
D I P L O M A R B E I T

Applied Augmented Reality in Education

Ausgeführt im Schuljahr 2034/24 von:

Recherche zu Varianten von Knapsack-Algorithmen und Umsetzung des Knapsack-Problems als AR-Anwendungsszenario inkl. Dokumentation || Erstellen/Auswerten eines Feedbackfragebogens zur Lernunterstützung

Moritz SKREPEK

5CHIF

Design und Umsetzung der 3D-Objekte zur AR-Abbildung || Analyse der Steuerungsmöglichkeiten (Menüführung, Gesten, ...) und Erstellen der Benutzeroberfläche für die AR-Applikation mit Fokus auf UX

Dustin LAMPEL

5CHIF

Erfassen realer Objekte und kontextgerechte Überlagerung der Realität mit AR-Device || Tagging v. realen Elementen mittels QR-Codes für Tracking || Unit-Tests für d. implementierten Knapsack-Algorithmus

Seref HAYLAZ

5CHIF

Evaluierung/Auswahl Laufzeit-/Entwicklungsumgebung für Umsetzung der Applikation und Integration mit AR-Device inkl. Recherche || Konzeption/Umsetzung des Anwendungsszenarios im Bereich Netzwerktechnik

Jonas SCHODITSCH

5CHIF

Betreuer / Betreuerin:

Mag. BEd. Reis Markus

Wiener Neustadt, am 29. März 2024

Abgabevermerk:

Übernommen von:

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Wiener Neustadt, am 29. März 2024

Verfasser / Verfasserinnen:

Moritz SKREPEK

Dustin LAMPEL

Seref HAYLAZ

Jonas SCHODITSCH

Inhaltsverzeichnis

Eidesstattliche Erklärung	i
Vorwort	vi
Diplomarbeit Dokumentation	vii
Diploma Thesis Documentation	ix
Kurzfassung	xi
Abstract	xii
1 Einleitung	1
1.1 Ausgangslage	1
1.2 Auslöser	1
1.3 Aufgabenstellung	1
1.4 Team	2
1.4.1 Aufteilung	2
2 Grundlagen	3
2.1 Vorgehensmodelle	3
2.1.1 Scrum	3
2.1.1.1 Die drei Rollen in Scrum	3
2.1.1.2 Scrum Meetings	4
2.1.2 Begründung der Auswahl	5
2.2 Projektmanagement-Tools	6
2.2.1 GitHub	6
2.2.2 Jira	7
2.3 Fragebögen	7
2.3.1 Konzeption von Fragebögen	7
2.3.2 Planung der Fragebogenkonstruktion	7
2.3.3 Formulierung der Fragen	9
2.3.4 Arten von Fragen	10
2.3.5 Struktur und Gliederung von Fragebögen	11
2.3.6 Mögliche Verfälschung des Resultats	11
2.3.7 Auswertung von Fragebögen	12
2.3.8 Selbst erstellter Fragebogen	12
2.3.8.1 Planung und Erstellung des Fragebogens	12
2.3.8.2 Struktur und verwendete Fragetypen	12
2.3.8.3 Pilotstudio des Fragebogens	13

2.3.8.4	Auswertung des Fragebogens	13
3	Produktspezifikationen	17
3.1	Eingesetzte Technologien	17
3.1.1	Microsoft HoloLens2	17
3.1.2	Kriterien	17
3.1.3	Game Engine	17
3.1.3.1	Unity	18
3.1.3.2	Unreal Engine	18
3.1.3.3	Game Engine Auswahl und Wechsel im Projektverlauf	19
3.1.4	Unity foundation packages	20
3.1.4.1	MRTK3	20
3.1.4.2	Microsoft OpenXR Plugin	20
3.1.4.3	Integrationsprozess der Plugins	21
3.1.5	Wahl des Modellierungsprogramm	21
3.1.5.1	Wie funktioniert Blender im Allgemeinen?	22
4	Feinkonzept und Realisierung	24
4.1	Entwicklungsumgebungen	24
4.1.1	Visual Studio 2022	24
4.1.2	Unity	24
4.1.2.1	Multidisziplinäre Unterstützung und Integration	24
4.1.2.2	Szenengestaltung und Asset-Management	25
4.1.2.3	Programmierung und Skripterstellung	25
4.1.2.4	Unterstützung für Augmented Reality (AR) und Virtual Reality (VR)	25
4.1.2.5	Erweiterte Debugging- und Profiling-Werkzeuge	26
4.1.2.6	Aufbau einer Unity-Applikation	26
4.1.2.7	Lebenszyklusmethoden in Unity	26
4.1.2.8	Assets	28
4.1.2.9	Unity Szenen	28
4.1.2.10	Unity Manager	28
4.1.2.11	Unity GameObjects und Komponente	29
4.1.2.12	Unity Prefabs	30
4.1.3	Deployment der Anwendung	31
4.1.3.1	Voraussetzungen für das Deployment	31
4.1.3.2	Deployment-Prozess	31
4.1.3.3	Erstmaliges Deployment	34
4.2	Objektdesign	34
4.2.1	Texturen	34
4.2.1.1	Bild-Texturen in der 3D-Modellierung	34
4.2.2	Mesh	36
4.2.3	Rolle von Polygonen in einem Modell	36
4.2.4	Optimierung der einzelnen Modelle	36
4.2.4.1	Polygonreduktion	36
4.2.4.2	Texturenoptimierung	37
4.2.5	Export- und Integrationsprozess	37
4.2.5.1	Dateiformat	38
4.2.5.2	Koordinatensysteme	38
4.2.6	Add-Ons und Plugins	39

4.2.6.1	Looptools: Optimierung von Topologie und Oberflächen	39
4.2.6.2	Images as Planes: Effiziente Integration von Texturen	40
4.2.7	Modi	40
4.2.7.1	Object-Modus	40
4.2.7.2	Edit-Modus	40
4.2.7.3	Texture-Paint-Modus	40
4.2.8	Hierarchie	41
4.2.9	Modifier	41
4.2.10	Modellierung von Gegenständen für das Projekt	42
4.2.10.1	Taschenrechner	43
4.2.10.2	Restliche Modelle	49
4.3	Hauptmenü	51
4.3.1	Erstentwurf	51
4.3.1.1	Probleme beim Erstentwurf	53
4.3.2	Finalversion des Menüs	54
4.3.2.1	Finalversion Menü - Hauptmenü	54
4.3.2.2	Finalversion Menü - Anwendungsszenario	56
4.3.2.3	Setzen des Menüs	56
4.3.3	Laden der Level	57
4.3.4	UI/UX	58
4.4	Nachrichtenaustausch Anwendungsszenario	58
4.4.1	Aufbau von Nachrichtenaustausch Anwendungsszenario	59
4.4.2	Text Mesh Pro	61
4.4.3	Transform Klasse	61
4.4.4	IEnumerator Klasse	61
4.4.5	Coroutines	61
4.4.6	Start der Szene	63
4.4.7	Kabelerkennung	64
4.4.8	Fotoaufnahme	65
4.4.8.1	Photocapture Klasse	65
4.4.8.2	Unity Canvas	65
4.4.8.3	Foto der Umgebung aufnehmen	66
4.4.9	Ladesymbol	67
4.4.10	Fotoverarbeitung	67
4.4.10.1	Unity Job System	67
4.4.10.2	Vector2 Klasse	68
4.4.10.3	Pixel Positionen finden	68
4.4.10.4	Kabel finden	69
4.4.11	Erstellen und Bewegung des Paketes	71
4.4.11.1	Definition eines Rays	71
4.4.11.2	TrackableType Klasse	71
4.4.11.3	ARRaycastHit Klasse	72
4.4.11.4	Entfernung messen	72
4.4.11.5	Erstellung und Positionierung des Paketes	74
4.4.11.6	Bewegung des Paketes	75
4.4.12	Anzeigen von Informationen der Nachricht	76
4.4.13	Andere Versuche / Probleme bei der Programmierung	77
4.4.13.1	Kabelerkennung Version 1	77
4.4.14	Debugging Optionen	78

4.4.14.1	Anzeigen der Kabelerkennung	78
4.4.14.2	Debug Funktion von Raycasts	79
4.4.14.3	Debug Funktion der Kamerabenutzung	79
4.4.15	Starten und Nutzen der Webseite	80
4.4.15.1	Webseite starten	80
4.4.15.2	CORS	82
4.4.16	Frontend der Webseite	84
4.4.16.1	Designprozess	84
4.4.16.2	Verwendete Programmiersprachen	86
4.4.16.3	Funktionen der Webseite	87
4.4.17	Backend der Webseite	88
4.4.17.1	Protokoll zur Datenübertragung	88
4.4.17.2	Bearbeitung von Anfragen	89
4.5	Knapsack Problem Anwendungsszenario	89
4.5.1	Knapsack-Problem Szenario Hirarchie	90
4.5.2	Nutzung von QR-Codes	91
4.5.2.1	Struktur und Inhalt eines QR-Codes	92
4.5.2.2	QR-Code-Tracking	93
4.5.2.3	Interaktion mit QR-Codes	95
4.5.2.4	Positionsbestimmung von QR-Codes	95
5	Zusammenfassung und Abschluss	96
5.1	Ergebnis	96
5.2	Abnahme	96
5.3	Zukunft	97
A	Mockups	98
A.1	Hauptmenu	98
A.2	Nachrichtenaustausch Anwendungsszenario	99
A.3	Knapsack-Problem Anwendungsszenario	100
B	Abläufe	101
B.1	Hauptmenu	101
B.2	Nachrichtenaustausch Anwendungsszenario	102
B.3	Knapsack-Problem Anwendungsszenario	103
C	Literatur	104

Vorwort

Die vorliegende Diplomarbeit wurde im Zuge der Reife- und Diplomprüfung im Schuljahr 2023/24 an der Höheren Technischen Bundeslehr- und Versuchsanstalt Wiener Neustadt verfasst. Die Grundlegende Idee zu dem arbeiten mit der Microsoft HoloLens2, gefördert durch das Förderungsprogramm des Land Niederösterreich "Wissenschaft trifft Schule", lieferte uns unser Betreuer Mag. BEd. Markus Reis. Das Ergebniss dieser Diplomarbeit ist eine Augmented Reality Applikation für die Verwendung innerhalb des Unterrichts und am Tag der offenen Tür.

Besonderer Dank gebührt unserem Betreuer Mag. Markus Reis für sein unerschöpfliches Engagement und seine kompetente Unterstützung. Weiteres möchten wir uns bei unserem Abteilungsvorstand Mag. Nadja Trauner sowie unserem Jahrgangsvorstand MSc. Wolfgang Schermann bedanken, die uns die gesamte Zeit an dieser Schule unterstützt haben.

Diplomarbeit Dokumentation

Namen der Verfasser/innen	Skrepek Moritz Haylaz Seref Lampel Dustin Schoditsch Jonas
Jahrgang Schuljahr	5CHIF 2023 / 24
Thema der Diplomarbeit	Applied Augmented Reality in Education
Kooperationspartner	Land Niederösterreich, Abteilung Wissenschaft und Forschung

Aufgabenstellung	Erklärung und Visualisierung von zwei ausgewählten IT-Grundprinzipien mittels der Microsoft HoloLens2.
------------------	--

Realisierung	Implementiert wurde eine in Unity entwickelte Augmented Reality Applikation für die Mircosoft HoloLens2. Um ein gutes Zusammenspiel zwischen Realität und Augmented Reality zu garantieren wurde Raumerkennung verwendet. Um mit den echten Objekten zu interagieren werden QR-Codes verwendet.
--------------	---

Ergebnisse	Planung, Design, Entwicklung und Test einer funktionsfähigen AugmentedReality-Applikation auf Basis des AR-Devices HoloLens2 von Microsoft, die es ermöglicht ausgewählte technische Themenstellungen im Bereich Informatik (Visualisierung des Nachrichtenaustausches zwischen PCs, Veranschaulichung Knapsack-Problem) für den Einsatz im Unterricht sowie beim Tag der offenen Tür visuell, interaktiv und spielerisch darzustellen.
------------	---

Typische Grafik, Foto etc. (mit Erläuterung)	<p>Das vorliegende Bild stellt das Logo der AR-Applikation dar.</p>  <p>Applied Augmented Reality IN EDUCATION</p>
--	--

Teilnahme an Wettbewerben, Auszeichnungen	
---	--

Möglichkeiten der Einsichtnahme in die Arbeit	HTBLuVA Wiener Neustadt Dr.-Eckener-Gasse 2 A 2700 Wiener Neustadt
---	--

Approbation (Datum, Unterschrift)	Prüfer Mag. Markus Reis	Abteilungsvorstand AV Mag. Nadja Trauner
--	--------------------------------	---

Diploma Thesis Documentation

Authors	Skrepek Moritz Haylaz Seref Lampel Dustin Schoditsch Jonas
Form	5CHIF
Academic Year	2023 / 24
Topic	Applied Augmented Reality in Education
Co-operation partners	Land Niederösterreich, Abteilung Wissenschaft und Forschung

Assignment of tasks	Representation of two selected basic IT principles using the Microsoft HoloLens2.
---------------------	---

Realization	An augmented reality application for the Microsoft HoloLens2 was implemented. In order to guarantee a good interaction between reality and augmented reality, spatial recognition was used. QR codes are used to interact with the real objects.
-------------	--

Results	Planning, design, development and testing of an augmented reality application based on the AR device HoloLens2 from Microsoft, which enables selected technical topics in the field of computer science to be illustrated (visualization of a ping, illustration of the knapsack problem) for use in lessons and on the open doors day of visual, interactive and playful way.
---------	--

Illustrative graph, photo
(incl. explanation)

This image represents the logo of the AR application.



Applied Augmented Reality IN EDUCATION

Participation in
competitions,
Awards

Accessibility of diploma
thesis

HTBLuVA Wiener Neustadt
Dr.-Eckener-Gasse 2
A 2700 Wiener Neustadt

Approval

(Date, Sign)

Examiner

Mag. Markus Reis

Head of Department

AV Mag. Nadja Trauner

Kurzfassung

Diese Diplomarbeit widmet sich der Entwicklung einer Lernapplikation für die HTL Wiener Neustadt unter Verwendung der Unity-Plattform. Die Umsetzung erfolgte in Form einer Augmented Reality (AR) Applikation, speziell für die Microsoft HoloLens 2.

Die Applikation besteht aus drei verschiedenen Szenarien. Darunter, dass Hauptmenu, das Nachrichtenaustausch und Knapsack-Problem Anwendungsszenario,

Die Applikation ermöglicht es den Schülern, während des Tages der offenen Tür zwei wesentliche Grundprinzipien der Informatik mithilfe von Augmented Reality auf spielerische und interessante Weise zu erkunden. Dies bietet den Schülern die Möglichkeit zu erfahren, ob sie ein Interesse an solchen Themen haben. Auch wird darauf abgezielt diese Applikation im Regelunterricht zu verwenden um so diese Prinzipien / Abläufe selbst anwenden und ausprobieren zu können.

Abstract

This thesis focuses on developing a learning application for HTL Wiener Neustadt using the Unity platform. The application is an Augmented Reality (AR) application designed for the Microsoft HoloLens 2.

The application comprises three distinct scenarios, namely the main menu, the message exchange and knapsack-problem scenario.

The application enables students to explore two fundamental principles of computer science in a playful and engaging manner using augmented reality during the open house event. This provides students with an opportunity to gauge their interest in such topics. Additionally, the aim is to integrate this application into regular classroom instruction, allowing students to apply and experiment with these principles and procedures themselves.

Kapitel 1

Einleitung

1.1 Ausgangslage

Um dem IT-Fachkräftemangel entgegenzuwirken, ist es entscheidend, die Ausbildung im MINT-Bereich attraktiver zu gestalten. Diese Diplomarbeit zielt darauf ab, einen wichtigen Beitrag zu diesem Ziel zu leisten, unterstützt durch das Förderprogramm *Wissenschaft trifft Schule* des Landes Niederösterreich. Dabei sollen exemplarische Anwendungen im Bereich Augmented Reality für die Vermittlung von Informatik-Lehrinhalten evaluiert und umgesetzt werden.

→ SKREPEK

1.2 Auslöser

Die Besucher des *Tag der offenen Tür* erhalten durch diese Applikation eine Vorführung der neuesten Technologien, was dazu beiträgt, dass sie erkennen, dass die Schule einen sehr hohen Technologiestandard aufweist. Dies soll sicherstellen, dass sich auch in Zukunft viele interessierte Schüler für eine Ausbildung an der Abteilung für Informatik entscheiden. Darüber hinaus stärkt dies den Ruf der Schule nach außen und präsentiert sie als attraktiven Ausbildungsstandort für zukünftige Mitarbeiter vieler Unternehmen.

1.3 Aufgabenstellung

Die Aufgabenstellung besteht darin, zwei Anwendungszenarien zu erstellen, die jeweils ein Grundprinzip der Informatik veranschaulichen. Im ersten Szenario wird das Grundprinzip des Nachrichtenaustauschs zwischen zwei PCs dargestellt. Das Ziel hierbei ist es dem Benutzer das eigentlich unsichtbare Nachrichtenpaket zu visualisieren. Der Benutzer soll hier auf einer Website eine Nachricht eingeben können, um diese dann an den anderen Computer zu senden und zusätzlich kann auch der Inhalt dieses Pakets durch draufdrücken visualisiert werden.

Im zweiten Szenario wird das in der IT bekannte und vielseitige verwendete Optimierungsproblem des Knapsack-Problems dargestellt. Mit Hilfe von Augmented Reality wird der Rucksack als Inventar angezeigt in welchem der Benutzer alltägliche Gegenstände platzieren kann. Die Aufgabe des Benutzers besteht darin, den Rucksack bestmöglich mit den verfügbaren Gegenständen zu füllen, um den Gesamtwert den Rucksacks zu maximieren. Zusätzlich zur Lösung des Problems kann der Benutzer durch einen Knopfdruck eine der berechneten perfekten Lösungen anzeigen.

1.4 Team

Das Diplomarbeitsteam besteht aus:

- Moritz SKREPEK
- Seref HAYLAZ
- Dustin LAMPEL
- Jonas SCHODITSCH

1.4.1 Aufteilung

Die Rolle des Projektleiters der Diplomarbeit nahm Moritz SKREPEK ein, da dieser die grundlegende Idee für die Darstellung zweier Anwendungsszenarios mittels der Microsoft HoloLens2 hatte.

Die entwickelte Applikation lässt sich in das Hauptmenü, das Nachrichtenaustausch-Szenario und das Knapsack-Problem-Szenario gliedern. Die Implementierung sowie die gesamte UI/UX und das Frontend der Website übernahm Dustin LAMPEL unter Verwendung des UX-Tools-Plugins, welches sämtliche UI/UX Elemente für Mixed Reality Anwendungen bereitstellt und HTML, CSS und JavaScript. Die Umsetzung des Nachrichtenaustausch-Szenarios übernahm SCHODITSCH Jonas mittels Objekt-Tracking und Foto-Aufnahmen. Für die Implementierung des Knapsack-Problem-Szenarios waren Moritz SKREPEK und Seref HAYLAZ mittels Verwendung von Plane-detection, QR-Code Tagging und Tracking, Knapsack Algorithmus und Unitests zuständig.

Kapitel 2

Grundlagen

In diesem Kapitel wird das Vorgehensmodell erläutert sowie alle Tools, die für die erfolgreiche Abwicklung des Projekts benötigt werden. Außerdem wird die Konzeption der Fragebögen behandelt.

2.1 Vorgehensmodelle

Vor der Implementierung des Projekts wurden verschiedene methodische Ansätze für das Projektmanagement untersucht. Das Projektteam entschied sich schnell für ein agiles Vorgehensmodell, um die Dynamik der Planung und Durchführung des Projekts zu optimieren. Zu Beginn des Projekts wurde Scrum als bevorzugtes Modell ausgewählt. Im folgenden Abschnitt wird eine detaillierte Erläuterung des Vorgehensmodells präsentiert, gefolgt von einer Begründung für unsere Wahl.

→ SKREPEK

2.1.1 Scrum

Scrum ist ein agiles Projektmanagement-Framework, dass sich auf die effiziente Entwicklung von Produkten und Software konzentriert. Es legt besonderen Wert auf Zusammenarbeit, Anpassungsfähigkeit und die kontinuierliche Bereitstellung funktionsfähiger Produktinkremeente innerhalb kurzer Entwicklungszyklen, die als Sprints bezeichnet werden.¹

Die zuvor skizzierte Definition bietet einen prägnanten Einblick in das agile Vorgehensmodell Scrum. Die herausragenden Merkmale dieses Modells umfassen:

- Definition von drei zentralen Rollen, die im Folgenden näher erläutert werden.
- Verwaltung des Product Backlogs, der sämtliche Anforderungen enthält.
- Iterative und zeitlich definierte Entwicklung von Produkten.
- Förderung der autonomen Arbeitsweise des Teams.
- Gewährleistung der Gleichberechtigung aller Teammitglieder.

2.1.1.1 Die drei Rollen in Scrum

Product Owner

Der Product Owner ist verantwortlich für die Pflege des Product Backlogs und vertritt dabei die fachliche Auftraggeberseite. Eine zentrale Aufgabe ist die Priorisierung der Elemente im Product Backlog, um den geschäftlichen Wert des Produkts zu maximieren und

¹Scrum.org **What is Scrum**

die Möglichkeit für frühe Veröffentlichungen essentieller Funktionalitäten zu schaffen. Der Product Owner nimmt nach Möglichkeit an den täglichen Scrum-Meetings teil, um Einblicke zu gewinnen. Er steht dem Team für Rückfragen zur Verfügung, um einen reibungslosen Informationsaustausch zu gewährleisten.²

Scrum Master

Der Scrum Master spielt eine zentrale Rolle im Scrum-Prozess und ist für dessen korrekte Umsetzung verantwortlich. Er fungiert als Vermittler und Unterstützer, um einen optimalen Arbeitsfortschritt zu erzielen und kontinuierliche Optimierung sicherzustellen. Ein zentrales Anliegen ist die Beseitigung von Hindernissen, um ein reibungsloses Voranschreiten des Teams zu gewährleisten. Der Scrum Master sorgt für einen effizienten Informationsfluss zwischen dem Product Owner und dem Team, moderiert Scrum-Meetings und behält die Aktualität der Scrum-Artefakte wie Product Backlog, Sprint Backlog und Burndown Charts im Blick. Darüber hinaus obliegt es seiner Verantwortung, das Team vor unbefugten Eingriffen während des Sprints zu schützen.³

Team

Das Team besteht idealerweise aus sieben Mitgliedern und setzt sich interdisziplinär aus Entwicklern, Architekten, Testern und technischen Redakteuren zusammen. Es agiert selbstorganisiert und übernimmt die Verantwortung als eigener Leiter. Es hat die Befugnis, autonom über die Aufteilung von Anforderungen in Aufgaben zu entscheiden und diese auf die einzelnen Mitglieder zu verteilen. Dadurch entsteht der Sprint Backlog aus dem aktuellen Teil des Product Backlogs.⁴

2.1.1.2 Scrum Meetings

Alle Anforderungen an das Produkt werden in sogenannten *User Stories* gesammelt, die vorrangig vom Product Owner im Product Backlog erstellt werden. Während eines Sprints werden die User Stories abgearbeitet. Die Projektentwicklung nach Scrum besteht aus fünf zentralen Elementen:

Sprint Planning Meeting

Im Sprint Planning Meeting wird das Ziel des folgenden Sprints definiert. Dabei werden die Anforderungen im Product Backlog, die in diesem Sprint umgesetzt werden sollen, in einzelne Aufgaben zerlegt und anschließend im Sprint Backlog gesammelt.⁵

Sprint

Ein Sprint ist eine Entwicklungsphase, in der eine voll funktionsfähige und potenziell veröffentlichte Software entsteht. Die Dauer eines Sprints beträgt typischerweise zwischen 1 und 4 Wochen und ist für alle Sprints gleich lang.⁶

²Scrum.org **What is a Product Owner**

³Scrum.org **What is a Scrum Master**

⁴Scrum.org **What is a developer**

⁵Scrum.org **What is Sprint Planning**

⁶Scrum.org **What is a Sprint**

Daily Scrum

Der Daily Scrum ist ein kurzes tägliches Teammeeting im Scrum-Framework. Es dient dazu, den Fortschritt des Teams zu synchronisieren und potenzielle Hindernisse frühzeitig zu identifizieren. Während dieses Meetings informieren die Teammitglieder die anderen über den Abschluss ihrer Aufgaben seit dem letzten Treffen, diskutieren, woran sie bis zum nächsten Treffen arbeiten werden, und geben Einblicke in eventuelle aktuelle Probleme oder Herausforderungen. Dieses regelmäßige Treffen stellt sicher, dass alle Teammitglieder stets auf dem aktuellen Stand sind. Es fördert eine effektive Zusammenarbeit und ermöglicht eine zeitnahe Lösung aufkommender Probleme.⁷

Sprint Review

Während des Meetings präsentiert das Entwicklungsteam den Stakeholdern die im Sprint abgeschlossenen Arbeitsergebnisse, wie zum Beispiel fertige Produktinkremente. Zu den Stakeholdern gehören typischerweise Produktbesitzer, Kunden, Führungskräfte und andere relevante Interessengruppen.⁸

Sprint Retrospektive

Die Sprint Retrospektive ermöglicht es dem Scrum-Team, bestehend aus dem Entwicklungsteam, dem Scrum Master und dem Product Owner, gemeinsam den abgeschlossenen Sprint zu reflektieren und Möglichkeiten zur kontinuierlichen Verbesserung der Projekteinheit zu identifizieren.⁹

Durch die Elemente des agilen Ansatzes, insbesondere die iterative Natur von Scrum, kann ein optimaler Projektablauf gewährleistet werden. Die kontinuierliche Zusammenarbeit mit dem Kunden ermöglicht es, Anforderungen und Erwartungen während des gesamten Entwicklungsprozesses anzupassen und zu verfeinern. Dies führt zu höherer Kundenzufriedenheit und einem Produkt, das besser den tatsächlichen Bedürfnissen entspricht.

Darüber hinaus bietet regelmäßige Kommunikation und Transparenz innerhalb des Teams und mit den Stakeholdern die Möglichkeit, Missverständnisse und Probleme frühzeitig zu erkennen und anzugehen. So kann das Team schnell auf Veränderungen reagieren und den Kurs des Projekts entsprechend anpassen. Dadurch können potenzielle Risiken minimiert und die Effizienz sowie die Qualität der Arbeit verbessert werden.

2.1.2 Begründung der Auswahl

Die Applikation *Applied Augmented Reality in Education* besteht aus drei verschiedenen Szenarien. Das Team, bestehend aus vier Schülern, übernahm jeweils einen Teilbereich oder arbeitete in Subteams an einem dieser Szenarien. Dabei erhielten sie Unterstützung vom betreuenden Lehrer, der stets für Fragen zur Verfügung stand und häufig beratend tätig war.

Als Vorgehensmodell wählte das Team das agile Modell Scrum. Die Scrum-Richtlinien konnten leicht eingehalten werden, da sich das Team täglich in der Schule traf und auch außerhalb der Schule privat in Kontakt stand. Diese regelmäßige Kommunikation ermöglichte es dem Team, Änderungen, Probleme und andere Angelegenheiten leicht zu kommunizieren und zu besprechen.

⁷Scrum.org What is a Daily Scrum

⁸Scrum.org What is a Sprint Review

⁹Scrum.org What is a Sprint Retrospective

Am Ende jedes Sprints wurden die erreichten Ergebnisse mit dem Betreuer besprochen und Neuerungen vorgestellt. Im Rahmen der Sprintreviews wurden Feedback zu den Ergebnissen gesammelt und neue Ansichten sowie Denkweisen vom Betreuer eingebracht und integriert.

Die Sprint Retrospektive ermöglichte den Schülern, einen größeren Mehrwert aus der Projektentwicklung zu ziehen, da sie neben der Anwendung des Scrum-Prozesses auch ihre Fähigkeiten in den einzelnen Bereichen verbessern konnten. Hierbei diskutierten und reflektierten sie positive und negative Aspekte.

2.2 Projektmanagement-Tools

Um einen positiven Verlauf des Projekts zu ermöglichen, benötigt man unterstützende Tools für das Projektmanagement sowie die Verwaltung von Sourcecode.

2.2.1 GitHub

Als Repository für die Source Code Dateien wurde Git in Verbindung mit seiner Webanwendung verwendet. Bei Projektbeginn galt es, die Entscheidung zu treffen, welche Technologie und welcher Anbieter für das Versionskontrollsystem am besten geeignet waren. Andere namhafte Anbieter solcher Plattformen für Verwaltungssysteme sind:

- GitLab¹⁰
- SourceForge¹¹
- BitBucket¹²

Die Wahl von GitHub war durch mehrere Aspekte begründet. Zum einen stellt GitHub eine kostenfreie Lösung dar, die es ermöglicht, ein privates Projekt mit bis zu drei Mitgliedern ohne Kosten anzulegen.¹³ Im Gegensatz dazu bieten einige Plattformen lediglich eine begrenzte Anzahl von Mitgliedschaften in kostenfreien Projekten an. Die Registrierung erforderte lediglich einen Account.

Darüber hinaus zeichnet sich GitHub durch eine benutzerfreundliche Oberfläche, eine breite Unterstützung für verschiedene Programmiersprachen und eine aktive Entwicklergemeinschaft aus.¹⁴ Dies erleichtert die Zusammenarbeit und den Informationsaustausch im Projektteam.

¹⁰GitLab.com **GitLab Website**

¹¹Sourceforge.net **SourceForge Website**

¹²BitBucket.org **BitBucket Website**

¹³GitHub Blog **Repositories und Mitglieder**

¹⁴GitHub-Dokumentation **GitHub Sprachunterstuetzung**

2.2.2 Jira

Jira wurde als Verwaltungstool für die Vorgänge im Projekt in Verbindung mit seiner Webanwendung verwendet. Auch hier stand zu Projektbeginn die Frage im Raum, welche Technologie und welcher Anbieter für das Aufgabenmanagement am besten geeignet waren. Neben Jira existieren noch weitere namhafte Anbieter solcher Tools wie beispielsweise:

- VivifyScrum¹⁵
- Trello¹⁶

Die Entscheidung für Jira basierte auf mehreren Überlegungen. Zum einen bietet Jira eine kostenfreie Lösung, die es ermöglicht, ein SCRUM Board mit mehreren Mitgliedern kostenfrei anzulegen.¹⁷ Ein weiterer entscheidender Faktor war die direkte Verbindung zu dem GitHub-Repository und die Möglichkeit, neue Branches und Commits direkt in Jira zu erstellen.¹⁸

Darüber hinaus bietet Jira eine umfassende Funktionalität für das Projektmanagement, einschließlich der Verfolgung von Aufgaben, der Planung von Sprints und der Erstellung von Berichten.¹⁹ Diese Features ermöglichen es dem Projektteam, den Fortschritt genau zu überwachen und eventuelle Herausforderungen frühzeitig zu identifizieren und anzugehen.

2.3 Fragebögen

→ SKREPEK

Um die erstellte Applikation anhand von User-Feedback zu verbessern wurde in dieser Diplomarbeit ein Fragebogen erstellt und eine Umfrage durchgeführt.

2.3.1 Konzeption von Fragebögen

Bei jeder Umfrage werden Informationen von Personen oder Personengruppen zu der allgemeinen Umsetzung und dem Verständnis der Applikation gesammelt. Diese werden im Anschluss ausgewertet und interpretiert. Wichtig ist hier den Zweck jeder Umfrage genau zu definieren. Durch präzise und detaillierte Zielsetzungen ist es später dann möglich, den Erfolg der Umfrage zu garantieren.

2.3.2 Planung der Fragebogenkonstruktion

Die sorgfältige Konzeption und Gestaltung eines Fragebogens sind grundlegende Schritte, die bei der Planung einer Erhebung unternommen werden. Durch eine sorgfältige Planung wird sichergestellt, dass relevante Daten erhoben werden und die spätere Auswertung erleichtert wird. Aus diesem Grund müssen bereits im Vorfeld verschiedene Entscheidungen getroffen und Definitionen festgelegt werden:

- **Inhalt:** Die Auswahl der Inhalte ist für die Qualität der erhobenen Daten entscheidend. Es sollte erwogen werden, bestehende Fragebögen wiederzuverwenden und gegebenenfalls an die spezifischen Anforderungen der Erhebung anzupassen. Um Missverständnisse zu vermeiden, sollten die Fragen klar und prägnant formuliert sein. Die Verwendung validierter Fragebögen kann bei der Gewährleistung der Vergleichbarkeit mit anderen Studien hilfreich sein.

¹⁵Vivifyscrum.com **VivifyScrum Website**

¹⁶Trello.com **Trello Website**

¹⁷Atlassian **Tarife und Preise vergleichen**

¹⁸Atlassian-Community **How to connect a GitHub account to Jira Software**

¹⁹Atlassian **Im Team schneller vorankommen; an einem Strang ziehen und besser entwickeln**

- **Umfang:** Ein wichtiger Faktor, der je nach Forschungsziel abgewogen werden muss, ist die Länge des Fragebogens. Das Ziel sollte ein ausgewogenes Verhältnis zwischen der Tiefe der Informationen und der Aufrechterhaltung der Teilnahme der Teilnehmer sein. Ein zu umfangreicher Fragebogen kann dazu führen, dass die Befragten ermüden und die Qualität der Antworten beeinträchtigt wird.
- **Ablauf und zeitlicher Rahmen:** Die Entscheidung bezüglich des Ablaufs und des Zeitrahmens der Befragung beeinflusst die Art der Datensammlung. Die Wahl zwischen einer postalischen und einer elektronischen Befragung wirkt sich auf die Antwortzeit und die Effizienz der Datenerhebung aus. Bei elektronischen Erhebungen liegen die Ergebnisse oft schneller vor, während bei postalischen Erhebungen längere Antwortzeiten möglich sind.
- **Zielgruppe:** Die Definition der Zielgruppe spielt eine wichtige Rolle für die Repräsentativität der Ergebnisse. Die Entscheidung für eine Vollerhebung oder eine Stichprobe ist eine Frage der zur Verfügung stehenden Ressourcen und der spezifischen Forschungsziele. Eine Vollerhebung kann eine umfangreiche Datenbasis liefern, während eine Stichprobenerhebung insbesondere bei großen Zielgruppen effizienter sein kann.
- **Fragearten und Antwortskalen:** Die Qualität der erhobenen Daten wird durch die Wahl der Art der Fragen und die Wahl der Antwortskalen beeinflusst. Geschlossene Fragen mit vorgegebenen Antwortmöglichkeiten erleichtern die quantitative Analyse, während offene Fragen die Möglichkeit bieten, qualitative Einsichten zu gewinnen. Die Wahl der Antwortskalen, unabhängig davon, ob es sich um Likert-Skalen oder numerische Bewertungen handelt, sollte auf die spezifischen Forschungsziele abgestimmt sein.
- **Ethik und Datenschutz:** Es ist von entscheidender Bedeutung, ethische Aspekte zu berücksichtigen, wie die Anonymität der Teilnehmer zu wahren und sensible Informationen zu schützen. Der Fragebogen sollte so konzipiert sein, dass die Integrität der Teilnehmer geschützt wird und keine unangebrachten persönlichen Informationen gesammelt werden.
- **Pilotstudie:** Es empfiehlt sich, vor der endgültigen Implementierung des Bogens eine Pilotstudie durchzuführen. Im Rahmen dieser Testphase können mögliche Probleme, Unklarheiten oder Missverständnisse bei der Formulierung der Fragen erkannt und behoben werden. Das Feedback der Testteilnehmer trägt dazu bei, den Fragebogen noch weiter zu optimieren.

Grundsätzlich werden die *quantitative* und die *qualitative Forschung* unterschieden, wobei Fragebögen zur *quantitativen Forschung* aufgrund der besseren Vergleichbarkeit leichter zu interpretieren sind.

Bei der Quantifizierung schließt man von der Stichprobe auf die Grundgesamtheit N, wie in Abbildung 2.1 gezeigt. Es ist wichtig, dass die ausgewählte Stichprobe repräsentativ ist. Das bedeutet, dass die Personen, die an der Stichprobe teilnehmen, die gleichen Voraussetzungen haben wie die Personen, die tatsächlich in der Grundgesamtheit vorkommen. Es handelt sich hierbei um numerische Daten, die erhoben werden und für die eine Auswertung vorgenommen wird. Von Bedeutung ist in diesem Zusammenhang das Verhältnis zwischen der untersuchten Stichprobe und der Grundgesamtheit (Population). In diesem Zusammenhang ist das Verhältnis zwischen der untersuchten Auswahl und der Population von Bedeutung. Alle Merkmale der Personen in der Auswahl müssen mit den Merkmalen der Personen in der Population übereinstimmen, zum Beispiel Alter und Geschlecht.²⁰

²⁰Vgl. Mayer, **Interview und schriftliche Befragung**, S. 57 ff.

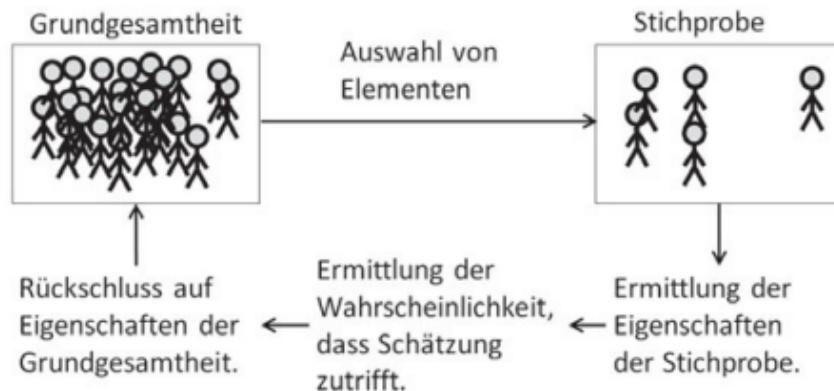


Abbildung 2.1: Zusammenhang zwischen der *Grundgesamtheit* und *Stichprobe*²¹

Die qualitative Forschung ist eine Methode zur Auswertung von Daten, die ausschließlich über Sprache (verbal) übermittelt werden. Diese Methode eignet sich vor allem zur näheren Beschreibung und Analyse von subjektiven Wahrnehmungen, persönlichen Einstellungen, Motiven und Meinungen der Befragten.²²

Am sinnvollsten ist eine Kombination von *qualitativen und quantitativen Ergebnissen*. Nach einer quantitativen Befragung sollte eine stichprobenartige qualitative Befragung durchgeführt werden, um die Ergebnisse besser interpretieren zu können.

Unabhängig davon, ob es sich um qualitative oder quantitative Forschung handelt, sind die drei Gütekriterien *Objektivität, Zuverlässigkeit und Validität* zu erfüllen.

Sie dienen, den Forschungsprozess zu steuern und zu kontrollieren. Die Validität ist ein Maß für die Brauchbarkeit der Methode und bezieht sich auf die tatsächliche Fähigkeit zur Messung des gewünschten Wertes. Das Ergebnis ist umso zuverlässiger, je klarer die Fragen formuliert sind. Entscheidend für die Validität der Analyse ist die Objektivität der Messung. Dabei ist sowohl die Durchführungsobjektivität des Befragers als auch die Auswertungs- und Interpretationsobjektivität des Analytikers zu beachten.²³

Die drei Gütekriterien stehen in einem wechselseitigen Zusammenhang. Nur, wenn die Objektivität gegeben ist, kann die Reliabilität (Zuverlässigkeit) gewährleistet werden. Ist die Reliabilität gering, kann die Validität nur mit einer gewissen Unsicherheit vorhergesagt werden.²⁴

2.3.3 Formulierung der Fragen

Um eine erfolgreiche Umfrage effizient durchführen zu können, ist eine sorgfältige Vorbereitung erforderlich. Die Erkenntnis, dass Umfragen nur bestimmte Aspekte eines Themenbereichs abdecken können, ist von entscheidender Bedeutung. Aus diesem Grund ist eine sorgfältige und präzise Definition dieser Aspekte erforderlich. Besonderes Augenmerk ist darauf zu richten, dass die Fragen klar formuliert sind.

Die zentrale Priorität bei der Formulierung der Fragen ist die Verständlichkeit und Eindeutigkeit. Die folgenden Formulierungsrichtlinien sollten unbedingt beachtet werden:

- Verwendung von einfachem Vokabular ohne Verwendung von Fachausdrücken, Fremd-

²¹Vgl. Brell, **Statistik von Null auf Hundert**, S. 122

²²Vgl. Mayer, **Interview und schriftliche Befragung**, S. 36

²³Vgl. Mayer, **Interview und schriftliche Befragung**, S. 54 ff., S. 88.

²⁴Vgl. Bühner, **Einführung in die Test- und Fragebogenkonstruktion**, S. 33 f.

wörtern oder Ausdrücken aus anderen Sprachen.

- Die Fragen prägnant formulieren.
- Belastende Begriffe wie *Ehrlichkeit* vermeiden.
- Hypothetische Formulierungen ausschließen.
- Fokussierung auf ein bestimmtes Thema für jede einzelne Frage.
- Vermeidung von Überforderung durch die Bereitstellung einer angemessenen Menge an Informationen pro Frage.
- Doppelte Verneinungen vermeiden.²⁵

Die genannten Kriterien sind besonders wichtig bei schriftlichen Befragungen. Um sicherzustellen, dass die Ergebnisse nicht verfälscht werden, darf der Interviewer keine zusätzlichen Fragen stellen oder die bereits gestellten Fragen ändern.

Direkte Fragen sind geeignet, um Fakten und Wünsche zu ermitteln, während formulierte Aussagen oder Feststellungen eher dazu dienen, die Bewertung durch die Befragten in Erfahrung zu bringen. Diese Techniken werden hauptsächlich zur Erfassung von Einstellungen, Wahrnehmungen und Meinungen eingesetzt.

2.3.4 Arten von Fragen

In Abhängigkeit von den Anforderungen der jeweiligen Evaluation können sowohl offene als auch geschlossene Fragen gestellt werden.

Bei offenen Fragen handelt es sich um Fragen, bei denen keine Antwortmöglichkeiten vorgegeben werden. Im Anschluss an die Frage sollte ausreichend Platz für die Beantwortung der Frage gelassen werden. Dieser Fragetyp sollte in den folgenden Fällen verwendet werden:

- Wenn die Anzahl der Antwortmöglichkeiten unbekannt ist.
- Wenn die Formulierung der Antwort des Auskunftspflichtigen für die Auswertung von Bedeutung ist.
- Wenn das Ziel der Erhebung darin besteht, die Unwissenheit und Meinung zu ermitteln.

Im Gegensatz zu offenen Fragen gibt es bei geschlossenen Fragen vordefinierte Antwortmöglichkeiten. Die Teilnehmerinnen und Teilnehmer wählen ihre Antworten aus einer vorgegebenen Liste aus oder entscheiden sich zwischen den vorgegebenen Optionen. Es gibt verschiedene Szenarien, in denen der Einsatz von geschlossenen Fragen sinnvoll ist:

- Wenn die Anzahl der möglichen Antwortalternativen begrenzt und bekannt ist.
- Bei Umfragen, die quantitative Daten für eine statistische Auswertung erfordern.
- Wenn die Standardisierung der Antworten wichtig ist, um eine konsistente Analyse zu ermöglichen.²⁶

In Bezug auf die geschlossenen Fragen ist es noch wichtig anzumerken, dass es im Wesentlichen drei Möglichkeiten für die Benennung oder Kennzeichnung gibt:

- **Numerische Benennung:** Die numerische Benennung ist ein klassisches Notationssystem mit semantischer Bedeutung. Jede Note oder Zahl ist eindeutig einer sprachlichen Formulierung zugeordnet. Der Abstand zwischen den einzelnen Noten ist dabei gleich groß.

²⁵Vgl. Mayer, **Interview und schriftliche Befragung**, S. 89.

²⁶Vgl. Scholl, **Die Befragung**, S. 157.

- **Kennzeichnung durch Formen:** Eine Möglichkeit, geschlossene Fragen zu kennzeichnen, besteht darin, bestimmte Formen wie Kreise, Kästchen oder grafische Skalen (Symbole) zu verwenden.
- **Sprachliche Benennung:** Die Benennung von geschlossenen Fragen erfolgt durch klare sprachliche Ausdrücke oder Texte, welche die verschiedenen Antwortoptionen definieren.²⁷

2.3.5 Struktur und Gliederung von Fragebögen

Die Struktur und Gliederung eines Fragebogens spielen eine entscheidende Rolle bei der Erhebung von Daten. Ein gut durchdachter Aufbau gewährleistet nicht nur eine klare und präzise Erfassung der benötigten Informationen, sondern erleichtert auch die Analyse der Ergebnisse. Bei der Gestaltung eines Fragebogens sollten mehrere wichtige *Schlüsselemente* berücksichtigt werden.

Fragearten sind ein zentraler Aspekt bei der Strukturierung von Fragebögen. Die geschickte Kombination von *geschlossenen* und *offenen* Fragen ermöglicht es, eine umfassende Datenerhebung durchzuführen und einen tieferen Einblick zu gewinnen.

Die *Reihenfolge* der Fragen sollte einer sinnvollen *Sequenz* und *Logik* folgen. Der Fragebogen sollte mit allgemeinen und weniger sensiblen Fragen beginnen, um das Vertrauen der Teilnehmer zu gewinnen. Danach sollten spezifischere und möglicherweise persönlichere Fragen gestellt werden.

Klarheit ist entscheidend. Klare Anweisungen, eine gut lesbare Schrift und genügend Leerraum tragen dazu bei, Missverständnisse zu vermeiden. Sie ermutigen die Teilnehmer, präzise Antworten zu geben.

Vor dem endgültigen Einsatz des Fragebogens empfiehlt sich die Erprobung des Fragebogens im Rahmen von *Pilotstudien*. Dadurch können mögliche Probleme in Bezug auf *Verständlichkeit*, *Länge* und *Schwierigkeitsgrad* der Fragen identifiziert werden, bevor der endgültige Fragebogen an die Zielgruppe verteilt wird.

Die Beachtung dieser Grundsätze bei der Strukturierung und Gliederung von Fragebögen trägt dazu bei, zuverlässige und aussagekräftige Daten für die Analyse zu gewinnen.

2.3.6 Mögliche Verfälschung des Resultats

Die Zuverlässigkeit von Umfrageergebnissen kann durch verschiedene Arten der Verfälschung beeinträchtigt werden. Zwei häufige Verfälschungsarten sind:

- **Simulation:** Teilnehmer neigen dazu, ihre Antworten absichtlich zu verfälschen, um ein bestimmtes Bild von sich selbst zu vermitteln. Dies kann dazu führen, dass die gegebenen Antworten nicht mit den tatsächlichen Meinungen oder Verhaltensweisen übereinstimmen und somit zu einer Verfälschung der Daten führen.
- **Dissimulation:** Es handelt sich hierbei um die bewusste Verzerrung von Informationen durch Teilnehmer mit dem Ziel, bestimmte Aspekte zu verschleiern oder zu verheimlichen. Diese Verzerrung kann dazu führen, dass die gewonnenen Daten nicht der Realität entsprechen und somit die Verlässlichkeit der Ergebnisse der Erhebung in Frage gestellt wird.²⁸

Eine Vielzahl von Faktoren kann Verfälschungen auslösen. Gesellschaftliche Normen üben oft Druck auf den Einzelnen aus, sich selbst in einem positiven Licht darzustellen,

²⁷Vgl. Scholl, **Die Befragung**, S. 164 ff.

²⁸Vgl. Bühner, **Einführung in die Test und Fragebogenkonstruktion**, S. 56.

was zu einem Verhalten führen kann, das eine Simulation darstellt. Auf der anderen Seite kann die Furcht vor sozialen Konsequenzen oder persönlichen Nachteilen dazu führen, dass Individuen Aspekte ihrer selbst zurückhalten oder verbergen.²⁹

Ein umfassendes Verständnis der Ursachen von Simulation und Dissimulation ist entscheidend für die Entwicklung wirksamer Strategien, mit denen diese Verfälschungen in der Umfrageforschung minimiert werden können.

2.3.7 Auswertung von Fragebögen

Die Hauptintention einer Fragebogenerhebung besteht darin, eine homogene Vergleichbarkeit der individuellen Antworten der Befragten zu gewährleisten. Dies ermöglicht eine fundierte statistische Auswertung. Eine unabdingbare Voraussetzung, um aus den erhobenen Fragebogendaten inhaltlich sinnvolle Aussagen ableiten zu können, ist die Umrechnung der qualitativen Antworten in quantitative Werte. Diese Umrechnung erfolgt insbesondere bei computergestützten Verfahren automatisch.

Offene Fragen erfordern eine inhaltliche Auswertung, die in der Regel in Form einer Häufigkeitsanalyse erfolgt, bei der ähnliche Antworten zu Kategorien zusammengefasst werden. Auf diese Weise ist es möglich, die Anzahl der Befragten zu ermitteln, die eine vergleichbare Aussage getroffen haben.

Die Auswertung geschlossener Fragen ist im Allgemeinen mit geringerem Aufwand verbunden, da die Antwortmöglichkeiten bereits vorgegeben sind und jede einzelne Antwort zu einer dieser vorgegebenen Möglichkeiten passen muss.

Für eine angemessene grafische Darstellung der analysierten Daten sind Kreis-, Balken- und Säulendiagramme besonders geeignet. Während Balken- und Säulendiagramme die absoluten Häufigkeiten der Antworten visualisieren und damit Unterschiede in der Anzahl der Antworten deutlich machen, eignen sich Kreisdiagramme besonders, um relative Häufigkeiten, ausgedrückt in Prozent, darzustellen.

2.3.8 Selbst erstellter Fragebogen

Im Verlauf dieser Diplomarbeit wurde ein Fragebogen gemäß den erläuterten Regeln und Grundlagen entwickelt. Der Fragebogen soll am *Tag der offenen Tür* an der HTBLuVA Wiener Neustadt in der Abteilung Informatik am 01.12.2024 durchgeführt werden.

2.3.8.1 Planung und Erstellung des Fragebogens

In der Planungsphase wurde das allgemeine Layout des Fragebogens festgelegt. Anschließend wurden im Projektteam Fragen zusammengestellt, die zur Weiterentwicklung der Applikation beitragen sollen. Der Fragebogen wurde mithilfe der Markup-Sprache *LaTeX* erstellt.

2.3.8.2 Struktur und verwendete Fragetypen

Eine klare und logische Struktur des Fragebogens sowie die Auswahl geeigneter Fragetypen spielen eine entscheidende Rolle, um es dem Befragten zu erleichtern, die Fragen zu beantworten, und seine Motivation aufrechtzuerhalten. Der Fragebogen beginnt mit einer kurzen Einleitung und Begrüßung, um den Befragten den Zweck des Fragebogens und den Grund für seine Durchführung zu vermitteln.

²⁹Vgl. Bühner, **Einführung in die Test und Fragebogenkonstruktion**, S. 59.

Bei der Auswahl der Fragetypen wurde bewusst entschieden, ausschließlich geschlossene Fragen zu verwenden. Diese Entscheidung basierte auf zwei Hauptüberlegungen. Erstens sollte die begrenzte Zeit am Tag der offenen Tür effizient genutzt werden, und es galt, die Besucher nicht zu lange aufzuhalten. Daher wurde es als taktisch sinnvoll erachtet, keine offenen Fragen einzubeziehen. Zweitens erleichtern geschlossene Fragen dem Projektteam die Auswertung und Interpretation der Ergebnisse. Außerdem ermöglichte diese Entscheidung dem Projektteam, spezifische Bereiche einzugrenzen, in denen bereits bekannt war, dass Verbesserungsbedarf besteht.

2.3.8.3 Pilotstudio des Fragebogens

Um Missverständnisse oder Unklarheiten bezüglich der gestellten Fragen zu vermeiden, wurde vor der endgültigen Durchführung der Befragung ein Pilotversuch in einer Abschlussklasse der HTBLuVA Wiener Neustadt, Abteilung Informatik, durchgeführt. Die Ergebnisse dieser Pilotstudie zeigten, dass die gestellten Fragen verständlich waren und ohne größere Probleme beantwortet werden konnten.

2.3.8.4 Auswertung des Fragebogens

Um die Ergebnisse detailliert zu präsentieren, wird in diesem Abschnitt der Fragebogen verwendet, der während des *Tags der offenen Tür* der HTBLuVA Wiener Neustadt, Abteilung Informatik eingesetzt wurde. Die Fragen werden zunächst aufgeführt, gefolgt von einer Analyse der Ergebnisse. Für diese Analyse wird das Tool *Excel* verwendet, um die Ergebnisse auszuwerten und ausführlicher darzustellen.

Diese Vorgehensweise ermöglicht es, die gesammelten Daten konkret zu veranschaulichen und deren Bedeutung für das Entwicklerteam besser verständlich zu machen.

Während des Tages der offenen Tür im Schuljahr 2023/2024 (25 Personen) und in einer Abschlussklasse 5CHIF 2023 / 24 (14 Personen) einer Schule mit Matura wurde eine Umfrage durchgeführt, um die Nutzermeinungen zur aktuellen Anwendung zu erfassen. Das Ziel dieser Umfrage bestand darin zu ermitteln, ob die zugrunde liegenden ausgewählten Konzepte der Informatik verstanden wurden und ob potenzielle Verbesserungsmöglichkeiten vorliegen.

Anschließend wurden sieben Personen (Stichprobe) der Abschlussklasse befragt, um potenzielle Verbesserungsvorschläge zu ermitteln. Das Ziel dieser Umfragen bestand darin, Personen zu befragen, die bereits über grundlegendes IT-Wissen verfügen und dadurch besser beurteilen können, welche Aspekte verbessert werden können.

Der durchgeführte Fragebogen bestand aus fünf geschlossenen Fragen, wobei bei zwei Fragen die Möglichkeit bestand, mehrere Antworten auszuwählen. Die Fragen samt Antwortmöglichkeiten und die zugehörige Auswertung sind wie folgt:

- **1. Frage:** Wie leicht war es für Sie, dass Prinzip des *Nachrichtenaustauschs* zwischen zwei PCs in der Applikation zu verstehen?

Antwortmöglichkeiten:

Sehr schwer(5), Schwer(4), Mittel(3), Leicht(2), Sehr leicht(1)

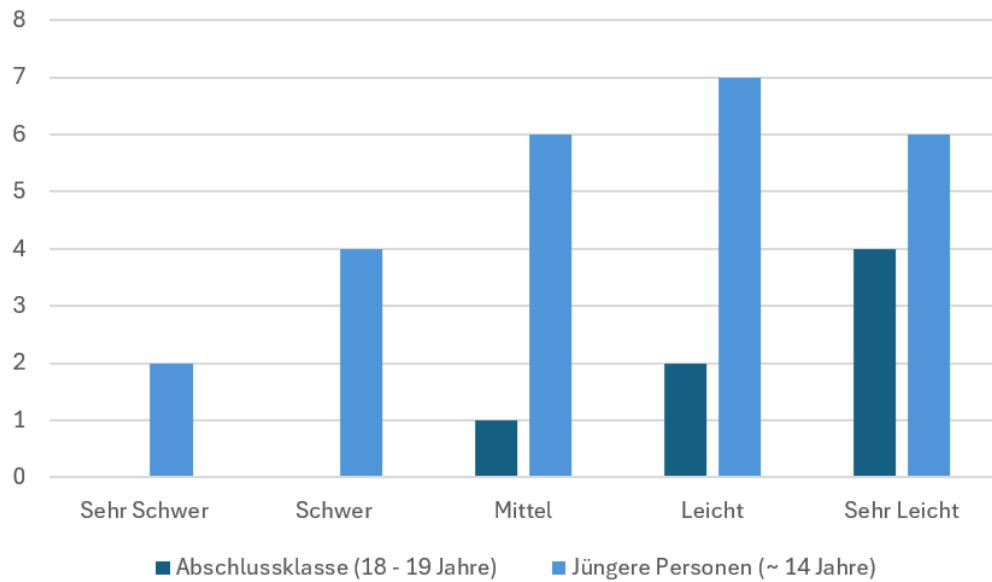


Abbildung 2.2: Auswertung Frage 1

- **2. Frage:** Konnten Sie das Knapsack-Problem in der Applikation ohne Probleme nachvollziehen?

Antwortmöglichkeiten:

Nein nicht wirklich(3), Ja mit einigen Schwierigkeiten(2), Ja problemlos(1)

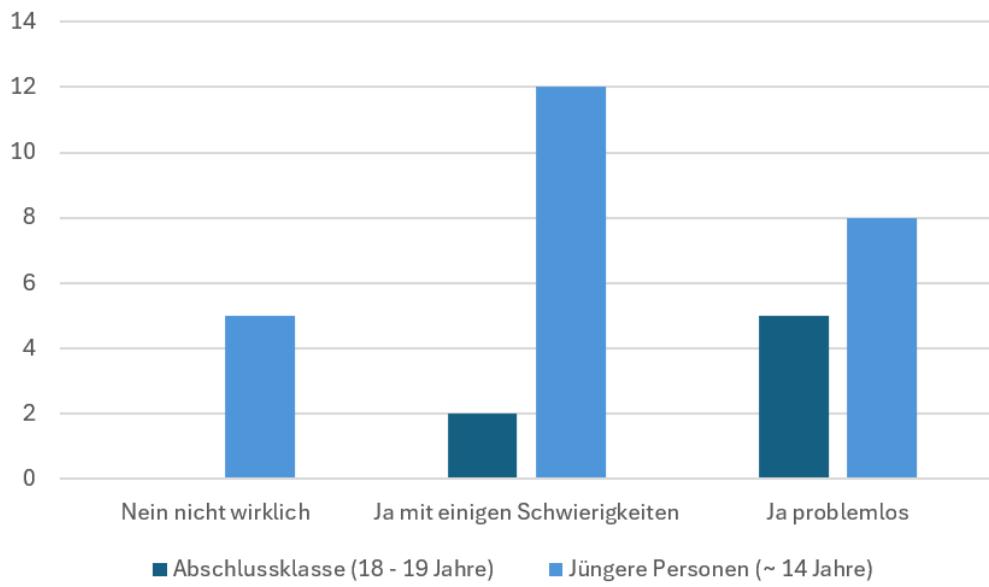


Abbildung 2.3: Auswertung Frage 2

- **3. Frage:** Wie sehr hat die AR-Technologie Ihnen dabei geholfen die Informatik Prinzipien zu verstehen?

Antwortmöglichkeiten:

Sehr negativ(5), Negativ(4), Neutral(3), Positiv(2), Sehr positiv(1)

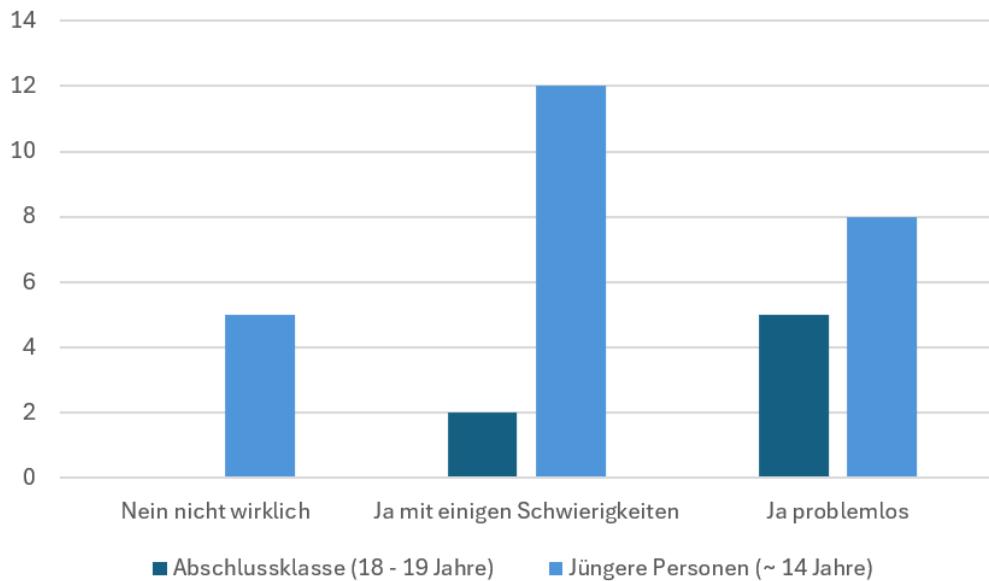


Abbildung 2.4: Auswertung Frage 3

- **4. Frage:** Welche der folgenden Aspekte haben Ihnen am meisten gefallen? (Mehr als eine Auswahl möglich)

Antwortmöglichkeiten:

Lernfördernd, Benutzerfreundlichkeit, Kreativität, Innovativ, Interaktiv

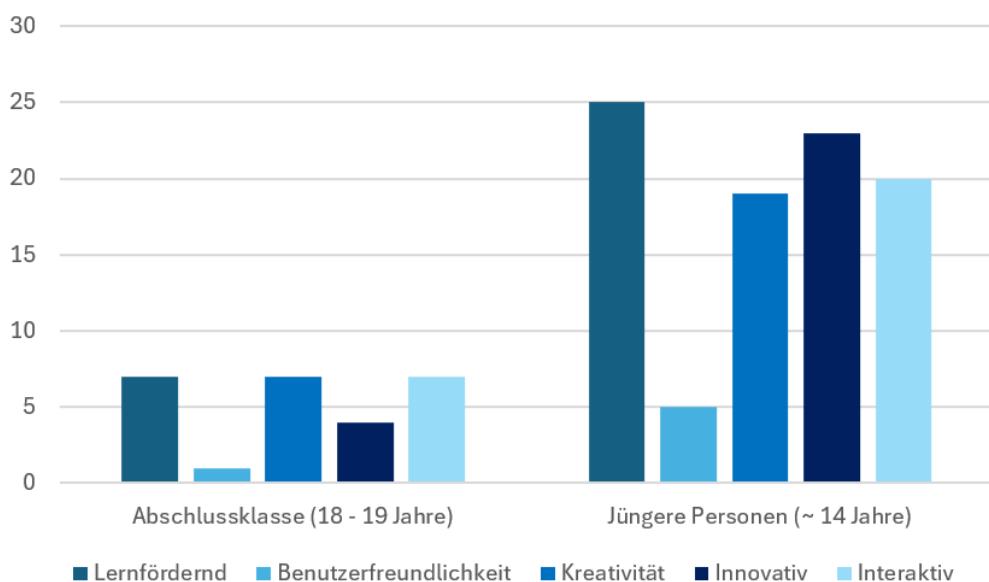


Abbildung 2.5: Auswertung Frage 4

- **5. Frage:** Welche der folgenden Aspekte sind noch verbesserungswürdig? (Mehr als eine Auswahl möglich)

Antwortmöglichkeiten:

Tips / Anweisungen, Leistungsfähigkeit, Inhalt, Stabilität, Übersichtlichkeit

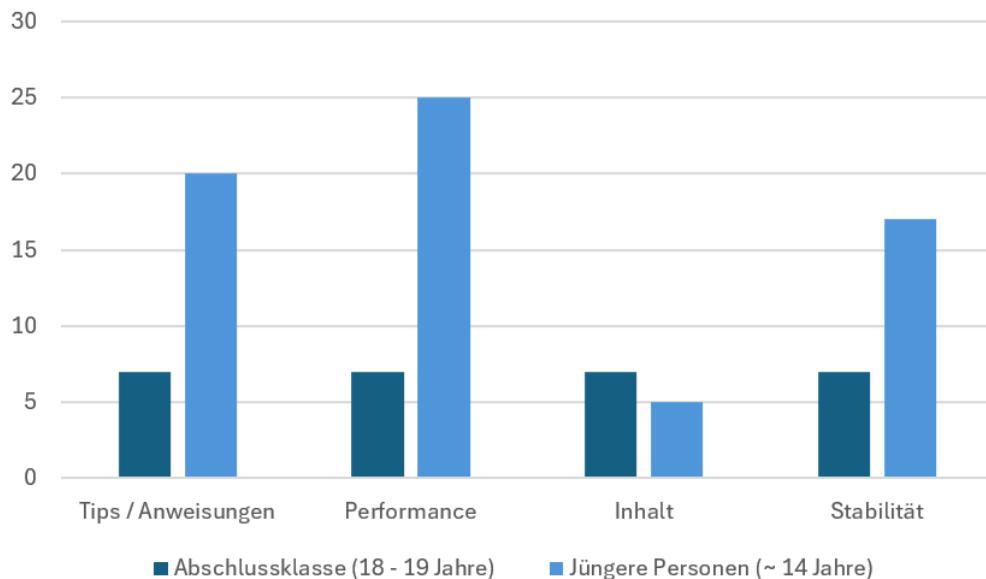


Abbildung 2.6: Auswertung Frage 5

Anhand der Auswertungen des Fragebogens wird deutlich, dass die Antworten der befragten Personen am Tag der offenen Tür und von den Personen der Abschlussklasse stark voneinander abweichen. Dies wird besonders bei den letzten beiden Fragen, die sich auf Aspekte, die dem Benutzer gefallen, und Verbesserungsmöglichkeiten beziehen, deutlich. Hier ist nämlich klar zu erkennen, dass die Personen der Abschlussklasse im Vergleich zu den am Tag der offenen Tür befragten Person signifikant mehr Angaben dazu gemacht haben. Dies lässt sich darauf zurückführen, dass sie über deutlich mehr Erfahrung im Bereich der Entwicklung von Applikationen verfügen und generell ein vertieftes Verständnis der Informatik besitzen. Ein erfreuliches Ergebnis ist jedoch, dass ein Großteil sowohl der am Tag der offenen Tür als auch der in der Abschlussklasse befragten Personen die vermittelten Prinzipien der Informatik gut verstanden hat und auch, dass der Einsatz von AR-Technologie dabei geholfen hat diese zu verdeutlichen.

Zu erwähnen ist, dass bei der Auswertung insbesondere bei der vorletzten Frage festgestellt wurde, dass die Applikation als nicht benutzerfreundlich bewertet wurde. Dieses Ergebnis war jedoch erwartet, da zum Zeitpunkt der Befragung das Projekt noch in einem sehr frühen Entwicklungsstadium war und daher einige essenzielle Funktionen und Erweiterungen wie Anweisungen und Tipps, Performance und Inhalt fehlten.

Darüber hinaus hat diese Befragung und Auswertung wichtige Ergebnisse geliefert, die dazu beigetragen haben, viele Aspekte der Applikation maßgeblich zu erweitern und zu verbessern.

Kapitel 3

Produktspezifikationen

Dieses Kapitel behandelt die Planung und Spezifikation des Projekts. Weiteres wird die verwendete Technologieauswahl begründet und mit Alternativlösungen verglichen.

3.1 Eingesetzte Technologien

In diesem Abschnitt werden die eingesetzten Technologien näher erläutert. Zusätzlich werden Aspekte die für und gegen die Einsetzung sprechen angeführt.

3.1.1 Microsoft HoloLens2

→ SKREPEK

Zentrale Plattform des Projekts ist die Augmented-Reality-Brille "HoloLens 2" von Microsoft. Dabei handelt es sich um eine bahnbrechende Augmented-Reality-Brille, die eine revolutionäre Verbindung zwischen der digitalen und der physischen Welt schafft. Mit einer Reihe hochentwickelter Sensoren, Kameras und einem hochauflösenden Display eröffnet die HoloLens 2 neue Horizonte für interaktive und immersive AR-Erlebnisse.

Die HoloLens 2 bietet eine beeindruckende Immersion, die es dem Nutzer ermöglicht, digitale Objekte nahtlos in seine Umgebung zu integrieren. Dank des fortschrittlichen Hand- und Blickverfolgungssystems können Benutzer mit den holografischen Elementen interagieren, als wären sie Teil ihrer realen Welt. Dies ermöglicht eine breite Palette von Anwendungen, von Unterhaltung über Bildung bis hin zu industriellen Anwendungen.

3.1.2 Kriterien

Bei der Auswahl der Technologien, die für die Entwicklung der geplanten Anwendung eingesetzt werden, war es besonders wichtig, dass diese über eine gute Dokumentation und eine große Entwicklergemeinschaft verfügen, falls Fragen auftauchen, für die nicht direkt im Internet eine Lösung gefunden werden kann. Außerdem sollten sie einfach zu bedienen sein und vor allem eine performante Nutzung der Anwendung gewährleisten.

3.1.3 Game Engine

Um einen reibungslosen Verlauf des Projekts zu gewährleisten, ist die sorgfältige Auswahl der richtigen *Game Engine* von entscheidender Bedeutung. Die Game Engine fungiert als fundamentale Plattform für die Entwicklung und Erstellung von Videospielen, indem sie eine umfassende Palette von Werkzeugen, Bibliotheken und Funktionen bereitstellt, die Entwicklern hilft, Spiele zu konzipieren, umzusetzen und zu optimieren. In diesem Abschnitt werden zwei potenzielle Game Engines, die für das Projekt in Betracht gezogen

wurden, eingehend untersucht und anschließend die Gründe für die getroffene Auswahl erläutert.

3.1.3.1 Unity

Unity ist eine Game Engine, die von Entwicklern weltweit für die Erstellung von 2D- und 3D-Spielen genutzt wird. Sie unterstützt verschiedene Plattformen wie PC, Konsolen, Mobilgeräte und AR/VR-Geräte¹. Unity bietet Entwicklern eine umfangreiche Sammlung von Werkzeugen und Ressourcen, um Spiele schnell zu prototypen und zu entwickeln.

Ein herausragendes Merkmal von Unity ist der Asset Store. Hier können Entwickler Assets wie 3D-Modelle, Texturen, Sounds und Plugins kaufen oder verkaufen. Dadurch können sie ihre Projekte mit hochwertigen Inhalten erweitern und verbessern, ohne alles von Grund auf neu erstellen zu müssen.² Unity bietet außerdem eine starke Community-Unterstützung mit Foren, Tutorials und Schulungen, was besonders für neue Entwickler hilfreich ist.

Die Programmierung in Unity erfolgt hauptsächlich durch die Verwendung von C-Sharp. Diese ist sowohl für erfahrene Entwickler als auch für Anfänger zugänglich. Unity ist aufgrund der Kombination von benutzerfreundlichen Werkzeugen, einer großen Community und einer breiten Plattformunterstützung eine beliebte Wahl für Indie-Entwickler sowie für große Studios.

3.1.3.2 Unreal Engine

Die Unreal Engine ist eine leistungsstarke Game Engine, die für ihre hochwertigen Grafiken und fortgeschrittenen Funktionen bekannt ist. Sie wird häufig für die Entwicklung von AAA-Titeln sowie für hochwertige VR-Erfahrungen verwendet. Die Engine bietet branchenführende Grafikfunktionen wie fortschrittliche Beleuchtung, Partikelsysteme, Physiksimulationen und Echtzeit-Rendering.

Eine bemerkenswerte Funktion der Unreal Engine ist das Blueprints-System (siehe folgender Absatz). Es ermöglicht Entwicklern, Spiele und interaktive Inhalte ohne herkömmlichen Programmiercode zu erstellen. Dadurch wird die Engine besonders zugänglich für Künstler und Designer, die möglicherweise keine tiefen Programmierkenntnisse haben.

Die Unreal Engine bietet eine umfangreiche Sammlung von vorgefertigten Assets und Werkzeugen sowie einen integrierten Marketplace, auf dem Entwickler zusätzliche Inhalte erwerben können. Sie unterstützt eine Vielzahl von Plattformen, darunter PC, Konsolen, Mobilgeräte und VR-Headsets.

Insgesamt ist die Unreal Engine eine leistungsstarke Wahl für Entwickler, die hochwertige Grafiken und fortschrittliche Funktionen in ihren Spielen und Anwendungen benötigen. Sie wird häufig von größeren Studios genutzt, die Zugang zu hochwertigen Tools und Support benötigen.³

Blueprint Visual Scripting

Das Blueprint Scripting System ist eine leistungsstarke visuelle Programmierumgebung innerhalb der Unreal Engine. Es ermöglicht Entwicklern, komplexe Logik und Interaktionen ohne traditionelle Programmierkenntnisse zu erstellen. Das System basiert auf dem Konzept von *Blueprints*, die visuelle Darstellungen von Logik und Funktionen sind.

¹Unity.com **Plattformen**

²Assetstore.Unity.com **Unity Asset Store**

³Unrealengine.com **Wir liefern die Engine. Sie machen sie Unreal**

Die Verwendung von Blueprints bietet eine Reihe von Vorteilen. Einerseits ermöglicht es Künstlern und Designern ohne tiefgreifende Programmiererfahrung, komplexe Interaktionen und Spielmechaniken zu implementieren. Durch das Drag-and-Drop-Interface können Benutzer Funktionsblöcke miteinander verbinden und so den Fluss der Spiellogik steuern.

Andererseits erleichtert das Blueprint-System die Iteration und Prototypisierung. Da Änderungen visuell vorgenommen werden können, können Entwickler schnell experimentieren und Anpassungen vornehmen, um das gewünschte Verhalten zu erreichen. Dies beschleunigt den Entwicklungsprozess und ermöglicht es Teams, flexibel auf Feedback und sich ändernde Anforderungen zu reagieren.

Das Blueprint Scripting System bietet zudem eine hohe Flexibilität und Erweiterbarkeit. Entwickler können benutzerdefinierte Blueprints erstellen und sie in anderen Projekten wiederverwenden, was die Effizienz erhöht und die Entwicklung beschleunigt.⁴

3.1.3.3 Game Engine Auswahl und Wechsel im Projektverlauf

Bei Projektbeginn wurde nach umfassender Recherche und Evaluierung verschiedener Optionen entschieden, die Unreal Engine als primäre Entwicklungsumgebung für dieses Projekt zu verwenden. Diese Entscheidung wurde aufgrund mehrerer überzeugender Faktoren getroffen, darunter insbesondere das beliebte Blueprint Visual Scripting, das die Entwicklung von interaktiven Inhalten erleichtert und auch für Personen ohne umfangreiche Programmierkenntnisse zugänglich macht. Zusätzlich zu diesem herausragenden Merkmal wurden weitere Aspekte berücksichtigt, die die Unreal Engine zu einer attraktiven Wahl machten, wie beispielsweise ihre leistungsstarken Grafikfunktionen, die branchenweit anerkannt sind, sowie ihre Unterstützung für die Entwicklung hochwertiger VR-Erfahrungen und AAA-Titel.

Herausforderungen im ersten Monat des Entwicklungsprozesses

Trotz der zu Beginn getroffenen Entscheidung für die Unreal Engine traten im Verlauf des ersten Monats des Entwicklungsprozesses bestimmte Herausforderungen auf. Diese Herausforderungen können als unerwartete Schwierigkeiten betrachtet werden, die während der Umsetzung des Projekts auftraten und zweifel an der ursprünglichen Entscheidung mit sich brachte. Im Einzelnen wurden folgende Herausforderungen identifiziert:

- 1. Mangelhafte Dokumentation für AR-Entwicklung in der Unreal Engine:** Die unzureichende Dokumentation für die Entwicklung von Augmented Reality (AR)-Anwendungen in der Unreal Engine erwies sich als erhebliche Hürde. Fehlende detaillierte Anleitungen und Referenzen für AR-spezifische Funktionen behinderten die effiziente Integration von AR-Elementen.
- 2. Begrenzte Verfügbarkeit von AR-spezifischen Online-Tutorials:** Ein Mangel an Online-Tutorials, die sich speziell mit der Entwicklung von AR-Anwendungen in der Unreal Engine befassten, führte zu einer beträchtlichen Lernkurve für das Entwicklerteam und verzögerte den Implementierungsprozess von AR-spezifischen Features.
- 3. Komplexität der AR-Entwicklung in der Unreal Engine:** Die Unreal Engine erwies sich als sehr anspruchsvoller für Neueinsteiger in Bezug auf die Umsetzung von AR-spezifischen Funktionen. Die Notwendigkeit, komplexe Skripte zu erstellen und vielfältige Einstellungen anzupassen, führte zu einem erhöhten Zeitaufwand für die Umsetzung von AR-Elementen.

⁴Unreal Engine Dokumentation [Blueprints Visual Scripting](#)

- 4. Eingeschränkte Community-Unterstützung für AR-Entwicklung:** Im Vergleich zu Unity war die Community-Unterstützung für die AR-Entwicklung in der Unreal Engine begrenzt. Die Verfügbarkeit von Ratschlägen und Lösungen für spezifische AR-Herausforderungen war eingeschränkt, was die Eigenständigkeit bei der Lösung von Problemen beeinträchtigte.

Diese Herausforderungen bildeten die Grundlage für die strategische Entscheidung des Projektteams, von Unreal Engine zu Unity zu wechseln. Der Wechsel ermöglichte eine effizientere und zielführende Entwicklung der AR-Applikation, gestützt durch Unity's umfassende Unterstützung, detaillierte Dokumentation und breite Community-Ressourcen.

3.1.4 Unity foundation packages

Um Augmented Reality (AR)-Applikationen in Unity erfolgreich zu entwickeln, sind bestimmte *Foundation Packages* erforderlich. Diese Pakete müssen in Unity importiert werden, um grundlegende Funktionalitäten bereitzustellen, die für die erfolgreiche Umsetzung einer AR-Applikation notwendig sind. Im Folgenden werden die beiden wesentlichen Pakete näher erläutert, die für die Funktionalität einer solchen Applikation unerlässlich sind.

3.1.4.1 MRTK3

Das Mixed Reality Toolkit 3 (MRTK3), welches sowohl für Unity als auch in der Unreal Engine anwendbar ist, ist ein leistungsstarkes Open-Source-Entwicklungstoolkit, das von Microsoft entwickelt und gepflegt wird. Es ist speziell darauf ausgerichtet, die Entwicklung von Mixed-Reality-Anwendungen zu erleichtern, indem es eine umfassende Sammlung von Komponenten, Skripten und Assets bereitstellt. MRTK3 bietet Entwicklern eine Vielzahl von Funktionen und Werkzeugen, die speziell für die Erstellung von Anwendungen für Augmented Reality (AR), Virtual Reality (VR) und Mixed Reality (MR) optimiert sind.

Das Toolkit umfasst eine breite Palette von Funktionen, darunter Interaktionselemente wie Hand- und Gestenerfassung, räumliches Mapping, Physiksimulationen für Objekte in der realen Welt, Benutzerschnittstellen-Design-Werkzeuge und vieles mehr. MRTK3 ist plattformübergreifend und unterstützt verschiedene AR-/VR-Headsets sowie andere Mixed-Reality-Geräte.

Die Bedeutung von MRTK3 für die Entwicklung von AR-Applikationen liegt in seiner Fähigkeit, Entwicklern eine solide Grundlage und eine Vielzahl von vordefinierten Komponenten und Tools zu bieten, die den Entwicklungsprozess beschleunigen und vereinfachen. Durch die Verwendung von MRTK3 können Entwickler komplexe AR-Anwendungen schneller prototypisieren und implementieren, da sie auf eine umfangreiche Bibliothek von Funktionen zugreifen können, die speziell für AR-Szenarien optimiert sind. Dies erhöht die Effizienz der Entwicklung und ermöglicht es den Entwicklern, sich auf die Gestaltung und Umsetzung innovativer AR-Erfahrungen zu konzentrieren, ohne sich um die Grundlagen der AR-Entwicklung kümmern zu müssen.⁵

3.1.4.2 Microsoft OpenXR Plugin

Das Microsoft OpenXR Plugin stellt eine bedeutende Komponente im Kontext der Entwicklung von Augmented Reality (AR)-Applikationen innerhalb der Entwicklungsumgebung dar. Entwickelt und bereitgestellt von Microsoft, ermöglicht dieses Plugin die nahtlose Integration des OpenXR-Standards in Unity. OpenXR, initiiert durch die Khronos Group, fungiert als branchenweiter Standard zur Vereinheitlichung der XR-Anwendungsentwicklung,

⁵Microsoft-Dokumentation Mixed Reality Toolkit 3

wobei XR für Extended Reality steht und sowohl Augmented Reality (AR), Virtual Reality (VR) als auch Mixed Reality (MR) umfasst.

Das Microsoft OpenXR Plugin erleichtert Entwicklern die Arbeit innerhalb von Unity, indem es eine reibungslose Integration des OpenXR-Standards ermöglicht. Hierdurch erhalten Entwickler Zugang zu einer breiten Palette von Funktionen und Möglichkeiten, die durch den OpenXR-Standard standardisiert sind, inklusive plattformübergreifender Kompatibilität sowie optimierter Leistung für XR-Anwendungen.

Die Relevanz des Microsoft OpenXR Plugins für die AR-Entwicklung liegt in seiner Fähigkeit, eine konsistente Entwicklungsumgebung für XR-Anwendungen innerhalb von Unity zu schaffen. Durch die Nutzung dieses Plugins können AR-Entwickler die Vorteile des OpenXR-Standards voll ausschöpfen, einschließlich verbesserte Kompatibilität, Leistung und Zukunftssicherheit ihrer Anwendungen.

Zusätzlich erlaubt das Plugin den Entwicklern den Zugriff auf spezifische Funktionen und Optimierungen, die von Microsoft speziell für die AR-Entwicklung entwickelt wurden. Dies beinhaltet beispielsweise Features zur Verbesserung der AR-Umgebungserkennung, Handhabung von Eingaben sowie Optimierung der Grafikleistung für AR-Anwendungen.⁶

3.1.4.3 Integrationsprozess der Plugins

Zur Integration der genannten Plugins in den Unity Editor, um diese verwenden zu können, wird das externe Tool namens *Mixed Reality Feature Tool* von Microsoft verwendet. Dieses Tool fungiert als umfassende Sammlung von Plugins und Erweiterungen im Bereich der Augmented und Virtual Reality Entwicklung für Unity. Neben den erwähnten Plugins umfasst es weitere wichtige Erweiterungen, darunter:

- Azure Mixed Reality Services
- Experimental
- Mixed Reality Toolkit
- Plattform Support
- Spatial Audio
- World Locking Tools
- Weitere Funktionen (Other features)

Um die Integration durchzuführen, müssen innerhalb des *Plattform Support* Bereichs des *Mixed Reality Feature Tools* sowohl das *Mixed Reality OpenXR Plugin* als auch das gesamte MRTK3-Paket ausgewählt werden. Nach der Auswahl dieser Pakete ist es erforderlich, sie zu validieren und anschließend in das Unity-Projekt zu importieren.

Dieser Prozess gewährleistet eine erfolgreiche Integration der benötigten Plugins und Erweiterungen, die für die Entwicklung von Augmented Reality-Anwendungen in Unity von entscheidender Bedeutung sind.

3.1.5 Wahl des Modellierungsprogramm

→ LAMPEL

Durch Unity und die Open-Source-lastige Modellierungscommunity gibt es verschiedene Möglichkeiten, um an die 3D-Modelle zu gelangen, die für die Anwendung der Szenarien benötigt werden.

Eine Möglichkeit besteht darin, auf weit verbreitete 3D-Modelle auf sogenannten *Asset Stores* zurückzugreifen, wie zum Beispiel dem von Unity selbst. Dort können sie entweder

⁶Khronos Group **OpenXR Plugin**

gekauft oder kostenlos heruntergeladen werden. Die verschiedenen Angebote unterscheiden sich in der Texturqualität und dem Modellierungsaufwand.⁷

Eine weitere Möglichkeit besteht darin, die Modelle selbst in einem Modellierungsprogramm zu entwerfen. Diese Variante ist zwar zeitintensiver und aufwendiger, führt jedoch zu maßgeschneiderten Modellen, die den Wünschen und Anforderungen des Projekts entsprechen. Es gibt verschiedene Programme, die die Anforderungen erfüllen würden. Beispielsweise würden Blender⁸, 3ds Max⁹ von Autodesk oder Cinema 4D¹⁰ von Maxon in Frage kommen. Diese Programme unterscheiden sich nur hinsichtlich der Bedienung, aber der eigentliche Zweck ist gleich.

Nach Absprachen innerhalb des Teams wurde beschlossen, das Modellierungsprogramm Blender zu benutzen, da eines der Teammitglieder bereits gewisses Wissen in der Bedienung und Modellierung mit diesem Programm angehäuft hat. Nicht nur das war der ausschlaggebende Punkt, sondern auch die große Benutzergemeinschaft in Blender mit zahlreichen Tutorials und Hilfestellungen für auftretende Herausforderungen.

Obwohl diese Variante einen höheren Zeitaufwand und zusätzliche Komplexität des Projekts mit sich bringt, erhält man am Ende eine Sammlung von Objekten, die vollständig den Anforderungen und Wünschen entspricht. Zudem ist dies die kostengünstigste Methode, da der Kauf von Modellen oder einer kostenpflichtigen Modellierungsplattform vermieden werden konnte, um hohe Kosten zu vermeiden.

3.1.5.1 Wie funktioniert Blender im Allgemeinen?

Die Applikation Blender ist sehr komplex. Daher werden in der folgenden Beschreibung nur die Schlüsselaspekte und die Funktionalität von Blender für unseren speziellen Anwendungsbereich hervorgehoben.

- **Benutzeroberfläche und Interaktion**

Die Benutzeroberfläche von Blender ist hoch anpassbar, obwohl sie komplex gestaltet ist. Sie enthält 3D-Modelle, Ansichten, Fenster und Panels. Benutzer interagieren mit Objekten und Werkzeugen über Maus- und Tastaturbefehle. Die Effizienz der Arbeit und die Modellierungsdauer hängen stark von der Erfahrung des Benutzers ab, insbesondere von der Nutzung von Shortcuts und Hotkeys.¹¹

- **3D-Modellierung**

Blender ermöglicht die Erstellung von 3D-Modellen durch die Verwendung von Primitiven wie Würfeln, Kugeln, Flächen und Kurven. Diese können bearbeitet und modifiziert werden, um komplexe Formen zu erstellen. Modellierungswerkzeuge wie Extrusion, Verschiebung, Skalierung und Rotation stehen zur Verfügung.¹²

- **Materialien und Texturen**

Um realistische Oberflächen zu erzeugen, können Materialien erstellt und Texturen auf Objekte angewendet werden. Blender ermöglicht die Feinanpassung von Materialeigenschaften wie Diffusreflexion, Glanz, Transparenz und Emission.¹³

- **Gemeinschaft und Ressourcen**

Blender hat eine engagierte Benutzergemeinschaft, die umfassende Dokumentation,

⁷Unity Asset Store

⁸Blender Blender Allgemein

⁹Autodesk 3DS Max

¹⁰Maxon Cinema 4D

¹¹Blender BenutzeroberflÄd'che

¹²Blender Toolbar

¹³Blender Materials

Tutorials und Foren bereitstellt. Diese Ressourcen erleichtern die Einarbeitung und Problemlösung.

Blender wird im gesamten Projekt eingesetzt, angefangen beim Entwurf und der Erstellung des Hauptmenüs bis hin zur Gestaltung jedes einzelnen Objekts. Es wird hauptsächlich für die digitale Modellierung der wichtigsten täglichen Gegenstände von Schülern verwendet. Das Ziel besteht darin, am Ende eine umfangreiche Sammlung von Objekten zu haben, um den Benutzern eine vielfältige und zahlreiche Auswahl an verschiedenen Modellen zu bieten.

Kapitel 4

Feinkonzept und Realisierung

Dieses Kapitel behandelt das Feinkonzept und die Realisierung der Anwendungsszenarien in einer Augmented Reality Applikation. Es wird auch auf die verwendeten Entwicklungsumgebungen, ihre Spezialitäten und besondere Merkmale eingegangen.

4.1 Entwicklungsumgebungen

Dieses Kapitel beschäftigt sich mit den verwendeten Entwicklungsumgebungen und ihren Spezialitäten.

→ SKREPEK

4.1.1 Visual Studio 2022

Visual Studio 2022 ist eine integrierte Entwicklungsumgebung (IDE) von Microsoft, die speziell für die Entwicklung von Softwareanwendungen, Webanwendungen und Desktop-Anwendungen konzipiert ist. Es handelt sich um eine umfangreiche Entwicklungsumgebung, die von Entwicklern weltweit für eine breite Palette von Anwendungsfällen eingesetzt wird.¹

4.1.2 Unity

Der Unity-Editor, entwickelt von Unity Technologies, fungiert als umfassende integrierte Entwicklungsumgebung (IDE) und zentrale Arbeitsumgebung für die Konzeption und Umsetzung von 2D-, 3D-, Augmented Reality (AR) und Virtual Reality (VR) Anwendungen und Spielen. Als Kernelement der Unity-Plattform spielt der Editor eine entscheidende Rolle in der Entwicklung von Projekten, die auf Unity-Technologien basieren.

Die Funktionalität des Unity-Editors erstreckt sich über verschiedene Aspekte der Softwareentwicklung, angefangen bei der visuellen Gestaltung von Szenen und Spielwelten bis hin zur Implementierung komplexer Logik und Interaktionen. Die folgenden Abschnitte vertiefen die Schlüsselmerkmale und Funktionen des Unity-Editors, die ihn zu einem essenziellen Werkzeug für Entwickler machen.

4.1.2.1 Multidisziplinäre Unterstützung und Integration

Der Unity-Editor zeichnet sich durch seine multidisziplinäre Unterstützung aus, die Entwicklern ermöglicht, kollaborativ an Projekten zu arbeiten. Künstler, Entwickler und Designer können innerhalb derselben Umgebung zusammenarbeiten, wodurch ein nahtloser

¹Microsoft.com **Visual Studio 2022**

Austausch von Assets, Szenen und Ressourcen (siehe Abschnitt 4.1.2.9) ermöglicht wird. Die Integration von Grafik-, Physik- und Audio-Engines erleichtert die Schaffung immersiver und ansprechender digitaler Umgebungen.

4.1.2.2 Szenengestaltung und Asset-Management

Ein zentrales Merkmal des Unity-Editors ist die intuitive Szenengestaltung, die es Entwicklern ermöglicht, 2D- und 3D-Szenen durch Drag-and-Drop-Operationen zu erstellen und anzupassen. Das Asset-Management ermöglicht eine effiziente Organisation von Ressourcen wie Modelle, Texturen und Audio-Dateien. Hierbei kommt dem Editor eine Schlüsselrolle in der Strukturierung und Verwaltung umfangreicher Projekte zu.²

4.1.2.3 Programmierung und Skripterstellung

Der Unity-Editor integriert leistungsstarke Programmierfunktionen, die Entwicklern erlauben, Skripte in C-Sharp oder JavaScript zu verfassen. Die Implementierung von Logik, Interaktionen und Funktionalitäten erfolgt durch die Integration von Skripten in Game-Objects und Szenen. Die Echtzeitansicht von Codeänderungen unterstützt einen iterativen Entwicklungsprozess.

4.1.2.4 Unterstützung für Augmented Reality (AR) und Virtual Reality (VR)

Der Unity-Editor ist essenziell für die Entwicklung von AR- und VR-Anwendungen. Durch die Integration von AR Foundation und XR Interaction Toolkit bietet der Editor leistungsstarke Werkzeuge zur Erstellung immersiver Erlebnisse. Die Möglichkeit, Szenen in Echtzeit in AR- und VR-Geräten zu überprüfen, unterstützt Entwickler bei der Feinabstimmung und Optimierung ihrer Projekte.³

AR Foundation

AR Foundation ist ein Framework innerhalb des Unity-Editors, dass eine plattformübergreifende Abstraktionsschicht für die Entwicklung von Augmented Reality (AR)-Anwendungen bereitstellt. Entwickler können durch die Integration von AR Foundation AR-Erfahrungen auf verschiedenen Geräten wie Smartphones, Tablets und AR-Brillen erstellen, ohne sich um die Unterschiede zwischen den AR-Plattformen wie ARCore für Android und ARKit für iOS kümmern zu müssen. Dies ermöglicht eine einheitliche Entwicklung von Augmented-Reality-Anwendungen unabhängig von der zugrunde liegenden Hardware.⁴

XR Interaction Toolkit

Das XR Interaction Toolkit ist ein Toolkit, das innerhalb der Unity-Entwicklungsumgebung verwendet wird und Entwicklern Werkzeuge und Komponenten zur Implementierung von Interaktionen in erweiterten Realitäts (XR)-Umgebungen bietet. Das Toolkit erleichtert die Erstellung komplexer Interaktionen wie das Greifen, Berühren und Manipulieren von virtuellen Objekten in VR- und AR-Szenarien. Der XR Interaction Toolkit unterstützt Entwickler bei der Erstellung immersiver und interaktiver XR-Erlebnisse durch die Bereitstellung plattformübergreifender Kompatibilität. Dadurch wird die Entwicklungseffizienz und -konsistenz verbessert.⁵

²Unity.com **Unity Asset Manager**

³Unity.com **Spiele in Augmented Reality und Virtual Reality**

⁴Unity-Dokumentation **About AR Foundation**

⁵Unity-Dokumentation **XR Interaction Toolkit**

4.1.2.5 Erweiterte Debugging- und Profiling-Werkzeuge

Der Unity-Editor stellt umfassende Debugging- und Profiling-Werkzeuge zur Verfügung, um die Leistung und Funktionalität von Anwendungen zu optimieren. Durch Echtzeit-Inspektion, Fehlerverfolgung und Ressourcenüberwachung unterstützt der Editor Entwickler bei der Identifizierung und Behebung von Problemen, um eine reibungslose Ausführung der Anwendungen sicherzustellen.⁶

4.1.2.6 Aufbau einer Unity-Applikation

Die Struktur einer Unity-Applikation ist entscheidend für eine effektive Entwicklung und Organisation von 3D-Anwendungen und Spielen. Eine typische Unity-Anwendung besteht aus verschiedenen Schlüsselementen, darunter Szenen, GameObjects, Komponenten, Skripte und Assets. Diese werden koordiniert durch die Hauptkomponente der Anwendung, die sogenannte *GameManager* oder *MainScene*. In diesem Abschnitt werden die grundlegenden Bausteine einer Unity-Anwendung sowie bewährte Praktiken für die Strukturierung und Verwaltung dieser Elemente beleuchtet.⁷

4.1.2.7 Lebenszyklusmethoden in Unity

Die Entwicklung von Augmented Reality (AR)-Applikationen in Unity erfordert ein tiefgreifendes Verständnis der Lebenszyklusmethoden, die in MonoBehaviour-Klassen implementiert werden können. Diese Methoden regeln den Fluss der Programmlogik und ermöglichen Entwicklern, spezifische Aktionen zu bestimmten Zeitpunkten im Lebenszyklus einer Anwendung auszuführen.⁸

- **Awake()**: Wird aufgerufen, wenn das Skript erstellt wird, und eignet sich für die Durchführung von Initialisierungen.
- **Start()**: Wird vor dem ersten Frame aufgerufen und ermöglicht Initialisierungsaufgaben.
- **Update()**: Wird in jedem Frame aufgerufen und ist ideal für kontinuierliche Logik.
- **LateUpdate()**: Ähnlich wie Update(), jedoch nachdem alle Update()-Methoden aufgerufen wurden, was sich für nachgelagerte Anpassungen eignet.
- **OnEnable() und OnDisable()**: Werden aufgerufen, wenn ein Skript aktiviert bzw. deaktiviert wird, und bieten die Möglichkeit, spezifische Aktionen auszuführen.

⁶Unity.com Tools for profiling and debugging

⁷Unity.com Was ist mit Quellcodeverwaltung gemeint?

⁸Microsoft Dokumentation Subsysteme MRTK3

Script Lifecycle Flowchart

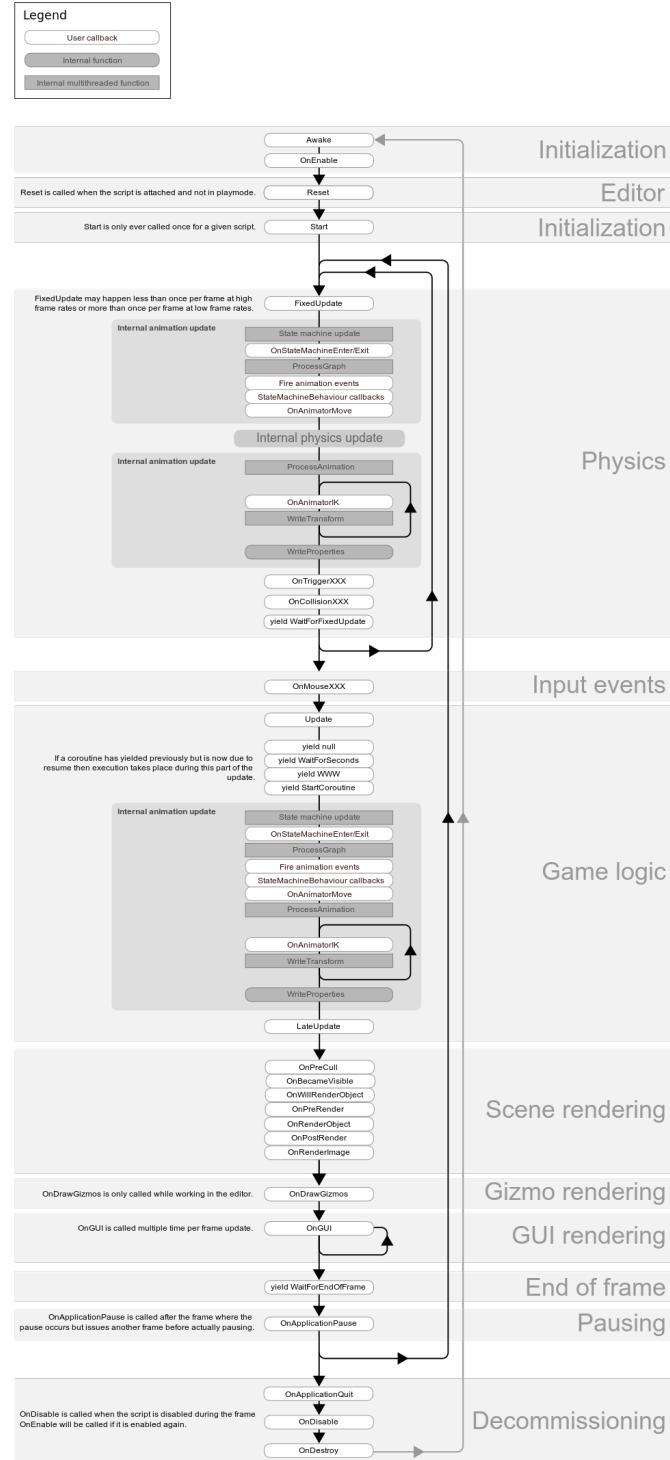


Abbildung 4.1: Ausführungsreihenfolge⁹

⁹Unity-Dokumentation **Order of execution for event functions**

MonoBehaviour Klasse

Die MonoBehaviour-Klasse ist eine Kernkomponente der Unity-Engine und dient als Basis für Skripte, die mit GameObjects in Unity interagieren. Sie ermöglicht die Implementierung von Verhaltensweisen und Funktionen, welche das Verhalten von GameObjects beeinflussen. Sie ermöglicht Entwicklern, bestimmte Aktionen zu bestimmten Zeitpunkten im Lebenszyklus eines GameObjects auszuführen. Die MonoBehaviour-Klasse wird verwendet, um benutzerdefinierte Logik und Interaktionen innerhalb einer Unity-Szene zu implementieren.¹⁰

4.1.2.8 Assets

Unity-Assets sind digitale Ressourcen, die in der Unity-Entwicklungsumgebung zur Gestaltung und Entwicklung von interaktiven 3D-Anwendungen verwendet werden. Sie umfassen 3D-Modelle, Texturen, Audio-Dateien, Skripte und andere Elemente, die zur Erstellung von visuellen und funktionalen Komponenten innerhalb einer Anwendung dienen. Durch die Verwendung von Assets können Entwickler ihre Arbeitsabläufe effizienter gestalten, da sie vorhandene Ressourcen wiederverwenden können. Dadurch wird die Entwicklungsgeschwindigkeit erhöht und die Qualität der Projekte verbessert.

4.1.2.9 Unity Szenen

Unity-Szenen (Siehe Abschnitt ?? oder 4.5 für Umsetzung) bilden das grundlegende Gerüst für die Gestaltung von Inhalten in der Unity-Entwicklungsumgebung. Sie stellen Assets dar, die alle oder einen Teil eines Spiels oder einer Anwendung enthalten. Szenen bieten eine strukturierte Möglichkeit, verschiedene Elemente wie *Umgebungen*, *Charaktere*, *Hindernisse*, *Dekorationen* und *Benutzeroberflächen* zu organisieren und miteinander zu verknüpfen.

Ein wichtiger Aspekt von Unity-Szenen ist ihre Flexibilität. In einem Projekt können beliebig viele Szenen erstellt werden, um die Organisation und Entwicklung des Spiels zu erleichtern. Durch das modulare Konzept von Szenen können Entwickler einzelne Teile des Spiels separat bearbeiten und optimieren, was die Zusammenarbeit im Team und die Wartung des Projekts vereinfacht.

Unity-Szenen dienen nicht nur der Darstellung von Inhalten, sondern auch der Steuerung des Spielablaufs. Durch die gezielte Aktivierung und Deaktivierung von Szenen können verschiedene Abschnitte des Spiels geladen und entladen werden, was die Leistung und Ressourcennutzung optimiert.

Insgesamt bieten Unity-Szenen eine leistungsstarke und flexible Möglichkeit, Spiele und Anwendungen zu strukturieren, zu organisieren und zu verwalten. Sie bilden das Grundgerüst für die Entwicklung von Inhalten in Unity und ermöglichen es Entwicklern, ihre Visionen zu verwirklichen und ansprechende Spielerlebnisse zu schaffen.¹¹

4.1.2.10 Unity Manager

Die präzise und immersive Umsetzung von Augmented-Reality-(AR-)Applikationen erfordert den Einsatz speziälisierter Manager, die grundlegende Funktionen bereitstellen, die für die erfolgreiche Umsetzung verschiedener Szenarien unerlässlich sind.¹² In dieser Applikation werden zwei Manager aus der breiten Palette von Unity bereitgestellten Managern verwendet. Diese sind die folgenden:

¹⁰Unity-Dokumentation **MonoBehaviour Class**

¹¹Unity-Dokumentation **Scenes**

¹²Unity-Dokumentation **Managers**

- **ARPlaneManager:** Der ARPlaneManager umfasst die automatische Erkennung von *horizontalen* und *vertikalen* Flächen in der Umgebung des Benutzers, was die präzise Platzierung virtueller Objekte auf diesen Flächen ermöglicht. Diese Flächen können verschiedene Strukturen wie *Böden*, *Tische* oder andere *flache Oberflächen* umfassen. Nach der Erkennung überwacht der ARPlaneManager kontinuierlich die Bewegungen der Flächen in Echtzeit, was essenziell ist, um die Stabilität virtueller Inhalte auf den realen Flächen zu gewährleisten. Erkannte Flächen können durch *Texturmarkierungen* visuell hervorgehoben werden, um dem Benutzer die Grenzen dieser Flächen deutlicher zu zeigen und die Integration von virtuellen Objekten zu verbessern. Zudem erleichtert der ARPlaneManager das Platzieren virtueller 3D-Objekte in der realen Welt, indem er eine Referenz für die Position und Ausrichtung der erkannten Flächen bereitstellt.¹³
- **ARRaycastManager:** Der ARRaycastManager in Unity ist eine wichtige Komponente für die Entwicklung von Augmented Reality (AR)-Anwendungen. Er ermöglicht es, *Raycasts* von einem festgelegten Ursprungspunkt aus durchzuführen, um Kollisionen oder Treffer mit Objekten in der AR-Umgebung zu erkennen. Diese Funktionalität ist entscheidend für die genaue Platzierung virtueller 3D-Objekte in der realen Welt, basierend auf den Interaktionen des Benutzers. Der ARRaycastManager bietet somit eine grundlegende Funktionalität zur nahtlosen Integration von virtuellen Elementen in die physische Umgebung.¹⁴

Die erfolgreiche Umsetzung der funktionalen Anforderungen in den spezifischen Augmented-Reality-(AR)-Anwendungsszenarien des *Knapsack Problems* sowie des *Nachrichtenaustauschs* hängt maßgeblich von der Integration und Anwendung der zwei genannten Manager, für die Schaffung einer qualitativ hochwertigen, präzisen und immersiven Benutzererfahrung.

Im Kontext des *Knapsack Problem* Anwendungsszenarios spielt der ARPlaneManager eine zentrale Rolle. Durch die Markierung von horizontalen Flächen in der Benutzerumgebung garantiert dieser eine präzise Platzierung des virtuellen Inventar-Objekts und gewährleistet dadurch eine stabile Integration in die reale Umgebung.

Im Kontext des Anwendungsszenarios *Nachrichtenaustausch* spielen beide Manager eine wichtige Rolle. Der ARPlaneManager erkennt die horizontalen ARPlanes in der Umgebung und der ARRaycastManager erkennt anhand eines Rays, auf welchem ARPlane das reale Kabel, über das das Nachrichten-Paket visualisiert wird, liegt.

Insgesamt sind diese Manager wichtige Ressourcen, da sie die technische Umsetzbarkeit und Effektivität von AR-Anwendungen maßgeblich beeinflussen. Durch ihre integrierte Anwendung wird eine nahtlose Verschmelzung von virtuellen und physischen Elementen realisiert, was eine immersive und präzise AR-Benutzererfahrung sowohl in dem Knapsack-Problem als auch bei dem Nachrichtenaustausch Anwendungsszenario gewährleistet.

4.1.2.11 Unity GameObjects und Komponente

Die Konzeption und Verwaltung von GameObjects stellt einen essenziellen Bestandteil der Entwicklungsumgebung von Unity dar. Ein *GameObject* repräsentiert in dieser Umgebung jede Entität innerhalb eines digitalen Szenarios, sei es ein Charakter, eine Umgebungskomponente oder ein Effekt. Diese grundlegenden Objekte agieren als Behälter für Komponenten, welche die Funktionalität und das Verhalten definieren.

Im Kontext von Unity bilden GameObjects die grundlegenden Bausteine einer Szene.

¹³Unity-Dokumentation **Plane Manager**

¹⁴Unity-Dokumentation **Raycast Manager**

Sie sind abstrakte Entitäten, die allein nicht aktiv handeln können, sondern erst durch das *Hinzufügen* von *Komponenten* zu funktionalen Einheiten werden. Die Zuweisung von *Eigenschaften* und *Verhalten* erfolgt durch das Anbringen spezifischer Komponenten an ein GameObject. Beispielsweise kann einem GameObject, das das Konzept einer Lichtquelle repräsentiert, eine Lichtkomponente zugewiesen werden.

Komponenten in Unity dienen dazu, die Eigenschaften und das Verhalten von GameObjects zu definieren. Sie können beispielsweise einer Lichtquelle die Fähigkeit verleihen, Licht zu emittieren, oder einem Charakter die Möglichkeit geben, sich zu bewegen und mit seiner Umgebung zu interagieren. Die Flexibilität von Unity zeigt sich in der Vielfalt der verfügbaren Komponenten, die sowohl vordefiniert als auch maßgeschneidert sein können. Entwickler können mithilfe der Unity Scripting API eigene Komponenten erstellen, um spezifische Verhaltensweisen zu implementieren und die Funktionalität ihrer GameObjects zu erweitern.

Insgesamt bilden GameObjects und deren Komponenten das Rückgrat der Entwicklung von Spielen und interaktiven Anwendungen in Unity. Ihr Verständnis und ihre effektive Verwaltung sind entscheidend für die erfolgreiche Umsetzung digitaler Szenarien und tragen maßgeblich zur Entwicklung innovativer und ansprechender Spielerlebnisse bei.¹⁵.

4.1.2.12 Unity Prefabs

→ HAYLAZ

Unity bietet mit *Prefabs* eine äußerst praktische Funktion zur Erstellung und Wiederverwendung von GameObjects. Prefabs sind modulare, wiederverwendbare Elemente, die als Blaupausen für die Konstruktion von GameObjects dienen. Sie ermöglichen Entwicklern, spezifische Objekte oder Objektstrukturen zu erstellen und zu katalogisieren. Diese Muster können dann in unterschiedlichen Szenarien oder Projekten repliziert werden, was eine effiziente Wiederverwendung und Konsistenz über verschiedene Projektumgebungen hinweg ermöglicht.

Prefabs bieten zahlreiche Vorteile für die Gestaltung und Entwicklung von Projekten. Ihr Hauptvorteil liegt in ihrer Fähigkeit, eine effiziente und konsistente Gestaltung von Projekten zu ermöglichen. Sie erleichtern die Wiederverwendung von Designelementen und tragen so zur Effizienz und Konsistenz des Designprozesses bei.

In komplexen Projekten, die eine Vielzahl von Objekten wie Charaktere, Umgebungen und visuelle Effekte beinhalten, erweisen sich Prefabs als äußerst nützlich. Sie dienen als wiederverwendbare Vorlagen, die es ermöglichen, einmal erstellte Elemente zu speichern und in unterschiedlichen Szenarien wiederzuverwenden. Dies eliminiert die Notwendigkeit, jedes Mal, wenn eine neue Szene erstellt werden muss, von Grund auf neu zu beginnen. Ebenso trägt die Verwendung von Prefabs dazu bei, eine konsistente Designästhetik über das gesamte Projekt hinweg zu gewährleisten.¹⁶¹⁷

Ein weiterer bedeutsamer Vorteil von Prefabs besteht in ihrer Fähigkeit, eine unkomplizierte Aktualisierung und Iteration von Projektobjekten zu ermöglichen. Bei notwendigen Modifikationen an einem spezifischen Objekt kann das zugehörige Prefab bearbeitet werden. Diese Modifikationen werden dann automatisch auf alle Instanzen dieses Prefabs angewendet, die in der Szene implementiert sind. Dies optimiert den Arbeitsablauf und vereinfacht den Prozess erheblich.

Innerhalb des Projekts wird die Prefab-Technologie in folgenden Bereichen eingesetzt:

- **3D-Modelle** 4.2.10.2: Die dreidimensionalen Modelle sind als einzelne Prefabs kon-

¹⁵Unity-Dokumentation **GameObjects**

¹⁶Unity-Dokumentation, **Prefabs**

¹⁷Unity-Dokumentation, **Prefabs-2**

zipiert. Diese Struktur ermöglicht eine einfache Bearbeitung und Anpassung spezifischer Eigenschaften wie Rotation, Position und Skalierung, ohne dass das gesamte Modell jedes Mal ersetzt werden muss. Diese Flexibilität erleichtert die Feinabstimmung der Modelle und ermöglicht eine schnelle Anpassung. Beispiele für die Anwendung dieses Prinzips sind im Abschnitt [4.2.10.2](#) zu finden.

- **QR-Codes 4.5.2:** In unserer Szene arbeiten wir gleichzeitig mit mehreren QR-Codes. Dank der Verwendung der Prefab-Technologie können wir diese mit minimalem Aufwand häufig instanziieren. Nach der Instanzierung wird die Arbeit mit diesen virtualisierten QR-Codes weiter vereinfacht. Durch den leichteren Zugriff auf die untergeordneten Objekte der Prefabs können wir die Informationen, die auf den QR-Codes angezeigt werden, mühelos aktualisieren.

Zusammenfassend kann gesagt werden, dass Prefabs ein unverzichtbares Werkzeug für die effektive Projektentwicklung in Unity sind. Sie ermöglichen es Entwicklern, Zeit zu sparen, die Konsistenz ihres Projekts zu gewährleisten und den Prozess der Aktualisierung und Iteration von Projektobjekten zu optimieren. Daher sind sie ein wesentlicher Bestandteil jeder effektiven und effizienten Projektentwicklungsumgebung.

4.1.3 Deployment der Anwendung

→ HAYLAZ

Die Entwicklung der Anwendung findet größtenteils auf den Laptops statt. Jedoch ist es auf diesen Geräten nur begrenzt möglich, die AR-Funktionalitäten zu testen. Um die Anwendung vollständig auf einem AR-fähigen Gerät zu überprüfen, muss sie auf dieses geladen werden. In diesem Abschnitt wird der genaue Prozess beschrieben, wann und wie die Anwendung auf ein AR-fähiges Gerät deployt wird.

4.1.3.1 Voraussetzungen für das Deployment

Damit das Deployment der Anwendung auf einem AR-fähigen Gerät erfolgreich verläuft, müssen bestimmte Voraussetzungen erfüllt sein:

- **Kompiliertes Unity-Projekt:** Alle erforderlichen Dateien und Ressourcen der Anwendung werden während des Build-Prozesses zu einem ausführbaren Unity-Paket kompiliert.
- **Netzwerkverbindung:** Es ist notwendig, dass sowohl das AR-fähige Gerät als auch der Computer, auf dem der Build durchgeführt wurde, sich im selben Netzwerk befinden. Nur so kann das Unity-Paket über das Netzwerk auf das AR-fähige Gerät übertragen werden.
- **Authentifizierung:** Der Computer, der sich mit der HoloLens verbinden und die Anwendung deployen möchte, muss von der Brille authentifiziert werden. Dies gewährleistet, dass nur autorisierte Geräte Zugriff auf die Brille haben und die Übertragung sicher erfolgt.

4.1.3.2 Deployment-Prozess

Die Entwicklung und Bereitstellung einer Anwendung für die HoloLens 2 unter Verwendung von Unity und der *Universal Windows Platform (UWP)* ist ein strukturierter Prozess, der mehrere Schritte umfasst, beginnend mit der Kompilierung des Unity-Projekts.

Kompilierung und Build-Einstellungen in Unity

Dieser Prozess findet auf dem Computer statt, auf dem sich das Unity-Build-Projekt befindet, und kann über das Tastenkürzel **Strg + Umschalt + B** aufgerufen werden, um die Build-Einstellungen in Unity zu öffnen.

In den Build-Einstellungen wird die Zielplattform festgelegt, auf die die Anwendung bereitgestellt werden soll. Für die HoloLens 2 ist die *Universal Windows Platform (UWP)* die geeignete Zielplattform aufgrund ihrer Kompatibilität mit diesem Gerät. Die UWP ermöglicht die Entwicklung von Anwendungen, die auf verschiedenen Windows 10/11-Geräten einschließlich der HoloLens 2 ausgeführt werden können.¹⁸

Es gibt verschiedene Build-Einstellungen, wie in Abbildung 4.2 dargestellt. Besonders relevant sind dabei die Einstellungen für die *Architektur* und die *Build-Konfiguration*. Die Architektur wird entsprechend der HoloLens 2 auf ARM 64-Bit festgelegt. Die Build-Konfiguration sollte auf *Release* eingestellt werden. Der Grund dafür ist die voreingestellte *Build-Konfiguration* von Unity, die eine Kompatibilität mit der HoloLens 2 gewährleistet. Die übrigen Optionen sollten entsprechend den Vorgaben in der Abbildung beibehalten werden. Diese Einstellungen sind wichtig, um in den nächsten Schritten eine funktionierende Übertragung zu garantieren.^{19 20}

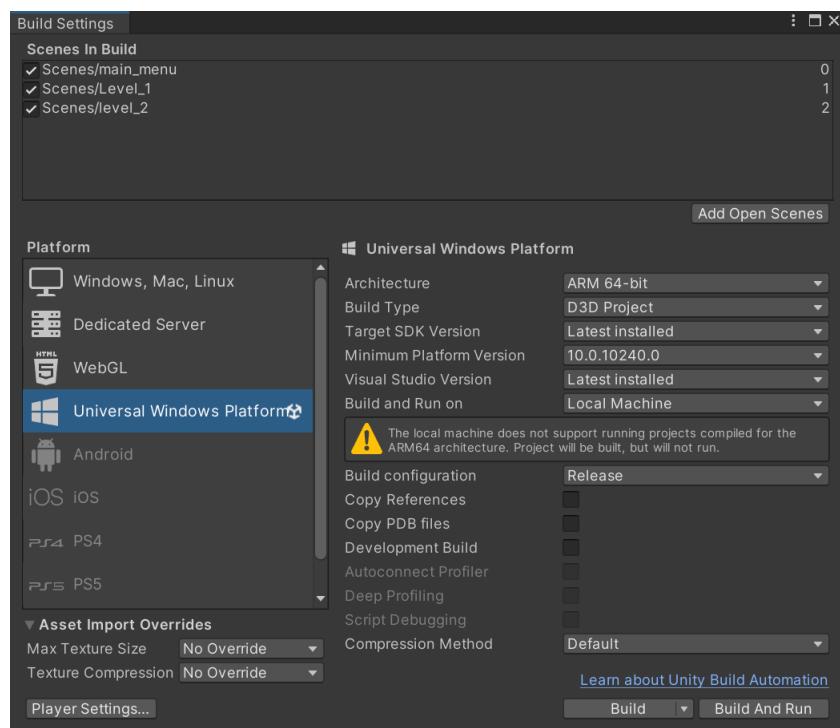


Abbildung 4.2: Build-Einstellungen in Unity fuer deployment

Nach einem erfolgreichen Build erhalten Sie einen Ordner mit den benötigten Dateien für das Deployment. Darin befindet sich unter anderem eine Visual Studio Solution (*VS Solution*), die Informationen zur Struktur des Projekts enthält und Build-Konfigurationen verwaltet.²¹

¹⁸Microsoft-Dokumentation, **Universal Windows Platform**

¹⁹Microsoft-Dokumentation, **Build-Tutorial**

²⁰Microsoft-Dokumentation, **Visual-Studio-Build-Settings**

²¹Microsoft-Dokumentation, **Visual Studio Solution**

Einstellungen in der Visual Studio Solution

Der nächste Schritt in dieser Visual Studio Solution, hier als *AARIE.sln* bezeichnet, umfasst die Vorbereitung der Umgebung für die Bereitstellung auf AR-fähigen Geräten.

Visual Studio ist eine von Microsoft entwickelte integrierte Entwicklungsumgebung (IDE), die hauptsächlich für die Softwareentwicklung verwendet wird. Innerhalb von Visual Studio können Entwickler Code schreiben, debuggen, testen und – in diesem speziellen Anwendungsfall – bereitstellen. ²²

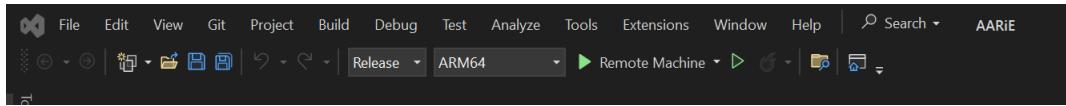


Abbildung 4.3: Toolbar – hier müssen die Einstellungen vorgenommen werden

Nach dem Öffnen der Datei müssen die Konfiguration auf *Release* und die Plattform auf *ARM64* eingestellt werden. Das Bereitstellungsziel ist von *Device* auf *Remote Machine* umzustellen. Diese drei Einstellungen sind in der *Toolbar* zu finden (4.3). Zusätzlich muss die IP-Adresse des AR-fähigen Geräts in den Projekteigenschaften unter Debugging gesetzt werden. Die IP-Adresse der HoloLens 2 kann auf der Brille in den Einstellungen unter Netzwerk gefunden werden. In den Projekteigenschaften ist noch folgende Einstellung zu treffen:

Der Authentication Mode *Universal (Unencrypted Protocol)* wird gewählt, um eine reibungslose Kommunikation zwischen dem Computer und der HoloLens 2 zu ermöglichen, insbesondere während des Deployments der Anwendung. Dieser Modus erlaubt eine unverschlüsselte Kommunikation zwischen den Geräten, was in diesem speziellen Anwendungsszenario aus verschiedenen Gründen vorteilhaft ist. Zum einen vereinfacht die Verwendung eines unverschlüsselten Protokolls die Konfiguration und ermöglicht eine schnellere Einrichtung der Verbindung zwischen dem Entwicklungscomputer und der HoloLens 2. Da dies ein Entwicklungs- und Testumfeld ist, wo Sicherheit weniger prioritär ist als Effizienz und Schnelligkeit, wird diese einfachere Konfiguration bevorzugt. Weiterhin, minimiert die Verwendung eines unverschlüsselten Protokolls das Risiko von Verbindungsproblemen und -verzögerungen, die bei verschlüsselten Kommunikationsprotokollen auftreten können, insbesondere in lokalen Netzwerken oder Umgebungen, in denen die Infrastruktur möglicherweise nicht optimal konfiguriert ist. In einer Entwicklungs- und Testumgebung, in der die Anwendung iterativ entwickelt und getestet wird, steht die Effizienz im Vordergrund. Die Verwendung eines unverschlüsselten Protokolls erleichtert den Entwicklungsprozess, da sie weniger Overhead und Komplexität mit sich bringt. Es ist jedoch wichtig zu beachten, dass in produktiven Umgebungen, in denen Sicherheit eine höhere Priorität hat, verschlüsselte Kommunikationsprotokolle verwendet werden sollten, um die Vertraulichkeit und Integrität der übertragenen Daten zu gewährleisten. ²³

Deployment auf die HoloLens2

Nachdem alle Einstellungen in Visual Studio vorgenommen wurden, kann die

Anwendung auf die HoloLens 2 deployt werden. Dazu muss die Solution mit *Start without Debugging* gestartet werden. Nach einem längeren Ladevorgang wird die Anwendung auf der HoloLens 2 geladen und gestartet.

²²Microsoft-Homepage, **Visual-Studio**

²³Microsoft-Dokumentation, **Universal Unencrypted Protocol**

Falls die Anwendung beendet wird, kann sie auf der HoloLens 2 unter *Start* → *alle Apps* → *AARIE* erneut gestartet werden. Falls Änderungen an der Anwendung auf den Computern vorgenommen wurden, muss der gesamte Prozess erneut durchgeführt werden. Es ist wichtig zu beachten, dass die Anwendung nicht automatisch aktualisiert wird (hot deployment).

4.1.3.3 Erstmaliges Deployment

Beim erstmaligen Deployment auf die HoloLens kann es zu Problemen kommen, die durch die Authentifizierung der HoloLens verursacht werden. In diesem Fall muss der "neue" Computer, von dem aus das Deployment durchgeführt wird, sich auf der HoloLens erst authentifizieren. Dazu werden Sie nach dem Starten des Deployments aufgefordert. Auf dem Computer wird ein Authentifizierungscode angezeigt, der auf der HoloLens unter *Einstellungen* → *Update & Sicherheit* → *für Entwickler* → *Koppeln* eingegeben wird. Nachdem die Authentifizierung erfolgreich abgeschlossen wurde, kann das Deployment erneut gestartet werden.

4.2 Objektdesign

→ LAMPEL

Die Bedeutung des Objektdesigns nimmt in der vorliegenden Arbeit einen zentralen Stellenwert ein, wie bereits im Abschnitt 3.1.5 erläutert wurde. Da sämtliche Modelle und Gegenstände eigenständig erstellt werden sollen, ist ein fundiertes Verständnis der grundlegenden Konzepte und Prozesse des Objektdesigns unerlässlich. Im nachfolgenden Abschnitt werden daher alle relevanten Begriffe erläutert, die beim Modellieren auftreten, und die einzelnen Schritte des Modellierungsprozesses werden in möglichst detaillierter Form durchgegangen.

Das Objektdesign umfasst nicht nur die Schaffung von ästhetisch ansprechenden und funktionalen 3D-Modellen, sondern auch die Optimierung dieser Modelle, um ein optimales Benutzererlebnis sicherzustellen. Dies beinhaltet Aspekte wie die Effizienz der Modelle in Bezug auf Rechenleistung und Speicherplatz, die Benutzerfreundlichkeit sowie die visuelle und funktionale Qualität der Modelle.

4.2.1 Texturen

Texturen sind ein wichtiger Bestandteil der Modellierung und verbessern die visuelle Erscheinung von Objekten, um sie realistischer wirken zu lassen.²⁴ Sie beschreiben die Oberflächenbeschaffenheit eines Objekts, einschließlich Eigenschaften wie Farbe, Rauigkeit, Reflexionsvermögen, Glanz, Lichtdurchlässigkeit und mehr. In Blender werden Texturen typischerweise als Grafiken oder Muster verwendet, die auf die Oberfläche eines 3D-Modells projiziert werden. Im Projekt wurde viel mit *Bild-Texturen* gearbeitet, um möglichst effiziente, realistische und ästhetisch ansprechende Modelle zu erzeugen.²⁵

4.2.1.1 Bild-Texturen in der 3D-Modellierung

Im folgenden Abschnitt wird erläutert, was unter dem Begriff *Bild-Textur* zu verstehen ist und wie die Umsetzung und Arbeit mit Bild-Texturen im Kontext von Blender funktioniert.

Diese Texturen können als Bilddateien importiert oder innerhalb des Programms selbst erstellt werden.

²⁴ Adobe What are 3d textures

²⁵ TechnicalProTips Textur

Die Verwendung von Bild-Texturen ermöglicht es Benutzern, eine Vielzahl von Oberflächeneffekten zu erzeugen, wie beispielsweise Holzmaserungen, Metalltexturen, Stoffmuster und mehr. Durch die Verwendung einer geeigneten Textur kann die visuelle Qualität eines Modells erheblich verbessert werden.

Im Blender-Programm können Bild-Texturen mithilfe des *Shader-Editors* auf das Material des 3D-Objekts angewendet werden. Hierfür wird eine *Image-Texture-Node* erstellt, welche den Farbwert der Textur an das Material übergibt. Es können verschiedene Einstellungen vorgenommen werden, um die Art und Weise der Texturprojektion zu steuern. Zum Beispiel kann die Projektionsart (Box, Flat, Sphere usw.) festgelegt werden, um anzugeben, wie die Textur auf die Oberfläche des Modells angewendet wird. Weitere Einstellungen können die Skalierung der Textur, die Wiederholungsmuster, die Mischmodi und andere Textureigenschaften umfassen.²⁶

Der Shader-Editor wird in Blender verwendet, um mithilfe von sogenannten *Nodes* Eigenschaften für Texturen und Materialien zu vergeben. Der Editor wurde im Projekt hauptsächlich für die Anwendung von Bild-Texturen genutzt.²⁷

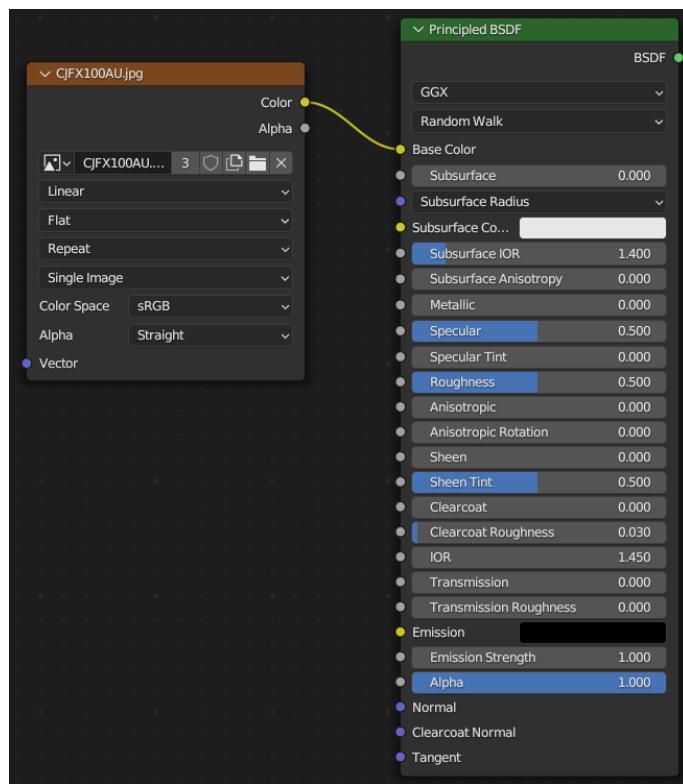


Abbildung 4.4: Beispielhafte Ansicht auf Image-Textur-Nodes im Shader-Editor

In Abbildung 4.4 ist zu erkennen, wie die Image-Textur-Node den Farbwert an das Material weitergibt. Als Beispiel wurde hier der Taschenrechner verwendet, der im Laufe des Kapitels noch näher erläutert wird. Die Abbildung zeigt die verschiedenen Konfigurationsmöglichkeiten, die bereits im Abschnitt 4.2.1.1 erläutert wurden.

²⁶artistischer Bild-Texturen

²⁷Blender Node-Introduction

4.2.2 Mesh

In der 3D-Grafik bezeichnet ein Mesh eine Sammlung von Eckpunkten, Kanten und Flächen, die die Geometrie eines 3D-Objekts definieren. Es bildet die Grundlage für die Darstellung von Objekten in einer virtuellen Umgebung. Ein Mesh kann aus einer beliebigen Anzahl von Polygonen bestehen, die die äußere Form und Struktur des Objekts definieren.²⁸

Die Rolle eines Meshes bei der Modellierung besteht darin, die Form, Oberflächenstruktur und Details eines Objekts zu definieren. Durch die Anordnung und Verbindung der Eckpunkten, Kanten und Flächen können komplexe geometrische Formen erzeugt werden. Polygone ermöglichen dabei die Darstellung von Objekten mit unterschiedlichen Detailgenauigkeiten. Je mehr Polygone ein Mesh hat, desto detaillierter und realistischer kann das Modell dargestellt werden. Allerdings führt dies auch zu einem höheren Bedarf an Rechenleistung und Speicherplatz.²⁹

Meshes werden in verschiedenen Bereichen der 3D-Grafik verwendet, wie zum Beispiel der Modellierung von Objekten, der Animation und der Visualisierung. Sie dienen als Grundlage für die Erstellung von digitalen Szenen.³⁰

4.2.3 Rolle von Polygonen in einem Modell

Polygone sind grundlegende Bausteine eines Meshes und spielen eine wichtige Rolle bei der Darstellung von 3D-Modellen. Sie bestehen aus einer Reihe von Eckpunkten (Vertices), Kanten und Flächen, die diese Punkte miteinander verbinden. In einem 3D-Modell definieren Polygone die Form, die Oberflächenstruktur und die Details eines Objekts.³¹

Durch die Anordnung und Verbindung von Polygonen entstehen komplexe Strukturen, die eine visuelle Darstellung von Objekten ermöglichen. Je mehr Polygone ein Modell hat, desto detaillierter und realistischer kann es dargestellt werden. Allerdings führt dies auch zu einem höheren Bedarf an Rechenleistung und Speicherplatz. Polygone sind somit ein wesentlicher Bestandteil eines Meshes und tragen maßgeblich zur visuellen Qualität und Komplexität eines 3D-Modells bei.³²

4.2.4 Optimierung der einzelnen Modelle

Der Entwurf von 3D-Objekten mit Blender erfordert ein systematisches Vorgehen, insbesondere im Zusammenhang mit der Optimierung dieser. Im Folgenden werden spezifische Aspekte dieses Prozesses beleuchtet, darunter die Polygonreduktion und die Texturoptimierung.³³

4.2.4.1 Polygonreduktion

Die Polygonreduktion ist eine Technik, die darauf abzielt, die Anzahl der Polygone in einem 3D-Modell zu reduzieren, um die Belastung der Hardware zu verringern, insbesondere bei Echtzeitanwendungen wie Computerspielen oder Simulationen. Eine effiziente Polygonreduktion ermöglicht eine bessere Leistung und eine schnellere Darstellung der Modelle auf verschiedenen Plattformen.³⁴

²⁸GeeksForGeeks Mesh Definition

²⁹TechTarget Mesh Usages

³⁰wedesignvirtual What are Meshes

³¹CGIFurniture What are Polygons

³²Adobe 3D Polygon Modeling

³³CGCookie Ways to optimize Game Assets

³⁴All3DP How to reduce Polygons

In der Computergrafik steht die Anzahl der Polygone eines Modells in direktem Zusammenhang mit dem benötigten Speicherplatz und der Rechenleistung. Je mehr Polygone ein Modell hat, desto mehr Daten müssen verarbeitet und gerendert werden, was zu höheren Anforderungen an die Hardware führt. Durch eine Reduzierung der Polygone können diese Ressourcen effizienter genutzt werden, ohne dass die visuelle Qualität des Modells wesentlich beeinträchtigt wird.

Werkzeuge wie der *Decimate Modifier* in Blender bieten die Möglichkeit, die Anzahl der Polygone automatisch zu reduzieren und gleichzeitig visuelle Artefakte zu minimieren. Artefakte in Bezug auf 3D-Modellierung sind unerwünschte visuelle Effekte oder Fehler, die während des Modellierungs- oder Renderingprozesses auftreten können.³⁵

Dennoch ist es ratsam, bereits während des Modellierungsprozesses darauf zu achten, keine unnötigen zusätzlichen Unterteilungen zu erzeugen, um eine optimale Ausgangsbasis für die Polygonreduktion zu schaffen. Eine Möglichkeit hierfür ist die regelmäßige Nutzung der *Merge-Funktion*, welche es ermöglicht, einzelne Punkte anhand ihres Abstandes zu kombinieren und einen einzelnen Punkt daraus zu erstellen.³⁶ Wenn dieser Vorgang auf das gesamte Modell angewendet wird, verhindert er außerdem ungewollte Kopien von Eckpunkten durch beispielsweise unbeabsichtigte und abgebrochene *Extrusion*.

Die Polygonreduktion ist ein wichtiger Bestandteil der 3D-Modellierung und -Optimierung, da sie dazu beiträgt, die Leistung von Anwendungen zu verbessern und die Benutzererfahrung zu optimieren. Durch eine sorgfältige Anwendung dieser Technik können qualitativ hochwertige 3D-Modelle erstellt werden, die sowohl ästhetisch ansprechend als auch effizient zu verarbeiten sind.

4.2.4.2 Texturenoptimierung

Die Optimierung von Texturen ist ein wesentlicher Bestandteil der Gestaltung von 3D-Modellen für Augmented Reality (AR)-Anwendungen, da sie einen direkten Einfluss auf die Benutzererfahrung haben. Bei der Auswahl geeigneter Texturen muss der Auflösung besondere Aufmerksamkeit geschenkt werden, insbesondere im Hinblick auf die begrenzte Leistungsfähigkeit von AR-Geräten wie der HoloLens 2.

Während des Texturierungsprozesses wurde stark auf die Leistung der HoloLens 2 und die Bilder pro Sekunde geachtet, um zu verhindern, dass zu hochauflösende Texturen, das Benutzererlebnis beeinträchtigen. Wurden Einbußen festgestellt, wurden entsprechende Anpassungen an den Texturen vorgenommen. Dies beinhaltete entweder die Suche nach alternativen Texturen oder die Komprimierung der vorhandenen Texturen, um eine geringere Auflösung zu erreichen.

Die Texturoptimierung ist ein iterativer Prozess, der eine ausgewogene Berücksichtigung der visuellen Qualität und der Leistungsfähigkeit der AR-Plattform erfordert. Durch die gezielte Optimierung von Texturen können AR-Anwendungen erstellt werden, die eine ansprechende visuelle Darstellung bieten und gleichzeitig ein flüssiges und immersives Benutzererlebnis gewährleisten.

4.2.5 Export- und Integrationsprozess

In diesem Abschnitt wird der Export- und Integrationsprozess für 3D-Modelle beschrieben, beginnend mit der Auswahl des geeigneten Dateiformats und der Berücksichtigung des Koordinatensystems.

³⁵Blender Dokumentation **Decimate Modifier**

³⁶Blender Dokumentation **Merge Funktion**

Der Export- und Integrationsprozess ist ein entscheidender Schritt bei der Übertragung von 3D-Modellen aus der Konstruktions- oder Modellierungssoftware in eine Zielumgebung, sei es eine Spiele-Engine, eine Virtual-Reality-Plattform oder eine AR-Anwendung. Dieser Prozess umfasst mehrere Schritte, um sicherzustellen, dass das Modell korrekt dargestellt und funktional in die Zielumgebung integriert wird.³⁷

4.2.5.1 Dateiformat

Als Dateiformat für den Export von 3D-Modellen in diesem Projekt wurde das von Autodesk entwickelte Filmbox-Format (FBX) gewählt. FBX wurde aufgrund seiner weit verbreiteten Unterstützung und seiner Fähigkeit, umfassende Informationen über geometrische Formen, Materialien, Animationen und andere Szenendaten zu speichern, ausgewählt.³⁸

FBX ist ein proprietäres Format, das speziell für den Austausch von 3D-Inhalten zwischen verschiedenen Anwendungen entwickelt wurde. Es bietet eine hierarchische Struktur, die es ermöglicht, komplexe Modelle (zum Beispiel von Blender) zu organisieren und zu übertragen. Diese Struktur umfasst Knoten, die verschiedene Elemente der 3D-Szene repräsentieren, wie Geometrie, Materialien, Animationen, Kamerassen und Lichtquellen.³⁹

Durch die Verwendung von FBX können 3D-Modelle nahtlos zwischen verschiedenen Softwareanwendungen und Plattformen ausgetauscht werden, was die Zusammenarbeit und Integration in Projekten wie diesem, das Unity als Engine verwendet, erleichtert. Die Wahl des FBX-Formats bietet somit eine solide Grundlage für einen effizienten Workflow und eine erfolgreiche Umsetzung des Projekts.

4.2.5.2 Koordinatensysteme

Die Berücksichtigung und korrekte Handhabung von dem Koordinatensystem während des Modellierungsprozesses ist ein entscheidender Aspekt für die nahtlose Integration von 3D-Modellen in verschiedene Anwendungen. In diesem Projekt wurden spezifische Maßnahmen ergriffen, um potenzielle Probleme im Zusammenhang mit Koordinatensystemen zu minimieren.

Während der Modellierung wurden alle Objekte im Koordinatenursprung platziert, um sicherzustellen, dass beim Export der Modelle nach Unity keine Komplikationen hinsichtlich Platzierung und Ausrichtung auftreten. Diese Vorgehensweise trägt dazu bei, mögliche Diskrepanzen zwischen den Koordinatensystemen der Modellierungssoftware und der Zielpflicht zu vermeiden, was den Integrationsprozess vereinfacht und beschleunigt.

Darüber hinaus wurden alle Modelle in der gleichen Ausrichtung modelliert, um zusätzliche Anpassungen in Unity zu vermeiden. Diese konsistente Orientierung stellt sicher, dass alle Modelle bereits in einer standardisierten Orientierung vorliegen, was die Notwendigkeit weiterer manueller Eingriffe minimiert und einen reibungsloseren Arbeitsablauf gewährleistet.

Für den Fall, dass Modelle dennoch mit einer falschen Rotation exportiert werden, wurden entsprechende Korrekturen direkt im Unity Inspector vorgenommen. Diese Nachjustierung ermöglicht es, eventuelle Fehler in der Ausrichtung der Modelle schnell und effizient zu beheben, ohne den Modellierungsprozess zu unterbrechen oder zusätzlichen Aufwand zu verursachen.

³⁷Blog CG Wire **Getting started with export in Blender**

³⁸GameDev-Profi Welches Dateiformat

³⁹Vection Technologies **Everything You Need to Know About FBX Files**

Insgesamt zeigt die Berücksichtigung von dem ausgewählten Koordinatensystem während des gesamten Workflows einen proaktiven Ansatz bei der Modellierung und Integration von 3D-Objekten. Die Umsetzung dieser Maßnahmen wird die Konsistenz und Effizienz des Projekts verbessern und gleichzeitig mögliche Komplikationen im Zusammenhang mit Koordinatensystemen effektiv vermeiden.

4.2.6 Add-Ons und Plugins

Während der Modellierungsphase wurden verschiedene Add-ons und Plug-ins genutzt, um die Modellierung effizienter zu gestalten und die Funktionalität von Blender zu erweitern. Im folgenden Abschnitt werden die wichtigsten Add-ons und Plug-ins erläutert, die zur Unterstützung und Verbesserung der Modellierungserfahrung eingesetzt wurden.

4.2.6.1 Looptools: Optimierung von Topologie und Oberflächen

Das Add-on *Looptools*⁴⁰ für Blender stellt eine wichtige Ergänzung für die Flächenmodellierung und Topologieoptimierung dar. Insbesondere bei der Umwandlung von rechteckigen oder quadratischen Flächen in kreisförmige Flächen erweist sich dieses Werkzeug als äußerst nützlich. Im *Stromverteiler-Modell* müssen beispielsweise die einzelnen runden Steckbuchsen aus der eckigen Basis heraus modelliert werden.

Als Hauptwerkzeug wurde das so genannte *Circle*-Werkzeug verwendet, das speziell zur Lösung des oben genannten Problems entwickelt wurde. Mit dem *Circle*-Werkzeug können rechteckige Flächen unter Berücksichtigung der Topologie und der Anzahl der Polygone nahtlos in kreisförmige Flächen umgewandelt werden.

Das *Circle*-Werkzeug bietet verschiedene Einstellungsmöglichkeiten, um die extrahierte Figur optimal für den jeweiligen Zweck zu konfigurieren. Die wichtigsten Konfigurationspunkte sind

- **Best Fit:** Dieses Werkzeug berechnet einen Kreis mit Hilfe einer nichtlinearen Methode der kleinsten Quadrate. Dadurch wird sichergestellt, dass der berechnete Kreis optimal zu den ausgewählten Eckpunkten passt.
- **Fit Inside:** Mit dieser Option wird der Kreis so berechnet, dass kein Eckpunkt vom Kreismittelpunkt entfernt wird. Dies ist besonders nützlich, wenn die Topologie des umgebenden Mesh erhalten bleiben soll.

Zusätzlich ermöglicht das *Circle*-Werkzeug die Anpassung weiterer Konfigurationsparameter wie Radius und Regularität, um die extrahierte Figur feiner abzustimmen und besser an individuelle Anforderungen anzupassen.⁴¹

Insgesamt trägt das *Looptools* Add-On dazu bei, den Modellierungsprozess effizienter zu gestalten und die Qualität der resultierenden Modelle zu verbessern, indem es eine Reihe präziser und flexibler Werkzeuge für die Topologie- und Flächenoptimierung bereitstellt.

Dieses Add-on wurde konkret bei den folgenden Modellen eingesetzt:

- **Handy-Modell:** Für die Modellierung der einzelnen Kameras sowie der runden Ein- und Ausgänge
- **Stromverteiler-Modell:** Zur Einarbeitung der einzelnen Steckbuchsen wurde bereits zu Beginn erwähnt.
- **USB-Stick-Modell:** Um den Übergang von der Halterung zum eigentlichen USB-Stick zu modellieren.

⁴⁰Blender **LoopTools**

⁴¹Blender **Circle**-Werkzeug

4.2.6.2 Images as Planes: Effiziente Integration von Texturen

Das Add-on *Images as Planes*⁴² spielt eine entscheidende Rolle in der Modellierungspraxis, insbesondere im Bereich der realistischen Modellierung.

Das Hauptanwendungsgebiet dieses Add-ons in dem Projekt liegt in der Möglichkeit, Vorschaubilder für die Modellierung hinter Objekten zu platzieren, um eine realistischere und präzisere Modellierung zu ermöglichen. Diese Funktion bietet die Möglichkeit, reale Bilder als Referenz in Blender-Szenen zu integrieren und als Hintergrund für die Modellierung zu verwenden. Durch die direkte Integration von Bildern in die Arbeitsumgebung können feine Details und Proportionen besser beurteilt und reproduziert werden, was zu einer verbesserten Qualität der Modelle führt.

Die Verwendung von *Images as Planes* trägt somit wesentlich zur Erhöhung der Genauigkeit und Realitätsnähe der Modellierung bei, indem sie eine effiziente Möglichkeit bietet, reale Referenzen in den Modellierungsprozess zu integrieren. Diese Funktion ist besonders nützlich bei der Modellierung von Objekten, die auf realen Vorbildern basieren, da sie es dem Modellierer ermöglicht, direkt aus Bildern heraus zu arbeiten und so einen höheren Detaillierungsgrad zu erreichen.

Dieses Add-on wurde in allen modellierten Objekten des Projekts eingesetzt, um anhand von Referenzbildern anschaulichere Modelle zu entwerfen.

4.2.7 Modi

In Blender stehen verschiedene Modi zur Verfügung, mit denen unterschiedliche Aspekte eines Objekts bearbeitet und manipuliert werden können. Diese verschiedenen Modi in Blender bieten eine umfassende Palette an Werkzeugen und Funktionen, mit denen die Benutzer ihre 3D-Modelle auf vielfältige Weise bearbeiten und verfeinern können.⁴³

4.2.7.1 Object-Modus

Der Object-Modus ist der Standardmodus in Blender und steht für alle Arten von Objekten zur Verfügung. In diesem Modus können grundlegende Transformationen wie Positionierung, Rotation und Skalierung sowie Duplizierung und andere Objekteigenschaften bearbeitet werden.

4.2.7.2 Edit-Modus

Der Edit-Modus ist ein spezialisierter Modus zum Bearbeiten der Form eines Objekts. Hier können mithilfe verschiedener Werkzeuge und Kontrollpunkte einzelne Knoten/(Eck-)Punkte, Kanten und Flächen des Objekts bearbeitet werden. Dies ermöglicht detaillierte Manipulationen und Anpassungen an der Geometrie des Objekts, was insbesondere für die Modellierung von entscheidender Bedeutung ist.

4.2.7.3 Texture-Paint-Modus

Der Texture-Paint-Modus ist ein spezieller Modus, der es ermöglicht, Texturen direkt auf das Mesh eines Objekts zu malen. Dieser Modus ist ausschließlich auf das Bearbeiten von Meshes beschränkt und bietet eine intuitive Möglichkeit, Texturen im 3D-Viewport zu zeichnen und zu bearbeiten. Durch die direkte Malerei auf dem Objekt können komplexe Texturen und Oberflächeneffekte einfach erstellt und angepasst werden.

⁴²Blender **Images as Planes**

⁴³Blender **Modi**

4.2.8 Hierarchie

In der 3D-Modellierung bezieht sich Hierarchie auf die strukturierte Organisation von Objekten innerhalb einer Szene oder eines Modells. Diese Hierarchie wird oft durch eine Baumstruktur dargestellt, in der übergeordnete Objekte untergeordnete Objekte enthalten können. Die Hierarchie spielt eine entscheidende Rolle bei der Verwaltung und Manipulation von Objekten sowie bei der Definition von Beziehungen zwischen ihnen.

- **Eltern-Kind-Beziehungen:** Übergeordnete Objekte werden oft als Eltern bezeichnet und enthalten untergeordnete Objekte, die als Kinder bezeichnet werden. Diese Beziehung ermöglicht es, Transformationen wie Verschieben, Drehen und Skalieren auf das übergeordnete Objekt anzuwenden, welche dann auf seine untergeordneten Objekte übertragen werden. Diese Technik ist besonders nützlich für komplexe Strukturen wie Roboterglieder oder hierarchische Modelle.⁴⁴
- **Gruppierung:** Objekte können hierarchisch gruppiert werden, um sie logisch zu organisieren und ihre Handhabung zu erleichtern. Durch die Gruppierung ist es möglich, mehrere Objekte gleichzeitig auszuwählen, zu verschieben oder zu bearbeiten, ohne jedes einzelne Objekt separat manipulieren zu müssen.⁴⁵

Die Hierarchie ist ein grundlegendes Konzept in der 3D-Modellierung. Sie ermöglicht eine effiziente Organisation und Manipulation von Objekten. Durch die kluge Nutzung von Hierarchien können komplexe Modelle erstellt und verwaltet werden. Dies führt zu einer effizienteren Arbeitsweise und einer verbesserten Qualität der Ergebnisse.

Im Projekt wurde dies beispielsweise verwendet, um im Taschenrechner-Modell die vielen Buttons von der Basisform logisch abzutrennen oder beim Handy-Modell die Basisform von den einzelnen Kameras und der Kameraauflagefläche abzutrennen.

4.2.9 Modifier

Modifier sind Werkzeuge oder Operationen, die auf Objekte in der 3D-Modellierung angewendet werden, um ihr Aussehen oder Verhalten zu verändern, ohne die zugrunde liegende Geometrie dauerhaft zu ändern. Sie ermöglichen es den Modellierenden, komplexe Effekte zu erzielen, ohne manuell jeden einzelnen Aspekt des Modells zu bearbeiten. Modifier sind ein wichtiger Bestandteil vieler 3D-Modellierungssoftware und bieten eine Vielzahl von Funktionen zur Verbesserung des Modellierungsprozesses.⁴⁶ Die Verwendung von Modifern bringt mehrere Vorteile mit sich:

- **Non-destructive Bearbeitung:** Modifier werden auf das Modell angewendet, ohne die ursprüngliche Geometrie zu verändern. Dies ermöglicht es, Änderungen vorzunehmen und bei Bedarf zum ursprünglichen Zustand zurückzukehren.⁴⁷ Als gutes Beispiel kann hier der *Subdivision*-Modifier herangezogen werden, welcher das gesamte Modell grundlegend einmal zerteilt, sodass man doppelt soviele Unterteilungen und feinere Details hat, bei Bedarf kann man die Zerteilung ausmachen oder noch mehr zerteilen. Wenn man händisch ohne Modifier alles subdividen würde, wäre es nicht möglich/ sehr schwer möglich das rückgängig zu machen
- **Effizienzsteigerung:** Durch die Verwendung von Modifern können komplexe Effekte und Veränderungen mit weniger manuellem Aufwand erreicht werden. Dies führt zu einem effizienteren Modellierungsprozess und spart Zeit und Ressourcen.

⁴⁴Blender Parents

⁴⁵Blender Groups

⁴⁶Blender Modifier

⁴⁷Blender Non-Destructive Modifier

Wenn man das ganze Modell abrunden (*Bevel-Modifier*) oder feiner/detailreicher (*Subdivision-Modifier*) gestalten möchte ist es deutlich zeitintensiver alle kanten und flächen einzeln händisch zu bearbeiten.

- **Experimentierfreude:** Da Modifier nicht-destruktiv sind, können sie leicht hinzugefügt, angepasst oder entfernt werden. Dies ermutigt dazu, verschiedene Optionen auszuprobieren und kreativ zu experimentieren, ohne Angst vor irreversiblen Änderungen haben zu müssen.

Beispiele für Modifier sind Subdivision Surface, Mirror, Bevel und Array (nähtere Erklärung zu diesem Modifier im Abschnitt 4.2.10.1). Jeder Modifier hat spezifische Anwendungsfälle und ermöglicht es den Modellierenden, eine Vielzahl von Effekten zu erzielen, von der Glättung von Kanten bis zur Erstellung von komplexen Wiederholungsmustern.

Modifier sind ein wichtiger Bestandteil der 3D-Modellierung. Sie bieten eine flexible und leistungsstarke Möglichkeit, Objekte zu bearbeiten und zu verbessern, ohne die Integrität des Modells zu beeinträchtigen. Die Verwendung von Modifizieren kann den Modellierungsprozess rationalisieren und die Kreativität der Modellierenden fördern.

Subdivision Surface Modifier

Dieser Modifier glättet die Oberfläche eines Modells, indem er die Anzahl der Polygon-subdivisionen erhöht. Dadurch entstehen weichere Kanten und eine insgesamt glattere Oberfläche. Der Subdivision Surface Modifier wird oft verwendet, um organische Formen zu erstellen oder die Details eines Modells zu verfeinern. Er kann auch helfen, die Qualität von Meshes zu verbessern, indem er unregelmäßige Geometrien ausgleicht und sie besser für das Rendern und die Animation geeignet macht.⁴⁸

Mirror Modifier

Der Mirror Modifier spiegelt die Geometrie eines Objekts entlang einer oder mehrerer Achsen. Dadurch können symmetrische Modelle erstellt werden, indem nur eine Hälfte modelliert werden muss. Der Modifier erzeugt automatisch die gespiegelte Seite des Objekts, wodurch Zeit und Aufwand gespart werden. Der Mirror Modifier wird häufig bei der Modellierung von symmetrischen Objekten wie Menschen, Fahrzeugen und Gebäuden eingesetzt. Im Projekt wurde er beispielsweise für das Modell des Taschenrechners eingesetzt, um die Basisform gleichmäßig auf beiden Seiten zu formen.⁴⁹

Bevel Modifier

Der Bevel Modifier fügt abgerundete Kanten zu den Ecken eines Objekts hinzu. Er erzeugt eine Fase oder eine abgeschrägte Kante, indem er die Kanten des Modells verändert und ihnen eine gewisse Breite verleiht. Dies hilft, harte Kanten zu glätten und das Aussehen des Modells zu verbessern. Der Bevel Modifier wird oft verwendet, um realistischere und ansprechendere Oberflächen zu erzeugen, insbesondere bei der Modellierung von mechanischen Teilen, Möbeln und Architektur.⁵⁰

4.2.10 Modellierung von Gegenständen für das Projekt

Dieser Abschnitt beschreibt die für das Projekt in Blender modellierten Gegenstände sowie die dabei aufgetretenen Herausforderungen.

⁴⁸Blender Subdivision Surface Modifier

⁴⁹Blender Mirror Modifier

⁵⁰Blender Bevel Modifier

Zu Beginn des Projekts musste eine wichtige Entscheidung getroffen werden, nämlich wie viele Objekte modelliert werden sollten. Es war wichtig, eine ausgewogene Anzahl zu wählen, die das Spiel nicht überladen, aber dennoch eine Herausforderung für die Spieler darstellen soll. Nach eingehenden Recherchen und internen Abstimmungen einigte sich das Team auf 11 Gegenstände. Diese sollten alltägliche Gegenstände eines Schülers der HTBLuVA darstellen und den Spielern einen Einblick in den Schulalltag bieten.

4.2.10.1 Taschenrechner

Um den Modellierungsprozess am Beispiel des Taschenrechners zu veranschaulichen, werden spezifische Schritte und Überlegungen während des Prozesses erläutert. Der Prozess begann mit der Idee, den Taschenrechner als Teil des Projekts zu integrieren. Anschließend wurde der Modellierungsumfang und die Art des Taschenrechners festgelegt. Dabei diente das Casio FX-991 ESPLUS Modell als Vorlage für das zu erstellende Modell. Als Referenzbild wurde nach kurzer Recherche ein geeignetes Bild mit hoher Auflösung aus dem Internet gefunden und ausgewählt.

Ausgangslage

Im Modellierungsprogramm Blender war das Standardprojekt als Startpunkt geeignet. Es wurde bereits geringfügig angepasst. Die Standardkamera und Lichtquelle, die für das Rendering innerhalb von Blender verwendet werden, wurden entfernt, da sie für das finale Projekt in Unity nicht benötigt werden und unnötige Komplikationen verursachen könnten. Je nach Anforderung des Modells wird eine geeignete Grundform wie beispielsweise ein Würfel, eine Fläche oder ein Zylinder erstellt, um darauf aufbauend mit der eigentlichen Modellierung zu beginnen.

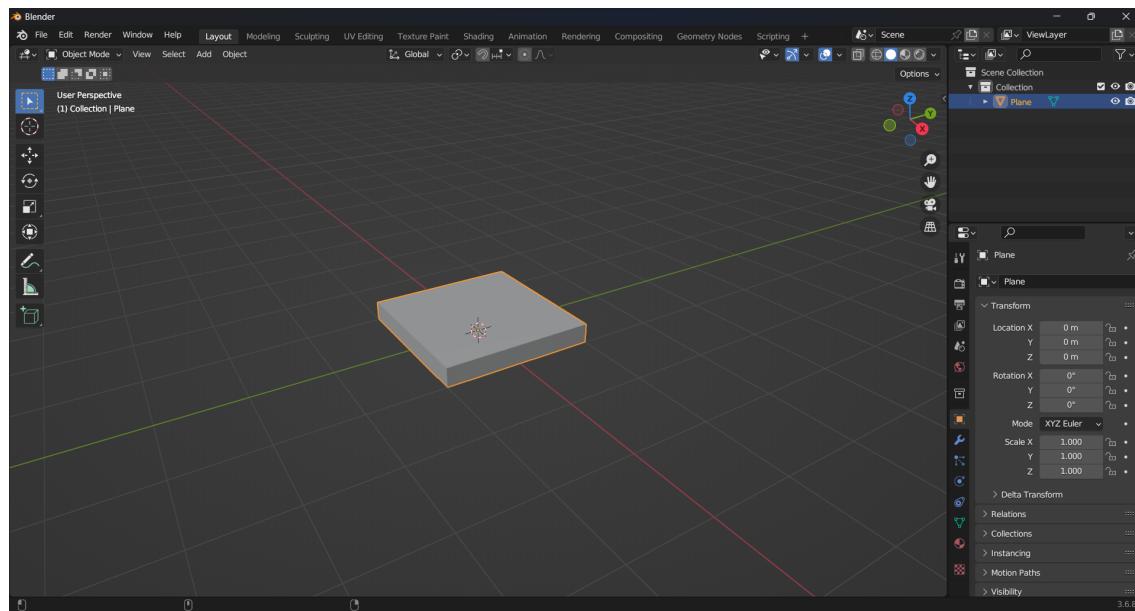


Abbildung 4.5: Ausgangslage innerhalb von Blender

Durch diesen Prozess der Initiierung wurde die Grundlage für die folgenden Schritte der Modellierung gelegt, die in weiteren Abschnitten detaillierter beschrieben werden.

Erste Schritte nach der Erstellung

Nach Auswahl eines geeigneten Referenzbildes für das Taschenrechnermodell wurde dieses mithilfe des Add-Ons *Images as Planes* in Blender eingefügt. Das Bild dient als Leitfaden für die Modellierung und wurde unterhalb des bereits vorhandenen Objekts platziert. Um eventuelle Unregelmäßigkeiten im Design zu vermeiden, wurde der Mirror Modifier aufgrund der symmetrischen Form des Taschenrechners angewendet. Durch diese Maßnahme werden alle Modellierungsaktionen, die auf der linken Seite durchgeführt werden, automatisch auf die rechte Seite gespiegelt. Dadurch wird die Effizienz und Genauigkeit des Modellierungsprozesses erhöht.

Extrahieren als Modellierungswerkzeug

Extrahieren⁵¹ ist ein Werkzeug im Edit-Modus von Blender. Es dient dazu, ausgewählte Punkte zu duplizieren und zu verschieben, während die ursprünglichen Punkte der Bearbeitungslinie erhalten bleiben. Für die Verwendung des Extrahieren-Werkzeugs ist ein Verständnis der grundlegenden Konzepte des Edit-Modus in Blender sowie eine präzise Auswahl der zu extrahierenden Punkte erforderlich, um die gewünschten Modifikationen am Modell vorzunehmen. Während des Modellierungsprozesses ist es von entscheidender Bedeutung, unbeabsichtigte Extraktionen zu vermeiden, da diese dazu führen können, dass duplizierte Vertices über bereits vorhandenen liegen. Dies kann nicht nur die weitere Bearbeitung des Modells beeinträchtigen, sondern auch zu einer unnötigen Zunahme der Anzahl an Eckpunkten führen.

Weiterführung beim Taschenrechnermodell

Im Anschluss begann die Modellierung durch Extrudieren eines Eckpunkts der Fläche, um grob der Form des Referenzbildes zu folgen.

⁵¹Blender Extrude



Abbildung 4.6: Taschenrechner mit Buttons in Blender

Nach Abschluss dieser groben Modellierungsschritte entstand das Grundgerüst des Taschenrechnermodells, wie in Abbildung 4.6 dargestellt. Die verschiedenen Eckpunkte (Vertices) werden im Edit-Modus von Blender als schwarze Punkte dargestellt. Durch den Mirror Modifier wird die Bearbeitung auf der linken Seite des Modells automatisch auf die rechte Seite gespiegelt, was eine symmetrische Form gewährleistet. Um dem Modell mehr Tiefe zu verleihen, wurde Extrudieren verwendet, um aus der ebenen Fläche eine dreidimensionale Struktur zu formen.

Als nächster Schritt steht die Modellierung der Buttons und anderer Funktionen des Taschenrechners an, um das Modell weiter zu verfeinern und seinem realen Gegenstück näherzukommen. Dazu wird der *Array-Modifier* verwendet, da dieser es ermöglicht, ein Button-Modell zu entwerfen und dieses dann zu vervielfältigen. Dadurch wird die Modellierungszeit enorm reduziert und die gleichmäßige Platzierung der Buttons verbessert.

Einschub Array-Modifier

Der Array-Modifier in Blender ermöglicht die Erstellung einer Reihe von Kopien eines Basisobjekts. Jede Kopie wird dabei auf eine vordefinierte Weise von der vorherigen Kopie versetzt. Der Modifier bietet eine Vielzahl von Optionen zur Anpassung der Positionierung und Ausrichtung der Kopien. Die grundlegenden Funktionen des Array-Modifiers umfassen die Möglichkeit, Eckpunkte in benachbarten Kopien zusammenzuführen, insbesondere wenn sie nahe beieinander liegen. Dadurch können nahtlose und konsistente Verbindungen zwischen den einzelnen Kopien hergestellt werden. Eine wichtige Option des Array-Modifiers ist der sogenannte *Fit Type*, der angibt, wie das Objekt dupliziert werden soll. Dies kann entlang einer Kurve, mit einer passenden Länge oder einer festgelegten Anzahl von Kopien erfolgen. Darüber hinaus kann eine relative oder feste Verschiebung entlang der x-, y- oder z-Achse definiert werden, um die Positionierung der Kopien weiter anzupassen.

Weiterführung beim Taschenrechnermodell

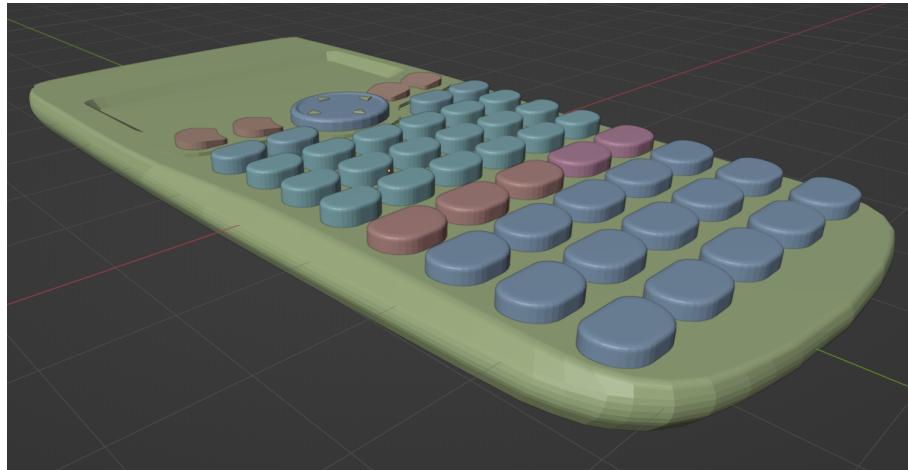


Abbildung 4.7: Taschenrechner mit Buttons in Blender

Abbildung 4.7 zeigt das Taschenrechnermodell in Blender mit den neu modellierten Buttons. Die dargestellten Farben dienen lediglich als visuelle Hilfestellung im Blender-Editor und stellen keine inhärenten Eigenschaften des Modells dar. Jedes neu erstellte Objekt wird automatisch mit einer zufälligen Farbe versehen, um eine einfachere Unterscheidung innerhalb der Szene zu ermöglichen.

Einige der Buttons im Modell wurden mithilfe des Array-Modifiers vervielfältigt. Dies gilt insbesondere für Buttons, die im Referenzbild die gleiche Größe, Funktion und Farbe aufweisen. Durch diese Technik ist es möglich, wiederkehrende Elemente effizient zu modellieren, indem eine einzelne Vorlage kopiert und entsprechend der gewünschten Anordnung angepasst wird.

Der aktuelle Stand des Modells lässt darauf schließen, dass die Modellierung größtenteils abgeschlossen ist. Der nächste Schritt besteht darin, dem Modell Farben oder Texturen hinzuzufügen, um ein realistischeres Aussehen zu erzielen. Dies kann durch Anwendung von Materialien und Texturen im Renderprozess erreicht werden, um dem Modell visuelle Tiefe und Detailtreue zu verleihen.

Die Fortführung des Modellierungsprozesses umfasst somit die Umsetzung der texturierten Oberfläche sowie mögliche Feinanpassungen, um das Modell weiter zu verbessern und an die Referenzvorlage anzupassen.

Hierarchie des fertiggestellten Modells

In Blender wird die Hierarchie eines Modells als Baumstruktur dargestellt. Dabei ermöglichen verschiedene Unterteilungen und Kategorien eine übersichtliche Organisation. Die Struktur wird in Abbildung 4.9 veranschaulicht. Das weiße Box-Symbol steht für eine Collection (Ordner).

⁵²Blender **Array Modifier**

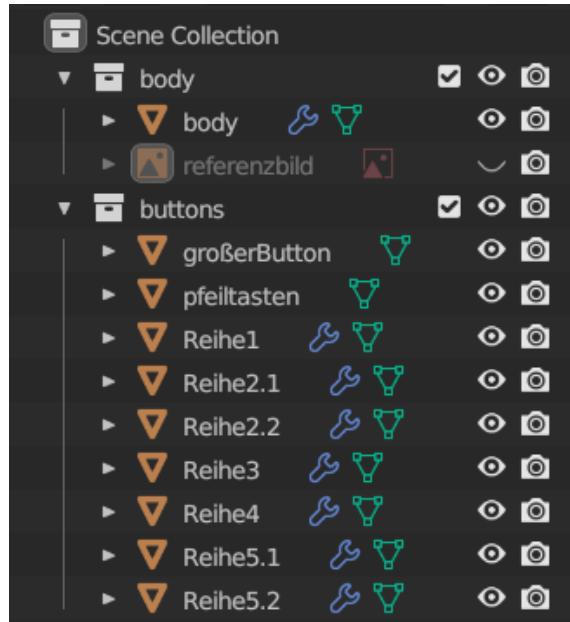


Abbildung 4.8: Hierarchie des fertigen Taschenrechnermodells

- Die **Scene Collection** ist die oberste Ebene, die das gesamte Modell enthält.
- Die **Body Collection** enthält das Hauptmodell des Taschenrechners, also die grundlegende Form.
- Die **Buttons Collection** umfasst alle Buttons, die auf dem Taschenrechner abgebildet sind.
- Die untergeordnete Struktur mit den orangefarbenen Dreiecken repräsentiert die eigentlichen Objekte innerhalb der jeweiligen Collections.
- Das **Referenzbild** ist ein importiertes Bild, das beispielsweise als Vorlage für das Modellieren verwendet wird.

In Abbildung 4.9 ist zu erkennen, dass ein Objekt ausgegraut ist. Dies bedeutet in dem Fall, dass das Objekt ausgeblendet wurde, da das Hauptmodell bereits fertiggestellt wurde und das Referenzbild nicht mehr benötigt wird. Das Referenzbild kann jedoch bei Bedarf aktiviert werden, beispielsweise zu Hilfe- oder Veranschaulichungszwecken.

Die hierarchische Darstellung der Elemente erleichtert die Organisation und Bearbeitung des Modells. Eine klare Strukturierung der verschiedenen Komponenten ermöglicht eine effiziente Arbeitsweise und hilft, den Überblick über die einzelnen Teile zu behalten.

Textur und Farbe des Taschenrechnermodells

Für die Texturierung und Farbgebung des Modells diente das Referenzbild zur Orientierung. Das *Eye-Dropper*-Werkzeug in Blender ermöglichte es, Farben gezielt aus dem Referenzbild auszuwählen und auf das entsprechende Modell anzuwenden. Durch Klicken auf einen bestimmten Bereich des Referenzbildes wurde die Farbe erfasst und dann auf den entsprechenden Bereich des Modells übertragen.⁵³

Dieser Prozess wurde für jedes Element des Modells durchgeführt, um eine genaue Anpassung an die Farben und Texturen des Referenzbildes zu erreichen. Die Verwendung des Eye-Dropper-Werkzeugs ermöglichte eine präzise und konsistente Umsetzung der Farbgebung und trug dazu bei, dass das Modell möglichst authentisch und realitätsnah aussieht.

⁵³Blender Dokumentation **Eyedropper Werkzeug**

Um noch bessere und realitätsnähere Textur zu gewährleisten, wurde ein Vorgang Namens *UV-Mapping* durchgeführt.

UV-Mapping ist ein wichtiger Schritt im Texturierungsprozess von 3D-Modellen. Dabei werden Texturen auf dreidimensionale Objekte projiziert. Die Bezeichnungen *U* und *V* stehen dabei für die beiden Koordinatenachsen im zweidimensionalen Raum. In Blender gibt es dafür einen eigenen Editor mit einer spezifischen Benutzeroberfläche.

Während der Modellierung des Taschenrechnermodells wurde *UV-Mapping* genutzt, um Texturen detailgetreu von einem Referenzbild auf die einzelnen Buttons und Tasten des Modells zu übertragen. Dieser Prozess ermöglicht es, die Beschriftungen und andere Details automatisch von der Textur abzuleiten, anstatt sie manuell für jedes Element erstellen zu müssen.

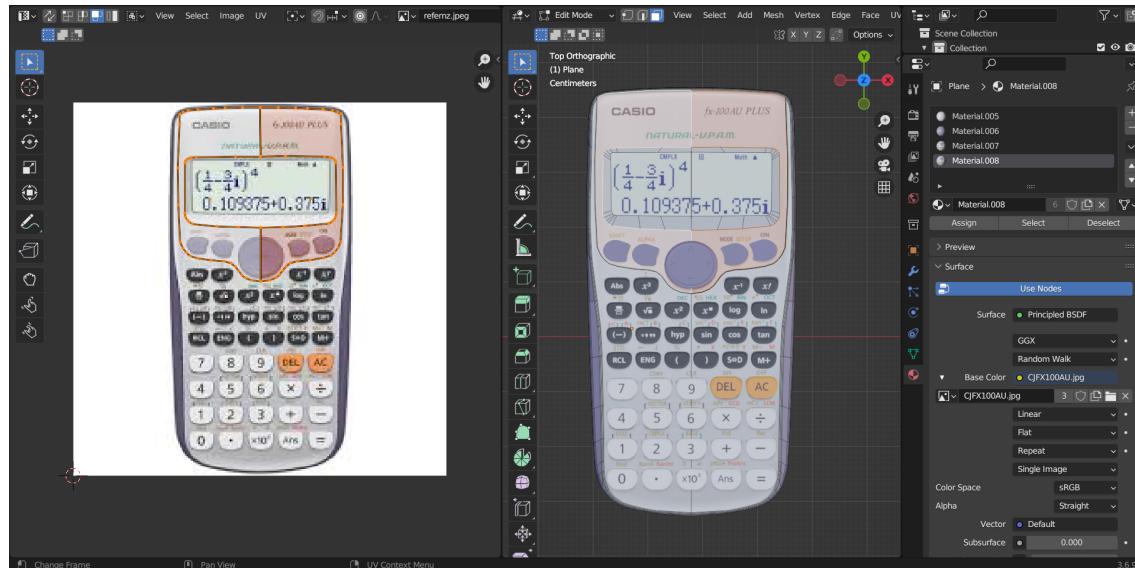


Abbildung 4.9: Ansicht des UV-Editing Editors in Blender

Die Abbildung zeigt den *UV-Editing-Editor* in Blender. Mit diesem Editor können die UV-Maps der 3D-Modelle bearbeitet werden, indem die 3D-Geometrie auf eine zweidimensionale Ebene projiziert wird, um die Positionierung der Texturen zu steuern.⁵⁴

Als Beispiel wurde der obere Teil des Taschenrechners rund um den Bildschirm ausgewählt. Zunächst wird dem Bereich ein neues Material zugewiesen, das die Basisfarbe des Referenzbildes enthält. Anschließend wird das Referenzbild als Texturhintergrund im Editor ausgewählt, um es anzuzeigen.

Der nächste Schritt besteht darin, die ausgewählten Flächen des Modells aufzuteilen, denn die Erstauswahl in Blender legt die einzelnen Flächen übereinander. Dies geschieht durch Auswahl der Flächen im Editor und Anwendung der *UV-Unwrap*-Funktion in Blender.

Nach dem *UV-Unwrappen* werden die Flächen im Editor positioniert, um die Texturen und Beschriftungen korrekt auf das Modell zu projizieren, ohne dass sie abgeschnitten oder verzerrt wirken. Dieser Prozess ist zeitaufwendig, da er für jedes einzelne Element des Modells wiederholt werden muss. Dabei muss auch auf korrekte Skalierungen geachtet werden, um eine einheitliche Darstellung zu gewährleisten.⁵⁵

UV-Mapping ist somit ein entscheidender Schritt bei der Erstellung realistischer 3D-

⁵⁴Blender Dokumentation **UV-Editing Editor**

⁵⁵all3dp **Blender UV Mapping Simply Explained**

Modelle, da es eine präzise Texturierung und Gestaltung ermöglicht.

4.2.10.2 Restliche Modelle



Ping-Paket-Modell

Das Ping-Paket-Modell wurde speziell für didaktische Zwecke auf das erste Anwendungsszenario konzipiert, um den Benutzern das grundlegende Konzept eines Daten-Pings zu veranschaulichen. Es dient dazu, die Übertragung von Daten zwischen zwei Endpunkten, beispielsweise Laptops, zu visualisieren. Das Modell besteht im Wesentlichen aus einem simplen Quader, ergänzt durch Details wie Paketklebeband, um eine realistische Darstellung zu erreichen. Bei der Texturierung wurde ein Amazon-Paket als Referenz verwendet.

Laptop-Modell

Der Laptop stellt einen unverzichtbaren Gegenstand im zweiten Anwendungsszenario dar und ist ein essentielles Werkzeug für Schüler ab der 3. Klasse. Bei der Modellierung wurden zusätzliche Details wie USB-Ports und der Laptopständer berücksichtigt. Um den Aufwand zu minimieren, wurde der Laptop im geschlossenen Zustand modelliert, wodurch die Komplexität von Tastatur und Bildschirm vermieden wurde. Die Referenz für das Modell lieferte ein reales Laptop-Exemplar eines Projektmitglieds.

Router-Modell

Die Modellierung des Routers war eine iterative Aufgabe, die mehrere Versionen erforderte, um ein realistisches und ansprechendes Modell zu erzielen. Das finale Modell repräsentiert einen Gaming-Router mit vielen Anschlüssen und Bedienelementen. Aufgrund der Vielzahl an Details war die Modellierung sehr anspruchsvoll und zeitaufwendig.

Maus-Modell

Die Modellierung der Maus war eine Standardaufgabe, bei der der Mirror Modifier effektiv eingesetzt wurde, um die symmetrische Natur des Objekts zu betonen. Eine einfache Büromaus diente als Referenz für das Modell.



Notizblock-Modell

Der Block ist ein wichtiger Gegenstand im schulischen Umfeld bis zur 3. Klasse, wenn noch keine Laptops verwendet werden. Während der Designphase ermöglichte der Array Modifier eine einfache Gestaltung der spiralförmigen Halterung der Blätter.

Stift-Modell

Der Stift wurde einfach modelliert und erhielt eine unkomplizierte Texturierung mit einfachen Farben. Ein Buntstift diente als Referenz für das Modell.

Kopfhörer-Modell

Die Modellierung der Kopfhörer erforderte aufgrund ihrer geschwungenen Form und der texturierten Oberfläche besondere Aufmerksamkeit. Das Modell durchlief zwei Phasen: zunächst eine Annäherung an das Referenzbild eines Gaming-Kopfhörers und dann eine Verfeinerung, einschließlich einer aufwendigen Lederstrukturtextur.

Dose-Modell

Die Modellierung der Dose eines Energydrinks begann mit einem einfachen Zylinder, der dann anhand eines Referenzbildes geformt wurde. Besonderes Augenmerk wurde auf die Textur gelegt, wobei ein bekanntes Produktlabel als Referenz diente.



USB-Stick-Modell

Der USB-Stick wurde mit einer integrierten Halterung für zusätzliche Komplexität modelliert. Die Texturierung betonte einen metallischen Effekt, insbesondere für die Halterung und das vordere Teil des USB-Sticks.

Stromverteiler-Modell

Der Stromverteiler wurde mit fünf Steckplätzen und einem Ein/Aus-Schalter modelliert, wobei der Fokus auf einfacher Funktionalität lag. Die Texturierung orientierte sich an einfachen weißen Steckerleisten.

Handy-Modell

Das Handymodell orientierte sich an modernen iPhones von Apple und erforderte aufgrund seiner zahlreichen Details wie Kameras, Tasten und Mikrofoneinlässe eine sorgfältige Modellierung und Texturierung, um einen realistischen Eindruck zu erzeugen.

4.3 Hauptmenü

Im folgenden Abschnitt werden die Entwicklungsphasen beim Entwurf des Menüs erläutert. Es wird erklärt, welche Probleme dabei auftreten und worauf bei der Benutzeroberfläche (UI) und der Benutzererfahrung (UX) geachtet werden muss. Es wird aufgeteilt in die Konzeption und Planung der Benutzeroberfläche bis hin zur finalen Umsetzung. Die iterative Entwicklung des Menüs durchlief mehrere Phasen, bei denen Entscheidungen zum Design des Layouts, des Farbschemas und der Interaktionselemente getroffen wurden. Während der Entwicklung wurden regelmäßig Überprüfungen und Anpassungen durchgeführt, um sicherzustellen, dass das Menü den gestellten Anforderungen entspricht und eine optimale Benutzererfahrung bietet. Dabei wurden auch Rückmeldungen und Tests von Nutzern in die Iterationsschleife einbezogen, um mögliche Verbesserungspotenziale zu identifizieren und zu berücksichtigen. Schließlich wurde das Menü in seiner endgültigen Version implementiert. Dabei wurde besonderes Augenmerk auf Funktionalität, Benutzerfreundlichkeit und Ästhetik gelegt. Durch diesen iterativen Entwicklungsprozess wurde sichergestellt, dass das Menü den Anforderungen entspricht und einen positiven Beitrag zur Gesamterfahrung des Spiels leistet.

4.3.1 Erstentwurf

Ursprünglich war geplant, das UI/UX-System mit einem sogenannten *Nahmenü* zu realisieren. Dieses folgt dem Nutzer in seiner Nähe und besteht in unserem Fall, aus drei primären Schaltflächen sowie einem Pin-Button. Das Nahmenü ist etwa auf Hüfthöhe des Benutzers positioniert und zeichnet sich durch eine vereinfachte Struktur aus, die eine intuitive Bedienung gewährleistet. Die Gestaltung des Menüs hat zum Ziel, Verwirrung zu

vermeiden und dem Nutzer ohne umfassende Vorabinformationen klar zu machen, wie er die Anwendung verwenden kann.

Nahmenü

Innerhalb von Unity bezeichnet der Begriff *Nahmenü* eine vordefinierte Konstruktion, die mit bestimmten Skripten ausgestattet ist, um sicherzustellen, dass das Menü dem Benutzer in alle Richtungen folgt. Es werden von Unity bereitgestellte Skripte verwendet, die es ermöglichen, das Menü dynamisch an die Position des Nutzers anzupassen.⁵⁶

Im Erstentwurf wurde das gesamte Menü ausschließlich innerhalb der Unity-Umgebung konstruiert. Dabei wurden sämtliche vorhandenen Schaltflächen und Funktionen aus den nativen Ressourcen von Unity zusammengesetzt. Dieser Ansatz führte zu einer konsistenten visuellen Gestaltung des Menüs. Allerdings waren die Möglichkeiten zur kreativen Gestaltung und Anpassung im Rahmen unseres individuellen Projekts stark begrenzt. Durch die ausschließliche Verwendung der internen Ressourcen von Unity wurden gewisse Einschränkungen hinsichtlich der Flexibilität und Individualisierungsmöglichkeiten des Menüs in Kauf genommen.

Obwohl diese Herangehensweise zu einem homogenen Erscheinungsbild des Menüs führte, war es wichtig, eine Balance zwischen visueller Einheitlichkeit und der Notwendigkeit individueller Anpassungsmöglichkeiten zu finden. Diese Herausforderung verdeutlicht die Bedeutung einer flexiblen und erweiterbaren Architektur für die Benutzeroberfläche, um den Anforderungen und Zielen des Projekts gerecht zu werden.



Abbildung 4.10: Darstellung des Erstentwurfs im Unity Editor

Die Abbildung 4.10 zeigt den ersten Entwurf des Menüs, das eine wichtige Schnittstelle für den Nutzer darstellt, um zwischen verschiedenen Szenarien zu navigieren. Die Struktur des Menüs bleibt unabhängig vom jeweiligen Spiellevel konstant, was zur Simplifizierung und Klarheit beiträgt. Die primären Schaltflächen auf der linken Seite dienen der Initiierung des Ladevorgangs für die verschiedenen Spielszenarien. Hervorzuheben ist der Pin-Button in Kombination mit dem darunterliegenden weißen Balken. Diese beiden Funktionen gemeinsam ermöglichen es dem Benutzer, das Menü an gewünschten Positionen zu verankern und bietet somit Flexibilität bei der Positionierung entsprechend individueller Präferenzen. Ein solcher Einsatz ist von Vorteil, wenn externe Objekte die Nutzbarkeit des Menüs beeinträchtigen könnten, wie zum Beispiel im Szenario des Platznehmens an einem Tisch, wo das Menü ungünstigerweise mit dem Tisch kollidieren könnte und somit die Nutzererfahrung beeinträchtigt werden würde.

⁵⁶Microsoft Dokumentation **Nahemenue MRTK2**

4.3.1.1 Probleme beim Erstentwurf

Problem 1 - Fehlende Dokumentation

Das Hauptziel bei dem Konzept des ersten Menüs, bestand darin, ein benutzerfreundliches Menü zu gestalten, das alle erforderlichen Funktionen enthält und dennoch übersichtlich ist. Nach mehreren Iterationen und Tests mit Testpersonen, die nicht mit dem Projekt vertraut waren, stellte sich heraus, dass der ursprüngliche Entwurf erhebliche Mängel bei der Dokumentation und Erklärung der verschiedenen Anwendungsszenarien und ihrer jeweiligen Aufgaben aufweist. Es wurde bemängelt, dass Benutzer lediglich zwischen den einzelnen Leveln wechseln können, ohne klare Informationen darüber zu erhalten, worum es in den einzelnen Leveln geht und welche Aufgaben diese beinhalten.

Diese Feststellung verdeutlicht eine wesentliche Lücke in der Benutzerführung und information innerhalb des Menüsysteams. Die unzureichende Dokumentation der Level und ihrer Ziele führt zu einer fehlenden Orientierung für die Benutzer und kann sich negativ auf ihre Erfahrung auswirken. Das Menü sollte nicht nur als Navigationswerkzeug dienen, sondern auch als Informationsquelle, die dem Benutzer eine klare Vorstellung über den Spielverlauf und die zu erreichen Ziele vermittelt.

Die Identifizierung dieses Problems betont die Wichtigkeit einer umfassenden Benutzerbeobachtung und -evaluation während des Designprozesses. Dadurch wird sichergestellt, dass das entwickelte Benutzeroberflächensystem den Bedürfnissen und Erwartungen der Zielgruppe entspricht. Eine zielgerichtete Überarbeitung des Menüs ist erforderlich, um die fehlende Dokumentation der Anwendungsszenarien und ihrer Aufgaben zu adressieren und somit die Benutzererfahrung zu verbessern.

Bei Testdurchführungen ohne Entwickler aus dem Projektteam kam es wiederholt zu Unklarheiten bei den Benutzern, insbesondere während des ersten Durchlaufs, bezüglich der zugewiesenen Aufgaben. Während der Nutzung der HoloLens-Anwendung gab es vermehrt Anfragen seitens der Benutzer an das Projektteam aufgrund von Unklarheiten. Dies lässt vermuten, dass die Anwesenheit von Entwicklern im Projektteam einen signifikanten Einfluss auf die Benutzererfahrung und Effektivität der HoloLens-Anwendung hat.

Problem 2 - Falsche Wahl des Menütyps

Während des Testprozesses stellte sich heraus, dass das Navigationsmenü oft eine Behinderung darstellte, insbesondere für Entwickler, die Funktionen wiederholt testeten und Anwendungsszenarien neu luden. Das ständige Verschieben des Menüs zur Seite erwies sich als zeitraubend und störte den Arbeitsfluss erheblich. Die ursprüngliche Intention, das Navigationsmenü zur Erleichterung der Interaktion zwischen Benutzer und Spielumgebung einzuführen, erwies sich somit als kontraproduktiv.

Es ist wichtig, die Auswirkungen von Benutzeroberflächenelementen auf den Entwicklungsprozess zu berücksichtigen, um ein reibungsloses Testen und Experimentieren zu gewährleisten. Eine effiziente und produktive Arbeitsweise des Entwicklerteams hängt davon ab. Die Notwendigkeit einer kontinuierlichen Evaluation und Optimierung von UI/UX-Komponenten während des gesamten Entwicklungszyklus wird durch dieses Problem verdeutlicht.

Dieses Problem verdeutlicht, dass der Erstentwurf und die ursprüngliche Konzeption hauptsächlich auf die Innovation und Neuheit des Menüdesigns abzielten, anstatt den Fokus auf die funktionale Nützlichkeit zu legen. Der einfache und kompakte Menütyp erwies sich als unzureichend, um eine umfassende und übersichtliche Dokumentation der Anwendungsszenarien zu integrieren. Folglich war eine Migration zu einem alternativen Menüansatz erforderlich. Diese Beobachtung verdeutlicht die Bedeutung einer ausgewogene-

nen Gestaltung von Benutzeroberflächen in Mixed-Reality-Systemen. Es ist wichtig, dass nicht nur ästhetische Innovationen verfolgt werden, sondern auch die tatsächliche Benutzererfahrung und -effizienz in Betracht gezogen wird. Die Reflexion über diesen Prozess bietet Einsichten in die iterative Entwicklung von Benutzerschnittstellen und die Anpassung an die Anforderungen und Rückmeldungen der Benutzer.

4.3.2 Finalversion des Menüs

Die endgültige Version des Menüs wurde entwickelt, um die Mängel des ursprünglichen Entwurfs anzugehen und idealerweise vollständig zu beheben. Dieses neue Konzept wurde durch eine umfassende Analyse populärer AR/VR-Spiele wie zum Beispiel *Beat Saber* geleitet, um die Gründe für die verbesserte Benutzererfahrung in diesen Kontexten zu untersuchen. Dabei wurden spezifische Merkmale und Designentscheidungen identifiziert, die einen signifikanten Einfluss auf die Zufriedenheit der Anwender haben.

Basierend auf vorangegangenen Recherchen wurde entschieden, das neue Menü in zwei klar definierte Bereiche zu unterteilen. Dies gewährleistet eine deutliche Unterscheidung des Hauptmenüs sowie eine klare Orientierung bezüglich der Navigation. Insbesondere innerhalb der Anwendungsszenarien wurde ein minimalistisches und unaufdringliches Menü implementiert, um den Fokus der Benutzer uneingeschränkt auf die auszuführenden Tätigkeiten zu lenken. Diese Designstrategie hat zum Ziel, die kognitive Belastung der Benutzer zu reduzieren und eine nahtlose Integration der Menüfunktionalitäten in den jeweiligen Nutzungskontext zu ermöglichen.

Das Ergebnis ist ein ansprechendes und benutzerorientiertes Menükonzept, das die Gesamterfahrung des Spiels verbessert und den Erwartungen der Zielgruppe gerecht wird.

4.3.2.1 Finalversion Menü - Hauptmenü

Auf dieser Grundlage wurde das Design des Menüsystems vollständig überarbeitet, wobei der erste Schritt bereits die vollständige Entfernung der Idee des Nahmenüs beinhaltete.

Anstelle des Nahmenüs wurden vorgefertigte Objekte in das Hauptmenü integriert, welche zuvor mit Blender modelliert wurden. Die Entscheidung, vorgefertigte Objekte aus Blender zu importieren, bietet eine breite Palette an Gestaltungsmöglichkeiten und ermöglicht es, das Menü visuell ansprechend und einladend zu gestalten. Außerdem trägt die Integration dieser Objekte dazu bei, das Menü intuitiver und zugänglicher zu machen, da sie dem Benutzer klare Anhaltspunkte und Handlungsmöglichkeiten bieten.

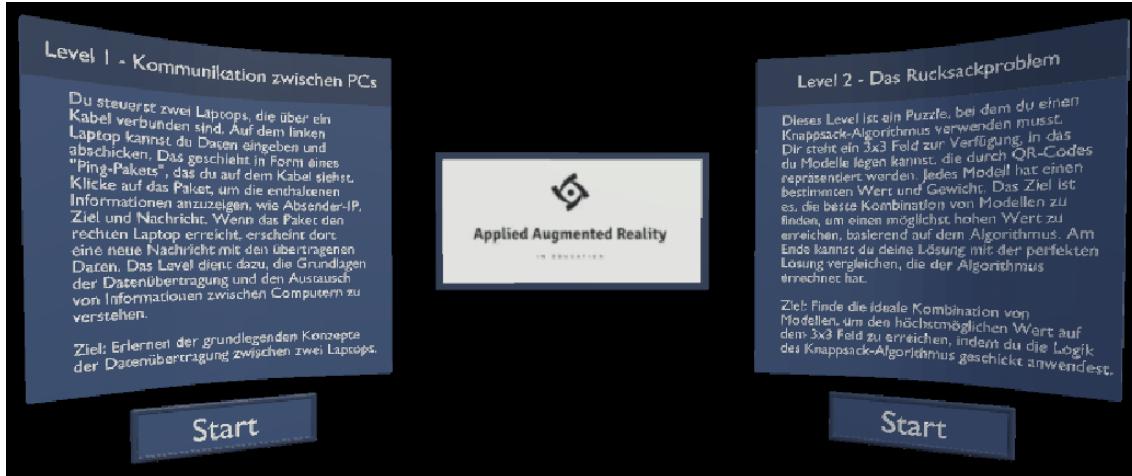


Abbildung 4.11: Darstellung des finalen Menüs im Unity Editor

Die in der Abbildung 4.11 dargestellten Objekte bestehen aus zwei separaten Tafeln, die durch unser Diplomarbeitslogo voneinander getrennt sind. Jede Tafel enthält den Titel des Anwendungsszenarios sowie einen vorgefertigten Startbutton. In Unity wurde ein unsichtbarer, aber dennoch klickbarer Button mit exakt denselben Dimensionen wie das Modell des Startbuttons an der entsprechenden Stelle platziert. Das Skript *SceneChange.cs*, welches für die Szenenwechsel-Funktionalität verantwortlich ist, ist diesem Button angehängt.

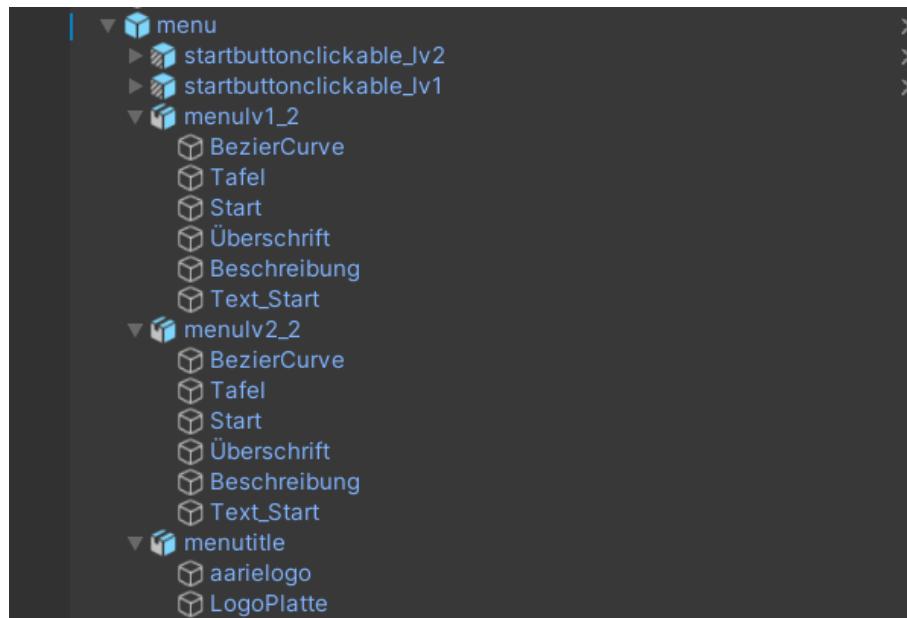


Abbildung 4.12: Darstellung der Hierarchie des finalen Menüs im Unity Editor

In der Abbildung 4.12 ist die Hierarchie des Hauptmenüs im Unity Editor zu sehen. Dem Prefab **menu** sind dabei 5 einzelne Objekte zugewiesen. Die Prefabs **startbuttonclickable_lv1/2** sind die in Unity erstellten unsichtbaren Buttons, die dafür zuständig sind den vormodellierten Buttons eine Funktion zu geben. Darunter zu sehen sind die 3 verschiedenen Assets, welche aus Blender importiert wurden. Diese bestehen jeweils aus den verschiedenen Einzelteilen, welche in Unity als Gameobjects dargestellt werden. Die **BezierCurve** ist eine spezielle Art von Kurve welche in Blender für die leichte Wölbung der Tafeln genutzt

wurde.⁵⁷

4.3.2.2 Finalversion Menü - Anwendungsszenario

Im Gegensatz zum Erstentwurf, der ein einheitliches Menü für alle Anwendungsszenarien vorsah, weist diese Version eine Differenzierung auf. Die Entscheidung, das Menü innerhalb der einzelnen Szenarien anzupassen, wurde getroffen, um die volle Aufmerksamkeit des Benutzers auf das jeweilige Level und die darin ausgeführten Aktivitäten zu lenken. Aus diesem Grund wurde ebenfalls das Nahmenü aus den Szenarien entfernt und durch ein simples Handmenü ersetzt.



Abbildung 4.13: Darstellung des finalen Menüs im Anwendungsszenario im Unity Editor

Wie in der Abbildung 4.13 gezeigt, besteht dieses Menü ausschließlich aus einem Zurück-Button, der den Benutzer zum Hauptmenü zurückführt. Das Handmenü wurde so konzipiert, dass es nur dann sichtbar ist, wenn der Benutzer es benötigt oder aktivieren möchte. Der Zurückbutton wird erst sichtbar, wenn der Benutzer seine linke Hand umdreht und auf die Handfläche schaut.

Diese Funktionalität hat zum Ziel, die Benutzerinteraktion intuitiver und kontextbezogener zu gestalten. Hierfür wird Bewegungserkennung und Blickrichtungserkennung genutzt, um das Handmenü nur dann einzublenden, wenn der Benutzer aktiv nach einem Navigationsmittel sucht. Dadurch wird die visuelle Ablenkung minimiert und die Benutzererfahrung optimiert, indem nur relevante Informationen und Interaktionselemente präsentiert werden, wenn sie benötigt werden.

4.3.2.3 Setzen des Menüs

Beim Erstentwurf gab es keine Probleme beim Wechseln zwischen den Szenen beziehungsweise dem Laden in das Hauptmenü, jedoch muss man in der neuen Version darauf achten das Menü richtig zu setzen. Dafür muss man die Modelle im richtigen Moment und an der richtigen Stelle in der richtige Szene selbstständig instanzieren, deshalb wurde extra für diesen Fall das *MenuSpawn.cs* Skript geschrieben.

```
1 public class MenuManager : MonoBehaviour{
```

⁵⁷Blender Bezier Curve

```

2     public GameObject menuPrefab;
3     public Transform cameraPosition;
4     private void Awake(){
5         Vector3 newSpawnPoint = new Vector3(cameraPosition.position.x, cameraPosition.
position.y, cameraPosition.position.z + 1);
6         menuPrefab.transform.position = newSpawnPoint;
7         menuPrefab.transform.localScale = new Vector3(0.25f, 0.25f, 0.25f);
8         menuPrefab.SetActive(true);
9     }
10 }
```

Codeabschnitt 4.1: Menueinstanzierung bei Neuladen.

Die Klasse `MenuManager` ist dafür zuständig, das Menü beim Neuladen der Anwendung zu instanziieren und zu platzieren. In der Methode `Awake()`, die bei jedem Start der Anwendung ausgeführt wird, wird zunächst eine neue Position für das Menü berechnet. Diese Position befindet sich leicht vor der aktuellen Kameraposition, indem der Wert der z-Achse um 1 erhöht wird. Anschließend wird das Menü auf diese neue Position verschoben und in der Größe angepasst, um es passend darzustellen. Schließlich wird das Menü aktiviert, sodass es für den Benutzer sichtbar und interaktiv wird.

4.3.3 Laden der Level

Um den Buttons eine Funktion zuzuweisen, wird ein Skript benötigt, das aktiviert wird, sobald ein Button gedrückt wird. Im vorliegenden Fall ist das Skript `SceneChange.cs` für den Wechsel zwischen den verschiedenen Anwendungsszenarien verantwortlich. Wenn einer der Buttons im Hauptmenü gedrückt wird, wird dieses Skript ausgeführt. Die Spielszene wird im *Unity Inspector* festgelegt. Der Code greift auf die Variable zu, die den Namen der Szene enthält, wie sie zuvor im Inspector benannt wurde.

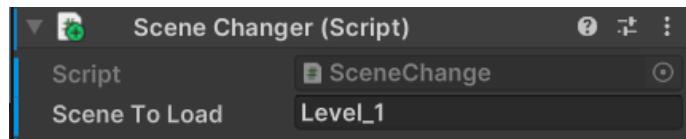


Abbildung 4.14: Darstellung der SceneToLoad Variable, anhand des Level1 Buttons.

Für den linken Button im Hauptmenü ist beispielsweise die Spielszene *Level1* vorgesehen, während für den rechten Button die Szene *Level2* zugewiesen ist. Durch diese Konfiguration im Inspector wird die Flexibilität gewährleistet, Szenen dynamisch anzupassen, ohne dass Änderungen am eigentlichen Skript vorgenommen werden müssen. Dies ermöglicht eine effiziente Verwaltung und Anpassung der Spielinhalte während des Entwicklungsprozesses.

```

1 public class SceneChanger : MonoBehaviour{
2     public string sceneToLoad;
3     public void ChangeScene(){
4         if (SceneManager.GetActiveScene().name != sceneToLoad){
5             PlayerPrefs.DeleteAll();
6             SceneManager.LoadScene(sceneToLoad, LoadSceneMode.Single);
7         }
8     }
9 }
```

Codeabschnitt 4.2: Auf Knopfdruck Szene wechseln.

Das vorliegende Skript hat eine vergleichsweise einfache Struktur. Die Variable `public string sceneToLoad` dient als Empfänger für den Namen der zu ladenden Szene, welcher im Unity-Inspector individuell für jede Szene festgelegt werden kann. Um den Szenenwechsel zu initiieren, wird die Methode `ChangeScene()` aufgerufen. Zunächst wird in dieser Methode geprüft, ob die aktive Szene, die durch `SceneManager.GetActiveScene()` ermittelt wird, mit der zu ladenden Szene übereinstimmt. Falls dies nicht der Fall ist, werden alle gespeicherten PlayerPrefabs mittels `PlayerPrefs.DeleteAll()` gelöscht. Dieser Schritt ist von entscheidender Bedeutung, um potenzielle Komplikationen im Zusammenhang mit dem HoloLens-Cache während des Szenenwechsels zu vermeiden.

Nachdem die Prefabs bereinigt wurden, wird die definierte Szene mit der Methode `SceneManager.LoadScene()` geladen. Dabei wird die Ladeoption `LoadSceneMode.Single` verwendet, um sicherzustellen, dass nur die angegebene Szene aktiv geladen wird und keine anderen Szenen im Hintergrund verbleiben. Diese präzise Steuerung des Ladevorgangs zielt darauf ab, die Performance und Stabilität der Anwendung auf der HoloLens-Plattform zu optimieren und potenzielle Konflikte zwischen Szeneninhalten zu vermeiden.

4.3.4 UI/UX

Die Gestaltung der Benutzeroberfläche und die damit verbundene Benutzererfahrung sind wesentliche Aspekte bei der Konzeption und Entwicklung von Software. Eine bewährte Strategie besteht darin, sich an populären und gleichgerichteten Anwendungen zu orientieren, insbesondere im Bereich der Augmented- oder Virtual Reality. Es wird darauf geachtet, komplexe und verwirrende Menüstrukturen zu vermeiden und eine konsistente Farbpalette über die gesamte Benutzeroberfläche hinweg zu verwenden.

Diese Herangehensweise basiert auf der Erkenntnis, dass erfolgreiche Anwendungen in ähnlichen Domänen oft bewährte Designprinzipien und Interaktionsmuster aufweisen, die sich positiv auf die Benutzerakzeptanz und -zufriedenheit auswirken können. Durch die Anpassung an gängige Standards und Trends in der Branche kann das Risiko von Benutzerirritationen und -ablehnungen minimiert werden. Gleichzeitig wird eine vertraute und angenehme Benutzererfahrung gefördert.

Es ist wichtig zu betonen, dass eine solche Inspiration von bestehenden Anwendungen nicht als direkte Kopie verstanden werden sollte, sondern vielmehr als Quelle für Anregungen und bewährte Praktiken, die in einem eigenständigen und angepassten Kontext implementiert werden können. Auf diese Weise kann die Effektivität und Attraktivität der Benutzeroberfläche optimiert werden, um die Bedürfnisse und Erwartungen der Zielgruppe bestmöglich zu erfüllen.

4.4 Nachrichtenaustausch Anwendungsszenario

In diesem Szenario wird das IT-Grundprinzip des Nachrichtenaustausches illustriert. Zwei PCs sind in einer realen Umgebung durch ein rotes Kabel miteinander verbunden, wobei auf jedem PC dieselbe Website geöffnet ist. Der Benutzer trägt eine HoloLens 2 und erhält die Aufforderung, das rote Kabel zu betrachten. Nach einem Countdown wird ein Foto der Umgebung aufgenommen und anschließend verarbeitet. Dabei werden Raycasts eingesetzt, um die exakte Position des roten Kabels in der physischen Welt zu bestimmen. Sobald die Position des Kabels erkannt wurde, kann der Benutzer über die Website auf den beiden PCs eine Nachricht senden. Diese Nachricht wird dann in Form eines Pakets für den HoloLens 2 Benutzer visualisiert. Der Benutzer hat außerdem die Möglichkeit, auf das Paket zu drücken, um Informationen wie Absender und Nachricht einzusehen.

(→
Schoditsch)

4.4.1 Aufbau von Nachrichtenaustausch Anwendungsszenario

In diesem Abschnitt wird detailliert beschrieben, wie die Unity-Szene aufgebaut ist, welche Rollen und Funktionen die einzelnen Spielobjekte haben und wie sie miteinander interagieren, um die gewünschte Funktionalität der Szene zu erreichen.

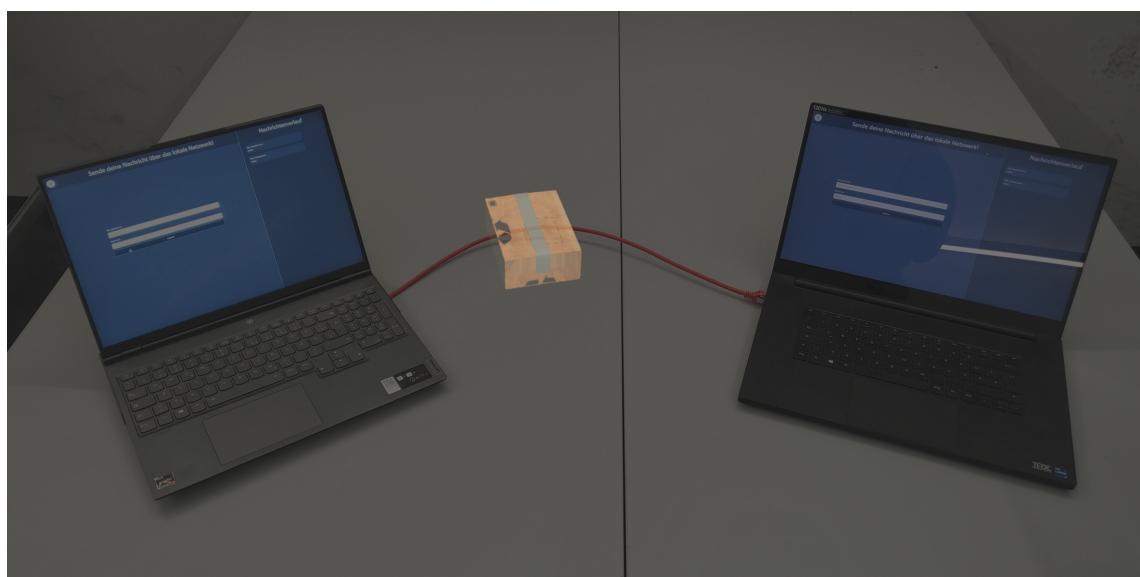


Abbildung 4.15: Aufbau in der realen Welt

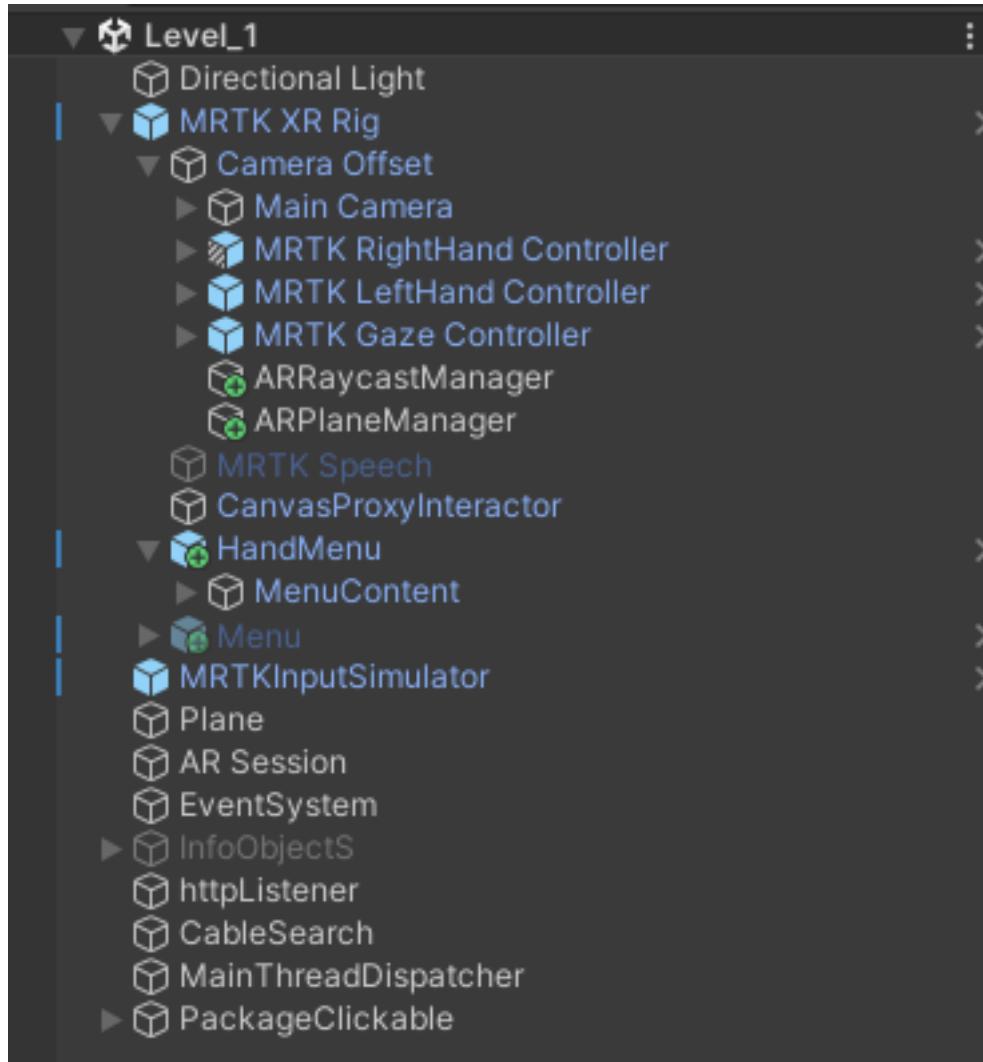


Abbildung 4.16: Hierarchie des Nachrichtenaustausch-Levels im Unity Editor.

- **Level-1:** Die Szene von Unity, in der alle Unity-Objekte enthalten sind.
- **UserInfo:** Spielobjekt, welches den HoloLens Benutzer einen Countdown angibt und von welchem aus daraufhin alle anderen Skripts ausgeführt werden.
- **LoadingCircle:** Während gewisse Prozesse von der Maschine durchgeführt werden, die länger benötigen könnten, wird dieses *Spielobjekt* angezeigt, um den Benutzer zu informieren, dass das Programm Berechnungen ausführt. Das Objekt beinhaltet ein *Canvas* (siehe Kapitel 4.4.8.2) mit einem *User Interface*.
- **Plane:** Ist ein Spielobjekt, welches den Code beinhaltet und ausführt. Es kann auch zum Debugging genutzt werden, da es die erkannten roten Pixel visuell darstellt.
- **InfoObjectS:** Beinhaltet die Informationen (Name und Nachricht) welcher der Benutzer auf der Website angibt. Diese Nachricht wird durch diesem Objekt angezeigt, sobald der Nutzer auf das virtuelle Paket drücken.
- **httpListener:** Verarbeitet die eingehenden http Anfragen der Website.
- **CableSearch:** Beinhaltet den Start der Unity-Szene und zeigt dem Benutzer Informationen an
- **MainThreadDispatcher:** Ein paar Funktionen, wie das Initialisieren eines Ob-

jets, nur im Haupt Thread erlaubt ist, um Fehler zu vermeiden, wird ein *MainThreadDispatcher* gebraucht. Das Skript innerhalb dieses Spielobjekts sorgt dann dafür, dass sobald der Haupt Thread verfügbar ist, dieser den Prozess durchführt.

- **PackageClickable:** Ist ein Objekt, welches die Textur des Paketes und zwei Code-Skripte beinhaltet. Eines dieser Code Skripte ist ein *BoxCollide*, welcher von Unity zur Verfügung gestellt wird, sowie ein Button Skript.

4.4.2 Text Mesh Pro

TextMeshPro ist eine leistungsstarke Text-Rendering, die in der Unity-Engine verwendet wird. Sie bietet verbesserte Kontrolle und Flexibilität bei der Darstellung von Text in Unity-Anwendungen. Im Gegensatz zur Standard-Textkomponente bietet TextMeshPro Unterstützung für eine Vielzahl von Schriftarten und Textstilen sowie eine präzisere Anpassung der Textdarstellung. Dies wird durch die Verwendung von Vektorgrafiken ermöglicht, die eine höhere Auflösung und Skalierbarkeit gewährleisten.⁵⁸

4.4.3 Transform Klasse

Die Transform-Klasse in Unity ist eine wichtige Komponente für die Platzierung, Rotation und Skalierung von Spielobjekten in einer 3D-Szene. Sie ermöglicht es ebenso innerhalb einer Hierarchie von Objekten, die Position, Rotation und Skalierung eines Spielobjekts relativ zu seinem Elternelement zu verändern.⁵⁹

4.4.4 IEnumarator Klasse

Die Klasse IEnumarator ist eine Schnittstelle in der Programmiersprache C#, die eine standardisierte Möglichkeit bietet, eine Sammlung von Elementen zu durchlaufen, ohne die interne Struktur der Sammlung offenzulegen. Sie ermöglicht das sequenzielle Durchlaufen von Elementen in einer Datenstruktur wie Arrays, Listen oder anderen Sammlungen.⁶⁰⁶¹

4.4.5 Coroutines

Coroutines erlauben es den für *Coroutines* verwendeten Code über mehrere Unity Frames vom Haupt Thread ausgeführt wird. Das bedeutet genauer gesagt einfach dass, wenn eine Coroutine aufgerufen wird, das die Funktion, welche als Coroutine ausgeführt werden soll, bis zu einem bestimmten Code Stück durchgeführt wird. Wenn es diesen bestimmten Code Stück erreicht hat, bearbeitet der Haupt Thread für einen Frame an dem Ablauf außerhalb von der Coroutine und geht dann wieder zur Coroutine Funktion zurück. Coroutines werden deswegen hauptsächlich in Methoden verwendet, welche eine Abfolge von Ereignissen über einen Zeitraum darstellen.

Zum Darstellen von *Coroutines* ist dieses kurze Skript, welches das Objekt, an den es angebunden ist, in eine Richtung bewegt.

```

1 public class CoroutineExample : MonoBehaviour {
2     // Das Zielobjekt, zu dem das Objekt bewegt werden soll
3     public Transform target;
4     // Die Geschwindigkeit, mit der das Objekt bewegt wird
5     public float moveSpeed = 1f;
```

⁵⁸UnityTextMeshPro

⁵⁹UnityTransform

⁶⁰Microsoft LearnIEnumaratorMicrosoft

⁶¹Medium IEnumaratorMedium

```

6   // Die Anzahl der Schritte fÃ¼r die Bewegung
7   public int numSteps = 10;
8
9   void Start() {
10    // Starte die Coroutine fuer die Bewegung
11    StartCoroutine(MoveToTarget());
12 }
13
14 IEnumerator MoveToTarget() {
15  // Die Startposition des Objekts
16  Vector3 startPos = transform.position;
17  // Die Zielposition des Objekts
18  Vector3 endPos = target.position;
19  // Iteriere ueber jeden Schritt der Bewegung
20  for (int i = 0; i <= numSteps; i++) {
21   // Berechne den Fortschritt der Bewegung (Wert zwischen 0 und 1)
22   float t = (float)i / numSteps;
23   // Bewege das Objekt entsprechend dem aktuellen Fortschritt
24   transform.position = Vector3.Lerp(startPos, endPos, t);
25   // Wartet fuer eine bestimmte Zeit, bevor der nÃchste Schritt ausgefuehrt
26   wird
27   yield return new WaitForSeconds(moveSpeed / numSteps);
28 }
29 }
```

Codeabschnitt 4.3: Coroutine Beispiel

- **moveSpeed:** Die Geschwindigkeit, mit der das Objekt bewegt wird. Diese Variable wird verwendet, um die Geschwindigkeit anzupassen, mit der das Objekt von seiner aktuellen Position zur Zielposition bewegt wird. Wenn keine Geschwindigkeit angegeben wird, wird standardmäig eine Geschwindigkeit von 1 verwendet.
- **numSteps:** Die Anzahl der Schritte, in denen die Bewegung stattfinden soll. Diese Variable steuert, wie viele Zwischenschritte während der Bewegung durchlaufen werden, um eine flüssige Bewegung zu erzielen. Ein höherer Wert führt zu einer genaueren Bewegung, aber möglicherweise zu einer längeren Bewegungsdauer.
- **target:** Das Zielobjekt, zu dem das Objekt bewegt werden soll. Diese Variable definiert die Endposition, zu der das Objekt bewegt werden soll. Sie speichert die Positionsinformationen als *Transform*, was in Unity üblich ist und die Position, Rotation und Größe eines Objekts speichert und ermöglicht, diese zu manipulieren.

Der Code verwendet eine Coroutine, um das Objekt von seiner aktuellen Position zur Zielposition zu bewegen. Coroutines werden in Unity verwendet, um asynchrone Operationen durchzuführen, ohne den Hauptthread zu blockieren. In diesem Fall wird die Coroutine *MoveToTarget* gestartet, die in einer Schleife jeden Schritt der Bewegung durchläuft. Für jede Iteration wird die aktuelle Position des Objekts mithilfe der Funktion *Vector3.Lerp* zwischen der Startposition und die Zielposition entsprechend dem Fortschritt der Bewegung gesetzt.⁶² ⁶³

Um zwischen den Schritten eine Pause einzulegen, wird die Coroutine mit *yield return new WaitForSeconds(moveSpeed / numSteps)* unterbrochen, was eine Wartezeit zwischen den Bewegungsschritten ermöglicht und eine flüssige und kontrollierte Bewegung des Objekts gewährleistet.

⁶²Log Rocket **Coroutine2**

⁶³Unity **Coroutine**

Falls man aus einer laufenden Coroutine beenden will, kann man auch *StopCoroutine* oder *StopAllCoroutines* verwenden. Man könnte den obigen Code folgendermaßen erweitern:

```

1 public void stopFunctionCoroutine() {
2     StopCoroutine(MoveToTarget)
3 }
4
5 public void stopAllCurrentCoroutine() {
6     StopAllCoroutines()
7 }
```

Codeabschnitt 4.4: Coroutine beenden

- *StopCoroutine*: Sorgt dafür, dass nur die *MoveToTarget* Funktion, welche für die *Coroutine* verwendet wurde, beendet wird.
- *StopAllCoroutines*: Sorgt dafür, dass alle Funktionen, welche für alle laufenden *Coroutine* verwendet wurden, beendet werden.

4.4.6 Start der Szene

Wenn der Benutzer von Hauptmenü in das Anwendungsszenario eins lädt, dann wird über das Spielobjekt *CableSearch* direkt das *CableSearch.cs* Skript gestartet, welches für die Erkennung des Kabels verantwortlich ist.

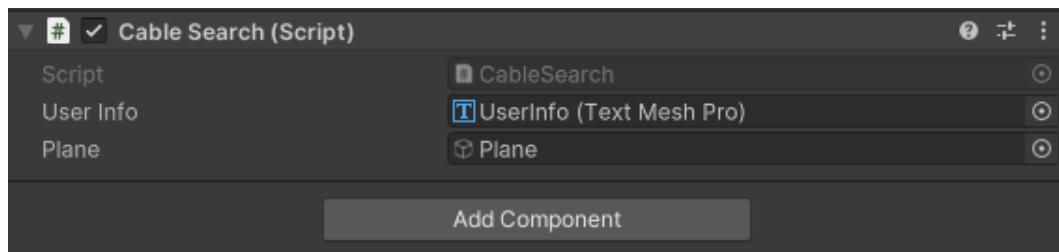


Abbildung 4.17: Komponenten des *CableSearch* Spielobjekts

Dem Skript wird zuerst ein *Text Mesh Pro* (siehe Kapitel 4.4.2) übergeben, dieser Text wird innerhalb von der Unity Szene angezeigt. Danach wird ein weiteres *Spielobjekt* mit den Namen *Plane*, welches ebenfalls in der Szene existiert, für das Debuggen (Siehe Kapitel 4.4.14.1) und ausführen vom weiteren Code übergeben.

```

1 void Start() {
2     // Text wird den Benutzer angezeigt sodas dieser auf das Kabel schaut.
3     userInfo.text = "Schauen Sie auf das Rote Kabel";
4
5     // Starte eine Coroutine, um einen verzögerten Start zu ermöglichen, sodas der
6     // nutzer zeit hat zu lesen.
7     StartCoroutine(DelayedStart());
8 }
9
10 // Coroutine funktion fñr einen verzögerten Start.
11 private IEnumerator DelayedStart() {
12     // Wartet 2 Sekunden, bevor der Startvorgang abgeschlossen ist.
13     yield return new WaitForSeconds(2.0f);
14
15     // Die Variable canStartScript wird auf true gesetzt, um anzuzeigen, dass das
16     // Skript gestartet werden kann.
17     canStartScript = true;
18 }
```

Codeabschnitt 4.5: Start des *CableSearch* Skripts

Beim Start des Skripts wird die Funktion *Start* aufgerufen. Diese modifiziert den Text von *userInfo*, welcher dem Benutzer angezeigt wird. Nach dieser Änderung wird eine *Coroutine*, (siehe Kapitel 4.4.5), gestartet und an einen neuen Thread namens *DelayedStart* übergeben. Die Funktion *DelayedStart* stellt sicher, dass der Benutzer die Nachricht zwei Sekunden lang lesen kann, bevor sie durch einen Timer ersetzt wird. Dieser Timer wird in der *Update*-Funktion von Unity verarbeitet, um eine zeitgesteuerte Aktualisierung zu gewährleisten.

```

1 void Update() {
2     // Ueberprueft, ob das Skript gestartet werden kann und ob das Skript noch schon
3     // ausgefÃhrt wurde
4     if (canStartScript && !objectPlaced) {
5         // Ueberprueft ob noch Zeit uebrig ist
6         if (timeRemaining > 1f) {
7             // Verringert die verbleibende Zeit basierend auf der verstrichenen Zeit
8             // seit dem letzten Update
9             timeRemaining -= Time.deltaTime;
10            // Berechnet die verstrichene Zeit in Sekunden
11            int elapsedTime = Mathf.FloorToInt(timeRemaining % 60);
12            // Aktualisiert den Text auf dem Bildschirm mit der verbleibenden Zeit
13            userInfo.text = elapsedTime.ToString();
14        }
15        else {
16            // Setzt die verbleibende Zeit auf 1 Sekunde, um zu verhindern, dass sie
17            // negativ wird
18            timeRemaining = 1f;
19            // Leert den Text auf dem Bildschirm
20            userInfo.text = "";
21            // wird auf true gesetzt um zu verhindern das weiter ueberpueft wird
22            objectPlaced = true;
23            // fuert den weiteren code aus
24            startPictureTaking();
25        }
26    }
27 }
28 }
```

Codeabschnitt 4.6: Update des *CableSearch* Skripts

Nach Ablauf der Wartezeit von zwei Sekunden wird eine Zeitmessung gestartet. Dabei wird zunächst geprüft, ob die Restzeit *timeRemaining*, die außerhalb der Funktion als Gleitkommazahl von sechs definiert ist, größer als eins ist. Ist diese Bedingung erfüllt, wird die Restzeit durch Abziehen der verstrichenen Zeit zwischen den *Updates* angepasst. Anschließend wird die verbleibende Zeit in Sekunden umgerechnet und dem Benutzer angezeigt. Nach Ablauf der fünf Sekunden Ladezeit wird die verbleibende Zeit auf eins gesetzt, um Fehler zu vermeiden. Danach wird der Text geleert und der boolesche Wert *objectPlaced* auf *true* gesetzt, um sicherzustellen, dass die Update-Funktion nicht weiter ausgeführt wird. Schließlich wird innerhalb der Funktion *startPictureTaking* das nächste Skript gestartet, um den Prozess fortzusetzen.

4.4.7 Kabelerkennung

Um das Visualisieren des Pakets bei der Übertragung der Nachricht über das Kabel zu gewährleisten, muss zunächst die Position des Kabels in der Umgebung genau bestimmt werden. Dies geschieht durch die Aufnahme eines Bildes mit der Kamera der HoloLens 2. Während dieser Aufnahmearaufgabe ist es wichtig, dass der Benutzer seinen Blick auf das jeweilige Kabel richtet. Nur so ist eine exakte Orientierung des Kabelverlaufs möglich.

4.4.8 Fotoaufnahme

Innerhalb von diesem Kapitel wird die Aufnahme des von rotem Kabel aufgenommenen Fotos, mithilfe des dementsprechenden Code Funktionen innerhalb von *PictureTaking.cs* beschrieben. Dazu werden auch die für den Prozess notwendigen Unity Klassen beschrieben.

4.4.8.1 Photocapture Klasse

Die *PhotoCapture* Klasse benutzt die *PhotoCapture* API, um ein Foto mit der Kamera des Gerätes zu machen. Damit dies möglich ist, muss man Unity die Erlaubnis für die *Webcam* und das *Microphone* erteilen. Diese gibt dann die Erlaubnis, wenn etwa ein Laptop verwendet wird, damit dieser über Unity Code Fotos/Videos sowie Sprachaufnahmen über die Kamera und Mikrofone aufnehmen kann. Dies trifft dann auch auf die HoloLens 2 zu. Dies kann unter *Edit* ⇒ *Project Settings* ⇒ *Player* ⇒ *Settings for Universal Windows Platform Rightarrow Publishing Settings* ⇒ *Capabilities* gemacht werden.

Die wichtigsten Funktionen von *PhotoCapture* sind:

- **PhotoCapture.CreateAsync():** Erstellt asynchron eine Instanz eines PhotoCapture Objekts, welches benutzt werden kann, um ein Foto zu schießen.
- **PhotoCapture.StartPhotoModeAsync(CameraParameter):** Dafür werden die Parameter der Kamera benötigt, die zur *CameraParameter*-Klasse instanziert werden.
- **PhotoCapture.TakePhotoAsync():** Macht das Foto.
- **PhotoCaptureFrame.UploadImageDataToTexture:** Kopiert das unverarbeitete Foto innerhalb einer Variable.
- **PhotoCaptureObject.StopPhotoModeAsync():** Beendet die Instanz des Photocapture und ruft danach die Funktion innerhalb der Klammer.⁶⁴

4.4.8.2 Unity Canvas

Ein Canvas stellt einen Bereich dar, in dem alle Benutzeroberflächen (kurz UI) enthalten sind. Die Hierarchie von so einem UI besteht aus dem *Spielobjekt* als sogenannten *parent*, welches als *child* (siehe Kapitel 4.2.8) die *UI* besitzt. Diese *UI* kann auch als weiteres child ein *Image* besitzen. Das Canvas wird als Rechteck dargestellt, was es erleichtert, die genaue Position des Objekts zu bestimmen.

Ein Canvas verfügt auch über verschiedene Rendermoduse:

- **Screen Space-Overlay:** Dieser Modus ermöglicht es, UI-Elementen über der Szene auf dem Bildschirm zu werden. Das bedeutet, dass das UI-Element unabhängig von dem, was in der Szene passiert, über der Szene gerendert wird.
- **Screen Space-Camera:** Dieser Modus funktioniert ähnlich wie *Screen Space-Overlay*, mit dem Unterschied, dass das UI-Element vor der zugewiesenen Kamera platziert wird.
- **World Space:** Dieser Rendermodus wird in diesem Projekt am häufigsten verwendet. In diesem Modus kann der Canvas, im Gegensatz zu den anderen Modi, verschoben werden, genau wie alle anderen Objekte in der Szene.⁶⁵⁶⁶

⁶⁴Unity, **Photocapture**

⁶⁵Unity **Canvas**

⁶⁶Unity **Renderer**

4.4.8.3 Foto der Umgebung aufnehmen

Das Aufnehmen des Fotos wird mithilfe der *PhotoCapture*-Klasse (siehe Kapitel 4.4.8.1) aufgenommen. Um zu gewährleisten, dass es funktioniert und mit der Kamera ein qualitativ gutes Foto zum Verarbeiten gemacht wird, müssen Einstellungen für die Kameraparameter eingestellt werden.

```

1 public void takingPicture() {
2     // ueberprueft, ob bereits ein Bild aufgenommen wird, um zu verhindern, dass die
     Methode erneut aufgerufen wird
3     if (!takingNewPicture) {
4         // Aktiviert das LoadingSymbol-Spielobjekt, um anzuzeigen, dass ein Bild
     aufgenommen wird
5         loadingCircle.SetActive(true);
6         // Holt sich das Canvas-Objekt innerhalb des LoadingSymbol-Spielobjekts
7         loadingCircleCanvas = loadingCircle.GetComponentInChildren<Canvas>();
8         // Startet die Ladeanimation im LoadingCircle
9         loadingCircleCanvas.GetComponent<LoadingCircle>().StartLoading();
10        // Erstellt asynchron eine Fotoaufnahme
11        PhotoCapture.CreateAsync(false, delegate (PhotoCapture captureObject) {
12            // ueberprueft, ob die Fotoaufnahme erfolgreich erstellt wurde
13            if (captureObject != null) {
14                // Speichert die Fotoaufnahme in einer Instanzvariablen
15                photoCaptureObject = captureObject;
16                // Konfiguriert die Kameraparameter fuer die Aufnahme
17                CameraParameters cameraParameters = new CameraParameters();
18                cameraParameters.hologramOpacity = 0.0f;
19                // Setzt die Aufloesung der Kamera
20                cameraParameters.cameraResolutionWidth = cameraResolution.width;
21                cameraParameters.cameraResolutionHeight = cameraResolution.height;
22                // Setzt das Pixelformat fuer die Aufnahme
23                cameraParameters.pixelFormat = CapturePixelFormat.BGRA32;
24                try {
25                    // Startet asynchron den Fotoaufnahmemodus
26                    photoCaptureObject.StartPhotoModeAsync(cameraParameters, delegate
(PhotoCapture.PhotoCaptureResult result) {
27                        // Nimmt ein Foto auf und speichert es im Arbeitsspeicher
28                        photoCaptureObject.TakePhotoAsync(onCapturedPhotoToMemory);
29                    });
30                }
31            });
32        });
33    }
34 }
```

Codeabschnitt 4.7: Bild einstellung und aufnahme

Zunächst wird in diesem Code mit dem Boolean *takingNewPicture* geprüft, ob bereits ein Bild aufgenommen wurde. Daraufhin wird sofort das Lade-Symbol (siehe Kapitel 4.4.9) gestartet und angezeigt. Anschließend wird eine asynchrone Methode *PhotoCapture.CreateAsync()* aufgerufen, um ein *Photo-Capture* zu erstellen. Diese Methode erstellt ein *PhotoCapture-Objekt*, das für die Aufnahme von Fotos verwendet wird. Die Methode akzeptiert den Parameter false, der angibt, dass kein Audio-Capture erstellt werden soll. Wenn die Aufnahme erfolgreich war, wird das PhotoCapture-Objekt gespeichert und die Kameraparameter für die Aufnahme konfiguriert. Anschließend wird der Photo-Capture-Modus mit *StartPhotoModeAsync* gestartet, und wenn dieser erfolgreich gestartet wurde, wird das Foto aufgenommen und mit *TakePhotoAsync* im Speicher abgelegt.

4.4.9 Ladesymbol

Das Ladesymbol ist eine visuelle Darstellung, die den Benutzer darüber informiert, dass Berechnungen oder Prozesse im Hintergrund ausgeführt werden. In erster Linie wird dieses Symbol während der Kabelsuche verwendet. Diese Animation des Ladesymbols wird durch einen entsprechenden Programmcode gesteuert, um dem Benutzer eine klare und konsistente Rückmeldung zu geben, dass der Prozess läuft. Im Design der Interaktion spielt das Ladesymbol eine wichtige Rolle, um den Benutzer über den aktuellen Status der Anwendung zu informieren.

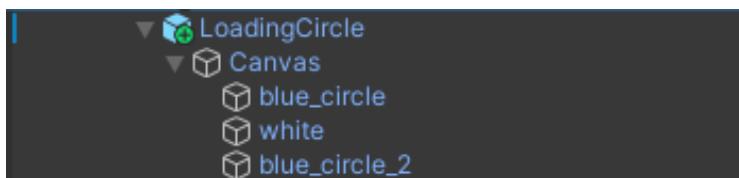


Abbildung 4.18: Hierarchie Lade Kreises im Unity Editor

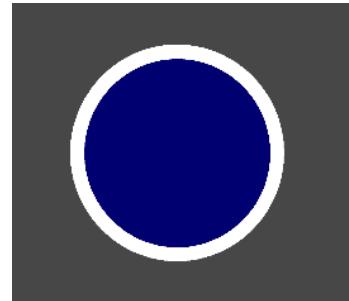


Abbildung 4.19: Lade Symbol

```

1 IEnumerator LoadingCoroutine() {
2     while (isRunning) {
3         if (img.fillAmount >= 1f) {
4             img.fillAmount = 0f;
5         }
6         else {
7             img.fillAmount += fillSpeed;
8         }
9         yield return new WaitForSeconds(0.1f);
10    }
11 }
```

Codeabschnitt 4.8: Code zum ausführen des lade Symbols

Bei der Initiierung dieser Funktion wird die Methode *StartCoroutine(LoadingCoroutine())* von Coroutine (siehe Kapitel 4.4.5) benutzt, durch diesen Aufruf wird sichergestellt, dass die Funktion nebenbei und asynchron zum Code ausgeführt wird. Zum Anzeigen von dem Ladesymbol wird ein Objekt *img* verwendet, dieses *img* beinhaltet ein Bild eines weißen Kreises (äußerer Kreis von Abbildung (4.19)) und stellt einen Fortschritt während der Berechnung dar.

4.4.10 Fotoverarbeitung

4.4.10.1 Unity Job System

Das Job-System von Unity bietet eine effiziente Möglichkeit zur Implementierung von Multithreading in Anwendungen, wodurch die Anwendung in der Lage ist, alle verfügbaren CPU-Kerne optimal zu nutzen. Im Gegensatz zu einem sequenziellen Durchlauf des Codes durch einen einzigen Thread, dem sogenannten Haupt Thread, ermöglicht das Job-System die parallele Verarbeitung von Codeabschnitten durch separate Arbeiter-Threads. Durch die Nutzung mehrerer Arbeiter-Threads können Aufgaben effizienter verteilt und gleich-

zeitig bearbeitet werden. Dies trägt wesentlich zur Verbesserung der Effizienz und der Gesamtdauer der Ausführung des Programms bei. Eine wichtige Eigenschaft des Job-Systems besteht darin, sicherzustellen, dass nur so viele Threads verwendet werden, wie von den verfügbaren CPU-Kernen unterstützt werden.

Ein weiteres wichtiges Konzept, das im Job System implementiert ist, ist das sogenannte "Work Stealing". Dabei handelt es sich um eine Planungsstrategie für Threads, bei der ein Thread, der seine aktuellen Aufgaben abgeschlossen hat, zusätzliche Aufgaben von anderen Arbeiter-Threads übernimmt und sie in seine eigene Warteschlange einfügt. Dies ermöglicht eine dynamische und effiziente Verteilung von Aufgaben zwischen den Threads und trägt dazu bei, die Last gleichmäßig auf alle verfügbaren Ressourcen zu verteilen.

Um das Schreiben von einem Multithread-Code zu erleichtern, verfügt das Job-System über ein Sicherheitssystem, das sicherstellt, dass potenzielle Wettlaufbedingungen eine sogenannte *Race Condition* zu vermeiden und somit können Fehler vermieden werden. Um zu gewährleisten, dass keine Wettlaufbedingungen auftreten können, erhält jeder Thread eine isolierte Kopie der Daten, die er verarbeiten kann.⁶⁷

4.4.10.2 Vector2 Klasse

Unity bietet dem Benutzer eine umfassende Funktionalität zur Darstellung und Manipulation von 2D-Positionen, -Richtungen und -Geschwindigkeiten innerhalb deren *Engine*. Mithilfe der *Vektor2D* Klasse von Unity lassen sich Bewegung, Interaktion und Objekte exakt steuern und simulieren. Diese Klasse besitzt die folgenden Eigenschaften:

- *x* und *y*: Punkt Koordinate in einem 2 dimensionalen Koordinatensystem
- *magnitude*: Besitzt die Größe des 2D Vektors. Die Größe des Vektors wird mithilfe von dieser Formel berechnet $\sqrt{x^2 + y^2}$
- *normalized*: Normalisierung bedeutet, den Vektor auf eine Länge von 1 zu bringen, während seine Richtung beibehalten wird.⁶⁸

4.4.10.3 Pixel Positionen finden

Nachdem ein Foto aufgenommen wurde, erfolgt die Datenverarbeitung mittels einer *Texture2D*-Klasse innerhalb des *Unity-Job-Systems* in paralleler Ausführung. Jeder Thread des Systems ist dafür zuständig eine festgelegte Anzahl von Pixeln im Bild zu überprüfen, um festzustellen, ob diese die Eigenschaft Rot aufweisen. Die Ergebnisse dieser Überprüfung werden in einem zweidimensionalen Boolean-Array gespeichert, wobei die x- und y-Koordinaten jeweils einer Dimension zugeordnet sind. Jeder Pixel im Array wird entsprechend seinem Farbzustand mit dem booleschen Wert *true* gekennzeichnet, falls er rot ist, andernfalls wird ihm der Wert *false* zugewiesen. Dieser Prozess wird durch das *Job-System* ausgeführt, wobei jeder einzelne Job eine gewisse Anzahl an Pixeln überprüft.

```

1 struct PixelJob : IJobParallelFor {
2     // Speichert die Pixel Farben des Bildes.
3     public NativeArray<Color> pixels;
4     // Speichert die Positionen der roten Pixel.
5     public NativeArray<Vector2> redPixelPositions;
6     // Die Breite des Bildes in Pixeln.
7     public int textureWidth;
8     // Anzahl der fuer jeden Job zu prueften Pixle
9     public int batchSize;
10

```

⁶⁷Unity Job System

⁶⁸Unity Vector2

```

11 // Methode, die von jedem Job ausgefuehrt wird, um Pixel zu verarbeiten.
12 public void Execute(int batchIndex) {
13     // Berechnet den Startindex fuer den aktuellen Batch.
14     int startPixelIndex = batchIndex * batchSize;
15     // Berechnet den Endindex fÃ¼r den aktuellen Batch und beruecksichtigt dabei
16     das Ende des Pixelarrays.
17     int endPixelIndex = Mathf.Min((batchIndex + 1) * batchSize, pixels.Length);
18
19     // Schleife ueber die Pixel im aktuellen Batch.
20     for (int pixelIndex = startPixelIndex; pixelIndex < endPixelIndex; pixelIndex
21        ++) {
22         // Berechnet die x- und y-Koordinaten des aktuellen Pixels.
23         int x = pixelIndex % textureWidth;
24         int y = pixelIndex / textureWidth;
25
26         // Ueberprueft, ob der Pixel rot genug ist.
27         if (pixels[pixelIndex].r > 0.7f && pixels[pixelIndex].g < 0.5f && pixels[
28             pixelIndex].b < 0.5f) {
29             // Setzt den Pixel auf Rot und speichert seine Position.
30             pixels[pixelIndex] = Color.red;
31             redPixelPositions[pixelIndex] = new Vector2Int(x, y);
32         }
33     }
34 }
35 }
36 }
```

Codeabschnitt 4.9: Rote Pixel suche

- **NativeArray:** bieten verbesserte Leistung im Vergleich zu standardmäßiger Arrays, da diese direkt auf native Speicherbereiche zugreifen können, was besonders wichtig ist, wenn große Datenmengen verarbeitet werden müssen und die Ausführung parallelisiert werden sollen.
- **pixels:** pixels ist ein *NativeArray*, welcher die aktuell noch unbearbeiteten Pixels beinhaltet.
- **batchSize:** Die *batchSize* beinhaltet die Anzahl von Pixel, die jeder *Job* durchgeht und überprüft.
- **texturewidth:** Wird vor dem Aufruf der *Jobs* berechnet, um die Koordinaten des roten Pixels herauszufinden.
- **batchIndex:** Nachdem der Prozess unter mehreren *Jobs* aufgeteilt wird, arbeiten alle *Jobs* von an demselben *NativeArray pixels*. Jeder *Job* bekommt einen *batchIndex* mit, dieser zeigt an, wo die Arbeit dieses *Jobs* innerhalb des *NativeArray* beginnt.
- **startPixelIndex:** Nachdem jeder Arbeiter Thread seine eigenen Batches besitzt, muss der Index für diese berechnet werden.
- **redPixelPositions:** Wird vor der Initialisierung in der Klasse mit derselben Länge wie Pixels erstellt.

4.4.10.4 Kabel finden

Nachdem das Foto verarbeitet wurde, wird nach roten Pixeln in dem Bild gesucht, um danach die noch fehlende Entfernung des Kabels, zu messen. Mit dieser hat man dann die genaue Position des roten Kabels in der echten Welt errechnet. Um dies zu erreichen, wird

nach roten Pixeln im Bild gesucht. Da es aber unsicher ist, ob diese roten Pixel zu einem Kabel gehören oder nicht, wird eine Methode angewendet, um die längste zusammenhängende Reihe von roten Pixeln zu ermitteln. Es kann auch dazu kommen, dass rote Pixel nicht erkannt werden, welche zum roten Kabel gehören. Deswegen wird ein Toleranzbereich festgelegt. Es wird ermittelt, ob in einem bestimmten Bereich, rote Pixel existieren, von welchen aus dann weiter ermittelt wird. Dadurch wird eine präzisere Erkennung und Zuordnung von relevanten roten Pixeln im Kontext des zu identifizierenden Kabels ermöglicht. Die methodische Vorgehensweise zur Berechnung gestaltet sich wie folgt:

```

1 List<Vector2> SearchForRedPixels(int startX, int startY) {
2     // Ist die momentane Line an der er Schaut ob sie einen weiteren Punkt findet
3     List<Vector2> currentLine = new List<Vector2>();
4     bool hasFound = false;
5     // foundX und foundY sind die momentare x und y positionen des Roten pixels
6     // von den man ausgeht
7     int foundX = startX, foundY = startY;
8     // heightY wird verwendet um zu ueberpruefen ob die gepruefte laenge nicht die
9     // exestierente uebertrifft
10    int heightY = redPixels.GetLength(1);
11    // widthX wird verwendet um zu ueberpruefen ob die gepruefte breite nicht die
12    // exestierente uebertrifft
13    int widthX = redPixels.GetLength(0);
14
15    do {
16        hasFound = false;
17        // Damit die funktion nicht in eine unendlich schleife bei den darueber
18        // und darunter ueberpruefen kommt wird die Position von redPixels auf false gesetzt
19        redPixels[foundX, foundY] = false;
20        currentLine.Add(new Vector2(foundX, foundY));
21        // ueberprueft ob 25 pixel "rechts" von den Pixel von den man ausgehen
22        // mindestens ein roter vorhanden ist
23        for (int i = 1; i < 25 && foundX + i < widthX; i++) {
24            if (redPixels[foundX + i, foundY]) {
25                foundX = foundX + i;
26                hasFound = true;
27            }
28        }
29
30        // ueberprueft ob 10 pixel ueber" von den Pixel von den man ausgehen
31        // mindestens ein roter vorhanden ist
32        for (int i = 1; i < 10 && !hasFound && foundY - i > 0; i++) {
33            hasFound = redPixels[foundX, foundY - i];
34            if (hasFound) {
35                foundY = foundY - i;
36            }
37        }
38
39        // ueberprueft ob 10 pixel ueber" von den Pixel von den man ausgehen
40        // ueberprueft ein roter vorhanden ist
41        for (int i = 1; i < 10 && !hasFound && foundY + i < heightY; i++) {
42            hasFound = redPixels[foundX, foundY + i];
43            if (hasFound) {
44                foundY = foundY + i;
45            }
46        }
47    } while (hasFound);
48    return currentLine;
49}

```

Codeabschnitt 4.10: Kabel suche

- **currentLine**: Liste von Vektoren, um Positionen der gefundenen Pixel zu speichern.

- **hasFound**: Boolean-Wert, der angibt, ob ein rotes Pixel gefunden wurde.
- **foundX**: Die aktuelle x-Position des gefundenen roten Pixels.
- **foundY**: Die aktuelle y-Position des gefundenen roten Pixels.
- **heightY**: Die Höhe des Arrays *redPixels*.
- **widthX**: Die Breite des Arrays *redPixels*.

Die Funktion sucht nach roten Pixeln, indem sie von einer Startposition ausgehend Pixel in der Nähe überprüft. Dabei wird eine Liste der gefundenen Pixel Positionen erstellt. Die Schleife wird, solange durchlaufen, bis kein rotes Pixel mehr gefunden wird. Innerhalb der Schleife wird zuerst das aktuelle Pixel als geprüft markiert und seiner Position zur Liste hinzugefügt. Dann wird nach roten Pixeln in der Nähe gesucht. Falls keines gefunden wird, wird die Schleife beendet. Am Ende wird die Liste der gefundenen Pixel zurückgegeben.

4.4.11 Erstellen und Bewegung des Paketes

Um das Paket über das Kabel gleiten zu lassen, muss zuerst der Abstand gemessen werden. Sobald diese gemessen wurde, sind alle wichtigen Punkte berechnet, über die das Paket gleiten soll. Dann muss das Objekt, welches das Paket darstellt, initialisiert und am Anfang des Kabels positioniert werden. Danach wird das Paket über die Punkte bewegt.

4.4.11.1 Definition eines Rays

Ein Ray ist ein abstraktes Konzept in der Computergrafik und Physiksimulation, das einen unendlich langen geraden Strahl repräsentiert. Dieser Strahl wird durch einen Ausgangspunkt definiert, der üblicherweise als *Ursprung* bezeichnet wird. Im Kontext von dreidimensionalen Szenen und Visualisierungen entspricht der Ursprung oft der Position einer *virtuellen Kamera* oder eines *Blickpunkts*. Der Ray erstreckt sich dann in eine bestimmte Richtung, die durch einen Vektor definiert wird. Diese Richtung kann durch verschiedene Methoden festgelegt werden, abhängig von der Anwendung, in der der Ray verwendet wird. Im Falle der Bildschirmkoordinaten kann die Richtung beispielsweise durch den Blick des Benutzers, bestimmt werden. In der Praxis wird der Ray häufig dazu verwendet, um *Kollisionen mit Objekten in einer Szene zu erkennen* oder *um Lichtstrahlen für die Beleuchtungsberechnung zu simulieren*. Durch das Schießen eines Rays in eine Szene und das Überprüfen auf *Kollisionen* mit den vorhandenen Objekten kann festgestellt werden, ob der Ray ein Objekt trifft und wenn ja, an welcher *Stelle* und unter welchem *Winkel*.

4.4.11.2 TrackableType Klasse

In Unity ist **TrackableType** eine Aufzählung, die verschiedene Arten von Objekten repräsentiert, die in Augmented Reality (AR) verfolgt werden können. Zu diesen Typen gehören Bilder, Objekte und Ebenen.

- **Bilder**: Unity ermöglicht die Erkennung und Verfolgung von Bildern in der realen Welt. Entwickler können spezifische Bilder definieren, die von ihrer AR-Anwendung erkannt werden sollen, um darauf basierend virtuelle Inhalte anzuziehen.
- **Objekte**: Dies bezieht sich auf die Verfolgung von physischen Objekten. Entwickler können bestimmte reale Objekte markieren, die von der AR-Anwendung verfolgt werden sollen, um interaktive Erfahrungen zu schaffen.
- **Ebenen**: Unity kann flache Oberflächen wie Böden, Tische oder Wände erkennen und als Grundlage für die Platzierung von AR-Inhalten verwenden. Diese Art von

TrackableType ermöglicht es, virtuelle Objekte stabil auf realen Oberflächen zu positionieren.⁶⁹

4.4.11.3 ARRaycastHit Klasse

Die *ARRaycastHit* Klasse in Unity wird verwendet, um die Ergebnisse eines *Raycasts* (siehe Kapitel 4.4.11.1) in Augmented Reality (AR) Anwendungen zu erhalten. Die Informationen des Raycasts werden dann von der *ARRaycastHit*-Klasse aufgefangen und zur Verfügung gestellt. Die *ARRaycastHit*-Klasse enthält mehrere Methoden, die es ermöglichen, auf die Details des Treffpunkts zuzugreifen und entsprechende Aktionen in einer AR-Anwendung auszuführen. Dazu gehören Informationen wie die Position des Treffpunktes und die Entfernung zum Startpunkt des Strahls. Durch die Verwendung der *ARRaycastHit*-Klasse ist es möglich, präzise Interaktionen in AR-Szenen zu implementieren, wie z. B. das Platzieren virtueller Objekte auf realen Oberflächen.⁷⁰ Diese Klasse besitzt die folgenden Eigenschaften:

- *distance*: Enthält die Entfernung vom Ursprung des Strahls bis zu dem Punkt, an dem er getroffen wurde.
- *hitType*: gibt Typ des Hits an, der durch eine Raycast Treffer bestimmt wurde. Sie ermöglicht die Unterscheidung zwischen verschiedenen Arten von Treffern, wie Kollisionen mit Meshes (Darstellung der Umgebung mit Dreiecken), Terrains (Unity Landschaft) oder Collidern (für physische Kollisionen). Diese Eigenschaft ist nützlich für die differenzierte Behandlung von Kollisionen in Unity-Anwendungen.

4.4.11.4 Entfernung messen

Da man noch die z-Achse, also die Tiefe ermitteln muss, um das visuelle Paket angemessen auf dem physischen zu platzieren und bewegen zu können, werden im späteren Verlauf Raycasts⁷¹ verwendet. Aufgrund der Ressourcenintensität, die mit dem Versenden eines Raycasts für jede Position verbunden wäre, um die z-Achse für jede x/y Koordinate zu erhalten, ist eine Optimierung notwendig. Zur Verbesserung der Effizienz werden die wichtigsten Positionen berechnet. Diese wichtigen Punkte sind der Anfangspunkt des Kabels (am ersten PC), der Mittelpunkt des Kabels, der Endpunkt des Kabels (am zweiten PC), sowie zehn andere Punkte, welche gleichmäßig über das gesamte Kabel verteilt sind. Diese Punkte werden in einer globalen *List<Vector3>* Variable namens *redPixelCoordinates* gespeichert, um auf alle 3 Koordinaten (x/y/z) zugreifen zu können. Diese Liste wird durch Iterieren und für jeden Punkt wird ein Raycast ausgeführt.

```
1 foreach (Vector2 redPixel in redPixelCoordinates) {
2     positionVirtualObject(redPixel, targetTexture)
3 }
```

Codeabschnitt 4.11: Iteration durch die Liste der Raycastpunkte

- Das *positionVirtualObject* ist die Funktion welche die folgenden Berechnungen für Raycasts errechnet und den Raycast ausführt.

Um Raycasts erfolgreich zu nutzen, greift Unity auf Bildschirmkoordinaten zurück. Jedoch liegen uns momentan nur die Koordinaten vor, die der Skalierung des Bildes entsprechen.

⁶⁹UnityTrackableType

⁷⁰UnityARRaycastHit

⁷¹Unity Raycast

Daher müssen diese in ein Format umgerechnet werden, das es ermöglicht, die Raycasts an die richtigen Positionen zu senden. Diese Umrechnung erfolgt im Code wie folgt:

```

1 Vector2 screenCoordinates = new Vector2(
2 redPixel.x * Camera.main.pixelWidth / image.width,
3 redPixel.y * Camera.main.pixelHeight / image.height
4 );

```

Codeabschnitt 4.12: Koordinaten Umrechnung

- Das **redPixel.x** und **redPixel.y** sind die Positionen, welche in *screenCoordinates* umgerechnet werden sollen.
- **Camera.main.pixelWidth** und **Camera.main.pixelHeight** sind die Breite und Höhe des Bildschirms, auf dem die Kameraansicht dargestellt wird.
- Unter **image.width** und **image.height** versteht man die gesamte Breite und Höhe des verarbeiteten Bildes vom roten Kabel in Pixels.

Sobald dann die *screenCoordinates* berechnet wurden, kann der Raycast auf die Position im Bildschirm aufgerufen werden. Dafür ist dieser Code teils von *positionVirtualObject* zuständig:

```

1 // Eine Liste von ARRaycastHit-Objekten wird erstellt, um Treffer von Raycasts zu
2 // speichern.
3
4 // Ueberpruefen, ob ein Raycast auf etwas trifft.
5 if (raycastManager.Raycast(screenCoordinates, hits, TrackableType.Planes)) {
6 // Schleife durch alle getroffenen Punkte.
7 for (int i = 0; i < hits.Count; i++) {
8 // Ueberpruefen, ob der Treffertyp nicht "PlaneWithinInfinity" ist.
9 if (hits[i].hitType != TrackableType.PlaneWithinInfinity) {
10 // Ueberpruefen, ob der Treffer nicht null ist.
11 if (hits[i] != null) {
12 // Die Position des Treffers um 0.05 Einheiten nach oben verschieben (fuer eine
13 // bessere anzeigt) und zur Liste der Kabelpositionen hinzufügen.
14 cablePositions.Add(new Vector3(hits[i].pose.position.x, hits[i].pose.position.y + 0.05
15 f, hits[i].pose.position.z));
16 // Die Schleife verlassen, nachdem ein gueltiger Treffer gefunden wurde.
17 break;
18 }
19 }

```

Codeabschnitt 4.13: Raycast schießen

Es wird eine Liste von **ARRaycastHit** Objekten erstellt, die den Namen *hits* trägt. In dieser Liste werden nach dem Ausführen einer AR-Raycast-Operation die Treffer gespeichert. Die Funktion **Raycast** erhält die *screenCoordinates*, die *hits*-Liste sowie die zu suchenden **TrackableType.Planes**. Wenn die Funktion mindestens einen Treffer erhält, werden alle Treffer überprüft. Diese Treffer werden dann darauf überprüft, ob ihr Typ nicht *TrackableType.PlaneWithinInfinity* ist, um sicherzustellen, dass der Raycast nicht ins Unendliche geht. Wenn ein Objekt getroffen wurde, wird geprüft, ob es Werte besitzt. Wenn alle diese Voraussetzungen erfüllt sind, wird es mit einer um 5 cm Erhöhung (damit das Paket nicht innerhalb von Kabel gleitet, sondern leicht darüber) zu *cablePositions* hinzugefügt. Unter *cablePositions* versteht man die Position, die das Paket später durchqueren wird.

4.4.11.5 Erstellung und Positionierung des Paketes

Nach der Bestimmung der kritischen Positionen mittels Raycast hat der Benutzer die Möglichkeit, über die Website eine Nachricht zu übermitteln. Die Übermittlung dieser Nachricht löst ein entsprechendes Ereignis aus, das zur Initialisierung und zum Absenden des Datenpakets führt.

Bei der *Start* Funktion, welche von Unity zur Verfügung gestellt wird, wird diese Zeile ausgeführt:

```
1 EventManager.OnMessageReceived += editAndSendPackage;
```

Codeabschnitt 4.14: Binden an der Methode

Diese Zeile sorgt dafür, dass beim Start der Unity *Scene* *editAndSendPackage* an den Event *OnMessageReceived* gebunden wird. Das bedeutet, dass wenn das Event *OnMessageReceived* erhalten wird, die Funktion *editAndSendPackage* ausgelöst wird. Die Funktion *editAndSendPackage* initialisiert das visuelle Paket am Anfang des Kabels, falls es bisher nicht erstellt wurde.

```
1 public void editAndSendPackage(string username, string message) {
2 // Ueberpruefen, ob sich die Operation am Ende befindet
3 if (isAtEnd) {
4 // Wenn ja, zuruecksetzen der Flags und Zähler um neues senden zu ermöglichen
5 isAtEnd = false;
6 moveToCounter = 1;
7 moveFromCounter = 0;
8 }
9 // Auf dem Haupt Thread die folgenden Operationen ausführen
10 MainThreadDispatcher.Instance().Enqueue(() => {
11 // Ueberpruefen, ob das Objekt bereits instanziert ist
12 if (isObjectInstantiated == false) {
13 // Wenn nicht, wird das virtuelle Objekt instanzieren
14 instantiatedObject = Instantiate(virtualObject, cablePositions[0], Quaternion.identity
    );
15 isObjectInstantiated = true;
16 }
17 else {
18 // Wenn ja, das aktuelle Objekt zerstören und neu instanziiieren
19 Destroy(instantiatedObject);
20 instantiatedObject = Instantiate(virtualObject, cablePositions[0], Quaternion.identity
    );
21 }
22 // Flag zum Bewegen des Objekts umschalten
23 shouldMove = !shouldMove;
24
25 // Ein Textobjekt für die Informationen erstellen und bearbeiten
26 EditTextOfInformationObject editTextOfInformationObject = new
    EditTextOfInformationObject(infoTextS);
27 editTextOfInformationObject.GetTextMeshProFromChild();
28 editTextOfInformationObject.EditText(username, message);
29
30 // Eine Nachricht über den EventManager senden
31 EventManager.SendMsg(username, message);
32 });
33 }
```

Codeabschnitt 4.15: Initialisierung und Bearbeitung des Paketes

Im Rahmen der Methode wird zunächst geprüft, ob ein Paket erstellt wurde und ob es sich bereits am Ende des vorgesehenen Weges befindet. Sollte dies der Fall sein, wird der

Zähler für die Bewegung des Pakets zurückgesetzt. Aufgrund der Tatsache, dass der *EventManager*, der die Funktion aufruft, nicht im Haupt Thread läuft und die Initialisierung des Pakets sowie die Funktion *EventManager.SendMsg* nur im Haupt Thread ausgeführt werden darf, ist die Verwendung des *MainThreadDispatcher* erforderlich. Dieser stellt sicher, dass dieser Codeabschnitt nicht von einem Neben Thread, sondern vom Haupt Thread ausgeführt wird. Dies ist notwendig, um sicherzustellen, dass die Unity-Szene synchronisiert ist, wie von Unity vorgegeben. Innerhalb des *MainThreadDispatcher* wird, wenn das Objekt bislang nicht erstellt wurde, dieses am Anfang des Kabels erstellt. Wenn das Paket jedoch bereits erstellt wurde, wird es zerstört und wieder auf die Startposition zurückgesetzt. Die Klasse *EditTextOfInformationObject* ist dafür verantwortlich, das *TextMeshPro* des Spielobjekts zu verwenden, um die korrekten Daten im Text anzuzeigen.

4.4.11.6 Bewegung des Paketes

Die Bewegung des Paketes läuft so ab, dass das die Nachricht nacheinander die Punkte durchläuft, welche davor berechnet wurden. Die Bewegung zwischen diesen Positionen wird innerhalb von Unity über die Funktion *Update* ausgeführt. Diese Funktion wird bei jedem Frame aufgerufen, um eine kontinuierliche Aktualisierung der Position des Pakets zu gewährleisten und somit eine nahtlose Bewegung darzustellen zu können.

```
1 void Update() {
2 // Ueberpruefen, ob sich das Objekt bewegt soll
3 if (shouldMove) {
4 // Ueberpruefen, ob der Index wo sich das Paket hinbewegt kleiner als die Anzahl der
5 // Kabelpositionen ist
6 if (moveToCounter < cablePositions.Count) {
7 // Das Paket in die richtung der naechsten position bewegeb
8 instantiatedObject.transform.Translate((cablePositions[moveToCounter] - cablePositions[
9 [moveFromCounter]) * 0.0075f);
10
11 // Ueberpruefen, ob das Objekt nahe genug an der Zielposition ist
12 if (Vector3.Distance(instantiatedObject.transform.position, cablePositions[
13 [moveToCounter]) < 0.00375f) {
14 // Das Objekt an die Zielposition setzen
15 instantiatedObject.transform.position = cablePositions[moveToCounter];
16 // Zaehler erhöhen
17 moveToCounter++;
18 moveFromCounter++;
19
20 // Ueberpruefen, ob das Objekt am Ende der Kabelpositionen ist
21 if (Vector3.Distance(instantiatedObject.transform.position, cablePositions[
22 [cablePositions.Count - 1]) < 0.00375f) {
23 // Wenn ja, markiere, dass es am Ende ist
24 isAtEnd = true;
25 }
26
27 // Wenn der Index zum Bewegen größer oder gleich der Anzahl der Kabelpositionen ist,
28 // beende das die Bewegung
29 shouldMove = false;
30
31 }
```

Codeabschnitt 4.16: Update Funktion

Das Skript enthält globale Variablen, die von verschiedenen Funktionen verwendet werden:

- *shouldMove*: Ein Boolescher Wert, der angibt, ob das Objekt sich bewegen soll oder nicht.
- *moveFromCounter*: Ein Zähler, der verfolgt, wie oft das Paket bereits von einer Position weg bewegt wurde.
- *moveToCounter*: Ein Zähler, der den Index für die nächste Position des Pakets zählt.

Der Code prüft zuerst, ob das Objekt sich bewegen soll. Dann wird überprüft, ob das Objekt bereits am Ende angelangt ist, indem der *moveToCounter* mit der Anzahl der Positionen in *cablePositions* verglichen wird.

Solange das Objekt nicht am Ende ist, wird die Bewegung ausgeführt. Die Bewegung selbst wird durch die Differenz zwischen der nächsten Position, zu der sich das Objekt bewegen soll, und der aktuellen Position, von der sich das Paket weg bewegt, definiert. Diese Bewegung wird mit einem Faktor multipliziert, um die Geschwindigkeit des Objekts anzupassen.

Nach jeder Bewegung wird überprüft, ob das Objekt nahe genug an der Zielposition ist. Sobald es sich innerhalb des zulässigen Bereichs befindet, wird seine Position festgelegt, um zu verhindern, dass es über das Ziel hinaus läuft. Dann werden die Zähler aktualisiert und der Vorgang wird wiederholt, bis das Objekt das Ende erreicht hat. Dies simuliert eine Bewegung entlang des Kabels.

4.4.12 Anzeigen von Informationen der Nachricht

Der Benutzer kann Mithilfe der Website, welche auf beiden Laptops (links und rechts) im Browser angezeigt wird, eine Nachricht innerhalb des Paketes senden. Diese Informationen werden dann angezeigt, wenn der Benutzer auf das virtuelle Paket in der HoloLens-App klickt.

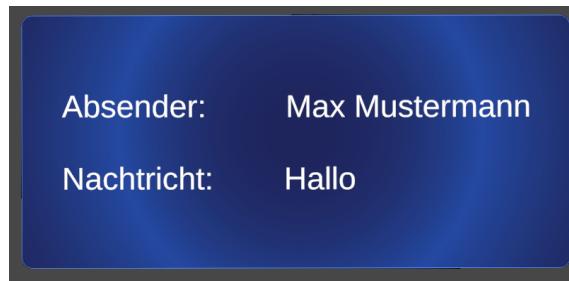


Abbildung 4.20: Anzeige Beispiel für die Nachricht

Dies wird mithilfe des Spielobjekt *InfoObject* umgesetzt, welches Voreinstellungen für den Hintergrund und Text besitzt, falls es zu Fehlern kommt.

Um den Text des Objekts zu ändern, muss auf das *Text Mesh Pro* Spielobjekt welches ein unter Objekt des *InfoObject* zugegriffen werden. Dies passiert durch diesen Code:

```

1 public class EditTextOfInformationObject : MonoBehaviour
2 {
3     // Referenz auf das Elternspielobjekt, von dem aus das Kindspielobjekt gefunden werden
4     // soll
5     public Spielobjekt parentGameObject;
6     // Referenz auf das TextMeshPro-Objekt, das bearbeitet werden soll
7     TextMeshPro textMeshPro;
8
9     // Methode zum Abrufen des TextMeshPro-Objekts vom Kindspielobjekt
10    public void GetTextMeshProFromChild() {
11        // Überprüfen, ob das Elternspielobjekt vorhanden ist
12    }
13}
```

```

11 if (parentGameObject != null) {
12 {
13 // Das Childobjekt mit dem Namen InformationTextSim Elternspielobjekt finden
14 Spielobjekt childGameObject = parentGameObject.transform.Find("InformationTextS").
    Spielobjekt;
15 // Ueberpruefen, ob das Kindspielobjekt gefunden wurde
16 if (childGameObject != null) {
17 // Das TextMeshPro-Komponente vom Kindspielobjekt abrufen
18 TextMeshPro tmp = childGameObject.GetComponent<TextMeshPro>();
19 // Ueberpruefen, ob das TextMeshPro-Komponente vorhanden ist
20 if (tmp != null) {
21 // Das TextMeshPro-Objekt der Klasse zuweisen
22 textMeshPro = tmp;
23 }
24 }
25 }
26 }
27 }

```

Codeabschnitt 4.17: Kind vom Spielobjekt bekommen

Die Klasse enthält zwei Variablen:

- `parentGameObject`: Eine öffentliche Variable vom Typ `Spielobjekt`, die als Referenz auf das Elternspielobjekt dient, von dem aus das TextMeshPro-Objekt gefunden werden soll.
- `textMeshPro`: Eine Variable vom Typ `TextMeshPro`, die das TextMeshPro-Objekt speichert, auf das zugegriffen werden soll.

Die Klasse enthält auch eine Methode namens `GetTextMeshProFromChild()`, die das TextMeshPro-Objekt aus dem Kindspielobjekt mit dem Namen *InformationTextS* im angegebenen Elternspielobjekt abruft und in der Variablen `textMeshPro` speichert.

4.4.13 Andere Versuche / Probleme bei der Programmierung

In diesem Kapitel werden neben den Prototypen und den festgestellten Problemen während der inkrementellen Entwicklung dieses Teilbereichs auch gescheiterte/alternative Versuche und Ansätze innerhalb des Projekts behandelt.

4.4.13.1 Kabelerkennung Version 1

Die erste Version der Kabelerkennung basiert auf der Analyse von einem Foto, die mit der HoloLens aufgenommen wurden.

```

1 void editTextureV2(Texture2D textureToEdit) {
2 int textureWidth = textureToEdit.width;
3 List<int> yCoordinates = new List<int>();
4 List<int> xCoordinates = new List<int>();
5 for (int y = 0; y < textureToEdit.height; y++) {
6 for (int x = 0; x < textureWidth; x++) {
7 int index = y * textureWidth + x;
8 // ueberprueft ob der gepruefte Pixel rot ist
9 if (pixels[index].r > 0.55f && pixels[index].g < 0.5f && pixels[index].b < 0.5f) {
10 yCoordinates.Add(y);
11 xCoordinates.Add(x);
12 }
13 }
14 }
15 int averageX = getCenterCoordinate(xCoordinates);

```

```

16 int averageY = getCenterCoordinate(yCoordinates);
17 // Fuegt den kleinsten X und den durchschnittlichen Y Punkt als Start Punkt zu
    // redPixelCoordinates hinzu
18 redPixelCoordinates.Add(getSideCoordinate(xCoordinates, true, averageY));
19 // Fuegt den durchschnittlichen X und den durchschnittlichen Y Punkt als Mittelpunkt
    // Punkt zu redPixelCoordinates hinzu
20 redPixelCoordinates.Add(new Vector2(averageX, averageY));
21 // Fuegt den hÃ¶chsten X und den durchschnittlichen Y Punkt als Mittelpunkt Punkt zu
    // redPixelCoordinates hinzu
22 redPixelCoordinates.Add(getSideCoordinate(xCoordinates, false, averageY));
23 }

```

Codeabschnitt 4.18: Erste Version der Kabel Erkennung

- **textureWidth**: Die Breite der bearbeiteten Textur.
- **yCoordinates**: Eine Liste, die die Y-Koordinaten der identifizierten roten Pixel speichert.
- **xCoordinates**: Eine Liste, die die X-Koordinaten der identifizierten roten Pixel speichert.
- **averageX**: Der Durchschnitt der X-Koordinaten der identifizierten roten Pixel.
- **averageY**: Der Durchschnitt der Y-Koordinaten der identifizierten roten Pixel.
- **redPixelCoordinates**: Eine Liste von Vektoren, die die berechneten Positionen der roten Pixel speichert.

Der Code analysiert jede Pixelposition einer Textur, um festzustellen, ob es sich um ein rotes Pixel handelt. Ein Pixel wird als rot betrachtet, wenn seine Rotkomponente über einem bestimmten Schwellenwert liegt und seine Grün- und Blaukomponenten unter anderen Schwellenwerten liegen. Die Koordinaten der identifizierten roten Pixel werden gespeichert, und dann wird der Durchschnitt der X- und Y-Koordinaten berechnet. Basierend auf diesem Durchschnitt werden drei Positionen berechnet, die den Verlauf des Kabels darstellen sollen. Diese Positionen werden in der Liste redPixelCoordinates gespeichert.

Es ist anzumerken, dass die sequenzielle Ausführung dieses Codes ineffizient ist und zu längeren Ladezeiten führen kann. Außerdem kann die Genauigkeit der berechneten Positionen begrenzt sein, was die Fähigkeit beeinträchtigen kann, das Kabel genau zu verfolgen.

4.4.14 Debugging Optionen

Während der Entwicklung dieses Projekts wurden spezifische Funktionen implementiert, um Fehler und Bugs im Code zu identifizieren. Zu diesem Zweck wurden zwei Funktionen mit den Bezeichnungen *debugRaycast* und *DebugCameraResolution* entwickelt. Weiterhin wurde ein *Plain Spielobjekt* namens *PixelDebug* erstellt, das als Hilfsmittel dient, um Fehler im Code zu diagnostizieren und zu beheben. Diese zusätzlichen Klassen tragen wesentlich dazu bei, die Fehlerfreiheit des entwickelten Systems sicherzustellen, indem sie eine strukturierte und systematische Fehlerbehandlung ermöglichen.

4.4.14.1 Anzeigen der Kabelerkennung

Das *Spielobjekt* mit dem Namen *PixelDebug* dient als Hilfsmittel zur Visualisierung der Erkennung roter Pixel. Diese Einrichtung ermöglicht die Darstellung aller identifizierten roten Pixel als 2D-Textur auf dem *Spielobjekt*, wodurch die vom Skript erkannten roten Pixel sichtbar werden. Die Steuerung der Aktivierung bzw. Deaktivierung der Objektdarstellung kann über das Skript *Deactivate Object* erfolgen. Diese Prozedur bietet eine effektive Methode zur Überprüfung der Pixelerkennung in der Umgebung.

4.4.14.2 Debug Funktion von Raycasts

Um während des Prozesses von Programmieren der Raycasts sicherzustellen, dass in der Funktion *positionVirtualObject* die korrekte Position getroffen wurde, kann man die Funktion *debugRaycast* verwenden. Diese Funktion dient der Visualisierung der getroffenen Punkte des Raycasts, indem es an den getroffenen Punkt ein Objekt anzeigt. Zur Ausführung dieser Funktion muss das *hit-Objekt* des Raycasts übergeben werden. Dies gewährleistet eine präzise Überprüfung und gegebenenfalls Anpassung der Raycast-Positionen für eine genauere Abbildung der realen Umgebung. Die Verwendung der *debugRaycast*-Funktion ermöglicht somit eine verbesserte Qualität und Zuverlässigkeit in Anwendungen, die auf Raycasting basieren.

```

1 void debugRaycast(ARRaycastHit hit) {
2     Debug.Log("Ray hit position: " + hit.pose.position);
3     Spielobjekt instantiatedObject = Instantiate(virtualObject, hit.pose.position,
4         Quaternion.identity);
4 }
```

Codeabschnitt 4.19: Debug Funktion von Raycasts

4.4.14.3 Debug Funktion der Kamerabenutzung

Diese Funktion kann verwendet werden, um sicherzustellen, dass die Anwendung korrekt auf die Kamera zugreift und die richtige Bildauflösung erhält.

```

1 void DebugCameraResolution() {
2     // Ueberpruefen, ob Kamerageraete verfuegbar sind
3     if (WebCamTexture.devices.Length > 0) {
4         // Das erste Kamerageraet auswaehlen
5         WebCamDevice frontCamera = WebCamTexture.devices[0];
6
7         // Eine WebCamTexture mit dem ausgewahlten Kamerageraet erstellen
8         WebCamTexture webcamTexture = new WebCamTexture(frontCamera.name);
9
10        // Die Textur abspielen, um sicherzustellen, dass die Kamera ordnungsgemaÈ
11        // initialisiert wird
12        webcamTexture.Play();
13
14        // Ueberpruefen, ob die Kamera eine gueltige Aufloesung zurueckgibt
15        if (webcamTexture.width > 0 && webcamTexture.height > 0) {
16            // Aufloesung protokollieren
17            Debug.Log("Selected Resolution: " + webcamTexture.width + "x" + webcamTexture.height);
18        }
19        else {
20            // Fehlermeldung ausgeben, wenn die Kamera keine gueltige Aufloesung zurueckgibt
21            Debug.LogError("Failed to get a valid camera resolution.");
22        }
23    }
24    else {
25        // Fehlermeldung ausgeben, wenn keine Kamerageraete gefunden werden
26        Debug.LogError("No camera devices found.");
27    }
}
```

Codeabschnitt 4.20: Debug Funktion von der Kamera Benutzung

Im Code wird die Klasse *WebCamTexture* verwendet, um auf die Kamera zuzugreifen und die Funktionalität zu überprüfen. Die *WebCamTexture* ist eine Klasse in Unity, die die Verwendung von Kamerabildern in Echtzeit ermöglicht. Sie ermöglicht es, Live-Bilder von

einer Kamera zu erfassen und sie als Textur zu verwenden, dazu beinhaltet es alle für Unity verfügbaren Kameras.

Zuerst wird geprüft, ob ein Kameragerät verfügbar ist. Wenn ja, wird die erste verfügbare Kamera ausgewählt und eine Textur mit dieser Kamera erstellt. Anschließend wird die Textur abgespielt, um sicherzustellen, dass die Kamera korrekt initialisiert wurde. Wenn die Kamera eine gültige Auflösung (größere Auflösung als 0,0) zurückgibt, wird diese Auflösung protokolliert. Ansonsten wird eine Fehlermeldung ausgegeben. Wird keine Kamera gefunden, wird ebenfalls eine Fehlermeldung ausgegeben.

4.4.15 Starten und Nutzen der Webseite

→ HAYLAZ

Dieser Abschnitt beschreibt den Prozess des Starts und der Nutzung der Webseite gemäß dem Anwendungsfall 4.4. Die Webseite spielt eine zentrale Rolle im ersten Anwendungsszenario, dem Nachrichtenaustausch. Ziel ist es, dass zwei Geräte miteinander kommunizieren können. Für eine optimale Benutzererfahrung bedarf es daher einer intuitiven Benutzeroberfläche. Die Entscheidung, ein Web-Interface zu verwenden, basiert auf diesen zwei Faktoren:

- **Breiter Geräte-Support:** Web-Interfaces sind auf einer Vielzahl von Geräten nutzbar, darunter Desktop-Computer und Laptops. Diese Vielseitigkeit gewährleistet eine breite Verfügbarkeit und Nutzbarkeit für Benutzer unabhängig von ihrem verwendeten Gerät, Betriebssystem oder der Rechenleistung.
- **Einfache Umsetzung:** Die Entwicklung eines Web-Interfaces gestaltet sich im Vergleich zu anderen Plattformen als verhältnismäßig unkompliziert. Dies liegt daran, dass Webtechnologien weitverbreitet, gut dokumentiert und leicht zugänglich sind.

Die gesamte Anwendung ist bewusst einfach und minimalistisch gehalten. Folglich befindet sich der gesamte Code der Webseite in einer einzigen Datei, *Level1-FrontEnd/index.html*. Im Folgenden wird erläutert, wie diese Datei ausgeführt wird, um die Webseite im lokalen Netzwerk bereitzustellen.

4.4.15.1 Webseite starten

Um die Webseite zu starten, wird ein lokaler Web-Server benötigt. Ein beliebter und einfacher Weg, dies zu erreichen, ist die Verwendung des Node.js basierten HTTP-Daemons namens *http-server*. Dieser HTTP-Server ist leichtgewichtig, einfach einzurichten und erfordert keine komplizierte Konfiguration. Durch die Verwendung von *http-server* kann die Webseite schnell lokal gehostet werden, was besonders praktisch ist, wenn sie während der Entwicklung getestet werden muss.⁷²

Installation des HTTP-Servers

Eine der einfachsten Möglichkeiten, den *http-server* zu installieren, ist die Verwendung des Node Package Managers (npm). npm ist ein leistungsfähiges Werkzeug, das es Entwicklern ermöglicht, JavaScript-Pakete einfach zu installieren, zu verwalten und zu aktualisieren. Es ist integraler Bestandteil der Node.js-Umgebung und bietet Zugang zu einer Vielzahl von Paketen und Bibliotheken, die von der Community entwickelt wurden. Durch die Installation des *http-server*-Pakets über npm können Entwickler schnell einen lokalen Web-Server einrichten, ohne sich um die Einzelheiten der Serverkonfiguration kümmern zu müssen.

Um den HTTP-Server mithilfe von npm auf einem Windows-Gerät zu installieren, muss ein Befehl in der Eingabeaufforderung (cmd) ausgeführt werden. Dieser Befehl ermöglicht

⁷²npmjs **http-server**

die Installation des Servers auf dem Gerät und lautet wie folgt: `npm install http-server -g`. Durch die Ausführung dieses Befehls wird der http-server global auf dem Windows-Gerät installiert, was bedeutet, dass er von überall im System aus aufgerufen werden kann.

Starten des HTTP-Servers

Nach der Installation des Servers kann die Webseite im Ordner, in dem sich die `index.html`-Datei befindet (standardmäßig im *Level1-FrontEnd* Ordner), mit dem Befehl `http-server` und den Flags `-c -1 -cors` gestartet werden.

Die Flags haben folgende Bedeutungen:

Das Flag `-c` legt die Cache-Zeit der Webseite fest; durch Kombination mit `-1` wird das Caching deaktiviert. Die Option `-cors` aktiviert CORS (Cross-Origin Resource Sharing) auf dem HTTP-Server. Weitere Informationen zu CORS finden Sie im Abschnitt [4.4.15.2](#).

Der vollständige Befehl lautet dann: `http-server -c-1 -cors index.html`.

In dem gegebenen Befehl wird `index.html` als Argument übergeben. Dies bedeutet, dass der `http-server` darauf eingestellt ist, die Datei `index.html` zu liefern, wenn die Wurzel-URL des Servers aufgerufen wird. Der Server wird den Inhalt dieser Datei dem Client zurückgeben, der die Anfrage gestellt hat. Wie bereits erwähnt ist der gesamte Inhalt der Webseite in dieser Datei enthalten, da es sich um eine einfache Single-Page-Applikation (SPA) handelt.⁷³

Der Server wird standardmäßig auf dem Port 8080 gestartet. Falls ein anderer Port gewünscht ist, kann dieser mit dem Flag `-p` und der gewünschten Portnummer geändert werden.

Konfigurationen auf der Webseite

Nach dem Starten des Webservers kann die Webseite in einem beliebigen Browser über die IP-Adresse des Geräts oder über `localhost` aufgerufen werden. Die IP-Adresse des Geräts kann mit dem Befehl `ipconfig` in der *Eingabeaufforderung (cmd)* abgefragt werden.

Um die Webseite erfolgreich zu verwenden, ist es zunächst erforderlich, die IP-Adresse der HoloLens einzugeben. Dies kann über den Button in der oberen linken Ecke der Webseite erfolgen, wie in Abbildung [4.21](#) dargestellt. Die IP-Adresse wird in das entsprechende Eingabefeld eingetragen, und durch Klicken auf den Button *Verbinden* kann die Verbindung getestet werden.

⁷³Bloomreach SPA

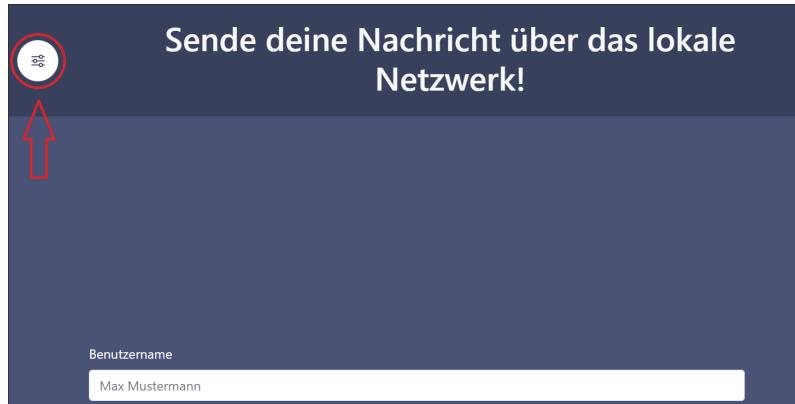


Abbildung 4.21: Position des Buttons



Abbildung 4.22: Fenster der Webseite Einstellungen

Um die Webseite in diesem Szenario mit zwei Geräten zu nutzen, ist es erforderlich, die Seite des Kabels festzulegen, an der sich das aktuelle Gerät befindet. Dies gewährleistet eine korrekte Simulation der Nachrichten in der virtuellen Umgebung. In der Dropdown-Liste, wie in Abbildung 4.22 dargestellt, kann zwischen *Rechte Seite* und *Linke Seite* gewählt werden, je nach Position des Trägers der Brille. Weitere Details zur Verarbeitung dieser Information sind im Abschnitt 4.4.17 zu finden.

Nach der erfolgreichen Konfiguration der Webseite können nun Nachrichten versendet werden.

4.4.15.2 CORS

CORS, oder Cross-Origin Resource Sharing, ist ein Sicherheitsmechanismus, der in modernen Webbrowsern implementiert ist. Seine Hauptaufgabe besteht darin, die Interaktionen zwischen verschiedenen Domains zu regulieren. In der Standardeinstellung erlauben Webbrowser aus Sicherheitsgründen nur Anfragen an Ressourcen, die sich auf derselben Domain befinden, von der die Anfrage stammt. Dies wird als Same-Origin-Policy bezeichnet.

CORS erweitert diese Grundfunktionalität, indem es den Zugriff auf Ressourcen von anderen Domains ermöglicht, vorausgesetzt, der Server, auf dem sich die Ressourcen befinden, erlaubt dies explizit. Dies ermöglicht eine sicherere und kontrolliertere Interaktion zwischen verschiedenen Domains.⁷⁴ ⁷⁵

⁷⁴ AWS CORS

⁷⁵ Mozilla CORS-2

Implementierung von CORS

In unserem speziellen Fall, da unsere Kommunikation ausschließlich im lokalen Netzwerk stattfindet, ist es nicht unbedingt notwendig, strenge Sicherheitsmaßnahmen zu ergreifen. Dennoch ist es wichtig, dass wir bestimmte Header in allen Antworten einfügen, die von der Brille zurückgeschickt werden. Alle Anfragen werden im *RequestManager.cs* Skript abgearbeitet. Genauere Informationen sind im Abschnitt 4.4.17 zu finden.

Diese Header welche einzubauen sind lauten:

```
1 context.Response.Headers.Add("Access-Control-Allow-Origin", "*");
2 context.Response.Headers.Add("Access-Control-Allow-Methods", "GET, POST, OPTIONS");
3 context.Response.Headers.Add("Access-Control-Allow-Headers", "Content-Type");
```

- **“Access-Control-Allow-Origin“, “*”**: Dieser Header gibt an, von welchen Ursprüngen aus die Ressource zugänglich ist. Durch das Einstellen von “*” wird allen Ursprüngen erlaubt, auf die Ressource zuzugreifen, was als “Wildcard” für alle Ursprünge fungiert.
- **“Access-Control-Allow-Methods“, “GET, POST, OPTIONS”**: Dieser Header gibt an, welche HTTP-Methoden für Cross-Origin-Anfragen erlaubt sind. Da wir nur GET, POST und OPTIONS-Anfragen an den Server senden, sind auch nur diese erlaubt, was bedeutet, dass Anfragen mit diesen Methoden von anderen Ursprüngen aus zugelassen werden.
- **“Access-Control-Allow-Headers“, “Content-Type”**: Dieser Header gibt an, welche zusätzlichen Header in einer Anfrage aus einem anderen Ursprung erlaubt sind. In diesem Fall ist es nur der Header “Content-Type”, was bedeutet, dass Anfragen von anderen Ursprüngen nur diesen spezifischen Header enthalten dürfen.

Durch das Hinzufügen dieser drei Zeilen zu unserer HTTP-Antwort fügen wir CORS-Header hinzu, die Cross-Origin-Anfragen für Ressourcen von unserem Server ermöglichen. Sie definieren auch genau, welche HTTP-Methoden und Header in diesen Anfragen erlaubt sind.

Umgang mit CORS-Fehlern

Falls beim Senden der Anfragen ein CORS-Fehler auftritt, liegt das vermutlich an der Konfiguration des Servers. Das Problem kann in Ausnahmefällen (zum Beispiel bei kleineren Testläufen) auf folgende Art und Weise umgangen werden:

- **Deaktivieren im Browser**: Die Deaktivierung von CORS im Browser würde das Problem beheben, birgt jedoch gewisse Risiken. CORS schützt den Benutzer vor schädlichen Webseiten, daher wird davon abgeraten. Die Deaktivierung erfolgt je nach Browser und ist nicht einfach zu finden. Stattdessen sollten die nachfolgende Lösung in Betracht gezogen werden.
- **Verwendung von Plug-ins**: Es gibt Plug-ins, die CORS-Header von der Clientseite hinzufügen. Wir haben das Plug-in *moesif-origin-cors-change* getestet, das nur für den Chrome-Browser von Google verfügbar ist. Nach der Installation kann das Plug-in bei Bedarf aktiviert/deaktiviert werden.

Es ist wichtig zu beachten, dass beide Lösungsvorschläge gewisse Sicherheitsrisiken bergen und nur im lokalen Netzwerk getestet wurden. Sie sind höchstens für erste Prototypen akzeptabel, wenn überhaupt. Nach der Verwendung unseres Programms sollten beide Konfigurationen wieder rückgängig gemacht werden.

4.4.16 Frontend der Webseite

→ LAMPEL

In diesem Abschnitt wird die Struktur und Entwicklung eines Frontends der Webseite zum Ping Anwendungsszenario erläutert. Besonderes Augenmerk wurde auf die zugrundeliegenden Entscheidungen bezüglich der Programmiersprachen sowie des Designprozesses gelegt.

Nach mehreren Entwicklungszyklen im ersten Anwendungsszenario wurde entschieden, eine Webseite in das Projekt zu integrieren. Die Funktionalität der Webseite sollte nahtlos in das Szenario integriert werden. Hierzu können Nachrichten und Absender direkt auf der Webseite eingetragen werden, um sie anschließend in die HoloLens-Applikation zu übertragen.

4.4.16.1 Designprozess

Beim Entwurf einer Webseite ist es essentiell, vor der eigentlichen Umsetzung die Farbpalette und das generelle Aussehen der Seite zu konzipieren.⁷⁶ ⁷⁷ Da der Fokus auf der Anwendung auf der HoloLens liegt, wurde bewusst darauf geachtet, der Webseite nicht zu viel Aufmerksamkeit zu schenken. Sie sollte lediglich die grundlegenden Funktionen erfüllen, ohne dabei komplex zu sein. Aus diesem Grund ist die Webseite simpel gestaltet und verfügt über wenige Eingabefelder und wenig Text.

Webseitenprototyp

Der Prototyp der Webseite, wie in Abbildung 4.23 dargestellt, gibt einen Überblick über das Frontend-Design. Das Zahnrad links oben öffnet ein Fenster, in dem die IP-Adresse der HoloLens eingegeben werden kann, um Nachrichten korrekt zu versenden. Auf der Hauptseite gibt es Eingabefelder für Benutzername und Nachricht. Der zeitlich sortierte Nachrichtenverlauf wird am rechten Rand angezeigt.

⁷⁶ Webdesign Journal **Website Konzept**

⁷⁷ Textstrategin **Aufbau einer Website Inhalte und Struktur richtig Planen und Erstellen**

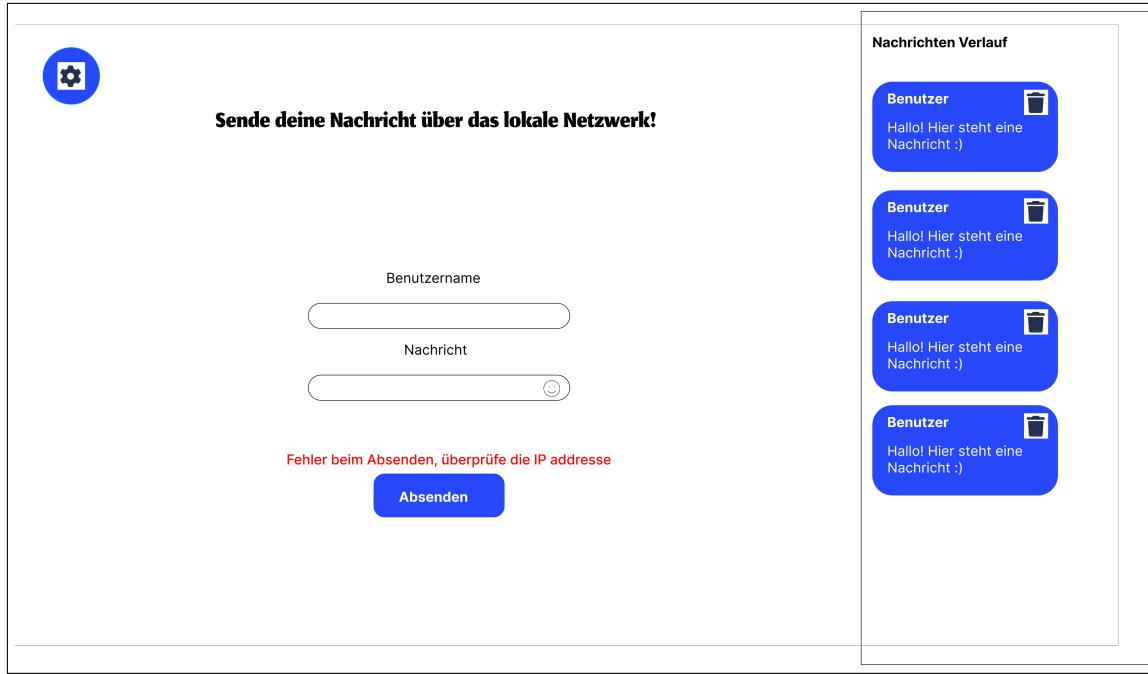


Abbildung 4.23: Darstellung der Webseitenansicht aus Benutzersicht (Mockup)

Ausprogrammierte Webseite

Die endgültige Version der Webseite, wie in Abbildung 4.24 gezeigt, enthält alle gewünschten Funktionen. Die Farbpalette orientiert sich stark am Hauptmenü und den Standardfarben der HoloLens. Es werden verschiedene Blautöne verwendet, um eine konsistente und benutzerfreundliche Benutzererfahrung zu gewährleisten.

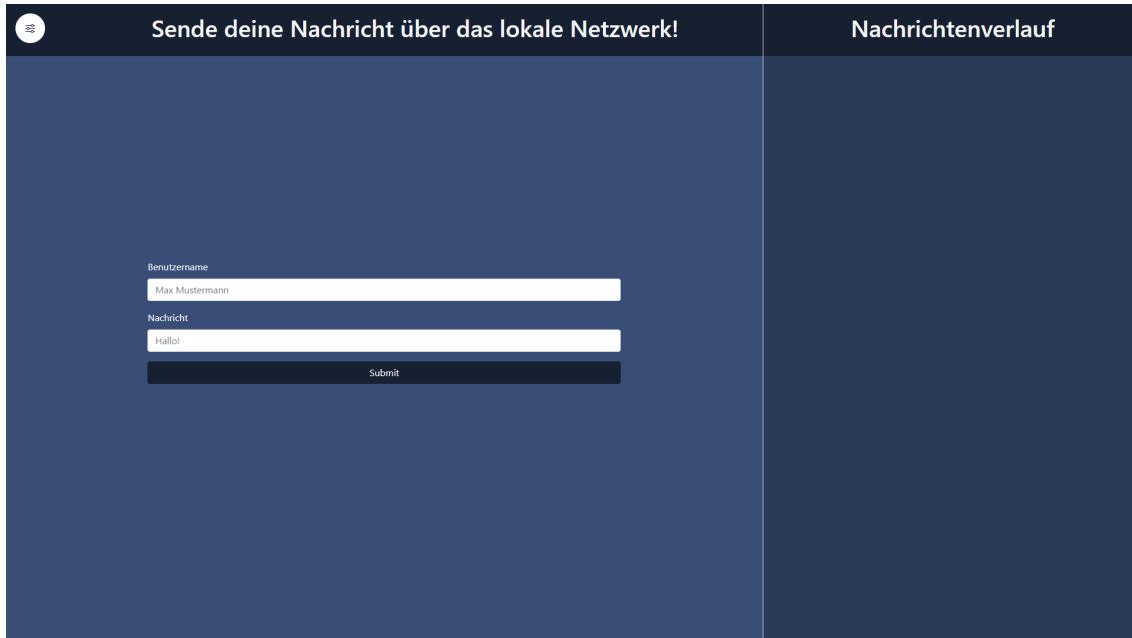


Abbildung 4.24: Darstellung der fertigen Webseitenansicht aus Benutzersicht

Die Umsetzung des Designs hat zum Ziel, Verwirrungen zu vermeiden und eine nahtlose

Integration der Webseite in das Gesamtsystem zu gewährleisten.

4.4.16.2 Verwendete Programmiersprachen

Für die Entwicklung der Webseite wurden hauptsächlich die Programmiersprachen *HTML*, *CSS* und *JavaScript* verwendet. Zur Verbesserung des Designs und der Benutzererfahrung wurde außerdem das *Framework Bootstrap* eingesetzt.

HTML (Hypertext Markup Language)

HTML ist die grundlegende Struktursprache des World Wide Web. Es ermöglicht die Erstellung von Webseiten durch die Verwendung von Strukturelementen (Tags) für Überschriften, Absätze, Listen, Links und vieles mehr. Browser können HTML-Inhalte dem Standard entsprechend für den Anwender darstellen.⁷⁸

CSS (Cascading Style Sheets)

CSS ist eine Stylesheet-Sprache, die verwendet wird, um das Aussehen und die Formatierung von HTML-Dokumenten zu steuern. Es ermöglicht die Definition von Farben, Schriftarten, Layouts und anderen visuellen Eigenschaften einer Webseite.⁷⁹

CSS funktioniert durch das Zuweisen von Regeln und Stilen zu HTML-Elementen. Diese Regeln können in einer externen CSS-Datei definiert oder direkt im HTML-Dokument eingebettet werden.

JavaScript

JavaScript ist eine dynamische Programmiersprache, die hauptsächlich für die clientseitige Entwicklung von Webanwendungen verwendet wird. Sie ermöglicht die Interaktion mit dem Benutzer, das Ändern von Inhalten in Echtzeit und die Steuerung des Verhaltens einer Webseite.⁸⁰

JavaScript verbessert die Funktionalität einer Webseite, indem es auf Benutzerinteraktionen reagiert, Formulare validiert, Animationen erstellt und vieles mehr.

Integration von Bootstrap

Bootstrap ist ein Open-Source-Framework für die Frontend-Entwicklung von Webseiten und Webanwendungen. Es bietet vorgefertigte HTML- und CSS-Vorlagen für Typografie, Formulare, Buttons, Navigation und andere UI-Komponenten.⁸¹

Bootstrap erleichtert die Erstellung responsiver und ästhetisch ansprechender Webseiten, indem es eine Reihe von vorgefertigten Designelementen und Layoutoptionen bereitstellt.

Die Integration von Bootstrap in eine HTML-Datei erfolgt durch das Einbinden der Bootstrap-Bibliotheksdateien in den <head>-Bereich des HTML-Dokuments. Dies kann entweder über ein CDN (Content Delivery Network) oder durch das Herunterladen und Einbinden der lokalen Bootstrap-Dateien erfolgen.⁸²

Anschließend können Bootstrap-Komponenten und -Klassen innerhalb der HTML-Datei verwendet werden, um das Design und die Funktionalität der Webseite zu verbessern.

⁷⁸Mozilla Developer Network **HTML**

⁷⁹Mozilla Developer Network **CSS**

⁸⁰Mozilla Developer Network **Javascript**

⁸¹Bootstrap-Dokumentation **Bootstrap**

⁸²Bootstrap Dokumentation **CDN-Links**

4.4.16.3 Funktionen der Webseite

Die Eingabefelder des Frontends brauchen Javascript Funktionen, um die gewünschten Tätigkeiten durchzuführen und so zu funktionieren wie eigentlich geplant.

```

1 function isNumberKey(evt) {
2     var charCode = (evt.which) ? evt.which : evt.keyCode;
3     if (charCode != 46 && charCode > 31 && (charCode < 48 || charCode > 57)) {
4         return false;
5     }
6     return true;
7 }
```

Codeabschnitt 4.21: Javascript | Ueberpruefung, ob die Eingabe eine Zahl oder . ist

Die Funktion `isNumberKey()` überprüft, ob die gedrückte Taste eine Zahlentaste ist. Sie wird auf der Webseite zur Überprüfung der Eingabe von der IP-Adresse genutzt, um sicherzustellen, dass nur Zahlen eingegeben werden können.

Die Funktion `isNumberKey()` nimmt ein Event-Objekt als Parameter entgegen, das Informationen über das Tastaturereignis enthält. Der `charCode` wird aus dem Event-Objekt extrahiert, der den Unicode-Wert der gedrückten Taste darstellt. Die Funktion überprüft, ob die gedrückte Taste eine Zahl ist (der Unicode-Wert liegt im Bereich von 48 bis 57) oder der Punkt ist (der Unicode-Wert ist 46). Falls die gedrückte Taste im Bereich von 48 bis 57 liegt oder der Punkt ist, gibt die Funktion `true` zurück, was bedeutet, dass die Eingabe akzeptiert wird. Andernfalls gibt die Funktion `false` zurück, was bedeutet, dass die Eingabe nicht akzeptiert wird.

```

1 async function validateAndSendMessage() {
2     var username = document.getElementById("username").value.trim();
3     var message = document.getElementById("message").value.trim();
4
5     if (username !== '' && message !== '') {
6         const messageSent = await sendMessage(username, message);
7         if (!messageSent) {
8             alert("Message could not be sent.");
9         }
10    } else {
11        alert("Please fill in both fields!");
12    }
13 }
```

Codeabschnitt 4.22: Javascript | Validierung und Senden der Message

Die Funktion `validateAndSendMessage()` in dem Codeabschnitt 4.22 validiert die Eingaben in den Feldern für Benutzername und Nachricht und sendet die Nachricht nur ab, wenn beide Felder ausgefüllt sind.

Zunächst ruft die Funktion die Werte der Formularfelder für Benutzername und Nachricht ab und entfernt führende und abschließende Leerzeichen mit der `trim()`-Methode. Anschließend überprüft sie, ob sowohl der Benutzername als auch die Nachricht nicht leer sind. Wenn beide Felder nicht leer sind, wird die Funktion `sendMessage(username, message)` aufgerufen, um die Nachricht zu senden. Wenn die Nachricht erfolgreich gesendet wurde, wird nichts weiter unternommen. Andernfalls wird eine Warnmeldung angezeigt, dass die Nachricht nicht gesendet werden konnte. Sollte eines der Felder leer sein, wird dem Benutzer eine Warnmeldung angezeigt, die ihn darauf hinweist, beide Felder auszufüllen.

4.4.17 Backend der Webseite

→ HAYLAZ

Nachdem im vorherigen Abschnitt (4.4.16) der Designprozess der Webseite erläutert wurde, wird nun das Thema durch eine Beschreibung des Backend-Bereichs erweitert.

Die Webseite fungiert als Schnittstelle für die Kommunikation zwischen den beiden Laptops, die für das Szenario benötigt werden. Diese Kommunikation erfolgt über die HoloLens, die gewissermaßen als Vermittler fungiert und die Datenübertragung ermöglicht.

4.4.17.1 Protokoll zur Datenübertragung

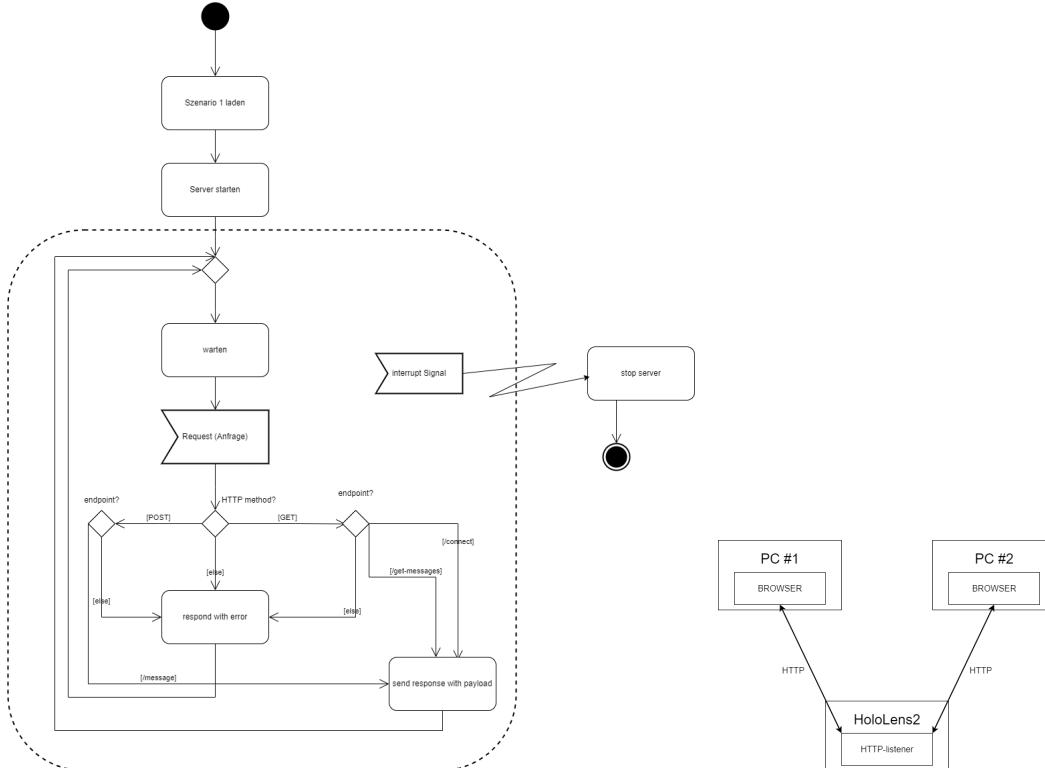


Abbildung 4.25: Aktivitätsdiagramm des Servers

Abbildung 4.26: Kommunikationsstruktur

Abbildung 4.25 zeigt den Ablauf aller möglichen Szenarien, wie eine solche Datenübertragung zustande kommen kann. Wie in Abbildung 4.26 zu erkennen ist, ist die HoloLens das Zentrum der Datenübertragung, über die die gesamte Kommunikation erfolgt.

Im Folgenden werden die einzelnen Schritte der Übertragung und ihre Funktionen beschrieben:

- **Start des Servers:** Auf der HoloLens befindet sich ein GameObject namens *httpListener* mit einem Skript namens *RequestManager.cs*. Beim Laden des ersten Szenarios wird ein HTTP-Listener in einem eigenen Thread gestartet. Erst wenn dieser läuft, kann auf Anfragen der Webseite reagiert werden. Eine ausführlichere Beschreibung der Funktionalität des Servers findet sich im Abschnitt 4.4.17.2.
- **Verbindungsaufbau:** Bevor ein Nachrichtenpaket gesendet werden kann, muss die IP-Adresse der HoloLens registriert werden. Obwohl die Verbindungstestfunktion nicht zwingend erforderlich ist, um Nachrichten an diese IP zu senden, kann ein visuelles Feedback dem Benutzer anzeigen, ob die eingegebene IP-Adresse gültig ist

und ob der Server erreichbar ist. Dies trägt zur Benutzerfreundlichkeit bei, indem es dem Benutzer ermöglicht, leichter zu erkennen, ob alles ordnungsgemäß funktioniert oder ob möglicherweise Probleme auftreten.

- **Versenden von Nachrichten:** Um eine Nachricht von einem Laptop auf den anderen zu senden, muss ein *HTTP POST-Request* an den Endpunkt */message* mit dem Benutzernamen, der Nachricht und die Position (Links oder Rechts) als Payload an die HoloLens gesendet werden. Nach Empfang der Anfrage auf der HoloLens wird das dementsprechende Event *ReceiveMsg* ausgelöst (siehe Abschnitt ??), welches wiederum die Simulation des Nachrichtenpakets (siehe Abschnitt ??) startet.
- **Empfangen von Nachrichten:** Auf der HoloLens wird eine Liste aktueller Nachrichten geführt. Beide Laptops rufen alle fünf Sekunden mit einem *HTTP GET-Request* an den Endpunkt */get-messages* diese Liste ab. Dies ermöglicht eine kontinuierliche Aktualisierung des Nachrichtenverlaufs (siehe Abschnitt 4.4.16, Abbildung 4.24) und zeigt dem Benutzer stets die neuesten Nachrichten an.

Es mag so aussehen, als ob das Senden von Requests für den Benutzer möglicherweise als unmöglich erscheint, aber im Abschnitt 4.4.16 wird gezeigt, dass dies alles durch eine benutzerfreundliche Benutzeroberfläche (UI) erreicht wird.

Es ist festzuhalten, dass die beiden Laptops niemals direkt miteinander kommunizieren, sondern alle Nachrichten über den Server, die HoloLens, laufen.

4.4.17.2 Bearbeitung von Anfragen

Die Bearbeitung von Anfragen erfolgt im Skript *RequestManager.cs*. Der Listener wird in einem eigenen Thread gestartet, da das Warten auf Nachrichten den Thread, auf dem der Listener läuft, blockiert. Nach dem Erhalt einer Anfrage wird ein neues *HttpListenerContext*-Objekt erstellt, und die Verarbeitung der Anfrage erfolgt in dem Thread, der diesen Kontext bearbeitet. Wenn eine HTTP-Anfrage empfangen wird, ist der erste Schritt, zu überprüfen, um welche Art von Anfrage es sich handelt. Ein POST und ein GET werden unterschiedlich verarbeitet. Bei Verwendung einer anderen HTTP-Methode wird ein *Method Not Allowed*-Fehler zurückgesendet. Bei der Wahl des Endpunkts passiert dasselbe; jede nicht unterstützte Anfrage auf einen Endpunkt wird mit einem *Not Found*-Fehler beantwortet.

Es werden in diesem Fall nur drei Endpunkte unterstützt:

- **Testen der Verbindung auf /connect**
- **Empfangen von Nachrichten auf /get-messages**
- **Senden einer Nachricht auf /message**

Um erfolgreich eine Anfrage an den Server zu senden, muss sie an die folgende Adresse gehen: <http://IP-Adresse der HoloLens:Port/Endpoint>. Die IP-Adresse der HoloLens wird manuell auf der Webseite eingegeben. Der Listener läuft standardmäßig auf dem Port 9090.

(Die erlaubten Endpunkte sind wie oben aufgeführt fallbasiert gelistet.)

4.5 Knapsack Problem Anwendungsszenario

→ SKREPEK

Im zweiten Anwendungsszenario dieser Applikation liegt der Fokus auf dem bekannten Problem des Knapsack-Problems. Ziel dieses Szenarios ist es, dieses Informatikproblem mithilfe von Augmented Reality (AR) visuell und spielerisch darzustellen. Das Knapsack-Problem ist ein klassisches Problem der Informatik, das nicht nur an der Höheren Technischen

Lehranstalt (HTL) vermittelt wird, sondern auch von Schülern eigenständig programmiert werden soll. Dieser Ansatz dient dazu, den Anwendern einen Einblick in die Informatik zu geben und möglicherweise Interessen für den Bereich zu wecken.

Das Unity-Anwendungsszenario für die HoloLens 2 bietet insgesamt eine interaktive und visuelle Erfahrung, bei der der Benutzer nicht nur das Knapsack-Problem verstehen, sondern auch praktisch anwenden kann. Die Integration von AR ermöglicht es dem Benutzer, das Problem in einer realen Umgebung zu erleben und die Optimierungsmöglichkeiten direkt zu visualisieren und zu manipulieren. Diese immersive Herangehensweise kann dazu beitragen, das Verständnis des Knapsack-Problems zu vertiefen und das Interesse an der Informatik zu fördern.

In diesem Abschnitt werden alle *GameObjects*, *Komponenten*, *Scripts* und *Klassen* genauer erklärt, die in der Unity-Szene des Knapsack-Problems verwendet werden, um dieses zu realisieren.

4.5.1 Knapsack-Problem Szenario Hirarchie

→ SKREPEK

Für einen sicheren und funktionalen Ablauf ist der Aufbau beziehungsweise die Hierarchie der Unity Szene von großer Bedeutung. In diesem Abschnitt wird anhand der Abbildung 4.27 verdeutlich wie die Unity Szene für die Implementierung des zweiten Anwendungsszenarios für das Knapsack-Problem aufgebaut ist und es wird zusätzlich kurz darauf eingegangen für was welches Objekt steht und welche Aufgabe dieses trägt.

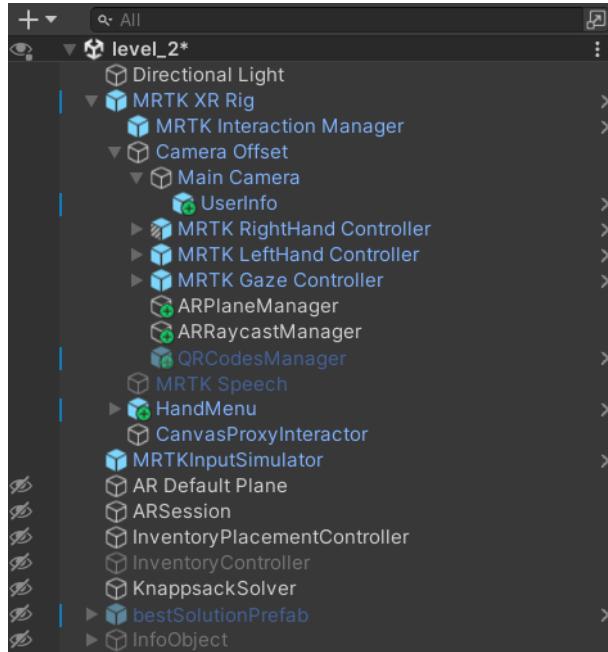


Abbildung 4.27: Knapsack-Problem Szenario Hirarchie im Unity Editor.

In dieser Abbildung ist die Hierarchie un der Inhalts des Knapsack-Problem Szenarios zu sehen. Anhand desen ist zu erkennen, dass diese Unity Szene aus vielen wichtigen Komponenten besteht, die alle zusammenspielen, um das gewünschte Ergebnis zu erzielen. Darunter sind folgende Objekte:

- **level-2:** Ist die Unity Szene selbst, in der alle GameObjects enthalten sind.
- **MRTK XR Rig:** Grundbaustein für das Entwickeln einer XR-Applikation. Enthält

wichtig Komponenten wie Controller für das tracken der Hände, für den User Gaze, etc.

- **UserInfo:** Text Objekt, dass in der *main camera* liegt, damit dieses ständig im Sichtfeld des Benutzers liegt.
- **Managers:** Die drei Manager *ARPlaneManager*, *ARRaycastManager* und *QRCodeManager* sind wichtiger Bestandteil um ARPlanes zu scannen, Raycasting durchzuführen und QRCodes zu erkennen.
- **HandMenu:** Button um in das Hauptmenu-Szenario zurück zu gelangen.
- **AR Default Plane:** Ist das Grund-Objekt für das markieren von erkannten AR-Planes.
- **ARSession:** Ist die Hauptkomponente, die die AR-Funktionalitäten steuert und koordiniert.
- **InventoryPlacementController:** Game Objekt, welches das platzieren des Inventar-Objekts handhabt.
- **InventoryController:** Game Objekt, welches das erkennen von neuen Objekten innerhalb des Inventar-Objekts handhabt.
- **KnapsackSolver:** Game Objekt, welches den Knapsack Algorithmus implementiert und das eigene Inventar berechnet.
- **bestSolutionPrefab:** Prefab, welches die perfekte Lösung wiederspiegelt.
- **InfoObject:** Dient der visualisierung von berechneten Werten und Fehlermeldungen.

In der Abbildung ist ausßredem zu sehen, dass ein Paar Game Objekte ausgegraut und nicht ausgegraut sind und, dass neben ein paar Game Objekten ein durchgestrichenes Auge zu sehen ist. Wenn ein Game Objekt im Unity Editor ausgegraut ist bedeutet das, dass dieses GameObject und somit alle angehängten Scripts von diesem Game Objekt deaktiviert sind. Das bedeutet, dass dieses Game Objekt samt allen Scripts zu Szenenbeginn nicht aufgerufen und somit auch nicht ausgeführt werden. Nicht ausgegraute Game Objekte wiederum sind daher genau das Gegenteil. Das beudetet, dass das Game Objekt selbst samt allen angehängten Scripts alle aktiviert sind und somit zu Szenenbeginn aufgerufen und ausgeführt werden.

Wenn neben einem Game Objekt das durchgestrichene Auge zu sehen ist bedeutet das nur, dass dieses Game Objekt im Unity Editor nicht zu sehen ist. Andererseits, wenn kein Zeichen neben dem Game Objekt zu sehen ist, ist dieses Objekt im Unity Editor sichtbar. Dies dient dazu, dass falls in der Unity Szene viele Game Objekte vorhanden sind, dass man diejenige ausblendet die nicht im Editor sichtbar sein müssen wie zum Beispiel Tesh Meshes oder Lables.

4.5.2 Nutzung von QR-Codes

→ HAYLAZ

Im vorherigen Abschnitt wurde bereits erwähnt, dass QR-Codes in diesem Level verwendet werden, um verschiedene Elemente zu repräsentieren. Diese QR-Codes spielen eine entscheidende Rolle, indem sie dazu dienen, vielfältige Informationen zu den einzelnen Gegenständen, die für das Knapsack-Inventar benötigt werden, zu speichern und sie anschließend in einer virtuellen Umgebung abzubilden. Im folgenden Abschnitt möchten wir näher darauf eingehen, wie genau diese QR-Codes generiert werden und welchen Zweck sie innerhalb der Augmented Reality (AR)-Applikation erfüllen. Dabei wird insbesondere betrachtet, wie die Codes generiert und auf welche Weise sie innerhalb der Anwendung zur Interaktion mit den realen Objekten verwendet werden.

4.5.2.1 Struktur und Inhalt eines QR-Codes

Die Informationen, die in einem QR-Code gespeichert werden können, sind begrenzt. In unserem Anwendungsfall wird lediglich eine einzelne Zahl im Bereich von 1 bis 11 abgespeichert. Diese Zahlen repräsentieren die 11 verschiedenen Modelle, die wir unterscheiden möchten. Da nur eine Zahl gespeichert wird, genügt ein QR-Code der Größe 21x21 Module (Version 1). Die geringe Anzahl von Modulen ermöglicht eine schnellere Erkennung, auch über größere Distanzen.⁸³

Die zugehörigen Zahlen werden in der Software, genauer gesagt in der Klasse *QRItem.cs*, mit Informationen verknüpft. Im folgenden Codeausschnitt wird dies verdeutlicht:

```

1 public class QRItem
2 {
3     public struct QRData
4     {
5         public int id;
6         public string name;
7         public Vector3 position;
8         public int weight;
9         public int value;
10    }
11
12    public QRData qrData;
13
14    public Dictionary<int, QRData> items = new Dictionary<int, QRData>()
15    {
16        {1, new QRData { id = 1, name = "Laptop", weight = 70, value = 100 }},
17        {2, new QRData { id = 2, name = "Router", weight = 25, value = 50 }},
18        {3, new QRData { id = 3, name = "Maus", weight = 20, value = 30 }},
19        // ...
20        {11, new QRData { id = 11, name = "Handy", weight = 30, value = 100 }}
21    };
22
23    public QRItem(int id)
24    {
25        items.TryGetValue(id, out qrData);
26    }
27}
```

Codeabschnitt 4.23: Codeabschnitt von QRItem Klasse

In dieser Klasse wird ein Dictionary verwendet, das den Gegenständen die folgenden Informationen zuordnet:

- **Id:** Die numerische Kennung im QR-Code.
- **Name:** Die Bezeichnung des Gegenstandes, das dieser QR-Code repräsentiert.
- **Position:** die Position des Gegenstandes in der virtuellen Umgebung.
- **Weight:** Das Gewicht des Gegenstandes.
- **Value:** der Wert des Gegenstandes.

Diese Informationen spielen eine wesentliche Rolle in der weiteren Anwendung des Knapsack-Algorithmus.

⁸³Microsoft-Dokumentation **QR-Code-Tracking**

4.5.2.2 QR-Code-Tracking

Das Tracking der QR-Codes erfolgt mithilfe des *QRCodeManager.cs* Skripts. Diese Klasse ist ein *Singleton*, das die Erkennung und Verfolgung der QR-Codes steuert.⁸⁴ ⁸⁵

Nach der Erkennung eines QR-Codes erfolgen eine Reihe von Schritten, um diese Informationen zu speichern, verarbeiten und zuletzt darzustellen. Hier eine kurze Übersicht:

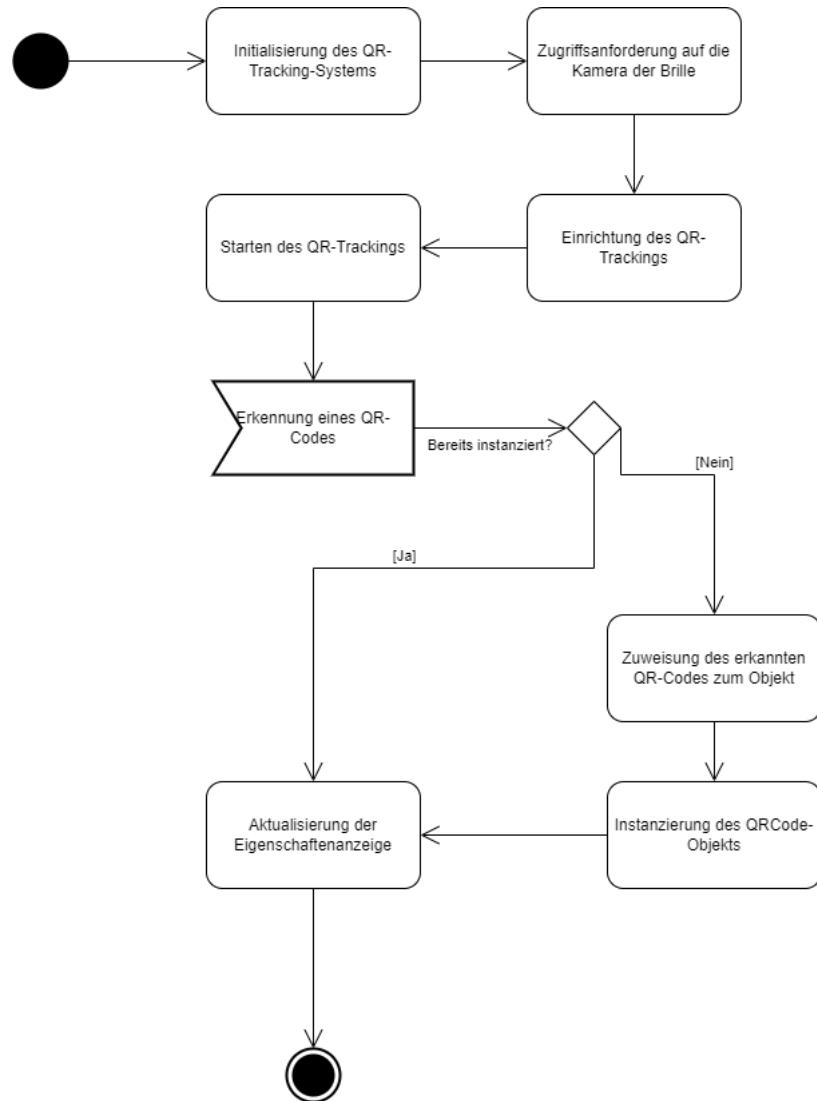


Abbildung 4.28: Ablaufdiagramm des QR-Code-Trackings.

- **Initialisierung des QR-Tracking-Systems:** Um das QR-Tracking-System zu aktivieren, wird die erforderliche Komponente gestartet, um die Erkennung von QR-Codes zu ermöglichen. Diese Komponente ist das QRCodeManager-GameObject, das weitere GameObject instanziert, wie den *QRCodes Visualizer*, der für die Visualisierung benötigt wird (siehe Abschnitt ??). Dabei werden Tracking-Algorithmen gestartet, die für die Lokalisierung und Identifizierung von QR-Codes in der Umgebung benötigt werden. Diese Initialisierung erfolgt zu Beginn der Anwendungsaktivität.

⁸⁴ Microsoft-Dokumentation **Singleton**

⁸⁵ Microsoft-Dokumentation **QR-Code-Tracking**

- **Zugriffsanforderung auf die Kamera der Brille:** Für das QR-Tracking wird der Zugriff auf die Kamera der Augmented-Reality-Brille benötigt. Eine Zugriffsanfrage wird gestellt, um die notwendigen Berechtigungen zu erhalten. Dieser Schritt ist entscheidend, um visuelle Daten von der Kamera zu erhalten und QR-Codes in der physischen Umgebung zu erkennen.
- **Einrichtung des QR-Trackings:** Die Einrichtung des QR-Trackings umfasst die Konfiguration von Parametern und Einstellungen, die für die korrekte Funktion des Tracking-Systems erforderlich sind. Dazu gehören Kalibrierungsschritte, die Anpassung an die Umgebung und die Festlegung von Erkennungsbereichen. Eine ordnungsgemäße Einrichtung gewährleistet eine zuverlässige und präzise Erkennung von QR-Codes.
- **Starten des QR-Trackings:** Das System sucht aktiv nach QR-Codes in der Umgebung, um sie zu erkennen. Die Kamera erfasst kontinuierlich visuelle Daten, welche von den Tracking-Algorithmen analysiert werden, um QR-Codes zu identifizieren. Das Starten des Trackings markiert den Beginn des fortlaufenden Erkennungsprozesses.
- **Erkennung eines QR-Codes (Event):** Sobald die Kamera einen QR-Code erfasst, wird dieser automatisch erkannt. Dieses Ereignis signalisiert die erfolgreiche Erkennung eines QR-Codes und liefert Informationen über den erkannten QR-Code, wie seine Daten und Position. Eine detaillierte Beschreibung des genauen Ablaufs der Erkennung und Positionsbestimmung ist im Abschnitt 4.5.2.4 zu finden.
- **Zuweisung des erkannten QR-Codes zum Objekt:** Nach der Erkennung wird überprüft, ob der erkannte QR-Code bereits in der Anwendung registriert ist. Falls dies der Fall ist, wird der erkannte QR-Code einem entsprechenden Objekt in der virtuellen Umgebung zugeordnet.
- **Instanziierung des QRCode-Objekts:** Nach dem Scannen eines QR-Codes wird ein QR-Code-Prefab erzeugt und in der Szene platziert.

Dieses Objekt dient als Repräsentation des gescannten QR-Codes und wird als QR-Objekt bezeichnet. Es enthält visuelle Darstellungen, Interaktionsmöglichkeiten und weitere relevante Informationen über den zugehörigen QR-Code. Die Instanziierung ermöglicht eine nahtlose Integration des gescannten QR-Codes in die virtuelle Umgebung. Weitere Informationen zur Visualisierung von QR-Codes finden Sie in Abschnitt ??.

- **Aktualisierung der Eigenschaften anzeigen:** Die Aktualisierung der Eigenschaftsanzeige dient dazu, visuelle und informative Darstellungen des erkannten QR-Codes zu aktualisieren. Hierbei werden die Position, Größe, visuelle Darstellung und zugehörige Informationen des QR-Codes aktualisiert. Durch die Aktualisierung wird sichergestellt, dass die Benutzer stets die neuesten Informationen über das durch den QR-Code repräsentierte Objekt erhalten.

4.5.2.3 Interaktion mit QR-Codes



Abbildung 4.29: Bauklötzte mit QR-Codes

Durch die Verwendung der HoloLens können wir dem Benutzer eine visuelle Darstellung einer virtuellen Welt bieten. Um eine Verbindung zwischen der realen und der virtuellen Welt herzustellen, nutzen wir QR-Codes. Diese dienen in der realen Welt als Repräsentationen jener Objekte, die wir in der virtuellen Welt darstellen möchten.

Wie in 4.29 zu sehen ist, sind die Bauklötzte mit QR-Codes versehen. Diese Bauklötzte repräsentieren die Gegenstände, die der Benutzer in sein Inventar aufnehmen kann. Wenn der Benutzer einen Bauklotz aufhebt und ihn der HoloLens nähert, wird der QR-Code gescannt. Dadurch werden Informationen wie das zugehörige 3D-Modell, der Wert, das Gewicht und der Name des Gegenstands angezeigt. Auf diese Weise können wir eine nahtlose Verbindung zwischen der realen und der virtuellen Welt herstellen.

Dem Benutzer wird ermöglicht, die Bauklötzte physisch zu berühren, aufzuheben und zu fühlen. Diese sensorische Erfahrung trägt dazu bei, die Immersion des Benutzers zu verbessern und ihm ein besseres Verständnis der virtuellen Welt zu ermöglichen. Zu diesem Zweck muss die Position der Bauklötzte und damit auch der QR-Codes laufend getrackt werden, wie im Abschnitt 4.5.2.4 beschrieben.

4.5.2.4 Positionsbestimmung von QR-Codes

Um die räumliche Position eines QR-Codes exakt zu bestimmen, ist ein spezieller Prozess notwendig. Dieser Schritt ist entscheidend, um physische QR-Codes in der virtuellen Realität erfassen und mit ihnen interagieren zu können. Die Positionierung der QR-Codes erfolgt mithilfe einer Klasse namens *QRWatcher*, die im SDK-Plug-in *Microsoft.MixedReality.QR* enthalten ist.⁸⁶

Kapitel 5

Zusammenfassung und Abschluss

5.1 Ergebnis

Die Implementierung des Projekts wurde erfolgreich abgeschlossen. Alle definierten Ziele wurden gemäß den vorgegebenen Spezifikationen erreicht. Das resultierende Produkt ist eine Augmented Reality (AR)-Applikation sowie eine begleitende Webseite. Die AR-Anwendung bietet eine innovative Möglichkeit, Lehrinhalte zu vermitteln, indem sie eine interaktive und immersivere Lernerfahrung ermöglicht. Durch die Integration von virtuellen Elementen in die reale Umgebung können komplexe Konzepte auf anschauliche Weise vermittelt werden. Die Webseite dient als ergänzendes interaktives Element, um dem Benutzer eine bessere Interaktion mit der realen Welt zu ermöglichen und somit das Verständnis des veranschaulichten Anwendungsszenarios zu vertiefen.

Das entstandene Produkt bietet vielfältige Einsatzmöglichkeiten im Bildungskontext. Es kann nicht nur als Lehrmittel im regulären Unterricht verwendet werden, sondern auch als Attraktion während Veranstaltungen wie dem Tag der Offenen Tür. Durch die Integration moderner Technologien wird die Lernumgebung bereichert und die Aufmerksamkeit der Zielgruppe auf innovative Weise auf sich gezogen. Dies trägt dazu bei, das Interesse an den behandelten Themen zu wecken und das Lernengagement zu steigern.

Die erfolgreiche Umsetzung des Projekts markiert einen bedeutenden Fortschritt in der Verknüpfung von technologischen Innovationen mit Bildungsprozessen. Die erzielten Ergebnisse bieten nicht nur einen Mehrwert für die Lehrenden und Lernenden an der HTBLuVA Wiener Neustadt, sondern stellen auch einen Beitrag zur Weiterentwicklung pädagogischer Ansätze dar, die auf den Einsatz digitaler Medien und interaktiver Technologien abzielen.

5.2 Abnahme

Die formelle Abnahme des Projekts fand am 20.03.2024 mit der Einreichung dieser Diplomarbeit statt. Die HTBLuVA Wiener Neustadt plant, die entwickelte Applikation in absehbarer Zeit in ihren Lehrplan zu integrieren und bei öffentlichen Veranstaltungen wie dem Tag der offenen Tür einzusetzen.

Die Entscheidung der HTBLuVA Wiener Neustadt, die Anwendung in ihr Bildungsangebot aufzunehmen, unterstreicht die Relevanz und den Mehrwert des Projekts für die Bildungseinrichtung. Die geplante Integration in den Lehrplan zeigt das Vertrauen in die Qualität und Effektivität der entwickelten Lösung.

Die Abnahme des Projekts durch die Bildungsinstitution markiert einen Meilenstein in der Validierung der Arbeit und der erzielten Ergebnisse. Es ist wichtig, dass die entwickelten Produkte den Anforderungen und Erwartungen der Zielgruppe entsprechen und den

angestrebten Bildungszielen dienen. Dabei sollte eine klare und logische Strukturierung der Informationen gewährleistet sein, um eine einfache Verständlichkeit zu gewährleisten.

5.3 Zukunft

Die Zukunftsperspektive bietet verschiedene Möglichkeiten zur Weiterentwicklung und Erweiterung der vorliegenden Projektarbeit. Ein möglicher Ansatz besteht darin, das vorhandene Wissen und die Technologie an ein nachfolgendes Team weiterzugeben, um zusätzliche Anwendungsszenarien zu entwickeln. Durch die Initialisierung des Projekts auf der HoloLens 2 durch das aktuelle Projektteam steht den Nachfolgern eine solide Grundlage zur Verfügung. Diese bietet bereits eine Vielzahl von behobenen Problemen und dokumentierten Lösungsansätzen. Dadurch können Zeit und Ressourcen gespart werden, und der Fokus kann verstärkt auf die Erweiterung und Verbesserung des bestehenden Systems gelegt werden.

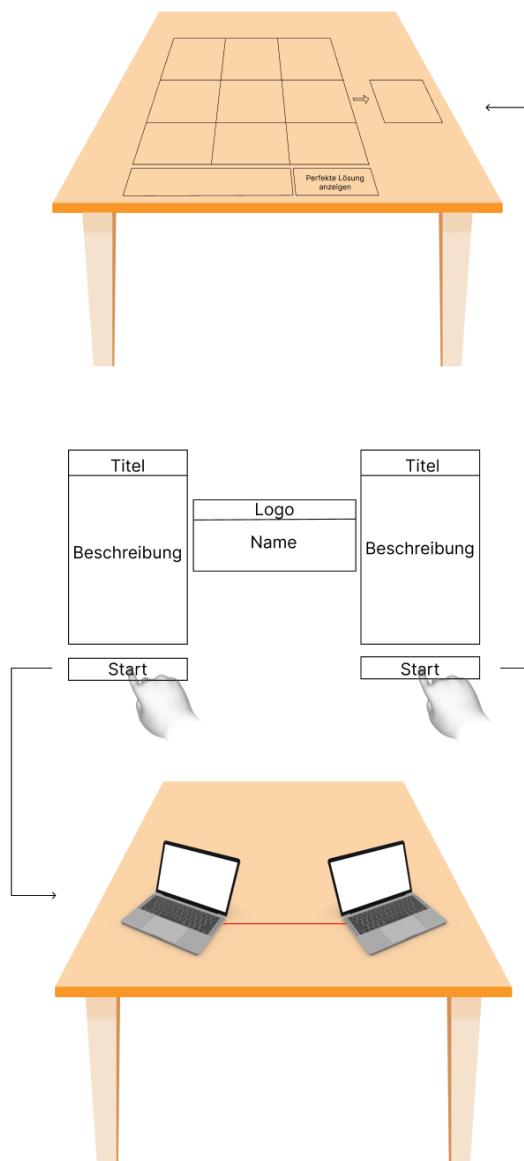
Während der Projektarbeit wurden im Team verschiedene Ideen für potenzielle Anwendungsszenarien diskutiert und konzipiert. Eine vielversprechende Idee wurde identifiziert, die das Spektrum der Anwendungsmöglichkeiten der HoloLens 2 erweitern könnte:

- **PC-Zusammenbau-Simulator:** Dieses interaktive Szenario ermöglicht es dem Benutzer, den detaillierten Zusammenbau eines PCs virtuell durchzuführen. Der Benutzer kann aus einem umfangreichen Katalog verschiedener Bauteile wählen und diese gemäß den Anweisungen in den PC einbauen. Das Szenario führt den Benutzer durch die einzelnen Schritte des Zusammenbaus und erklärt die Funktionen der verschiedenen Komponenten.

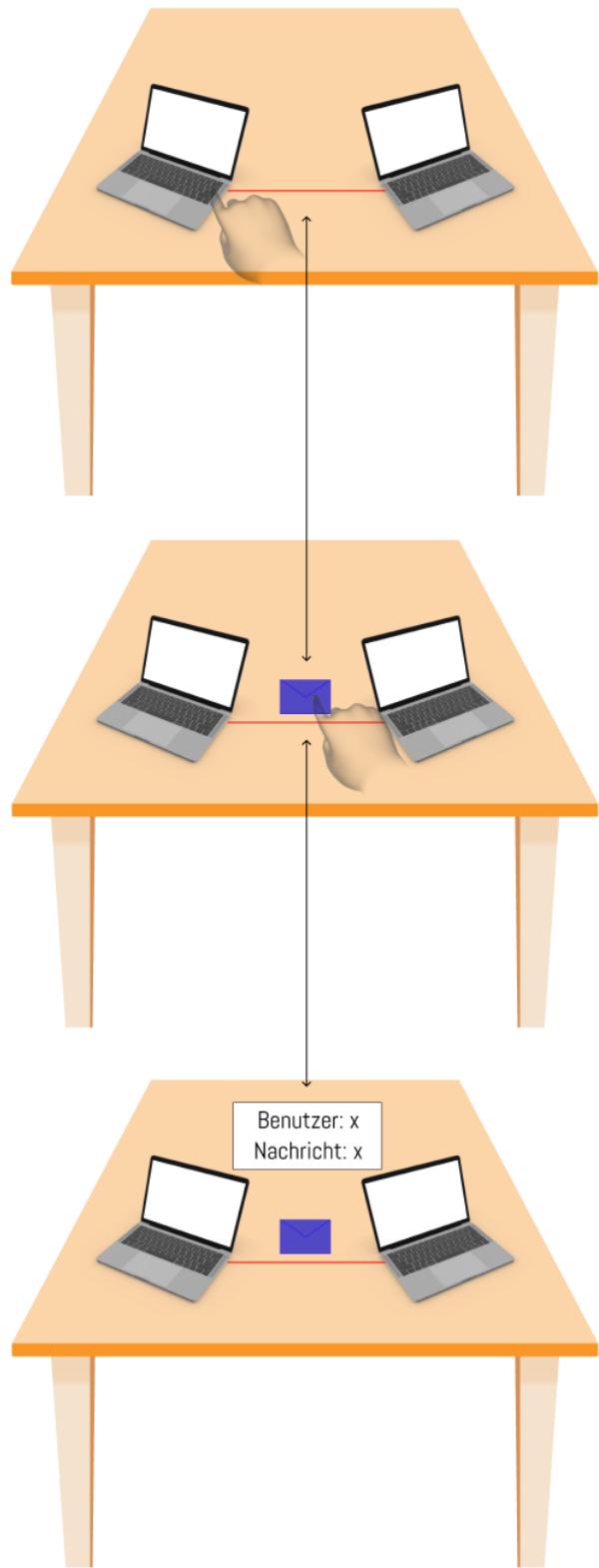
Anhang A

Mockups

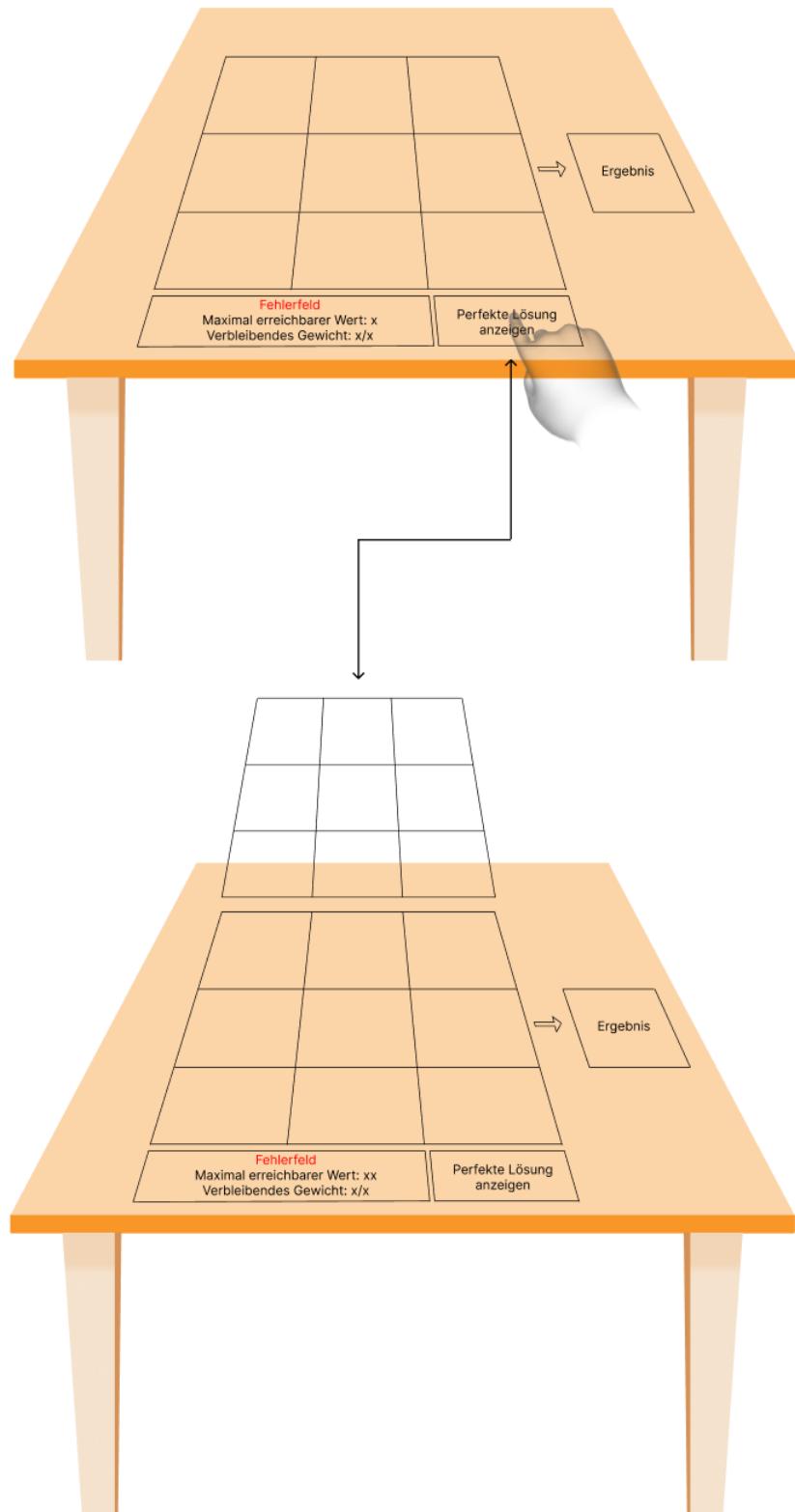
A.1 Hauptmenu



A.2 Nachrichtenaustausch Anwendungsszenario



A.3 Knapsack-Problem Anwendungsszenario



Anhang B

Abläufe

B.1 Hauptmenu

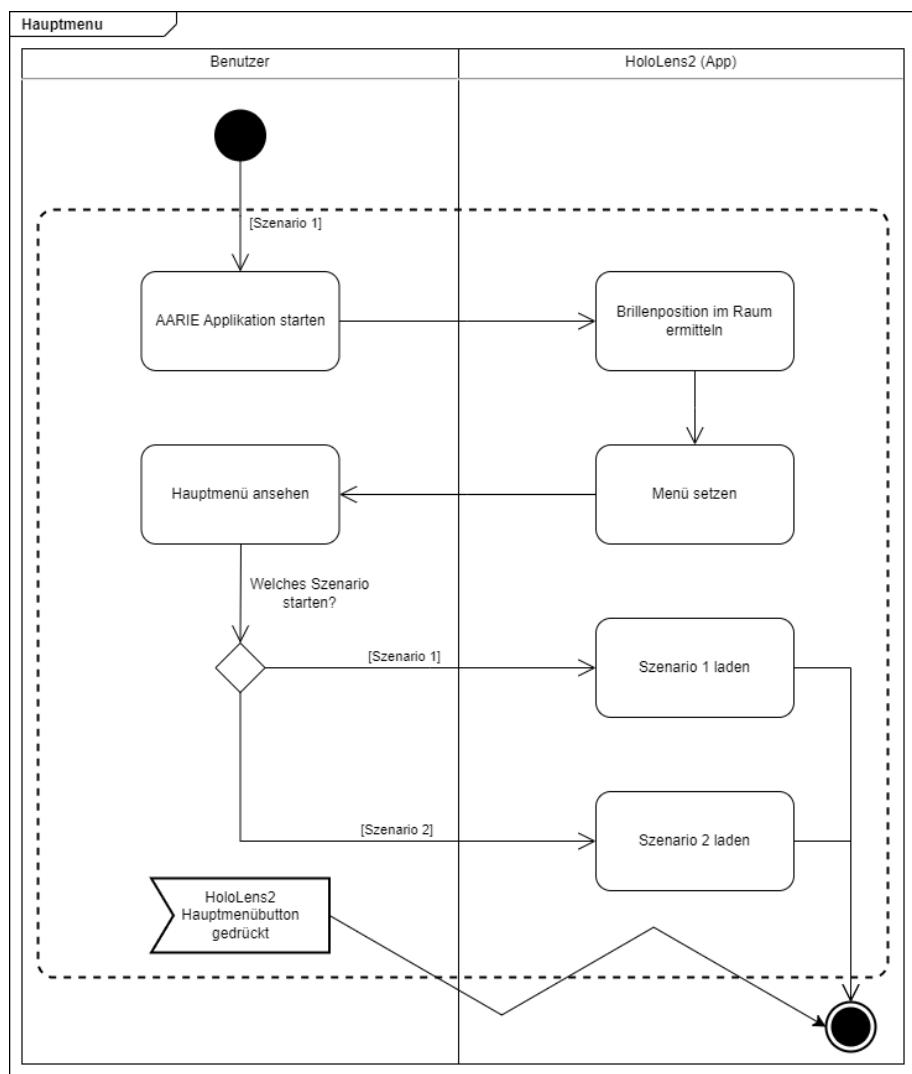


Abbildung B.1: Hauptmenu UML Aktivitätsdiagramm

B.2 Nachrichtenaustausch Anwendungsszenario

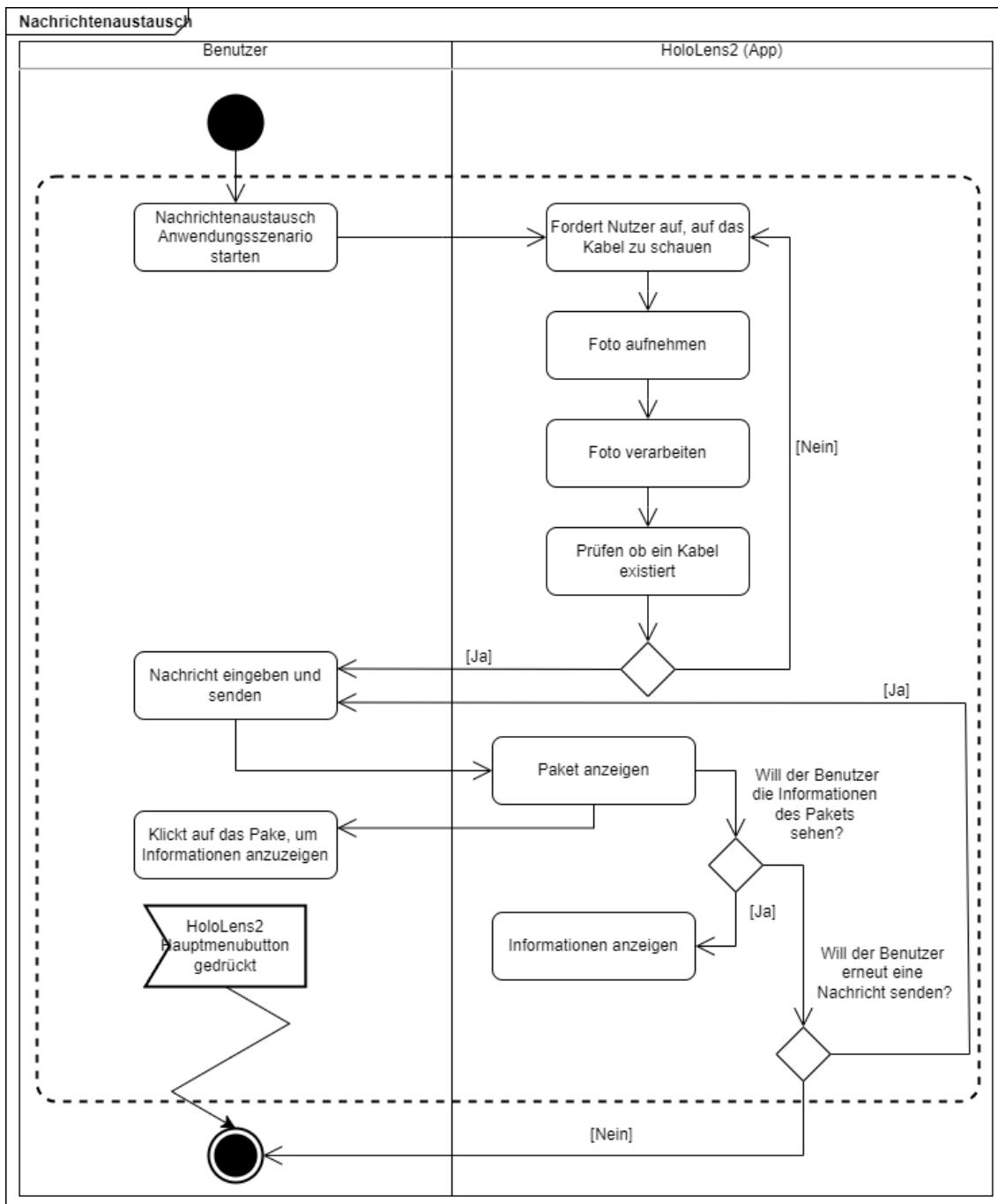


Abbildung B.2: Nachrichtenaustausch UML Aktivitätsdiagramm

B.3 Knapsack-Problem Anwendungsszenario

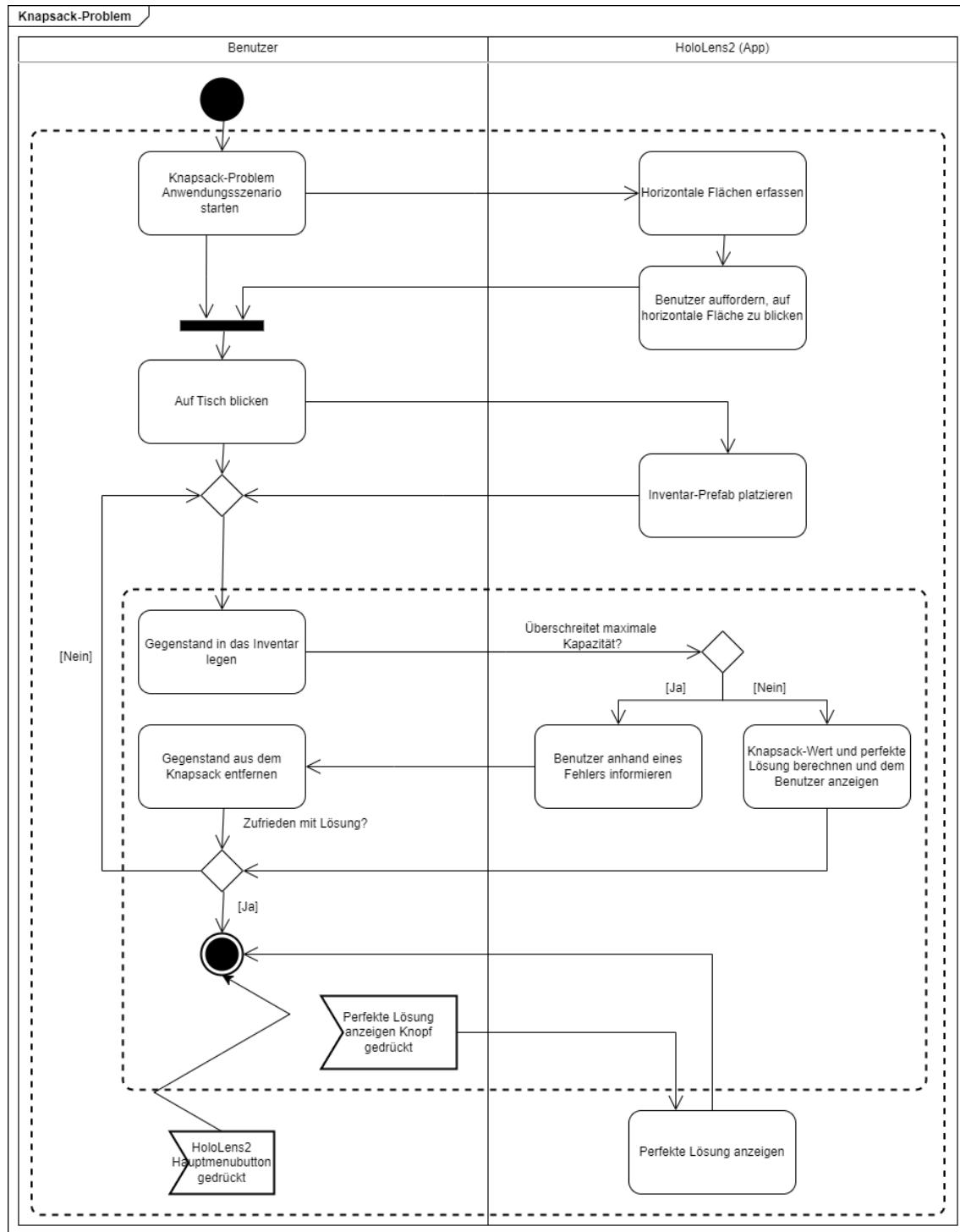


Abbildung B.3: Knapsack-Problem UML Aktivitätsdiagramm

Anhang C

Literatur

- Scrum.org - What is Scrum? URL: <https://www.scrum.org/learning-series/what-is-scrum/what-is-scrum> (besucht am 06.10.2023).
- Scrum.org - What is a Product Owner? URL: <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-team/what-is-a-product-owner> (besucht am 10.11.2023).
- Scrum.org - What is a Scrum Master? URL: <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-team/what-is-a-scrum-master> (besucht am 10.11.2023).
- Scrum.org - What is a Developer? URL: <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-team/what-is-a-developer> (besucht am 10.11.2023).
- Scrum.org - What is a Sprint? URL: <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-events/what-is-a-sprint> (besucht am 10.11.2023).
- Scrum.org - What is a Sprint Planing? URL: <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-events/what-is-sprint-planning> (besucht am 10.11.2023).
- Scrum.org - What is a Daily Scrum? URL: <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-events/what-is-a-daily-scrum> (besucht am 10.11.2023).
- Scrum.org - What is a Sprint Review? URL: <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-events/what-is-a-sprint-review> (besucht am 10.11.2023).
- Scrum.org - What is a Sprint Retroperspective? URL: <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-events/what-is-a-sprint-retrospective> (besucht am 10.11.2023).
- GitLab.com - About GitLab. URL: <https://about.gitlab.com/> (besucht am 20.03.2024).
- SourceForge.net - The Complete Software Platform. URL: <https://sourceforge.net/> (besucht am 20.03.2024).
- BitBucket.org - Code CI/CD built for teams using Jira. URL: <https://bitbucket.org/> (besucht am 20.03.2024).
- GitHub Blog - GitHub: Jetzt mit unbegrenzten, kostenlosen privaten Repositories sowie einem vereinheitlichten Enterprise-Angebot. URL: <https://github.blog/de/2019-01-07-new-year-new-github-jetzt-mit-unbegrenzten-kostenlosen-privaten-repositories-sowie-einem-vereinheitlichten-enterprise-angebot/> (besucht am 20.03.2024)
- Github-Dokumentation - Github Sprachunterstützung. URL: <https://docs.github.com/de/get-started/learning-about-github/github-language-support> (besucht am 20.03.2024).
- Atlassian - Tarife und Preise vergleichen. URL: <https://www.atlassian.com/de/software/jira/pricing> (besucht am 20.03.2024).
- Atlassian-Community - How to connect a GitHub account to Jira Software. URL: <https://community.atlassian.com/t5/Jira-Software-articles/How-to-connect-a-GitHub-account-to-Jira-Software/ba-p/1873994> (besucht am 20.03.2024).

- Atlassian - Im Team schneller vorankommen, an einem Strang ziehen und besser entwickeln. URL: <https://www.atlassian.com/de/software/jira> (besucht am 20.03.2024).
- Scholl, Armin. *Die Befragung. Sozialwissenschaftliche Methode und kommunikationswissenschaftliche Anwendungen*. 1. Aufl. Konstanz: UVK, 2003.
- Mayer, Horst. *Interview und schriftliche Befragung. Entwicklung, Durchführung und Auswertung*. 3. Aufl. München: R. Oldenbourg, 2006.
- Bühner, Markus. *Einführung in die Test- und Fragebogenkonstruktion*. 1. Aufl. München: Pearson Studium, 2004.
- Claus Brell. *Statistik von Null auf Hundert*. 2. Aufl. Berlin: Springer.
- Unity.com - Plattformen. URL: <https://unity.com/de/roadmap/unity-platform/platforms> (besucht am 20.03.2024).
- Assetstore.Unity.com - Unity Asset Store. URL: <https://assetstore.unity.com/> (besucht am 20.03.2024).
- Unrealengine.com - Wir liefern die Engine. Sie machen sie Unreal. URL: <https://www.unrealengine.com/de?hl=in&lang=de> (besucht am 20.03.2024).
- Unreal Engine-Dokumentation. Blueprints Visual Scripting. URL: <https://docs.unrealengine.com/5.3/en-US/blueprints-visual-scripting-in-unreal-engine/> (besucht am 13.03.2024).
- Microsoft-Dokumentation - Mixed Reality Toolkit 3. URL: <https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk3-overview/> (besucht am 05.11.2023).
- Khronos Group - OpenXR. URL: <https://www.khronos.org/openxr/> (besucht am 05.11.2023).
- Blender - Allgemein. URL: <https://www.blender.org/> (besucht am 20.03.2024).
- Autodesk - 3DS Max. URL: <https://www.autodesk.de/products/3ds-max/overview?term=1-YEAR&tab=subscription> (besucht am 27.02.2024).
- Maxon - Cinema 4d. URL: <https://www.maxon.net/de/cinema-4d> (besucht am 28.02.2024).
- Microsoft.com - Visual Studio 2022. URL: <https://visualstudio.microsoft.com/de/vs/> (besucht am 15.03.2024).
- Unity.com - Unity Asset Manager. URL: <https://unity.com/products/asset-manager> (besucht am 15.03.2024).
- Unity.com - Spiele in Augmented Reality und Virtual Reality. URL: <https://unity.com/de/solutions/vr> (besucht am 15.03.2024).
- Unity-Dokumentation - About AR Foundation. URL: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/index.html> (besucht am 15.03.2024).
- Unity-Dokumentation - XR Interaction Toolkit. URL: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.5/manual/index.html> (besucht am 15.03.2024).
- Unity.com - Tools for profiling and debugging. URL: <https://unity.com/de/how-to/profiling-and-debugging-tools> (besucht am 15.03.2024).
- Unity.com - Was ist mit Quellcodeverwaltung gemeint? URL: <https://unity.com/de/solutions/source-code-management> (besucht am 15.03.2024).
- Microsoft-Dokumentation - Subsysteme MRTK3. URL: <https://learn.microsoft.com/de-de/windows/mixed-reality/mrtk-unity/mrtk3-overview/architecture/subsystems> (besucht am 15.03.2024).
- Unity-Dokumentation - Order of execution for event functions. URL: <https://docs.unity3d.com/Manual/ExecutionOrder.html> (besucht am 15.03.2024).
- Unity-Dokumentation - MonoBehaviour Class. URL: <https://docs.unity3d.com/ScriptReference/MonoBehaviour.html> (besucht am 15.03.2024).

- Unity-Dokumentation - Scenes. URL: <https://docs.unity3d.com/Manual/CreatingScenes.html> (besucht am 13.12.2023).
- Medium - Creating Manager Classes in Unity. URL: <https://sneakydaggergames.medium.com/creating-manager-classes-in-unity-a77cf7edcba5> (besucht am 05.11.2023).
- Unity-Dokumentation - AR Plane Manager. URL: <https://docs.unity.cn/Packages/com.unity.xr.arfoundation@4.1/manual/plane-manager.html> (besucht am 05.11.2023).
- Unity-Dokumentation - AR Raycast Manager. URL: <https://docs.unity.cn/Packages/com.unity.xr.arfoundation@5.0/api/UnityEngine.XR.ARFoundation.ARRaycastManager.html> (besucht am 05.11.2023).
- Unity-Dokumentation - GameObjects. URL: <https://docs.unity3d.com/Manual/GameObject.html> (besucht am 18.02.2024).
- Unity-Dokumentation - Prefabs. URL: <https://docs.unity3d.com/Manual/Prefabs.html> (besucht am 15.01.2024).
- Unity-Learn-Dokumentation - Prefabs-2. URL: <https://learn.unity.com/tutorial/prefabs-e#> (besucht am 15.03.2024).
- Microsoft-Dokumentation - Visual Studio Solution. URL: <https://learn.microsoft.com/en-us/visualstudio/ide/solutions-and-projects-in-visual-studio?view=vs-2022> (besucht am 15.03.2024).
- Microsoft-Dokumentation - Universal Unencrypted Protocol. URL: <https://learn.microsoft.com/en-us/windows/uwp/debug-test-perf/deploying-and-debugging-uwp-apps> (besucht am 15.03.2024).
- Adobe - What are 3d textures. URL: <https://www.adobe.com/at/products/substance3d/discover/how-to-create-3d-textures.html> (besucht am 17.03.2024).
- TechnicalProTip - Textur. URL: <https://technicalprotips.com/the-importance-of-lighting-and-texturing-in-3d-modeling/> (besucht am 13.03.2024).
- Artisticrenderer - Bild Texturen. URL: <https://artisticrender.com/how-to-add-a-texture-to-an-object-in-blender/> (besucht am 13.03.2024).
- Blender - Node-Introduction. URL: <https://docs.blender.org/manual/en/latest/interface/controls/nodes/introduction.html> (besucht am 14.03.2024).
- GeeksForGeeks - Mesh Definition. URL: <https://www.geeksforgeeks.org/polygon-mesh-in-computer-graphics/> (besucht am 14.03.2024).
- TechTarget - Mesh Usages. URL: <https://www.techtarget.com/whatis/definition/3D-mesh> (besucht am 14.03.2024).
- wedesignvirtual - WhatareMeshes. URL: <http://wedesignvirtual.com/what-are-meshes-in-3d-modeling/> (besucht am 17.03.2024).
- CGIFurniture - What are Polygons. URL: <https://cgifurniture.com/what-are-polygons-in-3d-modeling/> (besucht am 14.03.2024).
- Adobe - 3D Polygon Modeling. URL: <https://www.adobe.com/de/products/substance3d/discover/3d-polygon-modeling.html> (besucht am 17.03.2024).
- CGCookie - Ways to optimize Game Assets. URL: <https://cgcookie.com/posts/3-ways-to-optimize-game-assets-in-blender> (besucht am 13.03.2024).
- All3DP - How to reduce Polygons. URL: <https://all3dp.com/2/blender-how-to-reduce-polygons/> (besucht am 14.03.2024).
- Blender-Dokumentation - Decimate Modifier. URL: <https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/decimate.html> (besucht am 17.03.2024).

- Blender-Dokumentation - Merge Funktion. URL: <https://docs.blender.org/manual/en/latest/modeling/meshes/editing/mesh/merge.html> (besucht am 17.03.2024)
- Blog CG Wire - Getting started with export in Blender. URL: <https://blog.cg-wire.com/getting-started-with-export-in-blender/> (besucht am 17.03.2024).
- GameDev-Profi - Welches Dateiformat. URL: <https://www.gamedev-profi.de/die-besten-quellen-fuer-3d-modelle/> (besucht am 11.03.2024).
- Vection Technologies - Everything You Need to Know About FBX Files. URL: <https://vectio-n-technologies.com/blog/Everything-You-Need-to-Know-About-FBX-Files-A-Comprehensive-Guide/#:~:text=FBX%20files%20are%20a%20type,texture%2C%20and%20skeletal%20animation%20data.> (besucht am 17.03.2024).
- Blender - Loop Tools. URL: <https://docs.blender.org/manual/en/latest/addons/mesh/looptools.html> (besucht am 20.11.2023).
- Blender - Circle-Werkzeug. URL: <https://docs.blender.org/manual/en/latest/addons/mesh/looptools.html#circle> (besucht am 20.11.2023).
- Blender - Images as Planes. URL: https://docs.blender.org/manual/en/latest/addons/import_export/images_as_planes.html (besucht am 05.12.2024).
- Blender - Parents. URL: <https://docs.blender.org/manual/en/2.79/editors/3dview/object/properties/relations/parents.html> (besucht am 05.12.2024).
- Blender - Groups. URL: <https://docs.blender.org/manual/en/2.79/editors/3dview/object/properties/relations/groups.html> (besucht am 05.12.2024).
- Blender - Modifier. URL: <https://docs.blender.org/manual/en/latest/modeling/modifiers/introduction.html> (besucht am 05.12.2024).
- Blender - Non-Destructive Modifier. URL: <https://docs.blender.org/manual/en/2.79/modeling/modifiers/introduction.html#the-modifier-stack> (besucht am 05.12.2024).
- Blender - Subdivision Surface Modifier. URL: https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/subdivision_surface.html (besucht am 05.12.2024).
- Blender - Mirror Modifier. URL: <https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/mirror.html> (besucht am 05.12.2024).
- Blender - Bevel Modifier. URL: <https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/bevel.html> (besucht am 05.12.2024).
- Blender - Extrude. URL: https://docs.blender.org/manual/de/dev/grease_pencil/modes/edit_point_menu.html#extrude (besucht am 20.2.2024).
- Blender - Array Modifier. URL: <https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/array.html> (besucht am 11.01.2024)
- Blender - Eyedropper Werkzeug. URL: <https://docs.blender.org/manual/en/latest/interface/controls/buttons/eyedropper.html> (besucht am 17.03.2024).
- Blender - UV-Editing Editor. URL: <https://docs.blender.org/manual/en/latest/editors/uv/introduction.html> (besucht am 17.03.2024).
- all3dp - Blender UV Mapping Simply Explained. URL: <https://all3dp.com/2/blender-uv-mapping-simply-explained/> (besucht am 17.03.2024).
- Microsoft-Dokumentation - Naehemenue MRTK2. URL: <https://learn.microsoft.com/de-de/windows/mixed-reality/mrtk-unity/mrtk2/features/ux-building-blocks/near-menu?view=mrtkunity-2022-05> (besucht am 17.03.2024).
- Blender - Bezier Curve. URL: <https://docs.blender.org/manual/en/latest/modeling/curves/primitives.html#bezier-curve> (besucht am 17.03.2024).

- Unity-Dokumentation - PhotoCapture. URL: <https://docs.unity3d.com/ScriptReference/Windows.WebCam.PhotoCapture.html> (besucht am 2.11.2023).
- Microsoft - Dokumentation - Foto-/Videokamera in Unity, URL: <https://learn.microsoft.com/de-de/windows/mixed-reality/develop/unity/locatable-camera-in-unity> (besucht am 2.11.2023)
- Unity-Dokumentation - Canvas. URL: <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/UICanvas.html> (besucht am 21.02.2024).
- Unity Dokumentation - Renderer. URL: <https://docs.unity3d.com/ScriptReference/Renderer.html> (besucht am 16.01.2024).
- Unity-Dokumentation - Job system overview. URL: <https://docs.unity3d.com/Manual/JobSystemOverview.html> (besucht am 21.02.2024).
- Unity-Dokumentation - Vector2. URL: <https://docs.unity3d.com/ScriptReference/Vector2.html> (besucht am 20.03.2024).
- Unity-Dokumentation - Coroutine. URL: <https://docs.unity3d.com/Manual/Coroutines.html> (besucht am 20.03.2024).
- Unity-Dokumentation - Coroutine2. URL: <https://blog.logrocket.com/using-coroutines-unity/> (besucht am 20.03.2024).
- Unity-Dokumentation - ARRaycastHit Doku. URL: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.0/api/UnityEngine.XR.ARFoundation.ARaycastHit.html> (besucht am 18.11.2023).
- Unity-Dokumentation - Raycast. URL: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.0/manual/raycast-manager.html> (besucht am 10.03.2024).
- npmjs - http-server. URL: <https://www.npmjs.com/package/http-server> (besucht am 15.03.2024).
- AWS - CORS. URL: [https://aws.amazon.com/de/what-is/cross-origin-resource-sharing/#:~:text=Cross%2DOrigin%20Resource%20Sharing%20\(CORS,Ressourcen%20in%20einer%20anderen%20Domain.](https://aws.amazon.com/de/what-is/cross-origin-resource-sharing/#:~:text=Cross%2DOrigin%20Resource%20Sharing%20(CORS,Ressourcen%20in%20einer%20anderen%20Domain.) (besucht am 15.03.2024).
- Mozilla - CORS-2. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS> (besucht am 15.03.2024).
- Webdesign Journal - Website Konzept. URL: <https://www.webdesign-journal.de/website-konzept/> (besucht am 19.02.2024).
- Textstrategie - Aufbau einer Website Inhalte und Struktur richtig Planen und Erstellen. URL: <https://textstrategie.de/2022/03/27/aufbau-einer-website-inhalte-und-struktur-richtig-planen-und-erstellen/> (besucht am 19.02.2024).
- Mozilla Developer Network - HTML. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (besucht am 19.02.2024).
- Mozilla Developer Network - CSS. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (besucht am 19.02.2024).
- Mozilla Developer Network - JavaScript. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (besucht am 19.02.2024).
- Bootstrap-Dokumentation. URL: <https://getbootstrap.com/docs/5.1/getting-started/introduction/> (besucht am 20.02.2024).
- Bootstrap-Dokumentation. CDN-Links. URL: <https://getbootstrap.com/docs/5.2/getting-started/introduction/> (besucht am 20.02.2024).
- Unity - QR-Code-Tracking. URL: <https://learn.microsoft.com/en-us/samples/microsoft/mixedreality-qrcode-sample/qr-code-tracking-in-unity/> (besucht am 2.11.2023).

- Microsoft-Dokumentation - Singleton. URL: [https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ee817670\(v=pandp.10\)?redirectedfrom=MSDN](https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ee817670(v=pandp.10)?redirectedfrom=MSDN) (besucht am 15.03.2024).
- home.earthlink.net - Singleton-2. URL: <https://web.archive.org/web/20202424122433/http://home.earthlink.net/~huston2/dp/singleton.html> (besucht am 15.03.2024).
- Microsoft-Dokumentation - Spatial Graph Node. URL: <https://learn.microsoft.com/en-us/dotnet/api/microsoft.mixedreality.openxr.spatialgraphnode?view=mixedreality-openxr-plugin-1.10> (besucht am 15.03.2024).
- Microsoft-Dokumentation - Lock-Statement URL: <https://learn.microsoft.com/de-de/dotnet/csharp/language-reference/statements/lock> (besucht am 20.03.2024).
- Unity-Dokumentation - Bounds. URL: <https://docs.unity3d.com/ScriptReference/Bounds.html> (besucht am 16.01.2024).
- Unity-Dokumentation - Transform.localPosition. URL: <https://docs.unity3d.com/ScriptReference/Transform-localPosition.html> (besucht am 20.03.2024).
- GeeksForGeeks - 0/1 Knapsack-Problem. URL: <https://www.geeksforgeeks.org/0-1-knapsack-problem-dp-10/> (besucht am 11.03.2024).
- GeeksForGeeks - Fraktionales Knapsack-Problem. URL: <https://www.geeksforgeeks.org/fractional-knapsack-problem/> (besucht am 11.03.2024).
- GeeksForGeeks - Begrenztes Knapsack-Problem. URL: <https://www.geeksforgeeks.org/extended-knapsack-problem/> (besucht am 11.03.2024).
- GeeksForGeeks - Unbegrenztes Knapsack-Problem. URL: <https://www.geeksforgeeks.org/unbounded-knapsack-repetition-items-allowed/> (besucht am 11.03.2024).
- Wikipedia - Mehrzieliges Knapsack-Problem. URL: https://en.wikipedia.org/wiki/Knapsack_problem#Multi-objective_knapsack_problem (besuscht am 11.03.2024).
- Wikipedia - Multidimensionales Knapsack-Problem. URL: https://en.wikipedia.org/wiki/Knapsack_problem#Multi-dimensional_knapsack_problem (besucht am 11.03.2024).
- Wikipedia - Mehrere Knapsack-Probleme URL: https://en.wikipedia.org/wiki/Knapsack_problem#Multiple_knapsack_problem (besucht am 11.03.2024).
- Wikipedia - Quadratisches Knapsack-Problem. URL: https://en.wikipedia.org/wiki/Knapsack_problem#Quadratic_knapsack_problem (besucht am 11.03.2024).
- Wikipedia - Geometrisches Knapsack-Problem. URL: https://en.wikipedia.org/wiki/Knapsack_problem#Geometric_knapsack_problem (besucht am 11.03.2024).
- Hindawi Journals - Solving the 0/1 Knapsack Problem Using Metaheuristic and Neural Networks for the Virtual Machine Placement Process in Cloud Computing Environment. URL: <https://www.hindawi.com/journals/mpe/2023/1742922/> (besucht am 11.03.2024).
- Wikipedia - Knapsack-Problem Anwendungen. URL: https://en.wikipedia.org/wiki/Knapsack_problem#Applications (besucht am 11.03.2024).
- Automationpanda - Arrange Act Assert. URL: <https://automationpanda.com/2020/07/07/arrange-act-assert-a-pattern-for-writing-good-tests/> (besucht am 15.03.2024).
- Microsoft-Dokumentation - Universal Windows Platform URL: <https://learn.microsoft.com/en-us/windows/apps/design/devices/designing-for-mr#overview> (besucht am 29.03.2024).
- Microsoft-Dokumentation - Build-Tutorial URL: <https://learn.microsoft.com/en-us/training/modules/learn-mrtk-tutorials/1-7-exercise-hand-interaction-with-objectmanipulator> (besucht am 29.03.2024).
- Microsoft-Dokumentation - Visual-Studio-Build-Settings URL: <https://learn.microsoft.com/de-de/windows/mixed-reality/develop/advanced-concepts/using-visual-studio?tabs=h12> (besucht am 29.03.2024).

- Microsoft-Homepage - Visual-Studio URL: <https://visualstudio.microsoft.com/de/> (besucht am 29.03.2024).
- Bloomreach - SPA URL: <https://www.bloomreach.com/en/blog/2018/what-is-a-single-page-application> (besucht am 29.03.2024).
- Microsoft-Dokumentation - QR-Code-Tracking-SDK URL: <https://learn.microsoft.com/de-de/windows/mixed-reality/develop/native/qr-code-tracking-cs-cpp> (besucht am 29.03.2024).
- Microsoft-Dokumentation - UnityTestFramework URL: <https://unity.com/how-to/automated-tests-unity-test-framework> (besucht am 29.03.2024).
- Bytehide - AttributesCSharp URL: <https://www.bytehide.com/blog/attributes-csharp> (besucht am 29.03.2024).