche Darstellung komplexer mathematischer Funktionen oder die Visualisierung des menschlichen Körpers sind dabei Beispiele für einen erfolgreichen Einsatz von Virtual Reality in der Lehre [42]. Der Einsatz von Virtual Reality im Unterricht wirkt motivierend für die Schüler, wodurch diese eine positive Einstellung gegenüber der Technologie in ihrem Lernprozess entwickeln. Virtual Reality fördert das Engagement der Schüler, da die Interaktion mit einer Anwendung aufregend und herausfordernd ist. Virtual Reality erlaubt, Objekte und Prozesse zu visualisieren, welche ansonsten nur unter großem Aufwand in einem realen Umfeld dargestellt werden können. Abstrakte Konzepte können mit interaktiven Modellen verständlich gemacht werden, wodurch die Vermittlung von Wissen durch einen konstruktiven Ansatz erleichtert wird. Dieser Ansatz ermöglicht, eine vollständige, auf die Schüler fokussierte Lernerfahrung zu schaffen, da die Schüler aktiv bei der Ausführung von Experimenten eingebunden sind und mit virtuellen Objekten trainieren, interagieren und lernen können [39]. Im Vergleich zu Augmented Reality Geräten wie der HoloLens 2 und der Magic Leap 1 ist der niedrigere Preis und die höhere Verfügbarkeit ein Vorteil von Virtual Reality Geräten. Der Preis einer Magic Leap 1 wird vom Hersteller auf 2995

Dollar angegeben [48], während Microsoft die HoloLens 2 für 3500 Dollar anbietet [49]. Virtual Reality Geräte und notwendiges Zubehör, wie zum Beispiel das Vive Cosmos Elite Headset, werden vom Hersteller für 899 Dollar angeboten [50]. Bezogen auf einige technische Spezifikationen sind bestimmte Virtual Reality Geräte den hier genannten Augmented Reality Geräten überlegen. Beispielsweise können virtuelle Objekte durch eine Auflösung von 2880 x 1700 Pixeln bei der Vive Cosmos Elite schärfer dargestellt werden als bei der HoloLens 2, welche eine Auflösung von 2048 x 1080 Pixeln besitzt. Das Vive Cosmos Elite Headset wird in Abbildung 28 dargestellt. Auch das *FOV* des Virtual Reality Headsets ist im Vergleich zu dem *FOV* der HoloLens 2 deutlich erhöht. Während die HoloLens 2 ein *FOV* von 52 Grad bietet, kann der Nutzer der Vive Cosmos Elite seine Umgebung mit einem *FOV* von bis zu 110 Grad betrachten [51].



**Abbildung 28:** Das Virtual Reality Headset Vive Cosmos Elite [50].

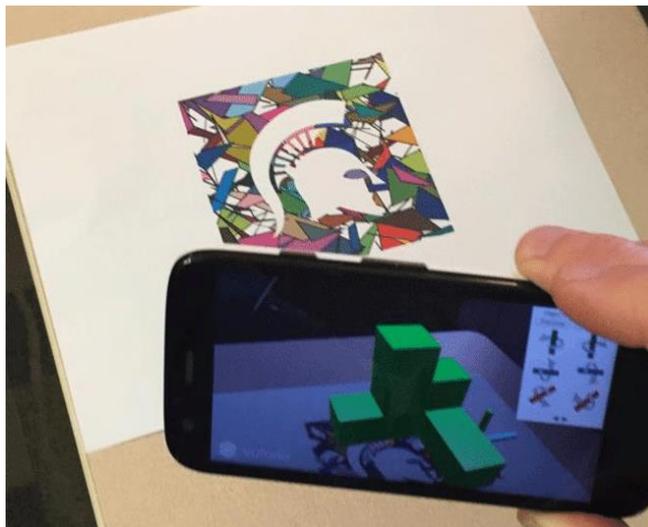
Andererseits kann der Einsatz von Virtual Reality in der Lehre Probleme und Risiken verursachen. Eine exzessive Nutzung kann zu einer Isolation von Schülern führen. Obwohl kollaborative Virtual Reality Anwendungen zur Verfügung stehen, sind diese im Moment weniger verbreitet als Anwendungen, bei welchen die Schüler isoliert von Mitschülern unterrichtet werden. Persönliche Interaktion ist beim Lernen jedoch wichtig, da diese zum Lernen durch Imitation, zum Lernen durch Zusammenarbeit und zum Lernen durch den Austausch von Erfahrungen führt. Ein anderes Risiko beim Einsatz von Virtual Reality in der Lehre ist die Annahme, dass die Technologie als eine endgültige Lösung in der Lehre, anstatt als ein Mittel zum Zweck, angesehen wird [42]. Ein anderes Problem beim Einsatz von Virtual Reality Geräten ist das Auftreten von *motion sickness* bei einigen Nutzern. Bei unterschiedlichen Anwendungen und bei bestimmten Nutzern besteht die Möglichkeit, dass diese unter unangenehmen Symptomen leiden, welche den Symptomen

von *motion sickness* ähneln. Dieses Phänomen wird auch als *cybersickness* oder *VR sickness* bezeichnet. Symptome der Krankheit sind die Ermüdung der Augen, Orientierungslosigkeit und Übelkeit. Durch diese unangenehmen Gefühle kann eine zukünftige Virtual Reality Erfahrung beeinträchtigt werden [52].

## 6.2 Handheld Augmented Reality

Als Handheld Augmented Reality Geräte können Smartphones oder Tablets verwendet werden. Viele dieser Geräte besitzen einen leistungsstarken Prozessor, einen großen Bildschirm und eine hochauflösende Kamera. Durch diese Funktionen verfügen die Geräte über die Fähigkeit, den Nutzern eine Augmented Reality Erfahrungen zu bieten [53]. Die Fortschritte von Prozessor- und Speichertechnologien erlauben die Rendering-Leistung der Mobilgeräte drastisch zu erhöhen. Dadurch können virtuelle Objekte mit einer hohen Framerate bei gleichzeitig hoher Auflösung dargestellt werden. Wie bereits erwähnt ist das beliebte Augmented Reality Spiel „Pokémon Go“ ein Beispiel dafür [38]. Die Beliebtheit mobiler Augmented Reality Anwendungen spiegelt sich in einer Vielzahl von Plattformen und Entwicklungswerkzeugen wider, auf welchen diese entwickelt werden können. Laine [54] zählt die Entwicklungswerkzeuge in einer Tabelle mit den zugehörigen *Software Development Kits (SDKs)*, der Zielplattform und der Tracking-Methode auf. Dabei werden Entwicklungswerkzeuge wie ARKit, ARToolkit und Vuforia SDK genannt. Die Vielzahl an Programmiersprachen, mit welchen die Anwendungen erstellt werden können, bietet den Entwicklern eine große Freiheit bei der Wahl eines Entwicklungswerkzeugs.

Eine gute Integration in existierende Spiele-Engines ist ein wichtiger Faktor für die Entwickler einer Augmented Reality Anwendung. Die Unity und Unreal Engine besitzen eine fortgeschrittene Integration für Handheld Augmented Reality Anwendungen. Durch diese Integration können Spieleentwickler ohne großen Aufwand eine Anwendung für den Augmented Reality Bereich entwickeln. Zur Verankerung virtueller Objekte in der realen Welt werden unterschiedliche Tracking-Methoden verwendet. Um die Hologramme stabil in der Welt einzubetten, kommen unterschiedliche Marker basierte Ansätze zum Einsatz. Dazu zählen Referenzmarker und Bildmarker. Abbildung 29 zeigt die Darstellung eines virtuellen Objekts, welches durch einen Papiermarker in der realen Welt verankert wird. Andere Augmented Reality Tracking-Methoden erkennen reale Objekte, das Gesicht einer Person oder stützen sich auf unterschiedliche Sensordaten des Mobilgeräts [54].



**Abbildung 29:** Ein Marker, welcher ermöglicht, ein virtuelles Objekt durch Handheld Augmented Reality in der realen Welt zu verankern und mit einem Smartphone zu betrachten [55].

### 6.2.1 Anwendungsbereiche

Die weite Verbreitung von Mobilgeräten, welche zur Darstellung von Augmented Reality fähig sind, kann in unterschiedlichen Bereichen Verwendung finden. Bernarduzzi et al. [56] erforschte beispielsweise den Einsatz unterschiedlicher Technologien wie Desktop-, Virtual Reality- und Handheld Augmented Reality Anwendungen in einem Museum. Um die Funktionalität dieser Technologien zu überprüfen, wurden Besuchern unterschiedliche Anwendungen zur Verfügung gestellt. Die Anwendungen hatten das Ziel, die Besucher über Ausstellungsstücke, wissenschaftliche Innovationen und bekannte Forscher zu informieren. Die Studie kommt dabei zu dem Schluss, dass sich Handheld Augmented Reality durch eine Vielzahl von Vorteilen am besten für einen Einsatz im Museum eignet. Dabei werden Vorteile, wie zum Beispiel das Hinzufügen von immersiven Informationen an reale Ausstellungsstücke und eine erhöhte Interaktion mit diesen von den Autoren genannt.

Der Einsatz von Handheld Augmented Reality in der Lehre wurde unter anderem von Reyes-Aviles et al. [57] erforscht. Dabei wurde eine Anwendung erstellt, um Studenten der Elektrotechnik über verschiedene Konzepte in diesem Fachbereich zu unterrichten. Um Informationen über einen sinnvollen Einsatz dieser Technologie in diesem Zusammenhang zu erhalten, wurde eine Evaluation durchgeführt. Die Studenten gaben bei der Evaluation an, dass der Einsatz von Augmented Reality in der Lehre ansprechend ist. Weiterhin ergab die Auswertung, dass die bereitgestellte Anwendung den Studenten nützliche Funktionen zur Verfügung stellt, welche das Erlernen komplexer Konzepte erleichtert. Die Aufmerksamkeit und der Lernwille der Studenten wurden ebenfalls gesteigert.

In der Industrie kann Handheld Augmented Reality eingesetzt werden, um die Effizienz eines Unternehmens zu verbessern. Große Industrieanlagen werden meist von einem zentralen Kontrollraum gesteuert. Dieser speichert Informationen, überprüft und steuert unterschiedliche

Prozesse. Die Kommunikation des Kontrollraums mit Mitarbeitern erfolgt meist verbal und ist damit limitiert. Durch die Ausstattung von Mitarbeitern mit Augmented Reality fähigen Mobilgeräten kann die Kommunikation verbessert und beschleunigt werden. So können beispielsweise durch die Darstellung virtueller 3D Objekte Teile von Anlagen visualisiert, und in Absprache mit den Mitarbeitern des Kontrollraums, vor Ort analysiert werden [58].

### 6.2.2 Vor- und Nachteile

Die Kompaktheit der Geräte, welche in der Lage sind, Handheld Augmented Reality Anwendungen auszuführen, erlaubt den Nutzern, diese ohne weiteren Aufwand an verschiedene Orte mitzunehmen. Im Vergleich zu am Kopf befestigte Augmented Reality Geräte, wie der HoloLens, sind Handheld Augmented Reality Geräte meist kostengünstig. Viele Menschen besitzen bereits ein Smartphone oder Tablet, wodurch die Verbreitung und Nutzung von Handheld Augmented Reality Anwendungen vereinfacht wird. Ein weiterer Vorteil der Mobilgeräte ist, dass in diesen standardmäßig Sensoren, zum Beispiel ein Gyroskop, ein Beschleunigungsmesser, eine Kamera und GPS-Funktionalität, verbaut sind. Diese Sensoren ermöglichen die Nutzung von Augmented Reality Anwendungen. Des Weiteren können diese Anwendungen vom Nutzer während des Gehens verwendet werden, da die Geräte nicht kabelgebunden sind [59].

Nachteile von Handheld Augmented Reality, im Gegensatz zu tragbaren *HMDs*, sind eine niedrigere Auflösung, eine kleinere Bildschirmgröße und eine höhere Latenz beim Rendering virtueller Objekte [59]. Trotz hochauflösender Kameras können Mobilgeräte nur einen kleinen Teil des Sichtfelds des Nutzers darstellen. Daher können wiederum virtuelle Objekte ebenfalls lediglich einen kleinen Bereich des Sichtfelds einnehmen [60]. Bei Handheld Augmented Reality müssen außerdem ergonomische Aspekte in Betracht gezogen werden. So müssen die Mobilgeräte vom Nutzer dauerhaft mit einer Hand gehalten werden. Eine Hand ist daher damit beschäftigt das Gerät zu halten, während die andere Hand unter Umständen mit der Anwendung interagiert [59]. Die Kombination von Gewicht und Größe eines Geräts, der Griff, mit dem der Nutzer das Smartphone oder das Tablet hält und die Körperhaltung können zu ergonomischen Problemen und Ermüdung führen [53]. Ein anderer Nachteil von Handheld Augmented Reality, beispielsweise an historischen Stätten, ist die Vielfalt an möglichen Markern zum Tracking. Eine Methode zur Standortverfolgung zu verwenden, wird durch unterschiedliche Lichtverhältnisse in Außen- und Innenbereichen erschwert oder sogar unmöglich. Dies führt dazu, dass eine Kombination aus den Tracking-Methoden verwendet werden muss, um eine konsistente Handheld Augmented Reality Erfahrung zu ermöglichen. GPS wird wegen der hohen Verfügbarkeit und der hohen Genauigkeit in einer Mehrheit der Anwendungen, welche für den Außenbereich entwickelt wurden, eingesetzt. Der GPS-Empfang und die Genauigkeit kann sich jedoch in städtischen Umgebungen, in welchen ein GPS Signal eventuell von umliegenden Gebäuden blockiert oder reflektiert wird, drastisch verschlechtern [60].

## 7 Zusammenfassung

In der vorliegenden Masterarbeit wurden zunächst die Leitlinien bezüglich des Komforts, der Interaktionsmöglichkeiten und dem UX-Design aufgelistet. Anschließend wurden die dokumentierten Leitlinien evaluiert, um festzustellen, durch welche Funktionen und Gameplay-Elemente diese in der implementierten Anwendung umgesetzt werden können. Dadurch konnte vor der Entwicklung ein Game Design Dokument angefertigt werden. Neben der Auflistung der erwähnten Leitlinien wurden für das Game Design Dokument typische Kategorien ausgefüllt. In Abschnitten ist darin das Gameplay, das Spielerlebnis, der Rahmen, der exakte Spielablauf, Aufgaben und Belohnungen, Mechaniken, die Steuerung, der Stil und das Level Design dokumentiert. Durch Nachforschungen zu historischen Burgen wurde eine Liste aus Gebäuden zusammengestellt, welche vom Nutzer der Anwendung platziert werden können. Die Beschreibungen der Gebäude, welche der Spieler als Info-Text angezeigt bekommt, wurden recherchiert und im Game Design Dokument notiert. Außerdem wurde für jeden KI-Charakter der Gebäude ein einfacher Ablauf entwickelt. Das Game Design Dokument wurde angelegt, um die zu implementierende Anwendung in einer strukturierten Art und Weise mit festgelegten Zielen umsetzen zu können.

Als Spiele-Engine wurde die Unreal Engine gewählt, da diese, wie in Abschnitt 2.1 ausgeführt, unterschiedliche Vorteile bietet. Zu Beginn des Projekts wurde ebenfalls die Spiele-Engine Unity in Betracht gezogen, da diese eine längere Unterstützung der HoloLens 2 und eine weiterentwickelte Version des *MRTK* Plugins besitzt. Da die HoloLens 2 dennoch ausreichend von der Unreal Engine unterstützt wird und das *MRTK* Plugin alle essenziellen Funktionen anbietet, wurde die Unreal Engine für die Entwicklung der Anwendung genutzt. Für die Erstellung von 3D-Modellen wurden die Open-Source Software Blender und für die Texturierung das Programm Substance Painter eingesetzt.

Um ein Gefühl für den Umgang mit Augmented Reality zu entwickeln, wurden bereits vorgefertigte Anwendungen auf der HoloLens 2 installiert und getestet. Zu diesen Anwendungen gehörten unter anderem das von Epic Games entwickelte „MissionAR“, welches mit fotorealistischen Texturen und Modellen das Wettrennen zum Mond und die Landung auf dem Mond zeigt und damit einen Lehranspruch verfolgt. Durch die unterschiedlichen Anwendungen konnte die Umsetzung der verschiedenen Leitlinien, wie zum Beispiel die Interaktionsmöglichkeiten, erlernt werden. Anschließend wurde sich mit der Entwicklungsumgebung vertraut gemacht. Dafür wurden die beiden Plugins, welche eine Entwicklung von Augmented Reality Anwendungen durch die Unreal Engine ermöglichen, verglichen und getestet. Zunächst wurde das OpenXR Plugin verwendet, da dieses mehr Funktionen und häufigere Aktualisierungen als das Windows Mixed Reality Plugin bietet. Da Microsoft außerdem die *UXT* und *GT* Plugins bereitstellt, wurde sich mit der Funktionalität dieser beiden Komponenten vertraut gemacht.

Basierend auf zu beachtenden Leitlinien und anderen Informationen, welche im Game Design Dokument festgehalten sind und dem Wissen über die Entwicklungswerkzeuge, konnte anschließend eine Augmented Reality Anwendung entwickelt werden. In dieser Anwendung kann der Nutzer mit unterschiedlichen Burggebäuden und *KI*-Charakteren interagieren und diese betrachten. In der Anwendung wurden zusätzlich einzelne Elemente für die Lehre implementiert, beispielsweise durch das Abbilden historischer Gebäude und der Vermittlung von Informationen zum Burgleben. Da sich während der Entwicklung Unterschiede zur im Game Design Dokument geplanten Implementierung herausstellten, wurde der Spielablauf und die Interaktionsmöglichkeiten angepasst. Zur Platzierung der Gebäude war die Auftrennung in unterschiedliche Modi, zum Beispiel in den Präsentations- und Platzierungs-Modus notwendig, da die Implementierung der unterschiedlichen Funktionalitäten in ein Objekt in einem unübersichtlichen und schwierig erweiterbaren Modul resultieren würde.

Zusammenfassend wurde in dieser Masterarbeit eine interaktive und immersive Augmented Reality Anwendung entwickelt, um das mittelalterliche Burgleben zu simulieren und damit eine Lernerfahrung zu schaffen. Dabei wurden die aktuell zur Verfügung stehenden Methoden, welche für eine optimale Erfahrung bei der Nutzung von Augmented Reality Anwendungen sorgen sollen, analysiert und erfolgreich eingesetzt.

## 8 Diskussion

Bei der Implementierung wurden unterschiedliche Einschränkungen bezüglich der Software und der Umsetzung der Leitlinien herausgearbeitet. Beispielsweise konnte kein *Nav-Mesh* aus der, von der HoloLens 2 erzeugten *spatial Map*, generiert werden. Der Grund dafür ist vermutlich, dass für die *spatial Map* keine Kollision generiert wird. Eine Kollision, beziehungsweise eine Oberfläche, wird jedoch für die Generierung eines *Nav-Meshs* benötigt. Im C++ Source Code des *GT Plugins* und im vom Plugin bereitgestellten Beispielprojekt wurde nach möglichen Lösungen recherchiert, hierbei erschien jedoch keine Option zielführend. Damit die *KI*-Charaktere sich dennoch durch die virtuelle Welt bewegen können, wurde als Lösung eine große Kollisions-Box unter dem Hauptgebäude befestigt, durch welche die Unreal Engine in der Lage ist, ein *Nav-Mesh* zu erzeugen. Diese Lösung kann jedoch zu anderen Problemen führen. Für den Fall, dass der Nutzer ein Gebäude in einer großen Entfernung oder ein Gebäude auf einer unterschiedlichen Ebene zum Bergfried platziert, ist keine *Nav-Mesh* und keine Kollision für die *KI*-Charaktere vorhanden. Diese fallen somit durch Oberflächen hindurch.

Da das OpenXR Plugin im Vergleich zum Windows Mixed Reality Plugin ein korrekt implementiertes Input-System besitzt und in zukünftigen Versionen weiter ausgebaut werden soll, wurde dieses Plugin zunächst gewählt, um damit die Anwendung für die HoloLens 2 in der vorliegenden Masterarbeit zu entwickeln. Dabei fiel jedoch auf, dass mit dem OpenXR Plugin keine *spatial Map* generiert werden konnte, obwohl dies offiziell unterstützt wird. Während der Bearbeitung dieser Arbeit wurde eine neue Version der Unreal Engine und des OpenXR Plugins veröffentlicht. Eine Aktualisierung des Projekts und des Plugins könnte das Problem der fehlenden Generierung der *spatial Map* möglicherweise lösen. Aufgrund eines gegebenen Risikos, dass durch die Verwendung der aktualisierten Versionen jedoch andere Funktionalitäten nicht mehr ordnungsgemäß ausgeführt werden, wurde auf eine Aktualisierung des bestehenden Projektes verzichtet und stattdessen das Windows Mixed Reality Plugin verwendet.

Für die eigene Anwendung wurde die Unreal Engine ausgewählt, da der Unreal Engine Editor über alle essenziellen Funktionen verfügt, welche zur Entwicklung einer Augmented Reality Anwendung benötigt werden. Das für die Unity Spiele-Engine erhältliche *MRTK* Plugin befindet sich jedoch länger in der Entwicklung als das für die Unreal Engine erhältliche Plugin. Dadurch fehlen Komponenten, welche für eine weitere Entwicklung der Anwendung hilfreich sein können. Beispielsweise ist in Unity Anwendungen ein Hand Coach verfügbar, welcher dem Nutzer vorführen kann, welche Gesten oder Interaktionen ausgeführt werden können. Dies kann zu einer Verbesserung der Benutzererfahrung führen. Ähnliche Funktionen müssen bei der Entwicklung mit der Unreal Engine eigenständig implementiert werden.

Die eigenständige Implementierung zeigt, dass durch kostenlos verfügbare Software, wie der Unreal Engine oder Blender, eine Augmented Reality Anwendung erstellt werden kann, welche außerdem die dokumentierten Leitlinien beachtet. Komfort-Empfehlungen können durch das Design und die Art der Anwendung selbst berücksichtigt werden. Die Anwendung kann dennoch

von zusätzlichen Funktionen profitieren, welche die Interaktion mit den virtuellen Objekten erhöht. Beispielsweise könnte die Simulation, beziehungsweise der Ablauf, der KI-Charaktere ausgebaut und damit interessanter gestaltet werden. Zusätzlich könnte ein Ressourcen-System eingeführt werden, bei dem der Nutzer für einen effizienten Ablauf und eine optimale Platzierung der Gebäude sorgen muss. Außerdem könnte die Nutzung von Sound ausgebaut werden, in dem neben Button-Sounds auch Umgebungs-Sounds, welche bei einer Burg vorkommen, abgespielt werden.

Die Entwicklung der Anwendung demonstriert, dass die zur Verfügung stehende Software eine effiziente Entwicklung von Augmented Reality Anwendungen ermöglicht. Durch das Streaming von einem PC zur HoloLens 2 kann durch die Unreal Engine außerdem eine schnelle Iteration und das Testen von entwickelten Komponenten bewerkstelligt werden.

Die Stärken von Virtual- und Augmented Reality, auf die in den Abschnitten 6.1.2 und 6.2.2 eingegangen wurde, legen nahe, dass Augmented Reality HMDs trotz diverser Vorteile nicht in jedem Bereich eingesetzt werden sollten. Beispielsweise können Kostenfaktoren eine große Rolle spielen und damit den Einsatz von Augmented Reality HMDs ausschließen. Bezogen auf die technischen Fähigkeiten der Technologien stellt sich außerdem die Frage, ob die im Rahmen dieser Masterarbeit implementierte Anwendungen nicht durch Virtual Reality oder durch Handheld Augmented Reality umgesetzt werden kann. Die Anwendung lässt sich nach eigener Einschätzung durch das aktuelle Design schwierig auf Handheld Augmented Reality übertragen, da auf einige Interaktionsmöglichkeiten verzichtet werden müsste, welche für das Gameplay benötigt werden und außerdem zu einer erhöhten Immersion führen. Dazu gehört zum Beispiel die Nutzung von direkter Manipulation durch die fehlende Erkennung von Gesten auf Mobilgeräten. Außerdem ist der Spielablauf vermutlich zu komplex, um auf einem Smartphone oder Tablet betrachtet werden zu können. In Virtual Reality hingegen können viele Elemente der Anwendung umgesetzt werden. Die direkte Manipulation könnte beispielsweise durch Controller ersetzt werden. Gebäude werden, statt auf einem realen Objekt, auf einem virtuellen Plateau platziert. Damit sich die Anwendung ausschließlich für Augmented Reality HMDs eignet, sollte diese um andere Funktionen erweitert werden. So kann zum Beispiel der Vorteil von Augmented Reality HMDs genutzt werden, dass Mitmenschen auch bei der Interaktion von virtuellen Objekten weiterhin sichtbar bleiben. Eine kollaborative Anwendung, bei der mehrere Nutzer die Burg aufbauen, verwalten und sich dabei absprechen müssen, wäre somit denkbar. Dadurch, dass Teile der realen Welt sichtbar bleiben, kann zudem die bei Virtual Reality häufiger auftretende *motion sickness* vermieden werden.

Um den Unterschied zu den berücksichtigten und tatsächlich erreichten Zielen, wie zum Beispiel den Design Leitlinien, der Vielfalt von Interaktionsmöglichkeiten und den ermöglichten Komfort, zu überprüfen, sollte eine Evaluation beispielsweise in Form einer Umfrage durchgeführt werden. Die in dieser Masterarbeit erarbeiteten Erfahrungen bei der Implementierung sind in weiten Teilen lediglich auf die Unreal Engine anwendbar. Während die verschiedenen Leitlinien auf andere Spiele-Engines übertragbar sind, trifft dies jedoch nicht auf die spezifische Umsetzung

einiger Funktionen zu. Des Weiteren könnte die im Rahmen dieser Masterarbeit entwickelte Anwendung auch einen praktischen Einsatz in der Lehre finden. Dazu sollte jedoch der Informationsgehalt der Anwendung weiter erhöht werden. Denkbar wären beispielsweise zusätzliche Informationen zu den Gebäuden und die historische Einordnung dieser, wie Bauort und Baujahr. Zusätzlich könnte näher auf die unterschiedlichen Berufe der mittelalterlichen Gesellschaft eingegangen und die Arbeitsabläufe detaillierter simuliert werden.

Um eine fundierte Aussage über die Interaktivität und das Potential der in dieser Arbeit vorgestellten, unterschiedlichen Technologien, wie Augmented Reality, Virtual Reality und Handheld Augmented Reality zu treffen, könnte in Zukunft für jede der Plattformen eine Anwendung erstellt werden, welche denselben Anwendungszweck besitzt. Dadurch könnte evaluiert und verglichen werden, welche der Technologien optimal für den Einsatz in unterschiedlichen Bereichen geeignet ist.

## 9 Abkürzungsverzeichnis

<b>Abkürzung</b>	<b>Bedeutung</b>
FOV	Field of View
FPS	Frames per Second
GT	Graphics Tools
HMD	Head-Mounted Display
HPU	Holographic Processing Unit
HRTF	Head Related Transfer Function
HUD	Heads-Up-Display
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
KI	Künstliche Intelligenz
MRTK	Mixed Reality Toolkit
PBR	Physically Based Rendering
SDK	Software Development Kit
UI	User Interface
UX	User Experience
UXT	User Experience Tools
VCS	Version Control System

## 10 Glossar

Begriff	Erklärung
Affordance	Hilfsobjekte, welche der Nutzer greifen kann, um eine Manipulation der Hologramme durchzuführen.
Air-Tap	Gestensteuerung, durch das Zusammenpressen des Daumens und des Zeigefingers kann eine Aktion ausgelöst werden.
Akkommodation	Akkommodation ist die Nah- und Ferneinstellung des Auges durch eine Veränderung der Linsenkrümmung. Dadurch können Objekte auf der Retina scharf abgebildet werden [61].
Armature	Kette aus virtuellen Knochen, durch welche zum Beispiel das Mesh eines virtuellen Charakters verformt und animiert werden kann.
Base Color Map	Textur, welche die Farbe einer Oberfläche ohne zusätzliche Informationen, wie zum Beispiel Schattenwurf, speichert.
Behavior-Tree	System, welches für die Erstellung von KI genutzt wird. Durch eine hierarchische Baumstruktur können Aufgaben und Entscheidungen eines virtuellen Agenten definiert werden.
Blueprint Visual Scripting	System in der Unreal Engine, welches ein visuelles Programmieren erlaubt.
Bounding-Box	6-seitiges Volumen, welches ein anderes Objekt komplett umschließt.
Chromatische Aberration	Wird am Rand einer Linse Licht mit unterschiedlicher Wellenlänge verschieden stark gebrochen und in die spektralen Bestandteile aufgespalten, tritt <i>chromatische Aberration</i> auf. Dadurch können Farbbräunen am Bildrand entstehen [62].
Clipping-Primitive	Durch Clipping-Primitive kann die Oberfläche eines 3D-Objekts transparent gerendert werden, um einen Blick in das Innere des Objekts zu ermöglichen.
Collider	Kollision, wodurch bei Überlagerung mit anderen <i>Collidern</i> ein Ereignis ausgelöst werden kann.
Color-ID Map	Textur, durch welche bestimmte Bereiche markiert und unterschieden werden können.
Commit	Wird bei der Verwendung eines VCS eingesetzt, um eine Änderung an Dateien zu dokumentieren.

Draw Calls	Anweisungen von der CPU an die GPU, wie viele Objekte gerendert werden sollen. Eine niedrige Anzahl sorgt für eine bessere Performance.
Eye-Gaze	Blickrichtung.
Eye-Tracking	Technologie zur Erfassung der Blickrichtung.
Focal Plane	Distanz, in der Hologramme gerendert werden sollten, um eine optimale Darstellung zu ermöglichen.
Hand-Tracking	Technologie zur Erfassung und Interpretierung von Handbewegungen.
Head-Gaze	Ausrichtung des Kopfes.
Head-Tracking	Technologie zur Erfassung der Ausrichtung des Kopfes.
Holographisches Frame	Nutzer betrachten durch ein rechteckiges Display die virtuelle Umgebung einer Mixed Reality Anwendung. Bei der HoloLens wird dieser rechteckige Bereich das Holographische Frame genannt [63].
Inside-Out Tracking	Kameras beobachten Merkmale der Umgebung, um die Position des Geräts bestimmen zu können [64].
Issue	Problem, beziehungsweise Aufgabe, welche bei einem Software-Projekt entstehen kann.
Issue-Tracker	Software, welche die Auflistung, Statusverfolgung und Priorisierung von beliebig vielen <i>Issues</i> erlaubt.
Konvergenz	Bewegung der Augen, bei denen sich die Fixierlinien dieser einwärts bewegen, wodurch sich beim Blick auf Objekte die Fixierlinien in einem Punkt schneiden [65].
Metallic Map	Textur, welche den Anteil metallischer Eigenschaften auf einer Oberfläche speichert.
Modifier	Funktionalität in Blender, wodurch nicht-destruktive Effekte, zum Beispiel Triangulierung, auf 3D-Objekte angewendet werden kann.
Modifier-Stack	Reihe aus <i>Modifiern</i> , welche aufeinander aufbauen, um einen bestimmten Effekt zu erzeugen.
Motion/Cyber/VR Sickness	Unwohlsein, welches unter anderem durch die Nutzung von Virtual Reality Geräten ausgelöst werden kann.
Multimediale Lerntheorie	Die <i>Multimediale Lerntheorie</i> besagt, dass Menschen durch Bilder und Wörter ein tieferes Verständnis des Wissens erlangen als von

	Wörtern allein. Der Einsatz mehrerer Medien fördert demnach die Lernerfahrung [66].
Multimodal	Die gleichzeitige Nutzung mehrerer Wahrnehmungssinne.
Nav-Mesh	Navigations-Mesh. Wird von <i>KI</i> -Charakteren verwendet, um eine Wegfindung zu berechnen.
Near Clipping Plane	Ebene, ab der eine virtuelle Kamera 3D-Objekte rendert.
Normal Map	Textur, welche Verformungen einer Oberfläche speichert und den Detailgrad von 3D-Objekten beim Rendering erhöhen kann.
Pie Menu	Kreisförmiges Menü, wobei Buttons mit Funktionalität radial angeordnet sind.
Profiling	Ermittlung der Performance einer Software-Anwendung.
Proximity-Shader	Shader, der Effekte basierend auf der Entfernung von zwei Objekten auslöst. Kann zum Beispiel für künstliche Lichtquellen eingesetzt werden.
Roughness Map	Textur, welche die Reflexions-Eigenschaften einer Oberfläche speichert.
Scene-Understanding Mesh	Mesh, welches Umgebungsmerkmale wie zum Beispiel die Oberfläche von Tischen, Wänden oder dem Boden nachbildet.
Skeletal Mesh	3D-Objekt, welches <i>Armature</i> besitzt und daher verformt und animiert werden kann.
Spatial Audio	Raumklang, beziehungsweise räumliches Audio, welches einem Punkt im Raum zugeordnet werden kann.
Spatial Mapping	Abbildung einer realen Umgebung in eine virtuelle Umgebung. Das Ergebnis ist das spatial Map Mesh.
Spatialization	Räumlichkeit, beziehungsweise räumliche Abbildung.
Static Mesh	3D-Objekt, welches spezifisch kein <i>Armature</i> zur Verformung besitzt.
Stereoskopisches Sehen	Räumliches Sehen, wobei durch die beidäugige Betrachtung von Objekten eine Tiefenwahrnehmung erreicht werden kann [67].
Tag-Along	Das „mitziehen“ eines Objekts. Ein <i>Tag-Along</i> Menu bewegt sich im virtuellen Raum analog zum Nutzer in der realen Welt.
Texture Samples	Texturen, welche im Pixelshader verwendet werden.

## 11 Literaturverzeichnis

- [1] Ivan Sutherland (Ed.), A head-mounted three dimensional display, ACM Press, New York, New York, USA, 1968.
- [2] H. Gieselmann, HoloLens: Augmented-Reality-Brille für Windows 10, 2015. <https://www.heise.de/newsticker/meldung/HoloLens-Augmented-Reality-Brille-fuer-Windows-10-2525390.html> (accessed 31 October 2021).
- [3] I.E. Sutherland, A head-mounted three dimensional display, in: A head-mounted three dimensional display, San Francisco, California, ACM Press, New York, New York, USA, 1968, p. 757.
- [4] P. Milgram, H. Colquhoun, A Taxonomy of Real and Virtual World Display Integration, in: Y. Ohta, H. Tamura (Eds.), Mixed Reality, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999, pp. 5–30.
- [5] R.T. Azuma, A Survey of Augmented Reality, Presence: Teleoperators & Virtual Environments 6 (1997) 355–385. <https://doi.org/10.1162/pres.1997.6.4.355>.
- [6] H. Swensen, Potential of Augmented Reality in Science Education - A literature review (2016).
- [7] Unreal Engine, Unreal Engine 4 support for HoloLens 2 released in beta, 2019. <https://www.unrealengine.com/en-US/blog/unreal-engine-4-support-for-hololens-2-released-in-early-access> (accessed 7 November 2021).
- [8] Microsoft - GitHub, Unreal UX Tools Plugin Repository, 2021. <https://github.com/microsoft/MixedReality-UXTools-Unreal> (accessed 25 October 2021).
- [9] Microsoft - GitHub, Unreal Graphics Tools Plugin Repository, 2021. <https://github.com/microsoft/MixedReality-GraphicsTools-Unreal> (accessed 25 October 2021).
- [10] K. Kyburz, Microsoft präsentiert die HoloLens 2 auf dem MWC 2019, 2019. <https://techgarage.blog/microsoft-praesentiert-die-hololens-2-auf-dem-mwc-2019/> (accessed 31 October 2021).
- [11] F. Moreu, Lippitt Chris, D. Maharjan, M. Agüero, X. Yuan, Augmented Reality Enhancing the Inspections of Transportation Infrastructure: Research, Education, and Industry Implementation.
- [12] Scooley, HoloLens 2-Hardware, 2021. <https://docs.microsoft.com/de-de/hololens/hololens2-hardware> (accessed 22 September 2021).
- [13] 4Experience Virtual Reality Studio, HoloLens 2 vs HoloLens 1: what's new? | 4Experience's AR/VR Blog, 2020. <https://4experience.co/hololens-2-vs-hololens-1-whats-new/> (accessed 22 September 2021).

- [14] Hartford Technology Rental | HTR, HoloLens 2 vs. HoloLens 1: What's New?, 2020. <https://hartfordrents.com/blog/hololens-2-vs-hololens-1/> (accessed 22 September 2021).
- [15] Microsoft, HoloLens 2 – Übersicht, Funktionen und Spezifikationen | Microsoft HoloLens, 2021. <https://www.microsoft.com/de-de/hololens/hardware> (accessed 21 September 2021).
- [16] Microsoft, HoloLens (1st gen) hardware, 2021. <https://docs.microsoft.com/en-us/hololens/hololens1-hardware> (accessed 23 September 2021).
- [17] Magic Leap, Magic Leap 1 - Overview and Specifications, 2021. <https://www.magicleap.com/en-us/magic-leap-1> (accessed 14 November 2021).
- [18] Third Eye Gen, Augmented Reality Technology Smart Glasses Developer Company | AR Enterprise, 2021. <https://thirdeyegen.com/x2-smart-glasses> (accessed 7 December 2021).
- [19] Third Eye Gen, X2 MR Smart Glasses Picture, 2021 (accessed 7 December 2021).
- [20] Third Eye Gen, X2 MR Glasses Specification Brochure, 2021. [https://www.dropbox.com/s/5k4uekgvpc6qn9/X2\\_MR\\_Glasses\\_Brochure.pdf?dl=0](https://www.dropbox.com/s/5k4uekgvpc6qn9/X2_MR_Glasses_Brochure.pdf?dl=0) (accessed 7 December 2021).
- [21] Erickjpaull, Comfort - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/comfort> (accessed 20 July 2021).
- [22] Kegodin, Audio in mixed reality - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/spatial-sound> (accessed 5 August 2021).
- [23] Microsoft, Spatial mapping - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/spatial-mapping> (accessed 13 July 2021).
- [24] Casey Meekhof, Direct manipulation with hands - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/direct-manipulation> (accessed 14 July 2021).
- [25] HakOn, Voice input - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/voice-input> (accessed 21 July 2021).
- [26] Casey Meekhof, Head and eye gaze in DirectX - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/develop/native/gaze-in-directx> (accessed 8 July 2021).
- [27] Sostel, Eye tracking - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/eye-tracking> (accessed 21 July 2021).
- [28] Sostel, Eye-gaze-based interaction - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/eye-gaze-interaction> (accessed 21 July 2021).
- [29] Mavitazk, Color, light, and materials - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/color-light-and-materials> (accessed 15 July 2021).

- [30] Shengkait, Scale - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/scale> (accessed 16 July 2021).
- [31] Kegodin, Use spatial sound in mixed-reality applications - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/spatial-sound-design> (accessed 5 August 2021).
- [32] Nbarragan23, Hand menu - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/hand-menu> (accessed 19 July 2021).
- [33] Cre8ivepark, Button - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/button> (accessed 15 July 2021).
- [34] Cre8ivepark, Tooltip - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/tooltip> (accessed 7 July 2021).
- [35] Cre8ivepark, Near menu - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/near-menu> (accessed 8 July 2021).
- [36] Caseymeekhof, Point and commit with hands - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/point-and-commit> (accessed 20 July 2021).
- [37] M. Billinghamurst, A. Clark, G. Lee, A Survey of Augmented Reality, *FNT in Human-Computer Interaction* 8 (2015) 73–272. <https://doi.org/10.1561/1100000049>.
- [38] J. Challenor, M. Ma, A Review of Augmented Reality Applications for History Education and Heritage Visualisation, *MTI* 3 (2019) 39. <https://doi.org/10.3390/mti3020039>.
- [39] J. Martín-Gutiérrez, C.E. Mora, B. Añorbe-Díaz, A. González-Marrero, Virtual Technologies Trends in Education, *EURASIA J MATH SCI T* 13 (2017). <https://doi.org/10.12973/eurasia.2017.00626a>.
- [40] M.C. Hsieh, J.J. Lee, Preliminary Study of VR and AR Applications in Medical and Healthcare Education, *J Nurs Health Stud* 03 (2018). <https://doi.org/10.21767/2574-2825.100030>.
- [41] H.-K. Wu, S.W.-Y. Lee, H.-Y. Chang, J.-C. Liang, Current status, opportunities and challenges of augmented reality in education, *Computers & Education* 62 (2013) 41–49. <https://doi.org/10.1016/j.compedu.2012.10.024>.
- [42] M. Fernandez, Augmented-Virtual Reality: How to improve education systems, *High. Learn. Res. Commun.* 7 (2017) 1. <https://doi.org/10.18870/hlrc.v7i1.373>.
- [43] Mehdi Mekni, Andre Lemieux, *Augmented Reality: Applications, Challenges and Future Trends*, 2014.
- [44] A. Tang, C. Owen, F. Biocca, W. Mou, Comparative effectiveness of augmented reality in object assembly, in: *CHI 2003 conference proceedings*, Ft. Lauderdale, Florida, USA, Association for Computing Machinery, New York, 2003, p. 73.

- [45] Google VR, Google Cardboard, 2021. <https://arvr.google.com/cardboard/> (accessed 4 November 2021).
- [46] The Official Samsung Galaxy Site, Samsung Gear VR with Controller, 2021. <https://www.samsung.com/global/galaxy/gear-vr/> (accessed 8 November 2021).
- [47] N.L. Yeo, M.P. White, I. Alcock, R. Garside, S.G. Dean, A.J. Smalley, B. Gatersleben, What is the best way of delivering virtual nature for improving mood? An experimental comparison of high definition TV, 360° video, and computer generated virtual reality, *Journal of Environmental Psychology* 72 (2020) 101500. <https://doi.org/10.1016/j.jenvp.2020.101500>.
- [48] Magic Leap, Magic Leap 1 - Shop and Pricing, 2021. <https://shop.magicleap.com/#/> (accessed 17 November 2021).
- [49] Microsoft, HoloLens 2—Pricing and Options | Microsoft HoloLens, 2021. <https://www.microsoft.com/en-us/hololens/buy> (accessed 17 November 2021).
- [50] VIVE, Find the best VR headset for you - VIVE Products, 2021. <https://www.vive.com/us/product/> (accessed 18 November 2021).
- [51] VIVE, VIVE Cosmos Elite Headset Specs | VIVE United States, 2021. <https://www.vive.com/us/product/vive-cosmos-elite-headset/specs/> (accessed 18 November 2021).
- [52] E. Chang, H.T. Kim, B. Yoo, Virtual Reality Sickness: A Review of Causes and Measurements, *International Journal of Human–Computer Interaction* 36 (2020) 1658–1682. <https://doi.org/10.1080/10447318.2020.1778351>.
- [53] M.E.C. Santos, T. Taketomi, C. Sandor, J. Polvi, G. Yamamoto, H. Kato, A usability scale for handheld augmented reality, in: *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology - VRST '14*, Edinburgh, Scotland, ACM Press, New York, New York, USA, 2014, pp. 167–176.
- [54] T. Laine, Mobile Educational Augmented Reality Games: A Systematic Literature Review and Two Case Studies, *Computers* 7 (2018) 19. <https://doi.org/10.3390/computers7010019>.
- [55] J. Bell, T. Hinds, S.P. Walton, C. Cugini, C. Cheng, D. Freer, W. Cain, H. Klautke, A Study of Augmented Reality for the Development of Spatial Reasoning Ability, in: *2017 ASEE Annual Conference & Exposition Proceedings*, Columbus, Ohio, ASEE Conferences, 6/24/2017 - 6/28/2017.
- [56] L. Falomo Bernarduzzi, E.M. Bernardi, A. Ferrari, M.C. Garbarino, A. Vai, Augmented Reality Application for Handheld Devices: How to Make It hAPPen at the Pavia University History Museum, *Sci. Educ. (Dordr)* (2021) 1–19. <https://doi.org/10.1007/s11191-021-00197-z>.

- [57] F. Reyes-Aviles, C. Aviles-Cruz, Handheld augmented reality system for resistive electric circuits understanding for undergraduate students, *Comput Appl Eng Educ* 26 (2018) 602–616. <https://doi.org/10.1002/cae.21912>.
- [58] B. Stutzman, D. Nilsen, T. Broderick, J. Neubert, MARTI: Mobile Augmented Reality Tool for Industry, in: 2009 WRI World Congress on Computer Science and Information Engineering, Los Angeles, California USA, IEEE, 3/31/2009 - 4/2/2009, pp. 425–429.
- [59] E. Song, N.M. Suaib, A.J. Sihes, R. Alwee, Z.M. Yunos, Design and development of learning mathematics game for primary school using handheld augmented reality, *IOP Conf. Ser.: Mater. Sci. Eng.* 979 (2020) 12014. <https://doi.org/10.1088/1757-899X/979/1/012014>.
- [60] S. Kurkovsky, R. Koshy, V. Novak, P. Szul, Current issues in handheld augmented reality, in: 2012 International Conference on Communications and Information Technology (ICCIT), Hammamet, Tunisia, IEEE, 6/26/2012 - 6/28/2012, pp. 68–72.
- [61] DocCheck Medical Services GmbH, Akkommodation (Auge) - DocCheck Flexikon, 2021. [https://flexikon.doccheck.com/de/Akkommodation\\_\(Auge\)](https://flexikon.doccheck.com/de/Akkommodation_(Auge)) (accessed 10 December 2021).
- [62] M.M. Volgger, chromatische Aberration, 2021. [https://www.univie.ac.at/mikroskopie/1\\_grundlagen/optik/opt\\_linsen/5c\\_chromatisch.htm](https://www.univie.ac.at/mikroskopie/1_grundlagen/optik/opt_linsen/5c_chromatisch.htm) (accessed 10 December 2021).
- [63] Cre8ivepark, Holographic frame - Mixed Reality, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/design/holographic-frame> (accessed 10 December 2021).
- [64] Hferrone, How inside-out tracking works - Enthusiast Guide, 2021. <https://docs.microsoft.com/en-us/windows/mixed-reality/enthusiast-guide/tracking-system> (accessed 10 December 2021).
- [65] Spektrum, Konvergenz, 2017. <https://www.spektrum.de/lexikon/optik/konvergenz/1650> (accessed 10 December 2021).
- [66] R.E. Mayer, Introduction to Multimedia Learning, in: R. Mayer (Ed.), *The Cambridge Handbook of Multimedia Learning*, Cambridge University Press, Cambridge, 2014, pp. 1–24.
- [67] Spektrum, Stereoskopisches Sehen, 2014. <https://www.spektrum.de/lexikon/geowissenschaften/stereoskopisches-sehen/15675> (accessed 5 December 2021).

## Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die Masterarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die aus fremden Quellen direkt oder indirekt übernommenen Gedanken als solche kenntlich gemacht habe.

Die Arbeit habe ich bisher keinem anderen Prüfungsamt in gleicher oder vergleichbarer Form vorgelegt. Sie wurde bisher nicht veröffentlicht.

---

Ort, Datum

---

Unterschrift

## Ermächtigung

Hiermit ermächtige ich die Hochschule Kempten zur Veröffentlichung einer Kurzzusammenfassung sowie Bilder/Screenshots und ggf. angefertigte Videos meiner studentischen Arbeit z. B. auf gedruckten Medien oder auf einer Internetseite der Hochschule Kempten zwecks Bewerbung des Bachelorstudiengangs „Game Engineering“ und des Masterstudiengangs „Game Engineering und Visual Computing“.

Dies betrifft insbesondere den Webauftritt der Hochschule Kempten inklusive der Webseite des Zentrums für Computerspiele und Simulation. Die Hochschule Kempten erhält das einfache, unentgeltliche Nutzungsrecht im Sinne der §§ 31 Abs. 2, 32 Abs. 3 Satz 3 Urheberrechtsgesetz (UrhG).

---

Ort, Datum

---

Unterschrift