



MASTERARBEIT | MASTER'S THESIS

Titel | Title

Navigation und geocodierte Informationsbereitstellung mittels
AR-Applikationen: Erstellung und Anwendung einer AR-
Applikation im Bereich von Zustelltätigkeiten mittels Realtime-
Engines

verfasst von | submitted by
Raphael Siegl BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of
Master of Science (MSc)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree
programme code as it appears on the
student record sheet:

UA 066 856

Studienrichtung lt. Studienblatt | Degree
programme as it appears on the student
record sheet:

Masterstudium Kartographie und Geoinformation

Betreut von | Supervisor:

Ass.-Prof. Mag. Dr. Andreas Riedl

Inhalt

Abbildungsverzeichnis	iv
Abkürzungs-/ Fremdwortverzeichnis	vii
Kurzfassung	ix
Abstract	x
1 Einleitung.....	1
1.1 Problemstellung und Relevanz	1
1.2 Forschungsstand und Tools	5
1.3 Struktur, Methodik und Ziele	8
2 Theoretische Grundlagen.....	11
2.1 GIS(c)	11
2.2 Navigation	13
2.3 Ortung.....	15
2.4 Extended-Reality (XR).....	17
2.5 Realtime-Engines (Game-Engines).....	19
2.6 Zustellbranche	21
2.6.1 Tätigkeitsablauf	22
2.6.2 Tour/ Rayon.....	24
2.6.3 Regionsabhängigkeit	24
2.6.4 Zustellbedingungen	26
2.7 Verknüpfung der Bereiche	27
3 Rundum AR-Systeme	28
3.1 Begrifflichkeiten & Konzepte	28
3.2 Definition, Geschichte & Abgrenzung von AR-Systemen	31
3.2.1 Abriss zur Entwicklungsgeschichte von AR.....	33
3.2.2 Abgrenzbarkeit von AR.....	34
3.3 AR-Applikationstypen und darauf aufbauende Einteilungen	35
3.3.1 AR-Applikationstypen basierend auf deren technischer Umsetzung.....	35

3.3.2	Versuch der Einteilung nach Anwendungsszenarien (Mehler-Bicher & Steiger, 2014).....	38
3.3.3	Weitere Prinzipien und Einteilungsmöglichkeiten	39
3.4	AR und Kommunikation	39
3.4.1	AR-Funktionsweise und häufige Kommunikationsformen	40
3.4.2	Interaktionsmöglichkeiten in und mit AR-Systemen.....	53
3.5	Entwicklung von AR-Applikationen.....	56
3.5.1	Grundlegendes zu Unity und ausgewählten (AR-)SDKs/ Erweiterungen	60
3.5.2	Räumlichkeit in einer AR-Applikation (am Beispiel einer künstlerischen AR-Installation & AR-Navigationsapplikation):	64
4	Augmented-Reality Navigationssysteme.....	66
4.1	Beispielhafte Umsetzungen aus der Wissenschaft	66
4.1.1	Applikationsdesign und Kommunikation von Inhalten	67
4.1.2	Nutzbarkeit, Navigationsqualität und Spatial Knowledge Acquisition	69
4.1.3	AR mithilfe von HUD in Fahrzeugen.....	71
4.1.4	Navigation mittels Head-Mounted-Displays	72
4.2	Beispielhafte Umsetzungen aus der Praxis	74
4.3	„Learnings“ aus den Grundlagen und vorangegangenen Arbeiten	77
4.3.1	Technische Aspekte	78
4.3.2	Gestalterische Aspekte	81
5	Use-Case: Erstellung einer AR-Applikation für Zustelltätigkeiten	84
5.1	Funktionsanforderungen.....	84
5.1.1	Funktionen.....	84
5.1.2	Art der Bereitstellung von Navigationshinweisen	85
5.1.3	AR-HMD Systeme	86
5.2	Design-Idee	89
5.2.1	Auswahl der Hard- und Software	89
5.2.2	Überlegungen bezüglich des Aussehens der Applikation.....	91
5.2.3	Theoretische Einordnung der Applikation.....	94

5.3	Erstellung der Applikation	95
5.3.1	Lern- und Versuchsphase	95
5.3.2	Planen & Verstehen	96
5.3.3	Applikationserstellung - Aufbau der Arbeitsumgebung	98
5.3.4	Applikationserstellung Phase I	102
5.3.5	Alpha-Testing	111
5.3.6	Applikationserstellung Phase II	112
5.3.7	Beta – Testing	115
5.3.8	Finalisierung	116
5.3.9	„Veröffentlichung“	116
5.4	Endprodukt	117
5.5	Probleme bei und Learnings aus der Erstellung	122
6	Use-Case: Anwendbarkeit der AR-Applikation	126
6.1	Testung der Applikation	128
6.1.1	Fußläufige Testung:	128
6.1.2	Fahrzeugbasierte Testung:	130
6.1.3	Analyse der Verortungs-Problematik	131
6.2	Mögliche Verbesserungen, Ideen für zukünftige Adaptionen und unlösbare Probleme	135
6.3	Fazit zur Anwendung, Potentiale und Limits	138
7	Conclusio	140
8	Appendix	142
	Kommentierter Code des MapHandler-Skripts	142
	Kommentierter (Objekt-)Aufbau der DelivAR-Szene	144
9	Literaturverzeichnis	146

Abbildungsverzeichnis

Abbildung 1: eigene schematische Darstellung der die Arbeit beeinflussenden Teilbereiche	11
Abbildung 2: Das Reality-Virtuality continuum von Milgram et al. (Milgram, et al., 1994) verändert dargestellt nach (Interaction Design Foundation - IxDF, 2022)	17
Abbildung 3: Extended-Reality, deren Teilbereiche und ihr Immersionsgrad (Tremosa, 2024).....	17
Abbildung 4: Degrees of Freedom nach (Barnard, 2023)	30
Abbildung 5: Einteilung von AR-Systemen (eigene Darstellung)	31
Abbildung 6: AR-Applikationstypen (verändert nach (Nextech3D.ai, 2022))	36
Abbildung 7: Monokular/ Binokular im Vergleich (verändert nach (Antakli, et al., 2017))	44
Abbildung 8: See-Through-HMD/ Smart-Glass: XREAL Air2.....	44
Abbildung 9: Non-See-Through-HMD: Meta Quest 3 (Painter, Lewis, 20204).....	44
Abbildung 10: Look-Around-HMD: Google Glass (Reckmann, 2014).....	44
Abbildung 11: Vergleichsmatrix ausgewählter AR-Systeme.....	48
Abbildung 12: Das Kommunikationssystem zwischen User und AR-Inhalten nach (Ghazwani & Smith, 2020).....	53
Abbildung 13: Ablauf der Erstellung einer AR-Applikation (verändert nach (Gerasymov, 2024)).....	58
Abbildung 14: Anzeige von POIs in der App von Mata & Claramunt (Mata & Claramunt, 2013).....	68
Abbildung 15: Anzeige von Navigationsanweisungen in der App von Mata & Claramunt (Mata & Claramunt, 2013).....	68
Abbildung 16: UI der AR-Applikation von Chu et al. (Chu, et al., 2017)	69
Abbildung 17: Umsetzung der AR-Navigationsapplikation bei Huang et al. (Huang, et al., 2012).....	70
Abbildung 18: Google Live View Navigation, verändert nach (I., 2019).....	74
Abbildung 19: Google Live View Navigation Zielanzeige, verändert nach (Ranieri, 2020)	74

Abbildung 20: Google Live View Search, verändert nach (Phillips, 2022).....	75
Abbildung 21: Darstellung des Parkassistenten, verändert nach (Audi AG, 2011)	76
Abbildung 22: AR-Navigation auf einem Display im Auto am Beispiel eines Mercedes GLC verändert nach (Eddy, 2022).....	76
Abbildung 23: schematische Darstellung eines AR-HUD mit Navigationsanweisungen, verändert nach (Volkswagen, kein Datum a).....	76
Abbildung 24: Zum Einsatz kommende Hardware (XREAL AIR 2 & Samsung Galaxy S21 5G).....	89
Abbildung 25: Skizze der Benutzeroberfläche.....	91
Abbildung 26: Ablaufdiagramm zur Erstellung der Applikation.....	95
Abbildung 27: Darstellung der wichtigsten im Rahmen der Applikation verwendeten Erweiterungen zur reinen Unity-Umgebung.....	99
Abbildung 28: Elemente des Aufbaus der initialen Startszene	100
Abbildung 29: Aussehen der Unity-Editor Oberfläche und seiner Teile	101
Abbildung 30: Original-Controller Gestaltung im Vergleich zur eigenen Umsetzung...	107
Abbildung 31: Testroute am Campus der Uni Wien, visualisiert mithilfe von Google Earth.....	108
Abbildung 32: Beispielhafte Liste mit Zustellhinweisen, visualisiert in Excel	109
Abbildung 33: Sygic Real View Navigation (historisch) (Sygic a.s., 2017).....	109
Abbildung 34: Sygic Real View Navigation (Sygic, 2024)	110
Abbildung 35: ARGPS Beispielsprojekt (Fortes, 2022)	110
Abbildung 36: Benutzeroberfläche nach dem Start	117
Abbildung 40: Settings Tab und dessen Einstellungsoptionen	118
Abbildung 39: Auswahl eines an einer bestimmten Position zu platzierenden Objekts .	118
Abbildung 37: Auswahl einer vorab definierten und in der App hinterlegten Route.....	118
Abbildung 38: Suche eines Zielpunkts und Auswahl des selbigen.....	118
Abbildung 41: Anzeige der Route im Map-Tab.....	119
Abbildung 42: Anzeige einer Benachrichtigung bei Start einer Funktion	119
Abbildung 43: Beispiele für die Anzeige von Navigationshinweisen mithilfe der Standardeinstellungen – Fußnote 262 beachten!	120

Abbildung 44: Anzeige des Commit Buttons am Controller	120
Abbildung 45: Von links nach rechts: Anzeige eines Navigationshinweises direkt an einem Kreuzungspunkt; Navigationsanzeige mit Zustellhinweis; Hinweisanzeige, wenn eine Route außerhalb des Blickfelds liegt – Fußnote 262 beachten!	121
Abbildung 46: Dialog zum Beenden der Applikation.....	121
Abbildung 47: Zwei Beispiele für Änderungen und deren Dokumentation im fremden Code.....	124
Abbildung 48: Route am Campus der Universität Wien, visualisiert auf Basis von OSM/ BING Maps.....	126
Abbildung 49: Liste der Orte mit Zustellhinweisen der Campus Route ²⁶⁹	127
Abbildung 50: Route in Oberösterreich, visualisiert auf Basis von OSM/ BING Maps .	127
Abbildung 51: Liste der Orte mit Zustellhinweisen der Oberösterreich-Route ²⁶⁴	127
Abbildung 52: Yaw, pitch, roll bei der Kopfbewegung von Menschen nach (Jantunen, 2016).....	128
Abbildung 53: Inkorrekte Anzeige multipler Hinweistafeln.....	129
Abbildung 54: ARLocationRoot Objekt mit dessen Komponenten und den hier einstellbaren Parametern.....	132
Abbildung 55: Soll-Situation der Navigation bei korrekter Berechnung und Positionierung.....	133
Abbildung 56: Situation bei falscher Verortung	134
Abbildung 57: Resultierende Situation aus falscher Verortung	134

Abkürzungs-/ Fremdwortverzeichnis

AR	Augmented-Reality
ARGPS	AR + GPS Location SDK für Unity von Daniel Fortes
emap (electronic map)	Elektronische Karte (umgs.: „Navi“)
GIS	Geoinformationssoftware
GI(Sc)	Geoinformation(swissenschaften)
(d)GPS NAVSTAR GPS	(Differentielles) Global Positioning System Navigational Satellite Timing and Ranging – Global Positioning System
(d)GNSS	(Differentielles) Global Navigation Satellite System
FPS	Frames per Second
HMD	Head-mounted Display
HUD	Head-up display
IMU	Inertial Measurement Unit / Inertiale Messeinheit
IT	Information Technology (dt. Informationstechnologie)
LBS	Location-Based Service
MR	Mixed-Reality
PKW	Personenkraftwagen
POI	Point of Interest
Post (AG)	Österreichische Post AG
RADAR	Radio Detection and Ranging
Rayon	Zustellbezirk

RE	Realtime-Engines
RTK	Real-Time-Kinematik/ Echtzeitkinematik
SDK	Software Development Kit
UI	User Interface
VR	Virtual-Reality (oftmals auch “Augmented-Virtuality”)
XR	Extended-Reality

Hinweise zu Begrifflichkeiten

- Es sei darauf verwiesen, dass im Rahmen der Arbeit GPS teils synonym zu GNSS Verwendung findet.
- Unter Kreuzungspunkten sind Wegkreuzungen und Straßenkreuzungen zu verstehen.

Kurzfassung

Die vorliegende Arbeit setzt sich zum Ziel, einen Beitrag zur Thematisierung der Anwendbarkeit von Extended Reality und insbesondere Augmented-Reality für die Geoinformationswissenschaften zu leisten, wobei weiters spezifisch auf die Anwendung von Augmented-Reality auf Head-Mounted-Displays im Bereich der Navigation während Zustellgängen fokussiert wird. Zu diesem Zweck wird in einem ersten Schritt ein grundlegendes Wissen rundum Extended- wie auch Augmented-Reality und der in diesem Bereich zum Einsatz kommenden Hard- und Software aufgebaut werden, um sich im Weiteren auf die Verknüpfung von Navigation und Augmented-Reality konzentrieren zu können. Für diese Verknüpfung werden in einem ersten Schritt diverse, aus Wissenschaft und Wirtschaft stammende und diese Thematik besprechende Ausarbeitungen diskutiert und als theoretische Grundlage für das eigene Forschungsprojekt herangezogen. In einem zweiten weitaus ausführlicheren Schritt wird darauf aufbauend eine auf die speziellen Navigationsanforderungen im Zustellbereich angepasste AR-Navigationsapplikation für Head-Mounted-Displays (XReal Air 2 mit Samsung Galaxy S21 5G) in Unity entwickelt und im kleinen Rahmen getestet. Die Testung wird dabei zweigeteilt und einmal für städtisches Gebiet in fußläufiger Form und einmal in ländlichem Gebiet bei fahrzeugbasierter Fortbewegung angelegt, um die Anwendbarkeit hinsichtlich beider Szenarien zu erörtern. Mit dieser Herangehensweise konnte herausgearbeitet werden, dass Extended- und im Speziellen Augmented-Reality bei der Repräsentation von Inhalten auf Head-Mounted-Displays ein durchwegs großes Potential für die Geoinformationswissenschaften und insbesondere auch die Navigation, unter anderem für Zustellbereiche, bieten, obwohl bei der hierzu zu verwendenden Technik momentan noch einige Limitationen vorliegen. Abschließend sei festgehalten, dass Augmented-Reality in den Geoinformationswissenschaften bei Einsatz für die Navigation auf Head-Mounted-Displays großes Potential birgt, auch wenn es zur Erschließung desselben noch weiterer Forschung und Ausarbeitung bedarf: Weitere Entwicklungen sind mit Spannung zu verfolgen.

Abstract

This thesis seeks to contribute to the discussion evolving around the usability of extended reality and especially augmented reality in the field of the geosciences, by focusing on a use case exemplifying the usability of augmented reality on head-mounted-displays in the field of navigation. To this end, the first sections of this thesis present basic information on extended and augmented reality as well as the hardware and software that are commonly used in these fields, before leading over to the linkage between augmented reality and the field of navigation. Aiming at preparing the ground for the research project at the core of this thesis, the next step highlights studies that discuss said connection between augmented reality and navigation in various areas, including the fields of both scientific and economy-directed research. In a second step, this research is taken as a basis for the central piece of this thesis, namely the presentation and discussion of an AR-based navigation application for head-mounted-displays (XReal Air 2 with Samsung Galaxy S21 5G), created in Unity to fit the specific requirements of delivery processes, and tested in a small-scale, dual testing procedure. The application and its usability are assessed in two highly diverging scenarios that simulate typical delivery situations in an urban area (walking) and a rural one (driving), with the test person crossing distances, firstly, by walking and secondly, in a car. Notwithstanding some limitations, this thesis shows that extended and especially augmented reality hold enormous potential with regards to the representation of content on head-mounted-displays for the field of the geosciences and navigation (especially delivery processes). Nonetheless, there is much room and need for future research and development, as there are still several issues to be considered: Future enterprises in this direction, however, are hoped to amend these.

1 Einleitung

Interessiert man sich für technologische Entwicklungen und Neuerungen rundum den Bereich der Geoinformationswissenschaften und insbesondere den damit verknüpften Bereichen der Informationsvermittlung und der Navigation, so kommt man rezent um zwei Themenfelder nicht herum: Realtime-Engines, wie beispielsweise die Unity- oder Unreal-Engine, so wie Extended-Reality mit all seinen Ausformungen von Augmented-, Mixed-, und Virtual-Reality. Ergänzt man dies dann weiters mit der Beobachtung der Zunahme von Zustelldienstleistungen, welche von postalischer Zustellung über Paketzustellung bis hin zu Essenslieferungen reichen, und mit den Problemstellungen in diesen Tätigkeitsfeldern, gelangt man relativ rasch zu der Frage, welche Möglichkeiten den Geoinformationswissenschaften in Verbindung mit Extended-Reality und Realtime-Engines – welche sozusagen als Mittel zum Zweck genutzt werden, auch wenn sie eine Vielzahl weiterer mächtiger Funktionen unterstützen – zur Verfügung stehen, um Zustelltätigkeiten zu beeinflussen und im besten Fall zu vereinfachen und zu verbessern. War man darüber hinaus, wie ich, im Tätigkeitsfeld von postalischer Zustellung tätig und hat ein weiterführendes Interesse in der Darstellung von niederschweligen geocodierten Informationen in Echtzeit mittels Realtime-Engines und Extended-Reality, so ergibt sich für die eigene auszuarbeitende, hier vorliegende Masterarbeit die Konsequenz, diese auf die erwähnten Bereiche zu fokussieren.

1.1 Problemstellung und Relevanz

Verfolgt man in den letzten Jahren und insbesondere seit der Corona Pandemie die aktuellen Nachrichten über den Zustellbereich, so ergibt sich ein relativ klares Bild, welches von einer Vielzahl an strukturellen Problemen geprägt ist. Aufgrund ihrer thematischen Fokussierung distanziert sich die vorliegende Arbeit jedoch davon, das Zustandekommen der verschiedenen Problemfelder¹ detailgetreu darzustellen, und begnügt sich stattdessen damit, einzelne Problemfelder und verbesserungswürdige Teilbereiche – wie Einschulungen, die Wegfindung neuer unbekannter Routen, das Merken von Zustelladressen /-bedingungen – herauszugreifen und zu besprechen, um in weiterer Folge Möglichkeiten zur Verbesserung dieser einzelnen Teilbereiche aufzuzeigen. Um anschließend relevante und gegebenenfalls mit den Möglichkeiten der Geoinformation und den damit verbundenen Disziplinen verbesserbare Bereiche in der Zustellbranche zu definieren, bieten sich zum einen einige (Zeitungs- und Video-) Berichte an, wobei diese hinsichtlich der daraus ableitbaren Informationen relativ schnell an ihre Grenzen stoßen. Darum werden diese durch die eigenen

¹ Als Bereiche, die nicht im Rahmen der Arbeit thematisiert werden, stellen sich etwa die folgenden dar: Anstellungsverhältnisse, Arbeitsrecht, Löhne, ...

Erfahrungen² aus dem Bereich der Brief- und Paketzustellung im städtischem, aber auch ländlichen Gebiet bei der Post AG ergänzt.

Wer kennt es nicht: Zusteller*innen, die nicht klingeln. Pakete oder Briefe, die nicht zugestellt werden. Briefe, Pakete und Zeitungen, die verspätet ankommen. Falsch adressierte Sendungen. Oftmals ärgert man sich über solche Situationen, ohne jedoch über die Hintergründe, die zu solchen Situationen führen, informiert zu sein. Einen medial wirksamen Anstoß in Richtung Bekanntmachung der prekären Situationen für im Zustellbereich beschäftigten Personen machte dabei Jan Böhmermann bereits im Jahr 2018 mit seiner satirischen Neuinterpretation des Kampflieds „*Die Internationale*“ mit dem Titel „*Wir sind Versandsoldaten*“ (Böhmermann, 2018). Auch wenn sich das Lied von Böhmermann, wie auch die meisten in Zeitungen zu findenden Artikel, wie etwa jene von *Der Standard*, sich hauptsächlich mit den Arbeitsbedingungen von (selbstständigen) Paketzusteller*innen auseinandersetzen und somit nicht direkt in einen Bereich, der von den GISc beeinflusst werden kann, fällt, so lassen sich doch einige Aspekte ableiten, die von den GISc angesprochen werden können (Bruckner & Laufer, 2020; Dang, 2021; Bruckner & Gepp, 2023). Unter Rücksichtnahme auf die partielle Adressierbarkeit der Probleme im Zustellbereich mithilfe der GISc und den mit diesen verwandten Disziplinen ist vor allem der Artikel von Dang mit dem Titel „*Wie sieht der Arbeitsalltag als Paketzusteller aus*“ (Dang, 2021) zu beachten, da in diesem ein Paketzusteller über einen Arbeitstag hinweg von einer Reporterin begleitet wurde und dezidiert Thematiken rundum die während eines Zustellgangs zurückgelegte Distanz und die Gebietskenntnis angesprochen werden.

In diesem Bericht werden vor allem zwei problematische und den Arbeitsalltag erschwerende Aspekte aufgegriffen, die auch gemäß eigener Erfahrung als durchwegs relevant definiert werden können: Die teils relativ große Distanz, die während eines einzelnen Zustelltags zurückgelegt werden muss, und die (Un-)Kenntnis über das Gebiet, in dem die Zustellung stattfindet. Betrachtet man die diversen Quellen dann noch weiter und verknüpft diese mit den eigenen Erfahrungen, so tritt ein weiterer auf den ersten Anschein nur indirekt für die Geoinformation relevanter Faktor zu Tage: Jener der Zeit. Aufgrund der teils relativ großen Distanzen, die Abhängigkeit der motorisierten Zustellung von der Flüssigkeit des Verkehrs und unter Umständen fehlenden Ortskenntnis, die sich daraus ergeben kann, dass etwa in Gebieten, in denen zuvor noch nicht gearbeitet wurde, ohne Vorbereitung zugestellt werden muss, kommt es teils zu merkbarem Zeitdruck.

Zusammengefasst stellen sich also auf den ersten Blick die folgenden drei Punkte als durch die GISc adressierbare Thematiken dar:

- **Zeit** als relevanter Faktor
- zurückgelegter **Distanz**
- **Ortskenntnis**

² Für eine genauere Besprechung dieser Thematik siehe Kapitel 2.6.

Diese Punkte können dann basierend auf den eigenen Erfahrungen um die folgenden Punkte ergänzt werden:

- **Kenntnis** über Zustelladressen/ Routen/ Gebieten (**Einschulung**)
- Merken der **zuzustellenden Sendungen (+ Zustelladresse)**
- Bedenken der **rechtlichen Zustellbedingungen**

An dieser Stelle ist darüber hinaus natürlich klar, dass etwaige Thematiken, wie eben der Faktor Zeit und die während eines Zustelltages zurückzulegende Distanz zum Teil, neben einer effektiven (zeiteffizient und möglichst geringe Distanz) Routenplanung³, welche zumindest bei der Post AG aktiv betrieben wird, nur indirekt beeinflusst werden kann. Der Grund hierfür liegt dabei darin, dass die für einen Zustellgang benötigte Zeit a) von der Anzahl an zuzustellenden Sendungen b) von der zu bewältigenden Route c) von der Art wie die Route bewältigt wird (zu Fuß, per Auto, ...) und d) von gegebenenfalls auftretenden unglücklichen Umständen (Staus, lange Wartezeiten auf Kunden, ...) abhängt. Weiters den Aspekt der Zeit indirekt beeinflussend und verkomplizierend kommen dann noch Punkte rundum Arbeitsbedingungen (Entlohnung, Arbeitszeiten, körperliche Belastung, ...) in diesem Bereich hinzu, welche ihrerseits wieder dazu führen, dass der Bereich oftmals an Arbeitskräftemangel leidet und somit die gleiche Menge an Tätigkeiten von einer unterbesetzten Mannschaft erledigt, werden müssen. Die anderen vier, beziehungsweise drei Punkte, wenn man jenen der Ortskenntnis und der Einschulung zusammenfasst, sollten jedoch in weiterer Folge im Rahmen dieser Arbeit mit dem „Know-How“ aus den GISc (und den Möglichkeiten der Extended-Reality) betrachtet, analysiert und bearbeitet werden.

Geht man nun neben der eingangs erwähnten Verbesserung der Arbeitsbedingungen, welche wie erwähnt nur zum Teil von der vorliegenden Arbeit adressiert werden können nochmals zur Frage, weshalb es von Relevanz ist die oberhalb herausgearbeiteten Punkte weiterführen zu adressieren, zu bearbeiten und wenn möglich zu verbessern, so begründet sich dies hauptsächlich mit der möglichen teilweisen Verbesserung der Arbeitsbedingungen und den damit eng in Kontakt stehenden Punkt des vorherrschenden Arbeitskräftemangels, welcher sicherlich zum Teil, auf in der Arbeit zu thematisierenden, aber auch abseits davon befindlicher Arbeitsbedingungen zurückzuführen ist.

Untermauert wird dieser Arbeitskräftemangel im Bereich von Zustelltätigkeiten und die dabei entstehende Abwärtsspirale⁴ etwa durch die rezente Forderung, den Job des/ der Paketzusteller*in in die Liste der Mangelberufe aufzunehmen: In weiterer Folge würde dies auch Abänderungen im Hinblick darauf, wer den Job ausüben dürfe, mit sich ziehen – eine

³ Diese wurde im wissenschaftlichen Bereich auch bereits in einer Vielzahl an Arbeiten mit unterschiedlichen Fokussetzungen thematisiert.

⁴ Weniger Arbeitskräfte führen zu mehr Arbeitsaufwand bei den verbliebenen Arbeitskräften, da die Positionen nicht nachbesetzt werden können. Dieser erhöhte Arbeitsaufwand führt seinerseits wiederum oftmals dazu, dass Personen den Bereich verlassen.

Thematik, die sich aber außerhalb des hier zu thematisierenden Bereichs befindet. (Redaktion, 2023)

Diskutiert man die Relevanz der im Rahmen dieser Arbeit zu verknüpfenden Bereiche (siehe Abb. 1) dann noch außerhalb der Grenzen der eigentlichen (Zustellungs-)Thematik, so zeigt sich diese in Bezug auf das in den letzten Jahren sehr aktive Forschungsfeld rundum die Verknüpfung von GIS und RE, sowie die Repräsentation von dreidimensionalen semantischen Geodaten in Realtime-Engines, welche unter anderem gerade durch esri in den letzten Jahren stark forciert wurde (esri, 2023a; esri, 2023b; Hansen, 2021). Ähnlich stellt es sich im Bereich der Navigation⁵ mithilfe von AR-Technologien dar, wo beispielsweise Google – ähnliches existiert vonseiten Sygic (Sygic, 2024) – in deren bekannten Maps Service vor kurzen die Möglichkeit zur Suche und Navigation mittels AR, genannt „Live View“, implementiert hat (Cantisano, 2022; Hall, 2023). Dieser Trend ging auch an den Automobilherstellern nicht vorbei: So bieten diese in ihren Fahrzeugen mittlerweile vermehrt die Möglichkeit, sich Navigationshinweise mittels eines sogenannten Head-Up Displays auf die Frontscheibe zu projizieren beziehungsweise sich tatsächlich mithilfe einer Kamera und eines Bildschirms im Auto Sachverhalte in der „realen Welt“ auf einem Bildschirm anzeigen zu lassen (Carter, 2023b; Ulrich, 2022). Problematisch hierbei ist jedoch, dass zwar eine nicht unmerkliche Zahl an praktischen Umsetzungen festgestellt werden kann, sich der Bereich der Forschung aber rezent der Betrachtung solcher Umsetzungen und den daraus abzuleitenden Konsequenzen (z.B. Verkehrssicherheit) und Potentiale (z.B. der Anwendung im Zustellbereich) für eine Vielzahl an Bereich nur in sehr geringem Ausmaß oder überhaupt nicht widmete.

Zusammenfassend kann man dieses einleitende Kapitel über die Relevanz und Positionierung der Arbeit im Forschungsfeld also damit schließen, dass sich die vorliegende Arbeit an einem Überschneidungsbereich diverser rezent sehr aktiver (etwa RE in den GISc) oder de facto beinahe nicht vorhandener Forschungsbereiche (Zustellbranche) ansiedelt, wobei eine Verknüpfung selbiger (Erstellung einer AR-Navigationsapplikation für Zustellbereiche), wie in dieser Arbeit forciert, bisher noch nicht durchgeführt wurde. Hinsichtlich der Behandlung der Problematiken der Zustellbranche muss aber einschränkend erwähnt werden, dass nur ein Teil selbiger, wie erläutert, mithilfe der GISc behandelt werden kann und andere außen vorgelassen werden müssen. Trotz der potenziellen Möglichkeit manche Problematiken in dem Bereich zu adressieren, lassen sich jedoch zumindest rezent – mit Ausnahme der Behandlung von Fragestellungen zum Routing⁶ – keine Arbeiten in diesem Bereich feststellen. Hinzu kommt dann noch der Faktor, dass die diversen aus der Praxis stammenden AR-Navigationsapplikationen in rezenter Zeit kaum auf deren Potentiale und Konsequenzen für verschiedenste Bereiche, etwa den

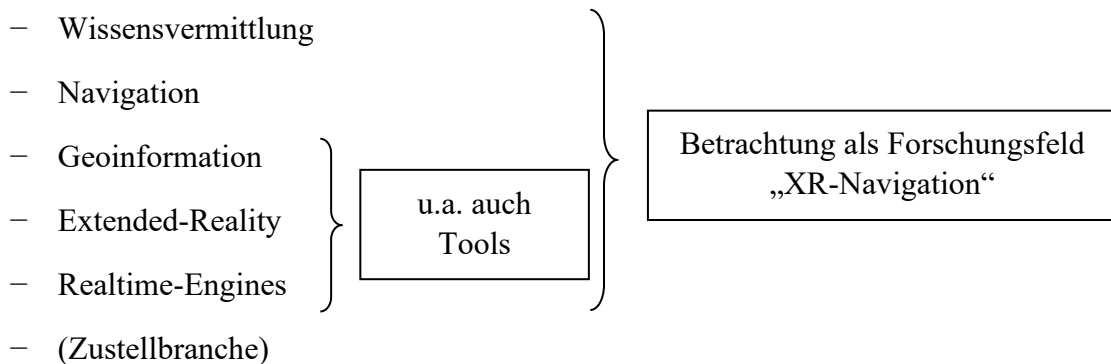
⁵ Navigation kann im Rahmen dieser Arbeit als ein sehr eng mit den GISc verknüpfter Bereich angesehen werden, auch wenn an dieser Stelle bewusst sein muss, dass die Navigation eine bei weitem längere Geschichte aufweist als die GISc. Darüber hinaus spielen in das Forschungsfeld der Navigation weitere sehr relevante Bereiche der geisteswissenschaftlich Forschungswelt, wie etwa die kognitive Rezeption und Wissensvermittlung hinein, wobei diese im Rahmen dieser Arbeit nur beiläufig Erwähnung und Betrachtung finden werden.

⁶ Für eine Thematisierung zum Routing allgemein siehe auch: (Golden, 2008; Toth & Vigo, 2014)

Zustellbereich hin untersucht wurden und sich somit im gesamten Bereich rundum AR-Navigation ein durchwegs großes Feld an bearbeitbaren Thematiken eröffnet.

1.2 Forschungsstand und Tools

Betrachtet man nun als nächstes den Forschungsstand, welcher für die vorliegende Arbeit von Relevanz ist, so müssen als erstes einmal die Hauptforschungsbereiche aus denen sich das eigene zu bearbeitende Forschungsfeld speist, welches in aller Kürze als „**XR-Navigation**“⁷ betitelt wird, definiert werden. Hierzu lassen sich die folgenden unterhalb angeführten fünf (beziehungsweise sechs) Bereiche⁸ feststellen, wobei drei davon nicht nur als Forschungsfeld, sondern als auch Tools betrachtet werden müssen:



Betrachtet man nun die Arbeiten und Publikationen zu den oberhalb dargestellten Bereichen, so kristallisiert sich relativ rasch heraus, dass es an dieser Stelle keinen Sinn ergibt die einzelnen Teilbereiche in ihrer Gänze zu thematisieren: zum einen um den Umfang der Arbeit nicht zu sprengen und zum anderen, um das Ziel nicht aus den Augen zu verlieren. Vielmehr werden nur die für den Bereich der „XR-Navigation“ direkt relevanten Beiträge diskutiert. Hierbei verhält es sich so, dass der Großteil der Arbeiten den Fokus auf dem Überschneidungsbereich der **Wissensvermittlung und der Navigation**, der **Navigation mithilfe von Extended-Reality** Formen (allen voran AR) oder der Realisierung von **dreidimensionalen semantischen Modellen in VR** mithilfe von GIS und RE legt. Auch wenn letzterer Verknüpfungsbereich im Rahmen dieser Arbeit nur in geringem Maße eine

⁷ Es wird an dieser Stelle XR anstelle von AR gewählt, da selbst wenn Navigation etwa in VR stattfände, ähnliche Gegebenheiten hinsichtlich der technischen Umsetzung wie auch der Bedürfnisse von Navigation in einer gewissen Gebietseinheit gegeben sein würden. Konkret bedeutet dies, dass etwa eine Navigationsapplikation für New York mittels AR denselben Bedingungen – abseits des Aufbaus eines kompletten Stadtmodells für VR – unterliegt wie wenn man selbiges in VR umsetzen möchte.

⁸ An dieser Stelle ist klar, dass sich die Bereiche überschneiden beziehungsweise ineinander wiederfinden (GISc-Navigation; Navigation-Wissensvermittlung), trotzdem wird es an dieser Stelle als sinnvoll angesehen, diese als eigene Bereiche herauszustreichen, da dies einen besseren Überblick über die das „XR-Navigations Forschungsfeld“ ausmachenden Thematiken zulässt.

Rolle spielt und sich im Rahmen dieser Arbeit eher nur dem Wissen (etwa rundum GNSS und Routing) aus den Geoinformationswissenschaften zur Umsetzung von für unter anderem in den GISc zu verortenden Arbeiten in Realtime-Engines bedient wird, sollte dieser doch zumindest kurz Erwähnung finden: Denn dieser Bereich stellte mehr oder minder den Startschuss für die Geoinformationswissenschaften dar, sich näher mit Realtime-Engines auseinander zu setzen und verlieh auch der Forschung hinsichtlich der Art der Bereitstellung von Navigationshinweisen neuen Schwung.

Weshalb die Bereiche der Geoinformation, Extended-Reality und Realtime-Engines oberhalb dann noch als Tools bezeichnet und betrachtet werden, liegt daran, dass diese oftmals in diversen Arbeiten, gerade wenn diese den Fokus auf eine der ersteren beiden oberhalb erwähnten Verknüpfungsbereiche setzen, nicht dezidiert erwähnt werden. Dies bedeutet an dieser Stelle jedoch nicht, dass diese nicht zur Erreichung der Ziele (beispielsweise mithilfe von aus diesem Bereich stammenden Wissen⁹, Soft- und Hardware¹⁰ oder ähnlichem) beigetragen haben – sie werden jedoch lediglich als Tools zur Erreichung der jeweiligen Ziele betrachtet und nicht als für die eigentliche Forschung relevant.

Für die Forschung im „XR-Navigationsbereich“ kristallisieren sich folgende Themenschwerpunkte heraus:

- Thematisierung der Art und Weise der Bereitstellung von Navigationsanweisungen und deren Limitierungen
- Thematisierung des Erfolgs des Erlernens von räumlichem Wissen bei der Nutzung von Navigationstools
- Thematisierung des Nutzens von XR bei der Navigation anhand praktische Anwendungsfälle
- Dreidimensionale (semantische) Darstellung von geocodierten Sachverhalten

Diese Themenschwerpunkte werden je nach Ausrichtung stärker oder schwächer behandelt und dabei oftmals zielgerichtet auf einen bestimmten Fokus (bspw. Fortbewegungsart¹¹, Art von XR, ...) hin weiter eingeschränkt. Interessant ist hierbei jedoch, dass sich die Arbeiten, welche sich dem Erlernen räumlichen Wissens widmen, teils in ihren Resultaten unterscheiden. So schließen etwa Huang et al. (Huang, et al., 2012), welche sich mit dem

⁹ Bspw. Wissen über Daten zum Straßennetz/Adressdaten und dessen Verwendung, GNSS, performantem Routing, Analysetools, Datenaufbereitung, ...

¹⁰ Klarere Weise stammt und existiert Soft- und Hardware nicht exklusiv für einen Bereich, jedoch kann deren Nutzung und der Grund für deren (Weiter-)Entwicklung oftmals einem Bereich zugeschrieben werden.
- bspw. Software: Den GISc entstammende Geoinformationssoftware, wie etwa QGIS/ ArcGIS und der digitalen Spielebranche entstammende Realtime-Engines, wie die Unity- und Unreal-Engine.
- bspw. Hardware: Ohne das Interesse an Extended-Reality (Anwendungen) für diverse Zwecke wären wohl eher kaum Head-Mounted-Displays entwickelt worden

¹¹ Diese sind in den Arbeiten immer entweder die Fortbewegung zu Fuß oder per Auto. Interessanterweise findet sich keine Thematisierung zur Fortbewegung mit öffentlichen Verkehrsmitteln oder die Kombination verschiedener Fortbewegungsarten.

Thema des Erlernens räumlichen Wissens mithilfe von digitalen Navigationskarten, auditiven Navigationshinweisen und AR-Navigationssystemen bei fußläufiger Bewegungsart beschäftigt haben, damit, dass sich hinsichtlich des Aufbaus von räumlichem Wissen keine signifikanten Unterschiede zwischen den Hilfsmitteln feststellen lassen. Yount et al. (Yount, et al., 2022) wiederum, die zu derselben Grundthematik – abgeändert hinsichtlich der Fokussierung der fußläufigen Fortbewegungsart zur Fortbewegung mit dem Auto und dem Ersetzen der auditiven Navigationshinweise mit der Verwendung von analogen Karten – arbeiten, konkludieren jedoch, dass zumindest in ihrem Testaufbau im Teilbereich der *survey-knowledge*¹² Unterschiede festgestellt wurden. Dies ist dabei jedoch nicht zwangsläufig auf einen besseren Wissensaufbau mit unterschiedlichen Navigationshilfsmitteln zurückzuführen, sondern kann unter anderem auch dem Testaufbau geschuldet sein. Generell wird aber in beiden Arbeiten auch darauf verwiesen, dass alle Methoden nicht den höchstmöglichen Lernerfolg im Hinblick auf das Erlernen einer Strecke bieten, da es sich eben um Hilfsmittel handelt und nicht um eine aktive Suche des Ziels ohne Hilfsmittel¹³. Besonders hervorzuheben ist an dieser Stelle noch das Faktum, dass Yount et al. nicht nur den Lernerfolg, sondern auch den Sicherheitsaspekt untersucht haben und dabei zu dem Ergebnis gekommen sind, dass AR-Navigation (im Vergleich zu *emaps* teilweise; im Vergleich zu analogen Karten in ganzer Linie) die Fahrsicherheit erhöht, was gerade im Hinblick der Bearbeitung der Fortbewegung zu Fuß und mit dem Auto in dieser Arbeit eine wichtige Erkenntnis darstellt.

Eine weitere wichtige Arbeit ist jene von Dünser et al. (Dünser, et al., 2012), in der untersucht wurde, wie es Probanden am einfachsten fällt sich zu einem bestimmten Ort zu navigieren: mithilfe einer digitalen Karte, einer digitalen Karte in Kombination mit AR-Navigationshilfestellungen oder nur AR-Navigationshilfestellungen. Auch wenn auf diese Fragestellung keine signifikante Antwort gegeben werden konnte, ist die Forschung gerade in Bezug auf das Nutzerempfinden und die Wegfindung von Interesse und bietet Einblicke für eine möglichst gute Umsetzung der eigenen Navigationsapplikation. Weiters finden sich einige Arbeiten zu praktischen Anwendungsfällen von AR, wie etwa zur Navigation durch einen Tourismusort, wie sie etwa von Pranoto et al. (Pranoto, et al., 2023) bearbeitet wurde, oder auch eine Vergleichsstudie zwischen AR und analoger Karte bei der Indoor-Navigation durch ein Museum von Rönisch (Rönisch, 2021), welche als Baustein zur Überblickung der Anwendungsmöglichkeiten und deren Performance in verschiedenen Anwendungsfällen im Rahmen der vorliegenden Arbeit dienen sollen.

Hinsichtlich des nur in Ausschnitten relevanten Bereichs der dreidimensionalen Darstellung geocodierter Sachverhalte in RE finden sich in den letzten Jahren einige Umsetzungen, wobei sich die abgedeckten Thematiken hierbei von Grundlagenarbeiten über die reine Darstellung von Sachverhalte, die Planung und Überwachung von Bautätigkeiten (inklusive „*participatory design*“) bis hin zu eher GIS-lastigen Aufgaben wie der Abfrage von semantischen Informationen an bestimmten Orten, dem Routing oder der Lärmausbreitung

¹² Sieh hierzu LRS-Modell in Fußnote 156.

¹³ ...ohne Hilfsmittel bedeutet in diesem Fall, dass zwar ggf. eine Karte vorliegen kann, jedoch keine Route darauf eingezeichnet ist.

erstrecken¹⁴. Ein Überblick zu dieser Thematik, dem Überschneidungsbereich von Game-/Realtime-Engines und GIS findet sich überdies in der unpublizierten Arbeit von Siegl (Siegl, 2022).

Betrachtet man nun abschließend den praxisorientierten Bezugspunkt (der Zustellbereich) der Arbeit und die wissenschaftliche, sowie mediale Auseinandersetzung mit diesem, so ist festzustellen, dass im ersteren keine und im zweiten einige, jedoch nur auf die Arbeitsbedingungen fokussierte, Artikel zu finden sind. Wie bereits eingangs erwähnt, beschäftigen sich die Artikel dabei mit dem Arbeitsalltag: Beispielsweise jener mit dem Titel „*Wie sieht der Arbeitsalltag als Paketzusteller aus?*“ (Dang, 2021), oder andere mit Bezugspunkt auf den Arbeitsbedingungen (Bruckner & Gepp, 2023), oder dem Mangel an Arbeitskräften (Redaktion, 2023), ohne jedoch die Betrachtung auf mögliche verbesserungswürdige (abseits von Arbeitszeitregelungen und Entlohnung) Bereiche – mit Ausnahme eines möglichst effizienten Routings – zu lenken, wie es zumindest für Teilbereiche in dieser Arbeit geschehen soll. Genau aus diesem Grund wird es im Rahmen der Arbeit auch unerlässlich sein, die durch die fehlende Thematisierung existierenden Lücken mit den eigenen Erfahrungen aufzufüllen.

1.3 Struktur, Methodik und Ziele

Nachdem in dem vorangegangenen Kapitel bereits kurz über die Fokussierung der Arbeit gesprochen wurde, soll an dieser Stelle nun die genaue Zielsetzung dargestellt werden. Begonnen wird dabei mit der Forschungs- sowie den Nebenforschungsfragen, um in weiterer Folge auf die Hypothesen überzugehen, welche die Forschung leiten. Im Anschluss folgt dann noch eine Betrachtung des strukturellen Aufbaus sowie den heranzuziehenden Methoden für die Beantwortung der Fragen und Hypothesen wie eine Besprechung der erwarteten Ziele.

Forschungsfrage:

Wie können AR-Technologien bei der Navigation und Bereitstellung geocodierter Informationen mit besonderem Hinblick auf Zustelltätigkeiten möglichst zielführend eingesetzt werden?

¹⁴ Zur Grundlagenforschung und einfachen Visualisierungen siehe: (Mohd & Ruzinoor, 2016; Ruzinoor, et al., 2014; Ruzinoor, et al., 2015; Ruzinoor & Mohd, 2018)

Zur Visualisierung von Modellen und Ableitung von Informationen aus Modellen siehe: (Huo, et al., 2021; Lütjens, 2018; Lütjens, et al., 2019; Thöny, et al., 2015)

Zu Visualisierung, Analysen & Participatory Design siehe: (Buyuksalih, et al., 2017; Herwig & Paar, 2002; Rafiee, et al., 2017; Stock, et al., 2008; Vletter, 2019)

Nebenforschungsfragen:

- Wie und wo kommen XR-Technologien zur Repräsentation geocodierter Informationen im Bereich der Navigation zum Einsatz?
- Welche Potentiale bieten RE und XR den GISc im Hinblick auf Navigationsfragestellungen und welche Beispiele lassen sich hierfür finden?
- Welche besonderen Anforderungen stellt der Bereich der Zustellung an Navigationsapplikationen bzw. in genereller Form an die Bereitstellung von relevanten (geographischen) Informationen?
- Welche Potentiale bietet eine Bereitstellung von Informationen mittels einer AR-Applikation dem Bereich der Zustellung und den darin beschäftigten Personen?

Hypothesen:

- AR-Navigationsapplikationen ermöglichen eine einfache unschwellige Informationsvermittlung im Bereich von Zustelldienstleistungen.
- Der Einsatz von AR-Navigation erleichtert den Arbeitsablauf von Zusteller*innen.
- Die Nützlichkeit von AR-Navigationsapplikationen im Zustellbereich unterscheidet sich zwischen städtischen und ländlichen Gebieten.
- Eine Repräsentation diverser geocodierter Informationen (Abstellgenehmigung, ...) birgt eine Vielzahl positiver Effekte bei Zustelldienstleistungen.

Um die oberhalb definierten (Neben-)Forschungsfragen wie auch Hypothesen bearbeiten zu können, ist es in einem ersten Schritt nötig, einen strukturellen Ablauf zur Beantwortung der selbigen auszuarbeiten, um sich Schritt für Schritt den Antworten zu nähern. Genau aus diesem Grund wird sich die Arbeit in weiterer Folge im nächsten Kapitel zuerst einmal mit den theoretischen Grundlagen (Kapitel 2) beschäftigen, um ein Verständnis ebendieser, deren Zusammenspiel und in weiterer Folge der Arbeit an sich zu erreichen. Anschließend wird in Kapitel 3 zu einer relativ umfangreichen Thematisierung von AR-Systemen per se, da es sich für die GISc mehr oder minder um neue Fahrwasser handelt und der Bereich in sich auch oftmals relativ viel Unklarheiten aufweist übergegangen. Das vierte Kapitel widmet sich dann konkreten, in der Forschung oder Praxis zu verortenden Arbeiten zum Thema AR-Navigationsapplikationen und den daraus (für diese Arbeit) abzuleitenden „Learnings“. Im fünften Kapitel wird zur konkreten Ausarbeitung einer eigenen AR-Navigationsapplikation für den Zustellungsbereich übergegangen, wobei die Struktur dieses Kapitels dabei so angelegt ist, dass zu Beginn nochmals kurz das theoretische Grundgerüst dargelegt wird, bevor zur eigentlichen Ausarbeitung und weiter zum Endergebnis und den „Learnings“ aus dem Erstellungsprozess übergegangen wird. Kapitel 6 wiederum bezieht sich auf die eigentliche im vorangegangenen Kapitel erstellte Applikation und deren

Nutzbarkeit, während das finale Kapitel dazu dient, abschließende Worte zu finden und ein Fazit einzuarbeiten.

Um alle diese einzelnen Schritte bewerkstelligen zu können, wird es in weiterer Folge für Kapitel 2, 3 und 4 nötig sein, sich mit den oberhalb bereits im Forschungsstand zum Teil kurz dargelegten theoretischen und praktischen Arbeiten und Technologien genauer auseinanderzusetzen und die aus diesen ableitbaren Informationen, Konzepte und Methoden zu diskutieren um in weiterer Folge die sich daraus ergebenden Resultate auf die eigene Fragestellung und geplante praktische Umsetzung umzulegen. Kapitel 5 wiederum beschäftigt sich mit der praktischen Erstellung einer AR-Applikation zur Navigation und den dahinterliegenden Methoden. Hierbei wird versucht, den Spagat zwischen technisch viel zu weit führender Erläuterung – etwa durch eine detailgetreue Erläuterung jeder Codezeile – und dem oftmals bei anderen Arbeiten mit Praxisteil aufscheinenden Überspringen des Prozesses der Erstellung und der reinen Präsentation des fertigen Produkts zu schaffen. In Kapitel 6 und 7 folgt dann anschließend an die Erstellung der Applikation noch ein kurzes eigenes Testing der zuvor erstellten Applikation und ein abschließendes Fazit hinsichtlich der Nutzbarkeit solcher Anwendungen im angegebenen Bereich (+ darüber hinaus) und die generellen für den Bereich der Geoinformationswissenschaften ableitbaren weiteren potenziell interessanten und noch zu beforschenden Themen im vorliegenden Forschungsbereich.

2 Theoretische Grundlagen

Nachdem eingangs bereits einige Worte zur Arbeit und den einzelnen Teilbereichen, aus denen sich diese zusammensetzen, verloren wurden (und somit ein gewisses Maß an Grundkenntnissen über die einzelnen Bereiche vorausgesetzt wurde), ist es nun an der Zeit, diese noch einmal im nötigen Umfang zu betrachten. Somit soll zum einen den Leser*innen ein thematischer Überblick geboten, als auch das theoretische Grundgerüst für die Arbeit geschaffen werden, um später darauf aufbauend das Potential einer Verknüpfung der Teilbereiche herausarbeiten zu können. Eine schematische Darstellung der Verknüpfung zwischen den einzelnen Bereichen und einigen hierfür relevanten Schlagwörtern kann Abbildung 1 entnommen werden.

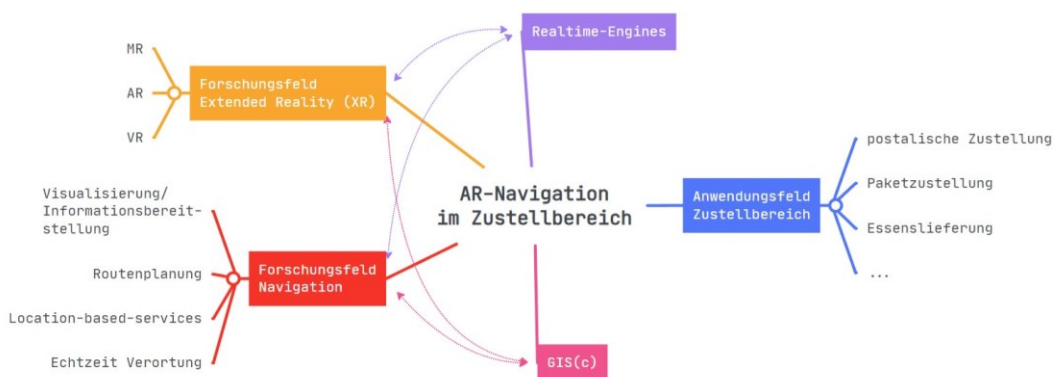


Abbildung 1: eigene schematische Darstellung der die Arbeit beeinflussenden Teilbereiche

2.1 GIS(c)

Begonnen sei an dieser Stelle zuerst einmal mit der Besprechung jener Forschungsrichtung, der die Arbeit an sich entstammt: Den Geoinformationwissenschaften (engl. Geographic Information Science) und dem unabdinglich damit verknüpften Bereich der Geoinformationssysteme (engl. Geographic Information Systems), auch wenn im Rahmen dieser Arbeit letztere keine praktische Anwendung finden werden.

Möchte man an dieser Stelle eine allgemein gültige Definition für die GISc und GIS aufstellen, so gestaltet sich das insofern schwierig, als dass sich die GISc aus allen raumbezogenen wissenschaftlichen Disziplinen speist. Im Umkehrschluss bedeutet dies aber ebenso, dass die GISc diese Bereiche wiederum mit Wissen aus deren Bereich, also beispielsweise räumliche Zusammenhänge, Muster und deren Bedeutung speist und somit einen großen Grad an Interdisziplinarität aufweist. Es handelt sich also um eine Disziplin, welche an den Schnittstellen zu vielen anderen Disziplinen steht und diese somit gut miteinander verknüpfen kann. Um Wissen in und durch die Disziplin der GISc generieren zu können, wird oftmals auf GIS, welche mit Geodaten, die wiederum aus verschiedenen fachfremden Disziplinen – oder dem facheigenen Bereich der Geodäsie – stammen, gespeist, um diese folglich zu analysieren und visualisieren. Somit sind GIS am ehesten „als ein

computergestütztes System zu verstehen, mit dem raumbezogene Daten (geographische Daten) gesammelt, verwaltet, ausgewertet und dargestellt werden können“ (Christiansen & Erb, 2023), ohne eine Aussage zur Datenherkunft treffen zu können. Diese Definition deckt sich darüber hinaus auch mit jenen der United States Geological Survey (USGS, kein Datum) und National Geographic (National Geographic, 2017). Weiters lässt sich aufbauend auf der Definition von GIS feststellen, dass die GISc oftmals als die ein GIS umgebende Wissenschaft definiert werden, welche sich mit der Konzeptionalisierung, den Theorien und der Entwicklung von GIS und der Interaktion der aus GIS erstellten Produkte mit dem Menschen beschäftigt (esri, 2023). Gut zusammengefasst wurde dies von der GISGeography, welche ein GIS so definiert, als dass es das „was?“ und „wo?“ beantwortet, während sich die GISc mit dem „wie?“ und der Entwicklung des GIS beschäftigen (GISGeography, 2023). Diese Definition greift jedoch insofern zu kurz, als dass sie den Bereich der Geodäsie, welcher in gewisser Weise auch zu den GISc zählt, und den hohen Stellenwert der Interdisziplinarität der GISc vollkommen außer Acht lässt.

Im Bezug auf die vorliegende Arbeit kommt dem Bereich der GISc insofern eine besondere Position zu, als dass sie aufgrund ihrer Interdisziplinarität mit den zwei im Anschluss zu betrachtenden Bereichen unweigerlich verknüpft ist, beziehungsweise diese teils inne hat: Der Navigation und der Echtzeitpositionierung/Ortung. Ohne diese drei Bereiche wäre etwa die Wegfindung in unbekannten Gebieten heutzutage kaum noch vorstellbar. So stellen etwa die Technologien, welche für die Echtzeitpositionierung Verwendung finden (Positions-)Daten bereit, welche (in einem GIS) verarbeitet und dann wiederum für die Navigation verwendet werden können. Das grundlegende Wissen hierzu, also wie etwa Echtzeitpositionierungssysteme (grundlegend) funktionieren – für eine detaillierte Funktionsweise muss hierzu oftmals tiefer in die jeweilige Technologie und beispielsweise unter anderem in die Physik eingetaucht werden –, oder wie Routen basierend auf verschiedensten Parametern berechnet werden und schlussendlich am besten für verschiedenste Anwendungszwecke visualisiert werden können, sammelt sich dabei in den Geoinformationswissenschaften und schafft es somit, einen Mehrwert aus den verschiedenen an sich noch „unveredelten“ Informationen zu generieren. Genau bei dem letzten Punkt der Präsentation von Informationen für eine*n Nutzer*in ergibt sich an dieser Stelle ein weiterer für die GISc interessantester Verknüpfungspunkt von GIS und RE und in weiterer Folge XR. So ermöglicht eine Verknüpfung von RE und GIS den GISc die performante (Echtzeit-) Darstellung von Informationen in verschiedenster Weise („3D“ am Bildschirm¹⁵; XR mit VR, AR, MR) und GIS SDKs wiederum ermöglichen es in einfacher Weise geocodierte Daten in RE, beispielsweise für die Spieleentwicklung, zu nutzen. Auf abstrakterer Ebene darf an dieser Stelle darüber hinaus ein weiterer Punkt nicht aus dem Auge verloren werden: Sobald eine Form von Interaktion mit realen räumlichen Sachverhalten auf der Erdoberfläche vorliegt – was zugegebenermaßen unabdinglich ist – befindet man sich im Feld der Geographie, wozu auch die GISc gehört. Somit zeigt sich, wenn wir an dieser Stelle über eine Navigationsapplikation nachdenken, die in einer

¹⁵ Zur Darstellung geocodierter Informationen im dreidimensionalen Raum sei an dieser Stelle angemerkt, dass dies auch direkt mit GIS möglich ist, jedoch die Performance und der Funktionsumfang in diesem Bereich bisher ein großes Manko darstellt.

Realtime-Engine erstellt wird und hierfür GPS und Routing eingesetzt wird – wie eben im Rahmen der vorliegenden Arbeit –, wiederum die Interdisziplinarität der GISc, ohne dass eine konkrete Verwendung eines GIS vorliegt.

2.2 Navigation¹⁶

Bei genauerer Betrachtung des Bereichs rundum den Begriff der Navigation, dessen teilweise Einbettung in die GISc bereits angemerkt wurde lohnt zuerst einmal ein Blick auf die Herkunft des Wortes. Das Wort Navigation leitet sich vom lateinischen „*navigatio*“, zu Deutsch „Schifffahrt“ ab, wobei sich dieses wiederum aus den beiden lateinischen Wörtern „*navis*“ („Schiff“) und „*agere*“ („führen/ lenken/ ...“) zusammensetzt und sich somit alleine daraus schließen lässt, dass es sich um einen bei weitem älteren Forschungsbereich als jenen der GISc handelt. Betrachtet man an dieser Stelle darüber hinaus die Verwendung des Wortes Navigation im heutigen Sprachgebrauch, so hat sich von der ehemaligen Bedeutung, zumindest im Deutschen und Englischen nur mehr der letztere Teil des „*agere*“, also „führen“, oder noch viel eher „führen lassen“ ohne die vormals exklusive Einschränkung auf Schiffe erhalten, wobei die Geoinformation in diesem Aspekt gerade in Bezug auf „führen lassen“ mit Routinganalysen und vielem mehr ansetzt.

Über die Zeit hinweg gleich geblieben sind also nicht unbedingt die Rahmenbedingungen und Disziplinen, in denen sich Navigation abspielt, jedoch aber die Tätigkeiten und Anforderungen, die mit ihr in Verbindung stehen: Es geht um die Notwendigkeit der Bestimmung der aktuellen Position, das Wissen über die Position des Zielortes, die Wegfindung und die Beobachtung des Wegs von der aktuellen Position zum Zielort¹⁷. An dieser Stelle soll ergänzend erwähnt werden, dass sich diese aus dem englischen stammende Klassifikation nach Downs and Stea (Downs & Stea, 1973) richtet, die sich seit 1973 im Großen und Ganzen nicht verändert hat und im Original in folgende Teilbereiche unterteilt wird:

- 1) **Orientation** (Orientierung – also Bestimmung der aktuellen Position)
- 2) **Route Decision** (Entscheidung über die zu wählende Route)
- 3) **Route Monitoring** (Beobachtung der Route während des Zurücklegens der selbigen)
- 4) **Destination Recognition** (Erkennung/ Bestimmung des Zielortes)

Hinsichtlich des Navigierens entlang der Route definieren Downs und Stea darüber hinaus noch drei verschiedenen Faktoren, wie etwa **Landmarks** (dt. Wahrzeichen bzw. Orte mit

¹⁶ Als Navigationsgeräte werden im Rahmen dieser Arbeit alle digitalen Hilfsmittel zur Navigation verstanden, welche a) die Möglichkeit der Ortung und b) die Möglichkeit zur Ausgabe von Navigationsanweisungen bieten.

¹⁷ Im Englischen wird Navigation dann weiters teilweise auch noch in zwei Teilbereichen unterteilt: Der Wegfindung (eng. „wayfinding“), also den kognitiven Bestandteil und die Bewegung (eng. „motion“ oder auch „travel“), den physischen Bestandteil. Für eine weiterführende Thematisierung siehe: (Darken & Peterson, 2001)

besonderem Wiedererkennungswert), **Street-Names** (dt. Straßenname) und **Distance** (dt. Distanz) welche für eine erfolgreiche Navigation in angeführter Reihenfolge von Relevanz sind (Downs & Stea, 1973).

Im Zuge dessen stellt sich die Frage, welche Möglichkeiten a) zur **Bestimmung der Position** und b) zur **Wegfindung** vom Ausgangspunkt zum Zielpunkt existieren. Geht man an dieser Stelle von zwei bekannten Positionen aus, die beispielsweise mithilfe einer Karte bestimmt worden sind, so gibt es eine Vielzahl an unterschiedlichen teils Jahrtausende alte Methoden, wie etwa das Navigieren anhand von Orientierungspunkten¹⁸ (eng. „landmarks“), mittels der Gestirne (Sonne und Sterne) oder mit einem Kompass, wobei es mit den letzteren beiden möglich wird die Position zu jedem beliebigen Zeitpunkt (unter Annahme der permanenten Beobachtung der sich ändernden Parameter der Richtung und Geschwindigkeit¹⁹) zu bestimmen. Betrachtet man nun die heutzutage am bekanntesten und häufigsten anzutreffenden Hilfsmittel (abseits von einer analogen Tourismus-/Straßenkarte und in der Welt vorzufindenden Wegweisern) wie etwa GNSS²⁰ und RADAR, so ist hierbei interessant, dass eine Bestimmung der aktuellen Position ad hoc, das heißt ohne die Notwendigkeit der Beobachtung von Richtung und Geschwindigkeit zu vorangegangenen Zeitpunkten, möglich ist und die Navigation somit um ein Vielfaches erleichtert wird. (National Geographic, 2023)

Weiters zeigen rezente Studien, dass es durch die Möglichkeit der Echtzeitpositionierung (bspw. mittels GNSS) und der breiten Verfügbarkeit von Geräten zur Berechnung und Repräsentation von Navigationsanweisungen²¹ (bspw. Navigationsgeräte oder Apps am Smartphone) so ist, dass ein großer Teil der Autofahrer*innen Navigations- oder Verkehrsmeldungsapps nutzt (Kamalanathsharma, et al., 2015), selbst wenn der Weg bekannt ist (Lo, et al., 2011). An dieser Stelle sollte darüber hinaus klar sein, dass anderen Mitteln zur Erleichterung der Navigation, wie etwa analogen Karten und/oder²² einem Kompass, in der Regel nur mehr eine sehr geringe Bedeutung zukommt.

¹⁸ Orientierungspunkte können sowohl natürliche Gegebenheiten darstellen, wie etwa Berggipfel, Wasserfälle oder Ähnliches, aber auch menschengebaute Objekte, wie Gebäude, Straßen und so weiter.

¹⁹ Richtung und Geschwindigkeit können mathematisch errechnet werden, wobei eine genaue Erläuterung des Vorgehens an dieser Stelle zu weit führen würde.

²⁰ Eine genauere Erläuterung zu Echtzeitpositionierung/ Ortung und somit auch GNSS folgt aufgrund der ausgesprochenen Wichtigkeit für die vorliegende Arbeit in Kapitel 2.3

²¹ Dieser Punkt stellt wiederum einen sehr engen Verknüpfungspunkt zu GIS dar, weil für die Navigation Geodaten, wie etwa Verkehrsnetzdaten, vorliegen müssen, die dann verarbeitet und analysiert werden können müssen.

²² An dieser Stelle soll mit der Formulierung darauf hingewiesen werden, dass in besiedelten Regionen eine Karte alleine mit den menschengebauten Informationen (bspw. Straßenschilder) in der Umwelt ausreicht, um zu navigieren und kein Kompass, wie etwa in Regionen ohne menschengebaute Informationen (Brachflächen, Wäldern, ...) oder anderer Landmarks (bspw. Wüsten), benötigt wird.

Routing (dt. Tourenplanung)

Ein sehr eng mit der Navigation verknüpfter und für die Arbeit relevanter Begriff ist Routing, wobei es sich hier im Sinne des Transports um den Prozess der Erstellung eines bestimmten (optimalen) Weges/ einer bestimmten Route (von A nach B) handelt (Cambridge Dictionary, 2014). Der optimale Weg hängt hierbei dann von diversen Faktoren ab, etwa ob es sich um die schnellste oder kürzeste Route handeln soll, ob Echtzeitinformationen zu Verkehr einbezogen werden, oder ob es darum geht, eine Vielzahl an Adressen optimal hintereinander miteinander zu verknüpfen, wie es etwa bei Zustelltätigkeiten der Fall ist.

2.3 Ortung

Der dritte an dieser Stelle für die Arbeit relevante und abermals mit den GISc eng verknüpfte Bereich beschäftigt sich mit der Frage, wie die Position/ der Standort eines Objekts – im Rahmen der Arbeit eines Geräts und somit die Position des/der Nutzer*in – festgestellt werden kann. Diese Thematik findet sich dabei sowohl in der Geoinformation als auch in der Navigation wieder und wird heutzutage, anders als bei historischen Verfahren²³, mittels Funkwellen in Echtzeit realisiert. Das bedeutet, dass zwischen Empfangsgerät und dem/ den weiteren Gerät(en) im Raum – etwa mittels Satelliten und/oder eingemessener fix verorteter Base-Stations – Funkwellen hin und her gesendet werden und es dadurch möglich ist den Standort, ohne eine merkliche Verzögerung in Echtzeit zu ermitteln. Als wichtigstes solcher Ortungssysteme haben sich sogenannte „*Global Navigation Satellite Systems*“ (kurz: GNSS) etabliert, welche aus einer Vielzahl an verschiedenen Systemen, wie etwa dem amerikanischen NAVSTAR-GPS, Galileo, GLONASS, Beidou und viele weiteren, bestehen. Zusätzlich muss an dieser Stelle erwähnt werden, dass umgangssprachlich zumeist nicht von GNSS, sondern von GPS quasi als pars pro toto gesprochen wird. Neben den GNSS und diese unterstützenden Technologien, wie etwa die Verwendung von GNSS in Kombination mit eingemessenen Bodenstationen (dGNSS), RTK, IMUs oder einer aktiven Internetverbindung am Empfangsgerät, existieren noch weitere Möglichkeiten zur Ortung in Echtzeit mittels Funkwellen: Etwa RADAR-Systeme oder Funkzellenortung, wobei diese in der Ausbreitung der Verwendung aufgrund diverser Umstände, auf die im Rahmen der Arbeit nicht weiter eingegangen werden wird, nur eine untergeordnete Rolle im Alltag spielen.

GNSS²⁴, kann dabei vereinfacht gesagt in drei Teile eingeteilt werden: Der erste Teil ist die Raumeinheit, umfasst also die Satelliten, der zweite die Bodenstationen und der dritte das Empfangsgerät. Wird nun vom Empfangsgerät der Standort angefordert, so benötigt dieses wiederum Kontakt zu mindestens drei Satelliten (jeder weitere Satellit dient der

²³ Für einen Überblick historischer Ortungsmethoden siehe auch: (National Geographic, 2023)

²⁴ Dieser Abschnitt soll nur eine kurze Erläuterung der Funktionsweise und keine allumfassende Erklärung der Funktionsweise der Systeme rundum GNSS darstellen. Für eine genauere Erläuterung siehe: (Groves, 2013)

Verbesserung der Genauigkeit), um den Standort mittels eines räumlichen Bogenschnitts der Laufzeit der übertragenen Signale berechnen zu können. Die Bodenstation dient dann weiters dazu, die Raumeinheit, wie etwa deren Flugbahn zu überwachen und Korrekturdaten zur Messung zur Verfügung zu stellen, um so die Genauigkeit der Position zu verbessern. Wird ein GNSS noch, wie eingangs erwähnt, in Kombination mit zusätzlichen fest eingemessenen Bodenstationen, welche ihrerseits deren koordinativ verorteten Standpunkt an das Empfangsgerät weitersenden oder RTK, also vereinfacht gesagt einer temporären Bodenstation, die ihrerseits Kontakt zu den Satelliten hat und die Werte ihrer Position auch ans Empfangsgerät weitersendet verwendet so erhöht sich die Genauigkeit der Verortung nochmals erheblich. Eine aktive Internetverbindung, wie sie bei Smartphones üblich ist, verbessert wiederum die Geschwindigkeit der initialen Ortsbestimmung und zu einem gewissen Grad auch die Genauigkeit. Falls verbaut, kann zudem eine IMU die Genauigkeit bei Bewegungen erhöhen, da diese durch die Messung der Geschwindigkeiten des Empfängergeräts genaue Bewegungsdaten vom letzten Messzeitpunkt bereitstellt. Insgesamt erhält man somit durch die Verortung eines Geräts mittels GNSS Informationen zur Position auf der Erde – als Zahlenwerte im WGS84 Koordinatensystem – und, sofern eine Überwachung der Bewegung passiert (insbesondere in Kombination mit einer IMU), zu den (Blick-)Richtungen. Das direkte Resultat umfasst somit sechs Parameter: geogr. Länge (in WGS84), geogr. Breite (in WGS84); geogr. Höhe, und die drei Drehwinkel (Yaw, Roll, Pitch) im Raum (siehe Abb. 52).

Location-Based Service²⁵

Ein wichtiger Begriff im Hinblick auf die Ortung von mobilen Geräten, welcher an dieser Stelle unbedingt erwähnt werden muss, ist jener der Location-Based Services (dt. standortbezogener Dienst). Grundsätzlich handelt es sich bei solchen Services um „[...] Dienste, die auf die aktuelle Aufenthaltsposition abgestimmte Informationen zur Verfügung stellen [...]“ (Markgraf, 2018). Diese Service basieren in ihrer Funktionsweise darauf, dass zum einen Informationen an einen/ mehrere Empfänger gesendet und von diesem/ diesen zurückerhalten werden, um daraus die aktuelle Position zu berechnen²⁶. Auf diesen Standortinformationen aufbauend, werden dann gewisse Inhalte, wie etwa die Ausgabe eines Navigationshinweises kurz bevor man eine Weggabelung erreicht (bei einer Navigationsapplikation) kommuniziert²⁷ (Froehlich, 2022).

²⁵ Manchmal findet sich hierfür auch der Begriff eines Location Dependent Service (LDS).

²⁶ Hierzu sei auf die Funktionsweise von GNSS, oder auch die Verortung mittels Mobilfunkmasten, WLAN, etc. verwiesen.

²⁷ Inhalte können in verschiedenster Weise, wie etwa auditiv, visuell, ... vermittelt werden.

2.4 Extended-Reality (XR)

Im Bezug auf die Betrachtung des Forschungs- und Technologiebereich Extended-Reality, ist es in weiterer Folge zuerst einmal nötig, sich mit dem Realitäts-Virtualitäts-Kontinuum (eng. Reality-Virtuality continuum) oder

auch seltener nach einem dem Begründer Paul Milgram benannten Milgram-Kontinuum (eng. Milgram continuum) auseinander zu setzen. Grundlegend definiert diese Aufschlüsselung in einfacher Weise die Vermischung von „realer“ und „virtueller“ Welt, wobei an beiden Enden eine Ausklammerung der jeweils anderen geschieht und in der Mitte die beiden Welten in unterschiedlich starker Weise vermischt werden: Bei Augmented-Reality dominiert die reale, bei Augmented-Virtuality (heutzutage zumeist als VR bezeichnet) die virtuelle Welt²⁸. Gerade wegen dieser relativ klaren Abtrennung der beiden äußeren Bereiche und den fluiden Übergängen zwischen diesen, den Mixed-Reality Bereich definierenden Bereichen, steht dieses Modell zwar in diversen Aspekten in der Kritik²⁹, nichtsdestotrotz stellt es aber zumindest momentan noch die „state-of-the-art“ Einteilung dar, wodurch dessen Betrachtung unerlässlich ist – wie auch eine kritische Diskussion desselben. (Interaction Design Foundation - IxDF, 2022)

Geht man an diesem Punkt nun dazu über Extended-Reality (XR) zu definieren, so ist dies ähnlichen Schwierigkeiten unterlegen wie das Reality-Virtuality Kontinuum. Insgesamt lässt sich jedoch sagen, dass es sich bei Extended-Reality um einen Sammelbegriff für die verschiedenen Arten der

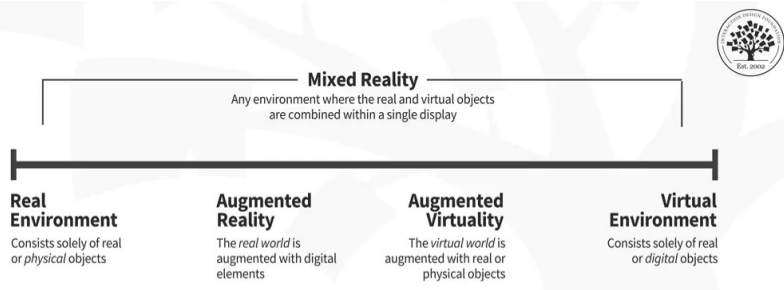


Abbildung 2: Das Reality-Virtuality continuum von Milgram et al. (Milgram, et al., 1994) verändert dargestellt nach (Interaction Design Foundation - IxDF, 2022)

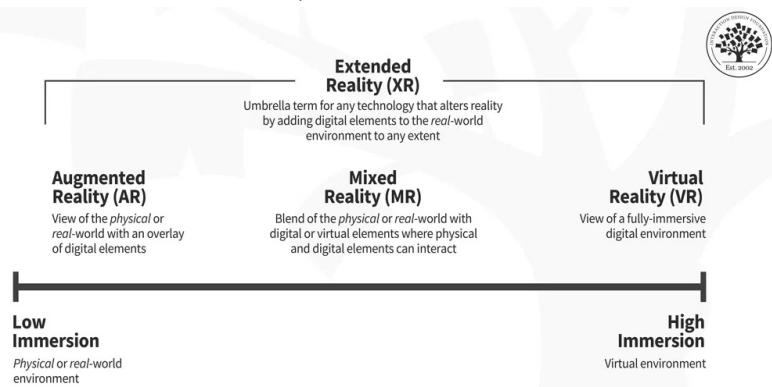


Abbildung 3: Extended-Reality, deren Teilbereiche und ihr Immersionsgrad (Tremosa, 2024)

²⁸ Nach Milgram et al. stellt die linke Seite des Kontinuums nur reale Objekte dar, welche von einem Beobachter – in diesem Sinne eine Person, aber auch etwa eine Kamera o.ä. – beobachtet werden, während die rechte Seite nur aus virtuellen Objekten aufgebaut wird (Milgram, et al., 1994).

²⁹ Etwaige Kritikpunkte sind zum einen der Punkt der kompletten Abtrennung des Virtual-Environments vom Real-Environment, was an sich aufgrund der räumlichen Gebundenheit von Lebewesen nicht möglich ist oder auch der Punkt, dass eine klare Abgrenzung von AR und VR in diesem Kontinuum nicht möglich ist. (Interaction Design Foundation - IxDF, 2022; Rauschnabel, kein Datum)

Vermischung der realen (physikalischen) Welt mit der virtuellen Welt in Form von Augmented-Reality (AR), Mixed-Reality (MR)³⁰ und Virtual-Reality (VR) handelt. In anderen Worten kann die Vermischung so beschrieben werden, dass neben dem uns umgebenden realen Raum bei einer XR-Applikation ein virtueller Raum erstellt wird, welcher dann mit dem realen Raum verknüpft wird. Die einzelnen Teilbereiche der Extended-Reality, also AR, MR und VR selbst unterscheiden sich dann wiederum je nach ihrem Anteil der Vermischung von realer (physikalischer) Information und virtueller Informationen (siehe Abb. 3), wobei VR in dieser Klassifikation die reale Welt nie zu 100% ausklammert. Die einzelnen Bereiche sind dabei durch den Grad der Vermischung zwischen realer (physikalischer) und virtueller Welt definiert, wobei an dieser Stelle ergänzend erwähnt werden muss, dass es im Gegensatz zur Abgrenzung zwischen AR und VR oftmals keine Einigkeit darüber gibt, wo genau sich MR im XR-Kontinuum wiederfindet³¹ und wie es sich von den anderen Typen abgrenzt. Problematisch in Hinblick auf MR ist zusätzlich, dass dieser Begriff, wie auch an Abbildung 2 und 3 zu sehen ist, zum einen als Überbegriff³² und zum anderen als zwischen AR und VR liegend³³ Anwendung findet, wobei dies so gelesen werden muss, dass es sich um unterschiedliche Dinge handelt, die ein und denselben Namen besitzen. Insgesamt muss Abbildung 2 also eher als konzeptionelle Skizze zur Unterteilung der Realitätssphären gelesen werden, während es sich bei Abbildung 3 um eine auf Abbildung 2 aufbauende und Kritik einarbeitende – beispielsweise im Hinblick auf die Unmöglichkeit Ausklammerung der realen Welt – und stark auf die Art der Darstellung fokussierte Einteilung handelt (Tremosa, 2024). Allgemein ist festzustellen, dass die Einteilung von Milgram et al. tendenziell eher in wissenschaftlichen Arbeiten und Einteilungen mit Ähnlichkeit zu jener von Tremosa eher in Arbeiten zur praktischen Umsetzung von Extended-Reality Erlebnissen verwendet werden.

Generell gesprochen muss an dieser Stelle auch noch ergänzend erwähnt werden, dass die verschiedenen Arten der Extended-Reality alle fünf Sinne – das heißt Tasten, Hören, Sehen, Riechen, Schmecken – ansprechen können, wobei in den meisten Fällen, wie auch den meisten Personen, die in irgendeiner Weise mit der Materie in Berührung gekommen sind, am ehesten der visuelle Aspekt geläufig ist und auch dieser Sinn hauptsächlich zur Übermittlung von Informationen bei diversesten Applikationen Verwendung findet (Rampolla & Kipper, 2012). Interessanterweise ist es in der Praxis darüber hinaus tatsächlich so, dass eigentlich nie von einer Extended-Reality Applikation gesprochen wird sofern nicht der Sehsinn in irgendeiner Form eingebunden wird, obwohl es sich tatsächlich

³⁰ Der Begriff Mixed-Reality kann an dieser Stelle irreführend sein. Deshalb sei auf die weitere Ausführung zu diesem Begriff verwiesen.

³¹ An dieser Stelle lohnt nochmals der Blick zurück an den Anfang dieses Unterkapitels, wo erwähnt wird, dass die Übergänge im Reality-Virtuality continuum fließend sind.

³² Dies kann etwa bei Milgram et al. gesehen werden, welcher den Begriff für die Gesamtheit aller Formen der Vermischung von realer und virtueller Umwelt anwendet.

³³ Dies kann etwa bei der Untergliederung von Tremosa und vielen weiteren landläufig im Internet aufzufindenden Untergliederung gesehen werden. Bezeichnet wird hierbei der Zwischenbereich zwischen Augmented- und Virtual-Reality, wobei diese beiden jedoch ausgeschlossen werden.

um eine solche handeln könnte. Ein Beispiel hierfür wäre etwa eine Übersetzungsapplikation, welche Live alle aufgefassten Sprachschnipsel in die eigene Muttersprache übersetzt und über einen Kopfhörer den/die Nutzer*in hören lässt. Man denke hier etwa daran, dass man sich durch eine spanischsprachige Stadt bewegt und alles zur Gänze normal wahrnimmt, bis auf den Punkt, dass die Sprache der einen selbst umgebenden Personen mit angewandter Applikation als die eigene Muttersprache (etwa Deutsch) über die Kopfhörer live ohne Verzögerung ausgegeben und somit verstanden wird, während die Personen tatsächlich Spanisch sprechen.

Abschließend für dieses Unterkapitel seien an dieser Stelle noch die sich aus den vorangegangenen Worten ergebenden Schlüsse – am Beispiel der später umzusetzenden praktischen Arbeit – kurz dargelegt.: Dadurch, dass im Rahmen der Arbeit geplant wird, Einblendung in Form einer auf der Straße/ dem Gehweg angezeigten Linie und an bestimmten Orten fixierten Schildern zu präsentieren, fällt die Applikation am ehesten in den AR Bereich. Gleiches würde gelten, wenn beispielsweise eine Einblendung am Briefkasten (etwa basierend auf der Erkennung des Namensschild) realisiert und somit zusätzliche Informationen zur Zustellung im Rahmen der eigenen Ausarbeitung präsentiert würden. Der Grund hierfür liegt darin, dass die Inhalte der realen Welt mit Inhalten aus der virtuellen Welt überlagert werden, oder in anderen Worten man sich gleichzeitig durch die reale und virtuelle Welt bewegt, wobei die virtuelle Welt nur einige die reale Welt ergänzende Informationen an fix bestimmten Orten enthält. Der VR-Bereich wäre dann ergänzend erwähnt jener, wenn der komplette Zustellvorgang im virtuellen Raum, also ohne jegliche Informationsvermittlung der physischen Welt – abseits der eigenen Existenz in bzw. das hindurchbewegen durch diese(r) –, stattfände. Handelt es sich jedoch nun um eine Einblendung, welche beispielsweise mittig auf einem HMD Display fixiert, basierend auf einer Ortsabfrage eingeblendet wird (beispielsweise beim Eintreten in ein Sperrgebiet), befindet man sich bereits in einem Zwischenraum, wobei diskutiert werden kann, ob es sich tatsächlich um AR beziehungsweise überhaupt eine Extended-Reality Realisierung handelt: Denn wenn sich solche eine Umsetzung überhaupt irgendwo im Extended-Reality Spektrum wiederfindet, geht diese noch stärker in Richtung Verringerung der virtuellen Inhalte (in Abb. 3 nach links) als AR.

2.5 Realtime-Engines (Game-Engines)

Hinsichtlich des, die Arbeit maßgeblich mitgestaltenden Bereichs der Realtime-Engines, auch bekannt als 3D- / Game-Engines, lässt sich bereits beim Blick auf den Namen erahnen, dass es sich hierbei um Softwarepakete handelt, die eng mit der Spieleindustrie verknüpft sind. Dies ist korrekt, wenn auch in den letzten Jahren nicht mehr in dem Ausmaß wie noch vor einigen Jahren, da die beiden größten unter ihnen, namentlich die Unity-Engine und Unreal-Engine, eine verstärkte Verbreitung in verschiedenen Bereichen, in der die Notwendigkeit der Realisierung von performanten dreidimensionalen (echtzeit) (XR-) Darstellungen besteht, erfahren haben. Unter anderem deshalb finden sich in rezenten Artikeln und Arbeiten mittlerweile vermehrt auch die Begriffe Realtime- oder 3D- und

weniger Game-Engines, wobei darum auch in dieser Arbeit der Begriff Realtime-Engine Anwendung finden soll³⁴.

Bevor nun in weitere Folge in aller Kürze Anwendungsfelder von RE besprochen werden, sollte kurz geklärt werden, worum es sich bei diesen eigentlich handelt. Dies gestaltet sich in ähnlichem Maße kompliziert, wie bei der Definition von GISc: An dieser Stelle hinkt der Vergleich jedoch etwas, da es sich bei den GISc um ein breites Wissenschaftsfeld und bei den RE um Tools und eher weniger ein wissenschaftliches (Forschungs-)Feld handelt. Beginnt man an dieser Stelle erstmal mit der Definition von Game-Engine, woraus sich die Realtime-Engines „herausentwickelt“ haben, so lautet dies von Unity vorgenommene Formulierung: „*Software that offers a suite of tools and features to game developers in order to build their games professionally and efficiently.*“ (Unity, kein Datum). Der Chipentwickler arm wiederum definiert Game-Engines als

„[...] a software development environment, also referred to as a ‘game architecture’ or ‘game framework,’ with settings and configurations that optimize and simplify the development of video games across a variety of programming languages. A gaming engine may include a 2D or 3D graphics rendering engine that’s compatible with different import formats, a physics engine that simulates real-world activities, artificial intelligence (AI) that automatically responds to the player’s actions, a sound engine that controls sound effects, an animation engine, and a host of other feature.“ (arm, kein Datum).

Betrachtet man diese beiden Definitionen etwas genauer und behält dabei im Hinterkopf, wie diese zustande kamen³⁵, erklärt sich auch, warum die Definitionen für Game-Engine sehr breit und wenig prägnant aufgestellt ist. Ähnlich stellt es auch Jason Gregory in seiner Ausführung „*What Is a Game Engine*“ in seinem Buch „*Game Engine Architecture*“ dar, wobei dieser nur zwei Punkte, durch welche sich Game-Engines definieren festhält: **Erweiterbarkeit** und **multiple Anwendbarkeit**³⁶ (Gregory, 2014).

Im Gegensatz dazu gibt es keinerlei explizite Definition zu Realtime-Engines. Behält man nun aber die groben Definitionen von Game-Engines im Hinterkopf und erweitert diese insofern, als dass auch die Nutzung abseits der Spieleindustrie Betrachtung findet, so könnte diese in eigenen Worten etwa wie folgt lauten:

Realtime-Engines sind Softwarepakete, welche dem Zwecke der Entwicklung verschiedener, voneinander unabhängiger Applikationen dienen und hierfür ein

³⁴ Nichtsdestotrotz kann es Teils sein, dass Game-Engines, Realtime-Engines und 3D-Engines synonym verwendet werden.

³⁵ In der Spielebranche war es lange Zeit üblich, für jedes Spiel eine eigene Engine, die sich auf gewisse Aspekte fokussierte (RPG, Shooter, ...), zu entwickeln. Unter anderem darum ist es bis heute so, dass eine Vielzahl an unterschiedlichen Game-Engines mit unterschiedlichen Fokussen existieren, weshalb es auch schwierig ist eine präzise allgemeingültige Definition aufzustellen (Gregory, 2014, pp. 11 - 13).

³⁶ Damit ist gemeint, dass eine Game-Engine die Basis für viele verschiedene Spiele bilden kann, ohne dass Veränderung an der Engine selbst vorgenommen werden müssen.

Grundgerüst an Funktionalitäten, welche **erweitert** werden können, aufweisen. Ziel einer Realtime-Engine ist es dabei, gewisse **Komponenten**, die in diversen Applikationen auftreten **wiederverwendbar** zu machen, um sie nicht für jede Applikation einzeln erstellen zu müssen und somit ein **professionelles und effizientes Arbeiten** zu ermöglichen.

Wird nun weiters die bereits vorhin angesprochene und für die Definition herangezogene Verwendung von Realtime-Engines abseits der Spieleentwicklung betrachtet, so stellt man gerade für den Bereich der GISc eine rezent sehr starke Fokussierung von esri hinsichtlich der Verknüpfung der beiden bekanntesten Engines (Anm. Unity und Unreal) mit ArcGIS Pro mithilfe deren Software Development Kits (SDK) fest. Das Ziel für die GISc einerseits ist es dabei, räumliche Sachverhalte dreidimensional besser und vor allem performanter (live) darstellen zu können und für die Spieleentwicklung andererseits echte räumliche Gegebenheiten, wie etwa Geländemodelle, Stadtmodelle und Ähnliches, verwendbar zu machen. (Furness & Hansen, 2022; Hansen, 2021; esri, 2023a; esri, 2023b)

Aber nicht nur die GISc, wie natürlich auch andere wissenschaftliche Bereiche (etwa die Archäologie), haben Interesse an Realtime-Engines gefunden, sondern auch große und wirtschaftlich bedeutende Bereiche, wie etwa die Automobilindustrie, der Einzelhandel, die Filmindustrie und nicht zu vergessen alle möglichen Bereiche zum Zweck von Schulungen, oder aber Serious-Games³⁷. Einen Auszug der Anwendungsmöglichkeiten, wobei diese unter dem Schlagwort „Digital Twin Applications“ firmieren, listet etwa Unity auf deren Webseite – vgl. (Unity, 2023) – auf. Alle diese Bereiche nutzen Realtime-Engines vor allem aufgrund folgender Punkte: Sie waren bis vor kurzem, a) abseits der kommerziellen Nutzung für Spiele kostenfrei, b) sparen, selbst wenn Lizenzen bezogen werden müssen, aufgrund der Tatsache, dass die Funktionen nicht selbst erstellt werden müssen Geld ein und c) bieten eine große Palette an Möglichkeiten zur Darstellung und Analyse von Sachverhalten in performanter dreidimensionaler Form, wobei diese je nach Belieben noch erweitert werden können.

2.6 Zustellbranche³⁸

Zu guter Letzt soll an dieser Stelle noch die Zustellbranche, auf welche im Rahmen der Arbeit dezidiert abgezielt wird, genauer betrachtet werden. Wie bereits erwähnt, distanziert sich die vorliegende Arbeit davon Anstellungsverhältnisse, Leistungserwartungen, Arbeitszeiten, Bezahlung und dergleichen zu thematisieren, da diese mitnichten das hierfür benötigte wissenschaftliche und wirtschaftliche Grundgerüst aufweist. Nun stellt sich aber logischerweise noch die Frage, welche Berufe/ Berufsgruppen denn überhaupt in den

³⁷ Serious Games sind Brett- / Computer/ ... -Spiele welche Lerninhalte auf unterhaltsame Weise vermitteln sollen. Siehe hierzu: (TOPSIM GmbH, 2024)

³⁸ Das Wissen für dieses Kapitel speist sich, falls nicht anders ausgewiesen, aus dem eigenen Wissen wie auch in kleinen Teilen durch den Bericht von Dang (Dang, 2021). Weiters erhebt dieses Kapitel keinen Anspruch auf hundertprozentige Aktualität und Korrektheit für alle im Zustellbereich tätigen Firmen und Tätigkeiten.

Bereich der Zustellbranche fallen, da keine Definition für letzteren festgestellt werden kann. Darum soll in weiterer Folge versucht werden selbst, eine zumindest für die vorliegende Arbeit gültige Definition mit Abgrenzbarkeit hin zu Transportlogistik, da es hier zumeist um den Transport von Gütern über weite Distanzen geht aufzustellen:

Die Zustellbranche definiert sich dadurch, dass von einem oder mehreren Orten eine oder mehrere Waren angenommen und zu einem oder mehreren Endkunden transportiert und dort abgeliefert werden, wobei die zurückgelegte Distanz relativ gering ist.

Hinsichtlich der mangelnden wissenschaftlichen Auseinandersetzung mit dem Thema und der daraus abzuleitenden schlechten Quellenlage muss an dieser Stelle in großem Maße auf eigene Erfahrungen und eigenes Wissen³⁹ zurückgegriffen werden, wobei bei der Besprechung der einzelnen Teilbereiche unterhalb etwaige bekannte Eigenheiten⁴⁰ verschiedener Betriebe angemerkt werden.

2.6.1 Tätigkeitsablauf

Begonnen sei an dieser Stelle mit der Besprechung eines normalen Arbeitstages im Bereich der Zustellung, wobei die folgend angeführten Inhalte dem eigenen Wissensstand nach für die meisten Betriebe im Bereich der Brief-, Zeitungs- und Paketzustellung zutrifft.

Beginnt man einen Job im Bereich der Zustelltätigkeiten (exkl. Lebensmittel- und Essenslieferungen), so ist es in den meisten Fällen so, dass die erste Zeit⁴¹ der Einarbeitung dient, wobei diese so von statten geht, dass man von einer Person, welche die Abläufe und Rayone⁴²/ Touren kennt, begleitet wird und man so die korrekte Handhabung mit den auszuliefernden Sendungen⁴³ erlernt. Anschließend an die Einschulung ist die Tätigkeit dann allein und selbstständig zu erledigen.

Der Alltag in der Brief-/ Paketzustellung sieht hierbei wie folgt aus: Der Arbeitsbeginn startet mit der Sortierung der für den zugeteilten Rayon vorhandenen Sendungen in der

³⁹ Im Rahmen der Sammlung von Berufserfahrung war ich zwei Mal für je einen Monat als sogenannte „Ferialkraft“ in der Zustellung von Brief- und Paketsendungen bei der österreichischen Post AG tätig. Das erste Mal 2018 in Pregarten (Oberösterreich) in einer ländlichen Region und das zweite Mal 2021 in Hernals (Wien) in städtischem Umfeld. Ebenso war ich im Rahmen meines Zivildienstes (2017) in einem Logistikunternehmen mit angeschlossener Post AG Servicestelle und auch DPD-Abholstation. Hierbei hatte ich neben der Post AG und DPD zusätzlich mit GLS zu tun, wodurch auch ein gewisses Wissen rundum die Anforderungen, Bedingungen und Arbeitsabläufen abseits der Post AG, vor allem bei DPD und GLS, herrscht.

⁴⁰ Hier soll dezidiert darauf hingewiesen werden, dass es weitere nicht aufgeführte Abweichungen geben kann!

⁴¹ Bei einer Ferialkraft bei der Post AG beträgt die Einschulungszeit in der Regel eine Woche.

⁴² Ein Rayon ist der Name für die Gebietseinheit/ den Zustellbereich einer einzelnen Zustellungskraft und weist zumindest bei der Post AG eine sogenannte Gangordnung auf (Details siehe Fußnote 45).

⁴³ Zur korrekten Handhabung der Sendungen siehe Kapitel 2.6.4 „Zustellbedingungen“.

Zustellbasis und dem Einscannen selbiger, sofern nötig. Nachdem dieser Schritt abgeschlossen ist, werden die Sendungen⁴⁴ zur einfacheren Handhabung bei der Auslieferung in der Reihenfolge der Gangordnung⁴⁵ – sofern eine solche existiert – in das Fahrzeug/ den Zustellwagen geladen. Nachdem das Fahrzeug/ der Zustellwagen beladen ist, beginnt der eigentliche Zustellgang⁴⁶. Hier geht man in der sogenannten Gangordnung vor – falls vorhanden, andernfalls muss man die Adressen falls unbekannt bspw. mittels Navigationsgerät suchen. Die Sendungen werden dann immer mit Hinblick auf besondere Anforderungen⁴⁷ ausgeliefert. Eine weitere Gegebenheit, die bei vielen Zustellern hinzukommt, ist jene, dass entlang der eigenen Route gegebenenfalls auch Sendungen wieder mitgenommen werden müssen⁴⁸. Wurde die letzte Adresse beliefert, so kehrt man wieder in die Zustellbasis zurück, liefert davor noch etwaige nicht zugestellte Sendungen an Adressen zur Abholung⁴⁹ ab (oder legt sie für einen zweiten Zustellversuch am nächsten Tag auf die Seite), räumt seinen Arbeitsplatz (sowohl Auto/ Zustellwagen, wie auch in der Zustellbasis) auf und beendet den Dienst.

2.6.1.1 Besonderheiten im Alltag

Eine an dieser Stelle noch zu erwähnende Besonderheit im Alltag der Zustellung, die dem eigenen Wissensstand zufolge sowohl bei der Post AG, wie auch bei DPD und GLS auftritt, ist das sogenannte „Mitbesorgen“. Hiermit ist gemeint, dass, wenn eine Person aufgrund von Krankheit oder ähnlichem nicht zum Dienst erscheint und kein Ersatz verfügbar ist (oder Unterbesetzung herrscht), das Gebiet der abwesenden Person unter den anderen Zusteller*innen aufgeteilt und beliefert werden muss. Dies hat zum einen beträchtlichen

⁴⁴ Bei der Briefzustellung kommen dann noch Zeitungen und Werbe-, wie auch Postwurfsendungen hinzu.

⁴⁵ Die Gangordnung gibt an, in welcher Reihenfolge Sendungen an gewissen Adressen zugestellt werden müssen. Bei der Briefzustellung gibt es hierbei bei der Post AG eine Zweiteilung, da zur Verringerung der Arbeitslast immer im Takte abwechselnd in nur einem Teil des Rayons Werbe- und Postwurfsendungen zugestellt werden (Briefe und Pakete müssen immer zugestellt werden). Dies hat den Effekt, dass Adressen, welche ohnehin an einem gewissen Tag keine Werbe- und Postwurfsendungen erhalten, gegebenenfalls, falls keine Briefsendung vorliegt, nicht beliefert werden müssen. Bei anderen Betrieben im Bereich der Zustellung, etwa Paketdienstleister, ist mir selbst nichts hinsichtlich einer Abwechslung der Gangordnung in bedientes/ unbedientes Gebiet bekannt, ich gehe aber davon aus, dass eine solche nicht existiert, da ohnehin immer wechselnde Zustelladressen existieren. Eine Gangordnung existiert jedoch auch bei Paketzustellern soweit mir bekannt.

⁴⁶ Je nach Rayon und dessen Größe wird dieser bei der Post AG mit Auto, oder mit einem Zustellwagen per Fuß erledigt. Bei anderen Betrieben kommen auch Fahrräder oder Lieferwägen zum Einsatz.

⁴⁷ Siehe Kapitel 2.6.4 „Zustellbedingungen“

⁴⁸ Bei der Post sind das beispielsweise Briefe aus Briefkästen oder Sendungen von Endkund*innen, die bereits vorfrankiert sind. Bei anderen Zustelldienstleistern, wie etwa DPD und GLS, sind dies oftmals Paketshops, an denen nicht abgelieferte Pakete abgegeben aber auch versendete Pakete entgegengenommen werden. Hinzu kommt bei DPD und GLS, dass diese bei großen Firmenkunden Pakete abholen müssen.

⁴⁹ Bei der Post AG sind dies entweder Post-Abholstationen oder direkt Post-Filialen. Bei DPD und UPS sind dies oftmals kleinere Betriebe, die den Abholdienst neben ihrer Haupttätigkeit erledigen.

Mehraufwand zur Folge und bedeutet darüber hinaus, dass man gegebenenfalls in einem Gebiet zustellen muss, dass man nicht kennt.

Als eine spezielle Besonderheit der Post AG sei an dieser Stelle zusätzlich erwähnt, dass es in ländlichen Gebieten üblich ist, dass Zusteller*innen sowohl die Brief- als auch die Paketsendungen zustellen, während diese beiden Sendungsarten in städtischen Gebieten getrennt voneinander (von unterschiedlichen Personen) zugestellt werden⁵⁰. Dies bedeutet in weiterer Folge aber nicht, dass man als Zusteller*in in der Stadt nicht auch Päckchen (sogenannte Kleinpakete) zustellt – diese machen jedoch nur einen kleinen Anteil aus.

2.6.2 Tour/ Rayon

Hinsichtlich der Tour/ des Rayons sei gesagt, dass es bei der Post AG, soweit bekannt, sowohl bei der Brief- als auch Paketzustellung so ist, dass grundsätzlich fixe Touren existieren, es jedoch nicht unbedingt der Fall ist, dass eine Person immer auf denselben Rayon mit gleichbleibender Tour arbeitet⁵¹. Personen, die regelmäßig wechselnde Rayone bedienen, werden dabei als sogenannten „Springer*innen“ betitelt, welche je nach Notwendigkeit (beispielsweise bei Krankheit oder Urlaub) unterschiedliche Rayone, gegebenenfalls sogar von unterschiedlichen Zustellbasen aus abdecken.

Ähnlich sieht es bei den Zustelldienstleistern DPD und GLS aus, hier herrschen soweit bekannt auch gleichbleibende Zustellgebiete, wobei nichts über mögliche „Springer*innen“ gesagt werden kann.

Weiters gibt es dann noch den Bereich von Lebensmittel- und Essenslieferungen, wobei die Lieferanten ihren Zielort erst erfahren, sobald eine Bestellung eingegangen ist. Dies bedeutet im Umkehrschluss, dass hier im Vergleich zu den fixen Touren bei der Post AG immer andere Routen bedient werden müssen.

2.6.3 Regionsabhängigkeit

Hinsichtlich der Regionsabhängigkeit gilt es zu erwähnen, dass es nicht nur zwischen den verschiedenen Rayonen einen beträchtlichen Unterschied gibt, sondern auch zwischen der Arbeit im ländlichen und städtischen Raum.

Zu ersterem sei die Struktur des jeweiligen Rayons erwähnt: Je nachdem, ob es sich eher um einen Rayon mit vielen Firmen, Wohnhäusern, oder relativ gut durchmischtes Gebiet handelt, haben verschiedene Zusteller*innen sehr unterschiedliche Ausgangspositionen.

⁵⁰ Dies trifft an dieser Stelle exklusiv für die Post AG zu, da andere in der Zustellbranche tätige Firmen sich auf eine Art der zuzustellenden Produkte spezialisiert (Essenslieferungen, Paketlieferungen, ...) haben.

⁵¹ Ein Rayon umfasst bei der Post AG bei der Briefzustellung und auch bei der gemischten Zustellung der eigenen Erfahrung nach, je nach Rayonaufbau, rund 200-300 Zustelladressen. Eine Zustelladresse kann hierbei eine Partei in einem Einfamilienhaus oder aber auch die jeweilige Partei in einem Wohnblock sein.

Denn allein dieser Faktor unterscheidet, insbesondere bei der Paketzustellung, ob Sendungen abgeliefert werden können, da Personen, wie es bei Firmen eigentlich immer der Fall ist, angetroffen werden und diese annehmen können und die Lieferungen somit nicht weiter mitgenommen und zu einem Abholort⁵² oder ähnlichem gebracht werden müssen. Hingegen ist es dann aber auch wieder so, dass bei einem Rayon mit vielen Firmen die Wahrscheinlichkeit steigt, Pakete wiederum annehmen⁵³ zu müssen. Bei Rayonen mit einem großen Anteil an Wohnhäusern wiederum stellt sich dann weiters die Frage, ob es sich um Mehrparteienhäuser oder Ein-/ Zweifamilienhäuser handelt. Dies macht insofern einen Unterschied, als dass, wenn viele Mehrparteienhäuser im Rayon liegen, man weniger Stopps hat, während die Menge an Zustelladressen⁵⁴ aber insgesamt höher ist: deswegen, weil es oftmals sein kann, dass mehr als 20 Parteien in einem solchen Haus leben. Im Gegensatz dazu stehen eine Vielzahl an Ein-/ Zweifamilienhäuser im Rayon, welche die Anzahl an Stopps und zumeist auch zurückgelegte Distanz erhöhen, jedoch insgesamt weniger Zustelladressen resultieren.

An dieser Stelle schließt auch direkt die Frage an, ob es sich um ein eher ländliches oder ein eher städtisches Zustellungsgebiet handelt, da dies stark die Art der Gebäude als auch Siedlungsstruktur und -dichte bestimmt. So zeigt sich in ländlichen Regionen logischerweise die Tendenz zu einer dünneren Besiedlung mit einem geringen Anteil an Mehrparteienhäusern und somit mehr Stopps und weniger Zustelladressen, jedoch müssen weitere Distanzen zwischen den einzelnen Stopps zurückgelegt werden. Gleiches in umgekehrter Form gilt an dieser Stelle für städtische Gebiete. Dies bedingt dabei auch die Fortbewegungsformen zwischen den einzelnen Zustelladressen, was insbesondere bei der Briefzustellung der Post AG sehr gut zu sehen ist. So werden in Gebieten, wo eine größere Distanz zwischen den einzelnen Zustelladressen vorliegt, PKWs eingesetzt, wodurch weniger Distanz fußläufig zurückgelegt werden muss, während in der Stadt auf Zustellwägen die per Hand oder elektrisch geschoben werden gesetzt wird, wodurch die gesamte Distanz zu Fuß zurückgelegt werden muss. Bei reinen Paketzusteller*innen ist es zwar so, dass diese unabhängig von der Distanz zwischen den Zustelladressen mit Lieferwägen unterwegs sind, jedoch ist es bei diesen – aufgrund von Parkplatzsituation und Zeitmanagement – auch so, dass diese in städtischen Gebieten mehr fußläufig unterwegs sind als in ländlichen⁵⁵.

⁵² ...insofern keine Abstellgenehmigung vorliegt (siehe hierzu Kapitel 2.6.4)

⁵³ Damit ist gemeint, dass ein/eine Kund*in der ausliefernden Person eine Sendung übergibt, die man in weitere Folge in der Zustellbasis zur weiteren Bearbeitung abgibt.

⁵⁴ Eine Zustelladresse ist die kleinste Einheit bei der Zustellung. So können etwa in einem Wohnblock mit 20 Parteien, sofern alle von unterschiedlichen Personen bewohnt werden, auch 20 Zustelladressen aufweisen. Im Umkehrschluss wiederum weist ein Einfamilienhaus nur eine Zustelladresse auf.

⁵⁵ Um das alles in ein Verhältnis zu setzen: In ländlichen Gebieten beläuft sich die zurückgelegte Strecke mit einem Auto aus eigener Erfahrung auf rund 100km, während in städtischen Gebieten die zurückgelegte Distanz, ebenso mit dem Auto, bei rund 30km liegt. Leider kann an dieser Stelle keine definitive Angabe zur fußläufig zurückgelegten Distanz gegeben werden.

Hinsichtlich Essens- und Lebensmittelzustellung kann zur Regionsabhängigkeit in Hinblick auf die zurückgelegte Distanz im Land-Stadt Vergleich ein ähnliches Fazit gezogen werden, im Bereich der Fortbewegungsformen bei der Zustellung muss jedoch mehr differenziert werden: Denn in diesen beiden Bereichen ist es so, dass gerade in städtischen Regionen verstärkt auf die Fortbewegung mit dem Fahrrad, beziehungsweise E-Rollern und in ländlichen Regionen zusätzlich auf Autos auf Mopeds gesetzt wird.

2.6.4 Zustellbedingungen

Nachdem oberhalb bereits kurz Besonderheiten und die korrekte Handhabung bei der Zustellung (mit Ausnahme jener für Essens- und Lebensmittelzustellung) angesprochen wurden, seien hier diese, vor allem für die Post AG, erläutert⁵⁶.

Der erste Punkt hierbei ist, dass alle Sendungen immer bei einer Person selbst, einer Person die in einem Kontakt zum/zur Empfänger*in⁵⁷ steht, oder an einem sicheren Ort⁵⁸ abgegeben werden müssen. Darüber hinaus besteht die Möglichkeit eine sogenannte „Abstellgenehmigung“ zu beantragen, mit welcher dem/ der Zusteller*in erlaubt wird die Sendung an einem bestimmten Ort, abseits der zuvor beschriebenen Auflagen abzulegen. Darüber hinaus besteht bei einer Vielzahl an Sendungen die rechtliche Verpflichtung seitens des/ der Zusteller*in eine Unterschrift des/der Empfänger*in, beziehungsweise der Person, welche die Sendung annimmt, für die Übergabe einzufordern: Solche Sendungen dürfen dann nicht in den Briefkasten oder eine Sendungsempfangsbox gegeben, jedoch bei einer Abstellgenehmigung an eben jenem Ort abgestellt werden⁵⁹. Weiters gibt es dann noch Pakete, die einer besonderen Vorsicht bedürfen (bspw. Glasprodukte, Batterien/ Akkus, ...), wobei diese keine allzu große Auswirkung im Alltag haben. Einen Einfluss hingegen haben Sendungen, die eine persönliche Übergabe verlangen, wo wirklich nur der/ die Empfänger*in die Sendung entgegennehmen darf, und Express-Sendungen⁶⁰, die zu einer bestimmten Zeit beim / bei der Kund*in zugestellt sein müssen und somit die gesamte Gangordnung abändern können.

⁵⁶ Hinsichtlich der Besonderheiten kann an dieser Stelle vornehmlich auf jene der Post AG eingegangen werden, da über etwaige Besonderheiten bei anderen Zustelldienstleistern nur geringes Wissen vorherrscht.

⁵⁷ Dies kann noch weiter differenziert werden, wie unterhalb bei der Post AG zu sehen sein wird. Zumeist reicht jedoch normaler Kontakt zum / zur Empfänger*in, also beispielsweise Nachbarschaft oder ähnliches aus.

⁵⁸ Als sicherer Ort gelten der Briefkasten oder Sendungsempfangsboxen.

⁵⁹ Diese Regelung gilt für die Post AG und kann sich bei anderen Zustelldienstleistern unterscheiden.

⁶⁰ Der Name für persönliche Übergabe und Express Sendungen unterscheidet sich je nach Zustelldienstleister und ist aus diesem Grund hier allgemein gehalten.

Weiters gibt es dezidiert bei der Post AG⁶¹ noch folgende Sonderregelungen:

- RSA-Briefe dürfen nur an den / die Adressat*in übergeben werden beziehungsweise an eine bevollmächtigte Person.
- RSB-Briefe dürfen nur an den / die Adressat*in, Bevollmächtigte und erwachsene Personen, die im gleichen Haushalt leben übergeben werden.
- Eingeschrieben Sendungen und Pakete dürfen, wie oberhalb beschrieben, an jede vertrauenswürdige erwachsene mit der Person im Kontakt stehende Person und wie oberhalb beschrieben bei einer Abstellgenehmigung abgeliefert werden.

Die Unterschrift für die meisten Sendungen wird dabei digital am PDA, nachdem die jeweilige Sendung eingescannt und Ausweisdaten eingegeben wurden, ausgefertigt, wobei es auch bei ausländischen, oder älteren Einschreibern noch der Fall sein kann, dass tatsächlich analog auf Papier die Ausweisdaten erfasst und unterschrieben werden muss – auf einer Vorlage, welche dem Absender zurückgesandt wird – und das Scannen der Sendung mittels PDA somit wegfällt.

2.7 Verknüpfung der Bereiche

Nachdem nun die einzelnen (Haupt-)Bereiche, aus denen sich die Arbeit speist, kurz dargelegt wurden, folgt nun, wie bereits eingangs erwähnt, eine Betrachtung der Überschneidungen der einzelnen Bereiche. Wie gezeigt wurde und im Rahmen dieser Arbeit ohnehin bekannt sein sollte handelt es sich bei den GISc und demnach auch der Geographie in weiten Teilen um sehr breit definierte wissenschaftliche, stark interdisziplinär ausgelegt Disziplinen, die von einer Vielzahl weiterer anderer (wissenschaftlicher) Disziplinen in starkem Ausmaß beeinflusst werden und nur sehr schwer abgrenzbar sind. Hierdurch ergibt sich auch ein großer Spielraum, wenn es darum geht, die Geographie, oder in diesem Fall die Geoinformationswissenschaften, mit anderen (wissenschaftlichen) Bereichen zu verknüpfen und Wissen in diese einfließen zu lassen. Genau diese beiden Punkte, die Verknüpfung der GISc mit und das Einfließen-Lassen von Wissen in und aus anderen Bereichen, wird im Rahmen dieser Arbeit essenziell sein. So wird etwa Wissen hinsichtlich der Ortung von Personen, wie auch zur Routenfindung oder ganz generell der „modernen“ Navigation und der Verortung von Sachverhalten, das zumindest teils direkt aus dem Bereich der GISc stammt, mit Kenntnissen über die Erstellung von performanten 3D-Applikationen aus dem Bereich der Realtime-Engines⁶² und Wissen über die Möglichkeiten, die XR-Applikationen (erstellt in RE) zur Präsentation von Informationen bieten, verknüpft werden, um zum einen die Potentiale einer solchen Verknüpfung herauszuarbeiten und zum anderen eine eigene AR-Navigationsapplikation auf diese Weise zu erstellen.

⁶¹ Bei diversen Zustelldienstleistern kann es solche auch geben, jedoch sind keine Informationen hierzu auffindbar.

⁶² Im Rahmen dieser Arbeit wird Realtime-Engines sowohl als Bezeichnung für die Software, als auch den Forschungs-/Anwendungsbereich verwendet, da die Verwendung von Game-Branche nicht (mehr) als korrekt anzusehen ist. Hierbei gestaltet es sich ähnlich zur Thematik der Verwendung des Begriffs Game-Engine zu Realtime-Engine.

3 Rundum AR-Systeme⁶³

Nachdem bereits eingangs kurz auf den Begriff Extended-Reality, wie auch Mixed-Reality, das Reality-Virtuality-Continuum und zu guter Letzt das Extended-Reality-Continuum eingegangen wurde, soll sich das nun folgende Kapitel aufgrund der Fokussierung der Arbeit – und der Unbekannt- wie auch Neuheit für die GISc – in gebührendem Ausmaß mit dem Bereich der Augmented-Reality beschäftigen. Hierzu sei auch direkt zu Beginn auf die Arbeit von Rampolla und Kipper mit dem Titel „*Augmented Reality: An Emerging Technologies Guide to AR*“ (Rampolla & Kipper, 2012) wie auch das Buch „*Augmented Reality: Theorie und Praxis*“ von Mehler-Bicher und Steiger (Mehler-Bicher & Steiger, 2014) verwiesen, welche einen guten Einstieg in den Themenbereich AR bieten.

Von den im Anschluss folgenden Unterkapiteln soll das erste (Kapitel 3.1) dazu dienen, Begrifflichkeiten und Konzepte, die mit AR und (AR-)Applikationen in Verbindung stehen, zu klären, während sich das zweite Kapitel (Kapitel 3.2) dann auf die Definition, die Geschichte und die Abgrenzung von AR-Systemen fokussiert: Es geht also darum zu erörtern, was AR-Systeme sind, wie diese in einem ersten Schritt unterschieden werden können, kurz die Entwicklungsgeschichte anzusprechen und abschließend die Abgrenzung von AR, etwa zu VR zu thematisieren. Die letzten drei Unterkapitel wiederum werden sich in ihrer Fokussierung hauptsächlich mit den Prinzipien hinter AR-Applikationen und deren Einteilung (Kapitel 3.3), den unterschiedlichen Arten der Kommunikation (Präsentation) von AR-Inhalten⁶⁴ (Kapitel 3.4) und zu guter Letzt der Realisierung/ Entwicklung einer AR-Applikation aus Sicht eines/einer Entwickler*in (Kapitel 3.5) widmen.

3.1 Begrifflichkeiten & Konzepte

Hardware: Unter Hardware werden alle physischen Komponenten eines Computers/ Laptops/ Tablets/ Smartphones⁶⁵ verstanden, wobei diese weiters in Hardware-Komponenten der **Zentraleinheit**, also Komponenten, die das Gerät funktionsfähig machen und die **Peripherie(-geräte)**, also Geräte, die an die Zentraleinheit angeschlossen werden,

⁶³ Ein großer Teil des folgenden Kapitels ist in dieser Form für alle im Bereich der XR liegenden Systeme gültig. Aufgrund der Fokussierung der Arbeit wird teils aber dezidiert auf AR eingegangen und somit werden gewisse Eigenheiten anderer XR-Formen außen vorgelassen.

⁶⁴ Aufgrund der raschen Entwicklungen im Bereich der AR-Applikationen und der Breite des Feldes wird davon Abstand genommen, eine hundertprozentig vollständige Auflistung aller Möglichkeiten zur Präsentation von AR-Inhalten als auch von der zur Verfügung stehenden Entwicklungsoptionen zu machen. Viel eher soll sich auf die (momentan) wichtigsten Bereiche konzentriert werden.

⁶⁵ Im Englischen wird für diese Auflistung zumeist der Begriff „computing device“ verwendet, welcher alle Arten von Geräten miteinschließen, die irgendwelche Berechnungen durchführen. Um diese breite Aufstellung im Rahmen der Arbeit im Deutschen auch abbilden zu können, wird in weitere Folge der Begriff „Berechnungseinheit“ Anwendung finden.

unterteilt werden können⁶⁶. Insgesamt bildet die Hardware, erst mitsamt dem Betriebssystem, von welchem diese die Anweisungen zur Ausführung gewisser Befehle und Funktionen erhalten, ein funktionsfähiges System. (Awati & Rosencrance, 2021; Münch, 2022)

Peripheriegeräte: Bei Peripheriegeräten handelt es sich, wie oberhalb angemerkt, um einen Teil der Hardware, welcher aber selbst wiederum in interne/ externe Peripherie und auch nach Art, also Eingabe- Ausgabe-, und Speichergerät, unterteilt wird⁶⁷. Zur ersteren Einteilung sei an dieser Stelle erwähnt, dass es hierbei um die Lage, also ob sich das Peripheriegerät innerhalb des Gehäuses mit den Komponenten der Zentraleinheit befindet, geht, während die Klassifizierung nach Art selbsterklärend sein dürfte. (Rouse, 2023)

Software: Als Software werden alle Programme verstanden, welche auf einem Computer ausgeführt werden und diesen gemeinsam mit der Hardware zu einem funktionsfähigen System machen. Es wird dabei zwischen **Anwendungsprogrammen**, also Programmen, welche einen bestimmten Zweck erfüllen (bspw. CAD-Programme, 3D-Zeichenprogramme, IDE⁶⁸), und **Systemprogrammen** unterschieden. Zu letzteren zählt auch das Betriebssystem, welche die „[...]computerinternen Abwicklung von Anwendungsprogrammen und der Datenverwaltung zu lösen hat[...]“ (Lackes & Siepermann, 2018b). (Lackes & Siepermann, 2018a; Lackes & Siepermann, 2018b; Lackes & Siepermann, 2018c; Lackes & Siepermann, 2018d)

Weiters sollte an dieser Stelle noch der Begriff einer **Applikation/ Anwendung** ins Feld geführt werden, da dieser oftmals Synonym zu Software verwendet wird. Generell handelt es sich hierbei aber lediglich um eine andere Bezeichnung für Anwendungsprogramme. (Gillis, 2020)

Betriebssystem: *„Sammelbegriff für Programme (Systemprogramme), die den Betrieb eines Computers erst möglich machen; auch als Operating System (OS) bezeichnet. Sie steuern und überwachen das Zusammenspiel der Hardwarekomponenten im Rahmen der Auftrags-, Daten-, Arbeitsspeicher- und Programmverwaltung (bes. die Abwicklung einzelner Anwendungsprogramme, den Zugriff von Prozessen auf bestimmte*

⁶⁶ Zu dieser Unterteilung muss hinzugefügt werden, dass es sich hierbei nicht um eine klare Abgrenzung handelt, da etwa eine Kamera in einem Smartphone/ Laptop nicht klar als Peripheriegerät bei diesem zu definieren ist, da es eine Kernfunktion eines Smartphones/ Laptops ermöglicht, wohingegen es bei einem regulären Computer definitiv als solches zu definieren ist.

⁶⁷ Natürlich kann es auch vorkommen, dass Peripheriegeräte gleichzeitig in zwei Bereiche fallen, wie es etwa bei Touch-Displays mit Eingabe- und Ausgabefunktion der Fall ist. Hier könnte man dann zwar den weiteren Schritt des Herunterbrechens eines Touchdisplays in seine einzelnen Komponenten vollziehen – dies erscheint jedoch an dieser Stelle weder sinnig noch zielführend.

⁶⁸ Unter einer/ einem IDE (Integrated Development Enviroment) versteht man, einfach ausgedrückt, eine Software, deren Oberfläche für Entwickler*innen wichtige Tools zur Erstellung neuer Software bereitstellt. (Luber, 2017)

Ressourcen) sowie der Systemsicherung (Fehlererkennung und -behebung). Das Betriebssystem macht ein Datenverarbeitungssystem erst bedienbar und beherrschbar.“ (Lackes & Siepermann, 2018d)

Plattform (IT): In der IT bezeichnet eine Plattform jegliche Art von Hard- oder Software, welche dem Zweck der Bereitstellung eines bestimmten Services oder Applikation dient. Weiters kann dabei eine Software (sowohl System- als auch Anwendungsprogramm) sowohl eine Applikation sein, welche einer Plattform benötigt, als auch selbst in einer anderen Fragestellung wiederum als Plattform dienen⁶⁹. Insgesamt ist die Frage nach einer Plattform also dann von Interesse, wenn es darum geht, ob eine gewisse Software eine gewisse Plattform für eine korrekte Ausführung benötigt. Um das Ganze an dieser Stelle etwas verständlicher zu gestalten: Die Software Unity bedient sich der Plattform Windows, während Unity selbst für SDKs die Plattform darstellt, wie es etwa bei der SDKs von *esri* der Fall ist. (Bigelow & Rouse, 2023)

SDKs: Bei sogenannten Software-Development Kits handelt es sich um Softwarepakete, die zumeist zur Einbindung in andere Programme (etwa Realtime-Engines) genutzt werden, wie im Fall der Spiele- oder Appentwicklung: Diese werden von Herstellern von Hard- und oder Software für Entwickler*innen oftmals kostenfrei zur Verfügung gestellt, um Funktionalitäten der Hard- oder Software des Bereitstellers des SDKs leichter nutzbar zu machen. (Yasar & Rosencrance, 2022)

Tracking: Unter Tracking wird das Erkennen und Nachverfolgen eines Objekts verstanden. Oftmals wird beispielsweise Hand- oder Augentracking eingesetzt, wobei ersteres zur Erkennung von Gesten, mithilfe derer Eingaben gemacht werden können und zweiteres, damit Inhalte dem Blickwinkel entsprechend angepasst werden zum Einsatz kommt.

Rendering: Rendering bezeichnet die Technik, welche hinter der Erstellung (Berechnung der einzelnen Bilder) des angezeigten (dynamischen/ statischen) Inhalts steht: Bei XR bezieht sich Rendering also auf die Berechnung der die reale Umwelt überlagernden Inhalte der virtuellen Umwelt.

Degrees of Freedom (DoF): Unter Degrees of Freedom⁷⁰ versteht man die Anzahl an Richtungen, in die sich ein Objekt in einem 3D-Raum bewegen kann. Die ersten drei Richtungen bilden dabei herkömmlicherweise die Rotationen in X-, Y- und Z- Richtung – oftmals auch als Yaw, Pitch und Roll (siehe Abb. 52) bezeichnet –, wobei diese Form dann als **3-DoF** bezeichnet wird. Den zweiten Teil bilden dann die Bewegungen entlang der X-, Y- und Z-

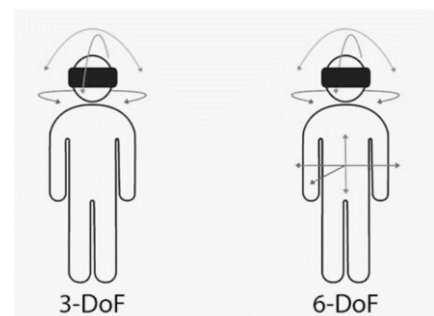


Abbildung 4: Degrees of Freedom nach (Barnard, 2023)

⁶⁹ Ein beliebtes Beispiel hierzu ist etwa SQL. Siehe hierzu: (Bigelow & Rouse, 2023)

⁷⁰ Am ehesten lässt sich Degrees of Freedom im deutschen als Anzahl der Bewegungsrichtungen bezeichnen.

Achse, wobei diese mitsamt der Rotationen entlang der Achsen dann den sogenannten **6-DoF** Typ definieren (Barnard, 2023).

An dieser Stelle lohnt es sich dann nochmals, die im Abschnitt zur Ortung dargelegten sechs von GPS und IMU ermittelbaren Werten in Erinnerung zu rufen und das alles in weiterer Folge auf XR-Applikationen und die Bewegung durch den virtuellen Raum umzulegen. Dies bedeutet also, dass eine XR-Applikation grundsätzlich 6-DoF, 3-DoF oder 0-DoF aufweist, wobei dies Folgendes bedeutet: Bei 6-DoF kann man sich komplett mit Bewegungen im realen Raum durch den virtuellen Raum bewegen. Bei 3-DoF hingegen kann man mittels Drehung und Bewegung des Kopfes die Blickrichtung im virtuellen Raum ändern, sich aber nicht durch Bewegung im realen Raum bewegen, sondern maximal durch Controllereingaben. Die letzte Form der 0-DoF hingegen lässt keine Bewegungen im virtuellen Raum durch Bewegung im realen Raum zu: Eine Bewegung durch Controllereingaben, wie etwa bei normalen Computermonitoren, ist aber trotzdem möglich.

Image⁷¹-/ Pattern-Recognition (dt. Bild- / Mustererkennung): Von Image-/ Pattern-Recognition wird gesprochen, wenn eine Maschine etwa mittels Kamera und Software (oftmals in Kombination mit AI) verschiedene Dinge, wie etwa Menschen, Orte, Fahrzeuge oder ähnliches (bei Image-Recognition) oder etwa gewisse ähnliche Muster (oder Abweichungen davon) in unterschiedlichen Bildern (bei Pattern-Recognition) erkennen kann. (Basseville-Lipka, 2020; Vogel, 1999; Yasar, 2023)

Virtueller-/ Realer-/ XR-/ AR-Raum: Wenn in weiterer Folge vom XR-/AR-Raum gesprochen wird, ist damit die Vermischung des realen und virtuellen Raums gemeint. Der reale Raum bezeichnet dabei unsere tatsächlich physisch vorhandene Umwelt mit allen in dieser enthaltenen Objekten. Der virtuelle Raum betitelt weiters den dreidimensionalen Raum, welcher von einer XR-/AR-Applikation erstellt wird, mit dem realen Raum verknüpft sein kann und die mithilfe der App anzuzeigenden Inhalte enthält.

3.2 Definition, Geschichte & Abgrenzung von AR-Systemen⁷²

Beginnen wir dieses Kapitel mit der Frage, worum es sich bei AR-Systemen eigentlich handelt, so muss zuerst einmal klargestellt werden, dass es sich bei einem System immer um etwas handelt, was aus mehreren

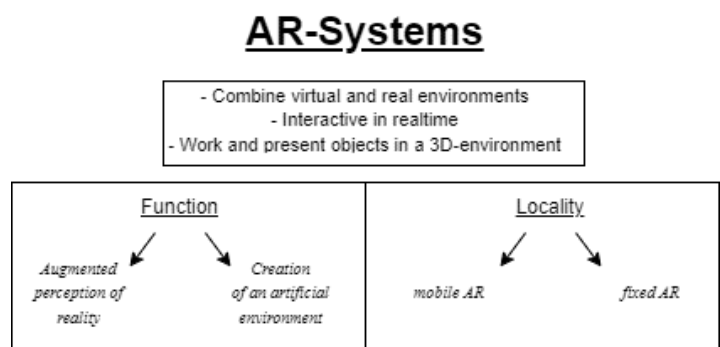


Abbildung 5: Einteilung von AR-Systemen (eigene Darstellung)

⁷¹ Auch: picture recognition oder photo recognition

⁷² Die Verwendung des Begriffs AR-System soll darauf hinweisen, dass es sich hierbei um alle für die Realisierung einer AR-Erfahrung benötigten Elemente – d.h. Soft- und Hardwarekomponenten – handelt und sich somit von AR-Programm / AR-Applikation abgrenzen, welche rein auf die Softwareseite hinweisen.

Komponenten aufgebaut ist: Dies gilt auch für AR-Systeme. Diese bestehen, wie auch jegliche anderen XR-Systeme, grundsätzlich aus zwei Teilen, dem software- und hardwareseitigen Teil⁷³, wobei ein solches Systems immer auf die Vermischung von virtueller und realer Umwelt abzielt. Aufbauend darauf kann eine AR-Applikation dann laut Azuma, noch mittels dreier Charakteristiken definiert werden: Erstens kombiniert **AR**, wie bereits erwähnt, a) **virtuelle und reale Informationen**, zweitens ist es b) **interaktiv in Echtzeit** und drittens c) registriert – soll heißen verknüpft, arbeitet und wird ausgeführt – in einer **3D-Umgebung** (Azuma, 1997). Aufbauend darauf definieren Rampolla und Kipper zwei nicht unbedingt immer klar abzugrenzende Funktionsweisen von AR: „*Augmented perception of reality*“ und der „*Creation of an artificial environment*“ (Rampolla & Kipper, 2012). Bei ersterer geht es hauptsächlich um die Darstellung von Zusatzinformationen der Umwelt, welche vom Menschen an sich so nicht erfasst werden können aber trotzdem existieren – gemeint ist hiermit etwa die Höhe eines Gebäudes, die Entfernung zu einem bestimmten Objekt, die Visualisierung des Innenlebens eines Geräts zum besseren Verständnis/ der Wartung, etc. Beim zweiten Typ wiederum geht es, anders als der Name im ersten Moment vermuten lässt, *nicht* darum, die reale Welt, wie etwa wie bei VR, durch eine virtuelle zu ersetzen, sondern viel eher darum, die reale Welt mit Inhalten aufzufüllen. Ein Beispiel hierfür wäre etwa die Positionierung von spezifischen, in der realen Welt nichtexistierenden Objekten an einer bestimmten Position in der realen Welt: beispielsweise ein Modell eines zu bauenden Hauses auf einem leeren Baugrundstück. Dies führt dann gleich auch weiter zur Unterscheidung von AR-Systemen basierend auf deren (non-) existenter Ortsgebundenheit, wobei in **mobile** und **fixe AR-Systeme** unterschieden wird⁷⁴. Grundsätzlich beziehen sich diese Begriffe auf die Frage, ob man sich mit dem AR-System durch den physischen Raum – Stichwort Degrees of Freedom – bewegen kann, wie es etwa mit AR-Applikationen auf Smartphones der Fall ist, oder ob man räumlich gebunden ist, weil etwa eine kabelgebundene Verbindung zu einem stationären Computer vonnöten ist. (Rampolla & Kipper, 2012)

Bezüglich der Untergliederung der AR-Systeme verhält es sich an dieser Stelle ähnlich, wie mit den in Kapitel 2.4 dargestellten Abgrenzbarkeiten der einzelnen das Reality-Virtuality und Extended-Reality Continuum ausmachenden Typen. Der Grund hierfür ist jener, dass auch hier die einzelnen Typen oftmals nicht klar voneinander abgrenzbar sind, sich überlappen und Diskussionsstoff aufwerfen. Nichtsdestotrotz wird es in weiterer Folge von Nöten sein, die Untergliederungsversuche und -arten aufzuzeigen, um ein generelles Verständnis für AR aufzubauen. Zuvor soll aber noch ein kurzer Blick auf die rezenten Entwicklungen⁷⁵ im Bereich von AR-Systemen geworfen werden.

⁷³ Im engeren Sinne wird der erstere softwareseitige Teil auch als AR-Applikation, AR-App, AR-Software oder ähnliches bezeichnet. Generell gestaltet es sich aber oftmals so, dass, obwohl im eigentlichen Sinne von einem AR-System gesprochen werden sollte, einer der oberhalb angeführten Begriffe verwendet wird.

⁷⁴ Im Englischen gibt es die Begriffe „mobile AR“ und „fixed AR“ wieder, die an dieser Stelle ins Deutsche übersetzt wurden.

3.2.1 Abriss zur Entwicklungsgeschichte von AR

Oftmals wird in Artikeln oder auch bei Gesprächen auf die Neuartigkeit von AR hingewiesen und dadurch auch die (bisherige) geringe Verbreitung begründet, wobei diese Aussage an dieser Stelle strikt widerlegt werden muss. Sieht man sich etwa die Arbeiten von Kipper und Rampolla oder auch Mehler-Bicher und Steiger zu Augmented-Reality an, machen diese die ersten Umsetzungen von AR-Applikationen in den 1950, beziehungsweise 1960er Jahren fest, wenn auch die mittlerweile geläufige Definition von Augmented-Reality zu diesem Zeitpunkt in der heutigen Form noch nicht aufgestellt war. Zudem weisen sie auf die positiven Synergien der Entwicklung von (leistungsfähigen) Smartphones, die für AR genutzt werden können, hin⁷⁵. Ab diesem Zeitpunkt kann auch von einer starken Zunahme des Interesses an, der Verbreitung von und einer Ankunft von AR-Systemen in der Mitte der Gesellschaft gesprochen werden. Der Grund hierfür liegt darin, dass Smartphones als eine All-in-One Lösung betrachtet werden können, also eine Kombination aus Hardware- und zusätzlich Systemsoftwarekomponente, die es ermöglicht AR-Applikationen einfach auf einem Gerät ohne Umstände auszuführen und AR-Lösungen dadurch einen merklichen Aufschwung erfahren haben, und somit ein Berührungspunkt von AR für ein breites Publikum geschaffen wurde. Ebenso nicht außer Acht gelassen werden darf in diesem Sinne die sich zuletzt beschleunigende Entwicklung und der preisliche Abwärtstrend von HMDs und Smart-Glasses, welche sich ihrerseits als mögliche Lösung zur Thematik, wie AR-Applikationen Nutzer*innen bereitgestellt und präsentiert werden können darstellen. Darüber hinaus bieten HMDs und Smart-Glasses, wie in weiterer Folge noch detaillierter gezeigt werden wird diverse Lösungsansätze für Problematiken, welche bei der Darstellung von AR-Inhalten auf Smartphones auftreten. Für eine genauere Auseinandersetzung mit der Entwicklungsgeschichte von AR (ab Anfang/ Mitte der 2010er Jahre) sei an dieser Stelle aber auf die jeweiligen Kapitel in „*Augmented Reality : An Emerging Technologies Guide to AR*“ von Rampolla und Kipper (Rampolla & Kipper, 2012), beziehungsweise „*Augmented reality : Theorie und Praxis*“ von Mehler-Bichler und Steiger (Mehler-Bicher & Steiger, 2014) verwiesen. An dieser Stelle sei jedoch kurz auf den in beiden Büchern Anfang beziehungsweise Mitte der 2010er Jahre aufgestellten Ausblick im Bezug auf die zukünftige Entwicklung von AR und deren Anwendungsgebiete eingegangen. So wurde in beiden Büchern die These aufgestellt, dass zum einen AR, AR-Anwendungsfälle und Bereiche in denen AR zum Einsatz kommt zunehmen werden und insbesondere auch HMD ein größerer Stellenwert zukommen wird. Ersteres lässt sich dabei ohne Frage mit Blick auf diverse Anwendungsfälle, wie etwa Pokemon GO⁷⁶, dem Heimeinrichtungstool von IKEA⁷⁷, den Gesichtsfiltren von Instagram (und diversen anderen Social Media Apps) und Navigationsapplikationen (z.B. Google Live View), gesehen werden kann bestätigen.

⁷⁵ Für einen groben Abriss der Entwicklung von AR bis Beginn der 2010er Jahre siehe Mehler-Bicher und Steiger (Mehler-Bicher & Steiger, 2014) oder Rampolla und Kipper (Rampolla & Kipper, 2012), da im Rahmen dieser Arbeit nur die rezenteste seit den 2010er Jahren vollzogenen Entwicklungen dargestellt werden.

⁷⁶ Siehe auch: (Niantic, 2024)

⁷⁷ Siehe auch: (IKEA, 2024)

Ebenso bestätigen lässt sich an dieser Stelle die Zunahme der Verwendung von HMDs, wenn an dieser Stelle etwa in Richtung Wartung von technischen Anlagen oder Einschulungszwecke geblickt wird. Dennoch ist man noch bei weitem nicht in einer Phase angelangt, wo von einer flächendeckenden Nutzung von AR-Systemen gesprochen werden kann.

3.2.2 Abgrenzbarkeit von AR

Nun noch einige Anmerkungen zur Abgrenzbarkeit von AR, also zur Frage, wo diese beginnt und endet. Wie in Kapitel 2.4 gezeigt wurde, ist es an sich klar, dass es sich bei AR um die Art von Extended-Reality handelt, welche ihrerseits direkt an das reine Erleben der realen Welt anknüpft. Dennoch ist es aber schwierig, AR in Richtung virtueller Welt abzugrenzen. Weil, wie erwähnt, eine Abgrenzung von MR ohnehin beinahe unmöglich scheint, soll sich weiters an dieser Stelle damit begnügt werden, AR zur realen Welt und zu VR abzugrenzen und Grenzfälle wie etwa das in Kapitel 2.4 angesprochene Beispiel einer Einblendung basierend auf einem Standort zu thematisieren. Im Hinblick auf die Abgrenzung von AR zu VR sei an dieser Stelle erwähnt, dass sich VR dadurch definiert, dass die reale Umwelt, mit Ausnahme des körperlichen Verbleibs in dieser, zur Gänze ausgeblendet ist, oder in anderen Worten die Immersion einen hohen Grad aufweist. Dies bedeutet für AR in dem Fall, dass dies die Grenze in Richtung virtuellen Raum für AR darstellt, was aber keine weiteren Schlüsse ermöglicht. Gehen wir an dieser Stelle nun nochmals auf die Klassifikation von Azuma für AR ein, wonach AR wie bereits erwähnt A) **virtuelle und reale Informationen verknüpft**, zweitens B) **interaktiv in Echtzeit ist** und drittens C) in einer **3D-Umgebung** registriert – soll heißen verknüpft, arbeitet und ausgeführt wird – (Azuma, 1997) und setzen diese in Kontext zu drei konkreten Beispielen: die Anzeige einer 2D-Karte auf einem HMD, Einpark- und Navigationshilfen in Autos (Abb. 21-23) und der in Kapitel 2.4 angesprochenen Einblendung eines textlichen 2D-Hinweises basierend auf einem Standort. Hierbei gestaltet es sich bei ersterem Beispiel, der Einblendung einer 2D-Karte auf einem Display, etwa einer Google Glass, so, dass zwar Punkt A und B erfüllt sind, Punkt C jedoch nicht und es sich demnach um die reine Anzeige eines Inhalts auf einem vor den Augen getragenen Display handelt. Beim zweiten Punkt ist dies hingegen schon nicht mehr so klar feststellbar, da hier zwar Punkt A und B wieder als erfüllt angesehen werden können, sich jedoch bei Punkt C Fragen ergeben: etwa ob die anzuzeigenden Inhalte wirklich in einem virtuellen 3D-Raum berechnet und angezeigt werden, oder ob es sich um 2D-Inhalte, welche angezeigt werden, handelt. Dies ist dabei jedoch ohne tiefergehenden Einblick in die eigentliche Umsetzung zu haben – da Dokumentation hierzu die Ausnahme darstellen – schwer oder aber gar nicht möglich festzustellen. Ein etwaiges Beispiel hierfür wären etwa Einparkhilfen. Hier kann es zum einen sein, dass tatsächlich Inhalte in einem virtuellen dreidimensionalen Raum berechnet und angezeigt werden, oder aber auch, dass einfach nur Kreisbögen in Abhängigkeit des Lenkwinkels angezeigt werden. Ein weiteres Beispiel wäre hierzu dann auch noch eine Navigationsapplikation, welche deren Inhalte als Pfeile auf einem HUD präsentiert, wobei es sich hier dann nicht um eine AR-Applikation, jedoch um eine Location-Based-Service

Applikation handelt. Einfacher festzustellen ob es sich bei einer Navigationsapplikation in einem Fahrzeug um eine AR-Applikation handelt oder nicht ist es hingegen bei Fahrzeugen bei denen die Navigationsapplikation mit Kamerasystemen arbeitet (Abb. 21), da man hier zumeist rasch erkennt, ob etwa Fahrtrichtungsanweisungen (Pfeile, Linien) entsprechend der Umgebung (auf der Fahrbahn, an der Kreuzung) korrekt überlagert werden, oder nicht.

Als letztes sei nun noch auf das Beispiel einer textlichen Einblendung bei einer Navigationsapplikation auf einem See-Through-HMD, wie sie in der praktischen Arbeit umgesetzt werden wird (siehe Abb. 45 Mittig) eingegangen. Wie auf Abbildung 45 Mittig (links oben) zu sehen, wird bei der konkreten Umsetzung tatsächlich nur ein zweidimensionaler (textlicher) Inhalt angezeigt, wobei die Umsetzung wie beim ersten Beispiel auch Punkt A und Punkt B erfüllt, jedoch Fragen bei Punkt C aufkommen. Dabei ist es tatsächlich so, wie in Kapitel 5 gezeigt werden wird, dass die Umsetzung der Einblendung zur Gänze in einem für die weiteren Navigationsfunktionen benötigten 3D-Raum erstellt und berechnet wird, selbst jedoch keine dreidimensionalen Informationen enthält und auch keine Bindung an den virtuellen Raum per se aufweist. Somit befindet man sich hier eigentlich an einem Punkt, wo es zum einen, aufgrund des Hintergrundwissens valide ist zu sagen, dass es sich um eine AR-Umsetzung handelt, jedoch auch, dass es sich wie im ersten Fall der Anzeige einer 2D-Karte rein um die Einblendung auf einem Display handelt und es somit keine AR-Umsetzung ist. Aus Sicht dieser Arbeit würde die Umsetzung an dieser Stelle, als AR-Umsetzung betitelt werden, was damit begründet wird, dass sie in eine an sich größere, ohne Frage als AR-Umsetzung zu betitelnder Applikation eingebunden ist. Würde sie hingegen für sich stehen wäre sie tendenziell, nach Ansicht dieser Arbeit nicht als AR-Umsetzung zu betiteln.

Insgesamt zeigt dieser Punkt der Abgrenzbarkeit abermals, und wie weiters auch noch in anderen Aspekten zu sehen sein wird, die Problematik im XR-Bereich auf: Unklarheiten bezüglich Definition und Abgrenzung. Aus Sicht dieser Arbeit sei an dieser Stelle hinzugesagt, dass die vorhin genannten Grenzfälle der simplen Einblendungen nicht als AR-Applikation, sondern als reine Anzeige einer Information auf einem Display angesehen werden: etwa Einblendungen textlicher Hinweise, die reine Einblendung von Kreisbögen bei Einparkhilfen oder die simplen Anzeige von Pfeilen als Navigationshilfe bei einem HUD ohne Bindung zur Kreuzung oder etwa einer Autobahnabfahrt.

3.3 AR-Applikationstypen und darauf aufbauende Einteilungen

3.3.1 AR-Applikationstypen basierend auf deren technischer Umsetzung

Anschließend an diese sehr grundlegendende Unterscheidung und den wirklich sehr kurzen geschichtlichen Abriss von AR-Systemen stellt sich nun natürlich die Frage nach den verschiedenen Methoden, Techniken und zur Umsetzung von AR-Applikationen benötigten Technologien, wie auch der Gliederung von AR-Applikationen. Hierzu lohnt in erster Linie

der Blick auf rezente Artikel im Internet, da aufgrund der in letzter Zeit durch Smartphones und HMDs beziehungsweise Smart-Glasses beschleunigten Entwicklung in herkömmlicher Form veröffentlichte Arbeiten (Bücher, Artikel etc.) in diesem Bereich oftmals dem Stand der Dinge hinterherhinken.

Die Darlegung der verschiedenen möglichen Typen der Umsetzung von AR-Applikationen soll an dieser Stelle für ein leichteres Verständnis mit drei weit verbreiteten und beliebten Beispielen begonnen werden: Pokemon GO, der IKEA Heimeinrichtungs-App und den Instagram Gesichtsfiltren. Bei ersterem Spiel werden in der realen Umwelt, basierend auf Positionsdaten des Geräts, auf welchem die Applikation ausgeführt wird, virtuelle Inhalte, nämlich Pokemons platziert, während bei zweiterem aufgrund der Erkennung von gewissen Strukturen im Raum Möbelstücke platziert werden können. Die dritte und letzte Anwendung wiederum ermöglicht es, basierend auf der Erkennung von Gesichtern, diese mit Objekten, wie etwa Hundeohren, Hüten oder ähnlichem zu überlagern. Der für diese Arbeit wichtige Punkt, dem sich der nächste Abschnitt nun auch widmen sollen, liegt dabei darin, wie und mit Hilfe welcher Technologien die verschiedenen Inhalte in dem von den Applikationen erstellten AR-Raum (= Vermischungen der virtuellen und realen Umwelt) platziert werden, weshalb in weiterer Folge genau dies betrachtet werden wird.

Als Erstes muss hierzu gesagt werden, dass die Art der Umsetzung von AR-Applikationen in einem ersten Schritt entsprechend der zur Anwendung kommenden Technologien in zwei Gruppen, nämlich die „**Marker-Based**“ (dt.: markierungsbasierte)⁷⁸ oder „**Markerless**“ (dt. markierungslose) AR-Applikation

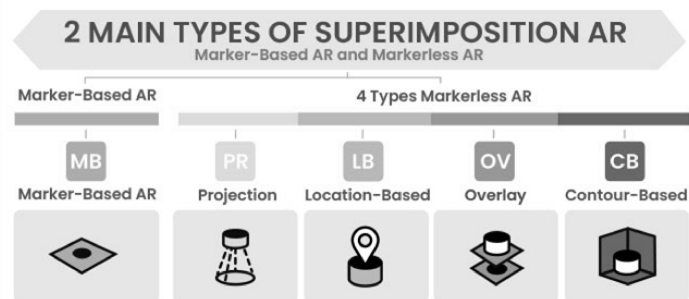


Abbildung 6: AR-Applikationstypen (verändert nach (Nextech3D.ai, 2022))

unterteilt werden können. Letztere unterteilt sich dann abermals in die vier Unterkategorien⁷⁹ **Projection**, **Location-Based**, **Overlay** und **Countour-Based AR**. Der Hauptunterschied zwischen Marker-Based und Markerless AR liegt dabei, wie bereits durch die Bezeichnung angedeutet, darin, dass für **Marker-Based AR** für eine Applikation lesbare Markierungen, zumeist QR-Codes⁸⁰ angebracht sein müssen, sodass die Applikation weiß,

⁷⁸ „Marker Based“ wird oftmals auch als „image-recognition“ (dt. Bilderkennung) oder „Pattern“ (dt. Muster) geführt. Siehe hierzu etwa: (Gerasymov, 2024; Hayes, 2009; Kipper & Rampolla, 2012)

⁷⁹ Hinsichtlich der Unterkategorien existieren verschiedene Name für die gleichen Typen von AR-Applikation. So stößt man zwar zumeist auf die beiden Typen „Projection AR“ und „Location-Based AR“, „Contour-Based AR“ und „Overlay AR“ hingegen treten oftmals auch mit dem Namen „Outlining AR“ für ersteres und „Superimposition-based AR“ für zweiteres auf. Siehe hierzu etwa (Gerasymov, 2024)

⁸⁰ An dieser Stelle muss hinzugefügt werden, dass ein eingescannter QR-Code natürlich in weitaus größerem Ausmaß für Dinge abseits der Nutzung für AR-Applikationen, wie etwa für die Verlinkung zu einer Speisekarte eines Restaurants, Verwendung findet, dies aber nichtsdestotrotz gerade zu Beginn von AR-Applikationen eine wichtige Möglichkeit zur Umsetzung von Funktionalitäten an bestimmten Orten diene.

welche Inhalte wo und wie angezeigt werden müssen. Dies bedeutet im Umkehrschluss aus technischer Sicht, dass neben der grundlegenden Plattform zur Lauffähigkeit der AR-Applikation nur eine Kamera benötigt wird – und nicht wie bei anderen Typen etwa auch GPS und IMU, oder Machine-Learning (etwa zur Erkennung von Räumen). Ein Beispiel hierfür wäre etwa ein künstlerisches Werk, welches mit QR-Codes versehen ist und sich am Display des Smartphones (beispielsweise farblich) verändert, sobald man diese einscannt. Bei **Markerless AR** hingegen handelt es sich im Gegensatz zu Marker AR tendentiell um komplexere Umsetzungen, wobei es sich bei ersterer der **Projection-Based Augmented Reality**⁸¹ um eine noch relativ einfache, eine Sonderstellung einnehmende Form des AR handelt, bei welcher Lichtstrahlen durch einen Beamer auf eine Fläche projiziert (beispielsweise Hologramme von Personen) werden. Diese Form wird dann weiters noch in zwei Unterarten, basierend auf der Frage, ob mit den projizierten Inhalten interagiert werden kann oder nicht untergliedert. Beispiele wären dabei zum einen die Projektion sich verändernder künstlerischer Lichteffekte auf eine Hausfassade bei einer nicht interaktiven Umsetzung und die Realisierung der interaktiven Sandbox der UCLA⁸², wobei das Kamerasystem die Höhenveränderungen (d.h. Eingaben der Nutzer*innen) wahrnimmt und dementsprechend die projizierte Hypsometrische Farbgebung der Sandkasten-Landschaft bei einer interaktiven Umsetzung anpasst. **Location-Based Augmented Reality** wiederum macht sich das globale Koordinatensystem (WGS84) zunutze, wobei hierfür beispielsweise die in Smartphones bereitgestellten GPS- und IMU-Sensordaten eines Geräts abgerufen werden und basierend auf diesen eine Umrechnung von Echtweltkoordinaten in Virtuelleweltkoordinaten genacht wird, um darauf aufbauend einen virtuellen Raum mit den anzuzeigenden Inhalten zu erstellen⁸³. Das prominente und bereits erwähnte Beispiel hierfür ist dabei Pokemon GO, bei welchem auf der Basis von fix koordinativ verorteten Punkten der realen Welt Pokemons im virtuellen Raum platziert werden und wie bei realen Objekten auch nur sichtbar sind, wenn man sich in der Nähe des Objekts befindet. Hinsichtlich der Funktionsweise gestaltet es sich dabei so, dass eine simple Koordinatenabfrage – die Umrechnung beider Koordinatenräume ineinander ungeachtet – durchgeführt wird und die darzustellenden Inhalte einfach in Abhängigkeit der Entfernung des Smartphones zum Ort der Positionierung des Pokemons skaliert (und in Abhängigkeit der Blickrichtung rotiert) wird. Der dritte Typ, **Contour-Based AR** oder auch Outlining AR, wiederum macht sich hauptsächlich die Kamera eines Geräts und Funktionen zur Erkennung von Kanten zu Nutze und wird beispielsweise in Autos zur Erkennung von Fahrbahnabgrenzungen oder Bordsteinkanten beim Einparken genutzt. Der vierte und letzte Typ, das sogenannte **Overlay AR** oder auch Superimposition-based AR, erkennt basierend auf dafür erstellten Algorithmen Objekte/ Räume und ermöglicht es, diese zu modifizieren beziehungsweise mit anderen Objekten zu ergänzen. Einige praktische Beispiele hierfür finden sich etwa im militärischen Raum, wobei gewisse Objekte (etwa feindliche Flugzeuge) erkannt und

⁸¹ Bei dieser Art von AR, wird ein zusätzliches Projektionsgerät (ugs. Beamer) benötigt.

⁸² Siehe: (UCLA, 2015)

⁸³ Für eine etwas genauere Beschreibung der Funktionsweise der Umrechnung von real-world coordinates und virtual-world coordinates siehe Kapitel 5.3.4.4

demnach (etwa mit einer Umrandung) markiert werden, oder aber auch im zivilen Bereich bei Möbelkonfiguratoren, welche in einfachster Form die Änderung der Farbe von Möbelstücken zulassen oder in komplexerer Form wie etwa bei der Umsetzung von IKEA den Raum „abscannen“ und es somit möglich machen, deren 3D modellierte Möbelstücke im Raum zu positionieren. Ein anderes Beispiel wären hierfür dann auch noch die diversen Apps wie Instagram, die es erlauben sogenannte „Gesichtsfilter“ oder Objekte wie Hüte auf Gesichter anzuwenden. (Gerasymov, 2024; Nextech3D.ai, 2022)

Das Problem bei dieser Untergliederung von AR ist jedoch abermals die Abgrenzbarkeit und der Umstand, dass es bei praktischen Umsetzungen zumeist so ist, dass nicht nur ein Typ, sondern eine Mischform zur Anwendung kommt, da dies oftmals zur Erreichung der gesteckten Ziele für eine Applikation von Nöten ist.

3.3.2 Versuch der Einteilung nach Anwendungsszenarien (Mehler-Bicher & Steiger, 2014)

Ein alternativer Zugang zur Unterteilung der verschiedenen Arten der Umsetzung von AR-Applikationen wird in weiterer Folge von Mehler-Bicher und Steiger aufgezeigt, welche AR-Applikationen basierend auf deren (möglichen) Anwendungsszenarien untergliedern. Diese Untergliederung basiert dabei auf der Überlegung der Abkehr der strikten Bindung der Typisierung von AR-Applikationen an technische Aspekte, worin die Autoren Probleme aufgrund der unklaren Abgrenzbarkeit der einzelnen Typen voneinander sehen: für die Kritik herangezogen wird an dieser Stelle die zur oberhalb nicht unähnlichen aber teils veraltete Klassifikation von Hayes (Hayes, 2009). Bedenken ergeben sich für Mehler-Bicher und Steiger auch in Hinblick auf die sich aus Hayes Klassifikation ergebenden Anwendungsszenarien. Die in ihrer Arbeit aufgestellte Klassifikation fußt darum, wie bereits erwähnt, rein auf Anwendungsszenarien und stellt den einzelnen Klassen das Wort „Living“, also zu Deutsch „lebend“, voran, wodurch darauf hingewiesen werden soll, dass für den/ die Nutzer*in Inhalte zum Leben erweckt werden. Weiters ergänzen sie hinsichtlich ihrer Klassifikation auch, dass grundsätzlich weitere Anwendungsszenarien existieren können, welche einer eigenen Klasse bedürften, diese aber (momentan noch) nicht abgebildet werden (können). Insgesamt stellen sie in ihrer Arbeit die folgenden acht Klassen auf:

1. Living Mirror
2. Living Print
3. Living Game mobile
4. Living Architecture
5. Living Poster
6. Living Presentation
7. Living Meeting
8. Living Environment

Auf eine weiterführende Besprechung der angeführten Klassen wird an dieser Stelle verzichtet und auf die Ausführungen seitens Mehler-Bicher und Steiger verwiesen. Es sei

jedoch kurz angemerkt, dass sich diese Klassifikation zumindest nach Einschätzung dieser Arbeit zu dezidiert auf Anwendungsfälle bezieht, was dem Fokus der Arbeit für „Technologen als auch Ökonomen [...] um beide gedanklich anzunähern und zu verbinden“ (Mehler-Bicher & Steiger, 2014) geschuldet sein dürfte. Der Grund hierfür ist zum einen der fehlende Spielraum, um neu aufkommende Realisierungen von AR abzubilden, aber ebenso, dass auch deren bisherige Klassifikation nicht als überschneidungsfrei bei momentan existenten Realisierungen beschrieben werden kann. Insgesamt würde es also immer wieder relativ rasch nötig werden, neue Klassen, die eine Kombination aus anderen Klassen darstellen, einzuführen.

3.3.3 Weitere Prinzipien und Einteilungsmöglichkeiten

Als weitere wichtige Einteilungsmöglichkeiten im Bereich von Augmented-Reality stellen sich an dieser Stelle noch einige eigentlich offensichtliche Formen, wie etwa die Einteilung, auf welcher Art von Geräten ein AR-System ausgeführt wird (beispielsweise Computer, Laptop, Smartphone, Tablet oder ähnliches), oder auf welchem Betriebssystem die jeweilige Applikation (also etwa iOS, Android, Windows, Linux, etc.) lauffähig ist dar. Hinsichtlich dieser beiden, auf den ersten Blick einfachen Einteilungen muss an dieser Stelle jedoch hinzugefügt werden, dass auch AR-Applikationen, welche im Webbrowser ausgeführt werden, existieren und eine Sonderstellung einnehmen, da sie auf allen der vorherig genannten Geräte/ Betriebssysteme laufen, solange der benötigte Webbrowser ausgeführt werden kann. Ohne diese Einteilung an dieser Stelle weiter auszuführen, können zudem noch etwaige weitere Einteilungsmöglichkeiten, basierend auf den technologischen Zugängen bei der Entwicklung, wie etwa welcher Entwicklungszugang oder welche Software gewählt wurde, gefunden werden. An dieser Stelle sollte jedoch dieser kurze Überblick genügen, um Platz für das nächste Kapitel rundum „AR und Kommunikation“, welches sich den Darstellungsmöglichkeiten, also den Geräten auf welchen AR-Systeme realisiert (und demnach eingeteilt) werden können, und den Interaktionsmöglichkeiten mit AR-Applikationen widmet zu lassen.

3.4 AR und Kommunikation

Nachdem nun bereits über Augmented-Reality per se und diverse Einteilungsmöglichkeiten gesprochen wurde, wird es an dieser Stelle nötig, sich mit der Frage der Kommunikation von AR-Inhalten zu beschäftigen. Kommunikation definiert sich dabei in der Kommunikationswissenschaft im weiteren Sinne als „Prozess der Übertragung von Nachrichten zwischen einem Sender und einem oder mehreren Empfängern“ (Maier, et al., 2018), wobei an dieser Stelle nicht definiert werden kann, was der Inhalt der Nachricht ist und in welcher Form⁸⁴ dieser übermittelt wird: beispielsweise etwa visuell, auditiv, haptisch oder olfaktorisch. Ebenso wenig kann über die Art des Senders/ Empfängers, also auf Basis

⁸⁴ Hiermit sind die fünf Sinneswahrnehmungen von Menschen (Sehen, Hören, Riechen, Tasten, Schmecken) gemeint.

der Frage, ob es sich um ein Lebewesen, eine Maschine oder anderweitiges handelt, über die Intention hinter der Kommunikation oder die grundsätzlichen Inhalte, die der Sender übermitteln wollte und der Empfänger verstanden hat, Aussagen getroffen werden. Natürlich gibt es an dieser Stelle noch eine Fülle weiterer Aspekte der Kommunikation und eine Vielzahl verschiedener Kommunikationsmodelle, wobei eine genauere Betrachtung dieser an dieser Stelle definitiv zu weit führen würde. Hierzu sei auf den breitgefächerten Bereich der Kommunikationswissenschaften verwiesen⁸⁵ und einschränkend erwähnt, dass im Rahmen der Arbeit nur in Bezug auf AR-Systeme und deren Kommunikation mit Menschen (und vice versa) relevante Aspekte aus den Kommunikationswissenschaften herausgepickt und kurz besprochen werden. (Kergel & Heidkamp-Kergel, 2023)

Als nächstes stellt sich nun die Frage, wie AR-Systeme mit den jeweiligen Nutzer*innen kommunizieren. Hier stellt es sich so dar, dass grundsätzlich, wie bei Kommunikation üblich, eine Vielzahl an Wege zur Verfügung stehen, wobei der am häufigsten genutzte sicherlich die visuelle (Sehsinn-basierte) Präsentation, gefolgt von auditiver Übertragung der Information ist. Natürlich gibt es auch Möglichkeiten zur Übertragung von Informationen mittels haptischen Feedbacks⁸⁶ oder Geruchs- und Geschmackssinn, wobei die letzteren beiden bei AR-Systemen soweit bekannt nicht zum Einsatz kommen. Der Einsatz eines haptischen Feedbacks wird überdies ebenso, mit Ausnahme der Nutzung von Vibration am Smartphone äußerst selten genutzt.

3.4.1 AR-Funktionsweise und häufige Kommunikationsformen

Bevor nun auf einige Beispiele zur visuellen Übermittlung von Informationen mittels verschiedener Geräte eingegangen wird, soll zuvor noch einmal kurz in genereller Weise besprochen werden, was für die Lauffähigkeit einer AR-Applikation gegeben sein muss. Laut der in Kapitel 3.1 angeführten Definitionen, entsprechend dem in Kapitel 3.2 dargelegten Aufbau von AR-Systemen und der in Kapitel 3.3 herausgearbeiteten Arten von AR-Applikationen muss zum einen Hardware⁸⁷, ein darauf funktionierendes Betriebssystem, wie auch die eigentliche AR-Applikation⁸⁸ gegeben sein.

⁸⁵ Einen kurzen Überblick über die wichtigsten Modelle der Kommunikationswissenschaften bieten, ohne ins Detail zu gehen, etwa Adams oder auch Keil in ihren Artikeln rundum Kommunikationsmodelle. Siehe (Adams, 2023; Keil, 2023)

⁸⁶ Ein Beispiel hierfür wäre etwa die Vermittlung haptischen Feedbacks über eigens dafür entwickelte Handschuhe (z.B. wenn etwas angefasst wird), oder aber in einfachster Form über die Vibration eines Smartphones (z.B. wenn man an einen Kreuzungspunkt bei Navigationsapplikationen angelangt).

⁸⁷ D.h. Hardware der Zentraleinheit (für das Ausführen der Software & Berechnungen) und Peripherie (etwa für die Anzeige von Inhalten oder Verortung), oder eine All-in-One Lösungen wie Smartphones.

⁸⁸ Die Kompatibilität mit Soft- und Hardware wird an dieser Stelle vorausgesetzt.

Die Anforderungen an die Hardware und das Betriebssystem unterscheiden sich dabei je nach (Art der) AR-Applikation, wobei ersterer Punkt – jener der Abhängigkeit von zur Verfügung stehender Hardware – sehr gut anhand von Location-Based AR-Applikationen darstellbar ist: Zur Realisierung einer solchen Applikation ist es zwangsläufig nötig, dass das eingesetzte Gerät eine Komponente zur Positionsbestimmung – zumeist einen GNSS-Empfänger – aufweist, wobei es je nach Art der Applikation möglich sein kann (beispielsweise beim Einsatz eines See-Through-HMD), dass keine Kamera benötigt wird. Genau gegenteilig verhält es sich bei den Marker-Based AR-Applikationen, wo die Positionsbestimmung irrelevant, die Aufnahme eines Markers mittels Kamera aber essenziell ist. Betrachtet man nun den Punkt der Abhängigkeit von AR-Applikationen von Betriebssystemen, so gestaltet sich dies so, dass Applikationen die für iOS/ Ubuntu zur Verfügung stehen nicht zwangsläufig für Android/ Windows (und vice versa) zur Verfügung stehen. Der Grund hierfür liegt darin, dass Applikationen, je nach der Plattform, auf welcher sie lauffähig sein sollen, anders aufgebaut werden müssen. Ein Beispiel aus dem eigenen Erfahrungsbereich bezieht sich etwa auf das Ansprechen und Abrufen von ein- und derselben Funktion in Unity: den GNSS-Chip bei iOS/ Android. Insgesamt bedeutet dies für eine AR-Applikation, dass entweder das korrekte für die Ausführung benötigte Set aus Hardware und Software (also Betriebssystem und ggf. weitere benötigte Pakete) vorhanden ist, oder diese schlichtweg nicht funktioniert. Im umgekehrten Fall wiederum bedeutet es, dass wenn eine Applikation entwickelt werden soll, man sich zuerst einmal mit den zur Verfügung stehenden Optionen hinsichtlich der Hard- und Software auseinandersetzen muss.

Anschließend an diese Erläuterung zur Notwendigkeit gewisser Hard- und Software für die Lauffähigkeit von AR-Applikationen und der großen Bandbreite an solcher – vor allem hinsichtlich der Hardware – sollen im Anschluss nun einige häufig für AR-Applikationen eingesetzte Geräte zur Kommunikation von Informationen – und abschnittsweise auch die hierfür benötigten Berechnungseinheiten und weiteren Hardwarekomponenten – thematisiert werden. Hierbei geht es vor allem um die Art, wie eine AR-Applikation dem/ der Nutzer*in oder einer Nutzergruppe angezeigt und diese somit erlebt wird/ werden kann, um etwaige Vor-/ Nachteile wie auch Anwendungsfelder sichtbar zu machen und diese abschließend vergleichen zu können. Konkret wird in weiterer Folge auf drei häufig bei AR-Anwendungen eingesetzte Geräte, nämlich Smartphones, HMDs und Projektionsgeräte, fokussiert werden, wobei immer im Hinterkopf zu behalten ist, dass das Präsentieren von AR-Inhalten der eine, das Berechnen der Informationen zur Präsentation ein anderer Punkt ist: Ein Smartphone etwa kann beide Tätigkeiten in einem Gerät vereinen, während es bei HMDs und Projektionsgeräten zumeist so ist, dass Berechnungseinheit und Präsentationseinheit getrennt (Stichwort Peripherie & Zentraleinheit) voneinander sind. Der konkrete Fall stellt sich etwa bei AR-Brillen (wie etwa der XREAL Air, oder der Rokid MAX) dar, welche im Prinzip nichts anderes sind als Brillen, welche mit Monitoren und Bewegungssensoren (IMUs) versehen sind. Sie besitzen jedoch keine Berechnungseinheit und müssen darum mit Berechnungseinheiten verbunden werden, um überhaupt Inhalte anzeigen zu können.

3.4.1.1 AR mittels Smartphones

Smartphones stellen in der Welt der AR-Applikationen mehr oder minder eine All-in-One Lösung dar: also Geräte, welche alle gängigen (software-)technischen Komponenten für die Umsetzung einer AR-Applikation in einem Gerät vereinen – in anderen Worten, die gesamte Peripherie ist intern. Die Einschränkung mit mehr oder minder wird oberhalb aufgrund zweierlei Gründe angeführt: Zum einen muss, wie bereits erwähnt, das Betriebssystem des jeweiligen Smartphones von der AR-Applikation unterstützt werden und, was exklusiv bei Android⁸⁹ zum Tragen kommt und insbesondere Entwickler*innen das Leben erschwert, muss ARCore vom Smartphonehersteller auf dem jeweiligen Gerät unterstützt werden. Zwar ist grundsätzlich eine Implementierung von AR-Applikationen ohne ARCore möglich, jedoch wird dies in der Praxis aufgrund eines enormen Mehraufwands zumeist nicht gemacht. Für eine genauere Besprechung der beiden SDKs ARCore und ARKit sei an dieser Stelle auf Kapitel 3.5.1 verwiesen.

Mit Blick auf den Aufbau von Smartphones als Plattform für AR-Applikationen kann festgehalten werden, dass diese den einfachsten Einstieg – da in breiter Masse vorhanden und demnach leicht verfügbar – zum Erleben einer AR-Applikation darstellen, weil keine extra Geräte angeschafft oder verbunden werden müssen, sondern das Smartphone alle Tätigkeiten von der Berechnung über die Bedienung der Applikation bis hin zur Präsentation der Inhalte übernehmen kann. Hierbei gestaltet sich die Funktionsweise so, dass simplifiziert dargestellt die Kamera die reale Welt aufnimmt und in weiterer Folge an die AR-Applikation weitergibt, welche ihrerseits die Aufnahmen in einer je nach Applikation unterschiedlichen gewünschten Weise modifiziert oder erweitert und dann am Display darstellt.

So gut nutzbar Smartphones als Plattform auch sind – unter anderem aufgrund einer Vielzahl an in diesen verbauten Sensoren, wie etwa GNSS-Empfänger, IMUs, ... –, so unterliegen sie doch bestimmten Limitationen, insbesondere bei den Arten der Visualisierung und dem Level der Immersion: Als limitierende Faktoren stellt sich dabei die Art wie ein Smartphone per se genutzt wird (Halteposition und Blickrichtung während der Nutzung) und die unabdingbare Nutzung der Kamera für die Realisierung von AR, wodurch wiederum die Halteposition des Smartphones bestimmt wird dar. Damit ist zum einen gemeint, dass der Ausschnitt des Blickfeldes, der bei der Nutzung eines Smartphones als Präsentationsmedium genutzt werden kann, naturgemäß relativ unter anderem im Hinblick auf die Brennweite beschränkt ist, und zum anderen Smartphones, selbst wenn die Brennweite kein Problem darstellen würde, diese nicht direkt vor die Augen gehalten werden und somit nur ein kleiner Teil des Blickfelds eingenommen wird. Hinsichtlich der Nutzung von Smartphones ist außerdem die Orientierung⁹⁰ zu bedenken, also wie das Handy gehalten wird, da immer nur gezeigt werden kann, worauf das Handy beziehungsweise die

⁸⁹ Bei iOS ist dies, aufgrund der geringeren Anzahl an Geräten mit diesem Betriebssystem, nicht der Fall. Hier hängt es rein von der Generation des iPhones und dessen Betriebssystemversion ab. Die Bibliothek, welche die Entwicklung von AR-Applikationen auf iOS ermöglicht, wird ARKit genannt.

⁹⁰ In diesem Sinne ist nicht die Orientierung im Portrait- oder Landschaftsmodus, auch wenn dies natürlich auch Einfluss auf die sichtbaren Inhalte hat, gemeint.

darin verbaute Kamera gerichtet ist. Ebenso unterliegen Smartphone-Display der Limitation, dass diese aufgrund der Tatsache, dass nur ein „normales“ Display existiert, keine echten dreidimensionalen Eindrücke vermittelt werden können⁹¹.

3.4.1.2 AR mittels HMDs ⁹²

Generelles zu HMDs

Bevor an dieser Stelle, wie bei Smartphones zuvor, kurz die Realisierung von AR-Inhalten mittels HMDs angerissen wird, ist es in weiterer Folge unabdingbar, HMDs an sich, also deren Aufbau, Arten und Typen etwas genauer zu beleuchten⁹³. Bei Head-Mounted-Displays handelt es sich, wie der Name bereits suggeriert, um Displays, welche am Kopf getragen werden. Diese stellen im Rahmen dieser Arbeit neben AR-Applikationen an sich sicherlich einen der sich zuletzt in ungeahntem Ausmaß und beeindruckender Geschwindigkeit entwickelnden Bereiche dar, wobei es gerade aufgrund dieser zwei Faktoren auch oftmals schwierig ist, den Überblick zu behalten und klare Kategorisierungen aufzustellen. Einen Anhaltspunkt zur Abgrenzung von Head-Mounted-Displays bietet jedoch zumindest in einem ersten Schritt die grobe Zuteilung der verschiedenen Geräte zu den verschiedenen Bereichen von Extended-Reality, also den verschiedenen Arten von XR-Erlebnissen, die mit selbigen realisiert werden können. Einschränkend muss hierbei aber auch direkt erwähnt werden, dass ein nicht unbeträchtlicher Teil der HMDs sowohl im AR- wie auch VR-Bereich eingesetzt wird und Applikationen dafür umgesetzt werden können. Ein zum Zeitpunkt der Ausarbeitung sehr aktuelles Beispiel hierfür ist etwa die Meta Quest 3⁹⁴, welche Inhalte über das gesamte Extended-Reality Spektrum hinweg abdecken kann. Da sich die vorliegende Arbeit aber hauptsächlich mit AR beschäftigt – und der Rahmen dieser Arbeit nicht gesprengt werden soll –, wird an dieser Stelle der Fokus auch auf eben

⁹¹ Zur Diskussion, wann Displays zweidimensionale Inhalte und wann sie dreidimensionale Inhalte präsentieren, sei erwähnt, dass ein echter dreidimensionaler Eindruck von Objekten Menschen nur vermittelt werden kann, indem dem linken und dem rechten Auge getrennt voneinander in bestimmter unterschiedlich voneinander aufgenommener Perspektive – d.h. Abstand und Winkel – Bilder vom darzustellenden Objekt zugespielt werden. Dieses Prinzip nennt sich „**stereoskopisches Sehen**“ und kann etwa über zwei gesonderte Displays oder über Farbfilter-, Polarisations-, Shutter- oder elektronische 3D-Brillen und ein Anaglyphenbild realisiert werden.

Zum Prinzip des stereoskopischen Sehens und Anaglyphen siehe: (Spektrum.de, 2000; Spektrum.de, 2001; Hauger, 2006)

Zu einer kurzen Einführung zu 3D-Brillen für die Stereoskopie siehe: (Blickcheck, kein Datum)

⁹² Im Weiteren werden Head-Mounted-Displays auch als Brillen und das Display/ die Betrachtungsfläche dieser als Brillengläser bezeichnet werden. Der Grund hierfür liegt zum einen in der Ähnlichkeit des Aufbaus mancher HMDs zu normalen Brillen, in der Tatsache, dass in der Literatur diese Begriffe auch Verwendung finden, und zu guter Letzt in der Vereinfachung der Sprache und einfacheren Vorstellbarkeit begründet.

Ebenso soll explizit darauf hingewiesen werden, dass an dieser Stelle auch Smart-Glasses zu HMDs gezählt werden.

⁹³ Der Grund hierfür liegt darin, dass HMDs und deren Funktionen im Gegensatz zu Smartphones und deren Funktionen, in der breiten Masse relativ unbekannt sind.

⁹⁴ Für detaillierte Informationen zu dieser siehe: (Meta, 2024; Pichler, 2023)

jene in diesem Bereich vorzufindenden Geräten gelegt werden, auch wenn klarerweise eine Vielzahl der Aspekte auch für HMDs im VR-Bereich, genauso wie für HMDs die beide Bereiche bedienen können, zutreffen.

Betrachtet man nun also Head-Mounted-Displays, die dem AR-Bereich zugeordnet werden können, so kann man diese in einem ersten Schritt in **monokulare** und **binokulare Head-Mounted-Displays** unterteilen. Der Unterscheidungsgrund bei diesen beiden Typen liegt hierbei im Aufbau der Brille selbst, wobei monokulare Brillen nur ein Display besitzen und aufgrund dieser Tatsache nur 2D-Darstellungen realisiert werden können. Binokulare Brillen hingegen besitzen zwei Displays und können durch das Prinzip des stereoskopischen Sehens/ der stereoskopischen Bildbetrachtung⁹¹ dreidimensionale Eindrücke und Bilder vermitteln. (ALEGER, 2024)

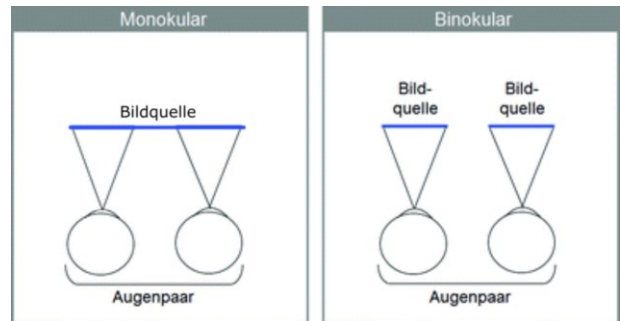


Abbildung 7: Monokular/ Binokular im Vergleich (verändert nach (Antakli, et al., 2017))



Abbildung 8: See-Through-HMD/ Smart-Glass: XREAL Air2



Abbildung 9: Non-See-Through-HMD: Meta Quest 3 (Painter, Lewis, 20204)



Abbildung 10: Look-Around-HMD: Google Glass (Reckmann, 2014)

Weiters können HMDs dahingehend unterteilt werden, ob die vor den Augen liegende Betrachtungsfläche undurchsichtig oder durchsichtig ist, wobei sich die einzelnen Arten dann in sogenannte **See-Through-HMDs**, **Non-See-Through-HMDs** und **Look-Around-HMDs**⁹⁵ einteilen. Bei ersterem Typ handelt es sich um Brillen, die normale Brillengläser (und ggf. davor liegende Projektionsflächen) besitzen, auf welche mittels im Rahmen befindlicher Displays und Umlenkspiegel die virtuellen Inhalte projiziert werden: Ein Beispiel hierfür ist beispielsweise die auf Abbildung 8 ersichtliche und im Rahmen der Arbeit Anwendung findende XREAL AIR 2. Wie bei normalen optischen Brillen nehmen See-Through-Brillen auch einen Großteil des Sichtfeldes ein und eignen sich somit optimal zur Darstellung von Informationen. Der zweite Typ (Non-See-Through-HMDs⁹⁶) blendet in einem ersten Schritt die Außenwelt komplett aus (wie etwa die Meta Quest 3 – siehe Abb. 9), da die Brillengläser undurchsichtig sind. Erst dadurch, dass diese Brillen eine Kamera besitzen, die die Außenwelt sichtbar machen kann, kann diese Art von Brillen für AR-Applikationen

⁹⁵ Dieser Typ existiert i.d.R. nur monokular.

⁹⁶ De facto handelt es sich vereinfacht gesagt um eine Brille, bei welcher die beiden Brillengläser mit Displays ersetzt wurden.

Verwendung finden. Der dritte und letzte Typ, jener der Look-Around-HMDs (wie etwa die Google Glass – siehe Abb. 10) ist wiederum durchsichtig, nimmt aber nur einen kleinen Teil des Sichtfelds in Anspruch. Dadurch, dass nur ein kleiner Teil des Sichtfeldes benutzt wird, kann dieser Typus aber auch nur wenige Informationen darstellen – und diese auch nur in 2D, da es sich um eine monokulare Lösung handelt. (Seydel, 2023)

Eine weitere wichtige und auf den ersten Blick oftmals nicht klare Unterscheidung, welche einer Erwähnung bedarf, ist dann noch jene der AR/VR-Brillen⁹⁷ und der sogenannten Smart-Glasses. So definiert Carter in ihrem Artikel auf XR Today die beiden Typen etwa wie folgt:

„VR/AR headsets are mixed reality or extended reality headsets that combine functionality for both virtual and augmented reality experiences. Smart glasses are AR or MR wearables that allow users to see digital content superimposed over their surroundings.“ (Carter, 2023a)

Funktionsweise von AR mit HMDs:

Hinsichtlich der Verwendung von HMDs für die Realisierung einer AR-Applikation muss an dieser Stelle zuerst einmal die Frage gestellt werden, ob das jeweilige HMD sowohl die Berechnung der anzuzeigenden Inhalte, also im Sinne der Hardware Peripherie und Komponenten der Zentraleinheit in einem „All-in-One“ Gerät, wie etwa ein Smartphone vereint und somit sowohl für Visualisierung als auch Berechnung verwendet wird, oder ob es sich nur um ein Peripheriegerät handelt, welches rein die Visualisierung und gegebenenfalls einige Aufgaben der räumlichen Verortung (etwa über eine IMU) übernimmt. Im zweiten Fall wird also ein Gerät benötigt, welches die Aufgaben der Bereitstellung der Informationen für das Peripheriegerät, in anderen Worten, eine Kombination aus Komponenten der Zentraleinheit und Betriebssystem, auf welchem die AR-Applikation lauffähig ist, übernimmt. Üblicherweise werden hierzu entweder Smartphones, Tablets, oder aber, wenn auch selten, Laptops beziehungsweise Computer herangezogen.

Zusammengefasst muss das Gerät, welches zur Berechnung der Inhalte verwendet wird, wie bereits im Kapitel oberhalb beschrieben und für Smartphones erläutert, zum einen die AR-Bedingungen – Stichwort Betriebssystem und ARCore/ ARKit – im Generellen erfüllen. Zusätzlich zu diesem Punkt ist dann natürlich auch noch die Unterstützung des jeweiligen HMDs – also etwa die Anbindung mittels USB Typ C mit DP-Alt Modus – nötig. Konzeptionell einfacher und eigentlich genau so wie bei Smartphones verhält es sich mit HMDs, welche eine All-in-One Lösung darstellen. Jedoch weisen diese eine nicht zu geringe Anzahl anderer negativer Faktoren auf: wie etwa den höheren Preis, proprietäre Software, die es schwierig macht, selbst Applikationen für diese zu entwickeln, und die Erweiterbarkeit um Peripheriegeräte, falls nötig. Hinsichtlich der einzelnen in HMDs

⁹⁷ An dieser Stelle soll nochmals darauf hingewiesen werden, dass HMDs oftmals sowohl den AR- als auch den VR-Bereich bedienen können.

verbauten Hardwarekomponenten verhält es sich überdies so, dass diese mitdefinieren, welche Applikationen mit diesen wie realisiert werden können.

Als Beispiel sei etwa ein See-Through-HMD mit beziehungsweise ohne eingebaute(r) Kamera in Blickrichtung angeführt. Hierbei ist es zwar möglich, beide für AR-Applikationen nutzbar zu machen, die Möglichkeiten variieren jedoch: Bei einem HMD mit Kamera⁹⁸ kann diese etwa genutzt werden, um alle AR-Typen abseits von Projected-AR umzusetzen: ähnlich wie bei einem Smartphone. Der Unterschied hierbei liegt simplifiziert nur darin, dass sich die zu verwendende Kamera und das Display für die Anzeige am HMD befinden und dies bei der Entwicklung berücksichtigt werden muss. Bei See-Through-HMDs ohne Kamera⁹⁸ hingegen können nur Location-based AR-Inhalte realisiert werden, da für die anderen Formen eigentlich immer eine Kamera zur Erkennung und Erfassung von Objekten/ Muster benötigt wird. Natürlich kann an dieser Stelle argumentiert werden, dass mit der Zuhilfenahme eines weiteren Geräts – wie bereits erwähnt werden ohnehin oftmals Smartphones/ Tablets, die eigentlich immer eine Kamera besitzen, zur Berechnung der Inhalte für HMDs verwendet – auch die anderen Typen realisiert werden können, jedoch ist dies keine realistisch gangbare Lösung, weil die Smartphonekamerablickrichtung in Einklang mit der Blickrichtung des HMDs zu bringen wäre, um Objekte korrekt im Blickfeld der das HMD nutzenden Person anzuzeigen. Der Grund hierfür liegt darin, dass bei der Verwendung der Smartphonekamera zur Aufnahme von Inhalten und Anzeige der veränderten Inhalte auf einem HMD logischerweise nur der durch die Smartphonekamera aufgenommene und mittels der AR-Applikation veränderte Aufnahmebereich auf dem HMD angezeigt werden könnte und dieser natürlich davon abhängt, worauf diese gerichtet ist. In anderen Worten: Die Blickrichtung der Kamera wäre ausschlaggebend für die zu sehenden Inhalte und nicht die Blickrichtung der das HMD benutzenden Person. An dieser Stelle sei hierzu gesagt, dass es natürlich unter bestimmten Umständen gewünscht sein kann, solch eine Lösung anzustreben, es jedoch als eher unwahrscheinlich angesehen wird.

Insgesamt stellt der vorangegangene Block nur die Spitze des sprichwörtlichen Eisbergs dar und wirft bei einigen Leser*innen sicherlich die Frage auf, welche Geräte für bestimmte Anwendungsbereiche und -fälle existieren. Von der Beantwortung einer solchen Frage wird jedoch an dieser Stelle bewusst Abstand genommen, da es sich, wie bereits erwähnt, um einen sich sehr schnell entwickelnden Bereich mit immer neu am Markt befindlichen Geräten handelt. Darüber hinaus ist es nicht möglich, eine generelle Empfehlung auszusprechen, da es hierbei, wie bereits dargelegt, sehr stark darauf ankommt, was das Ziel der eigentlichen AR-Applikation sein soll und welche Funktionen das HMD (ggf. in Kombination mit einem Gerät zur Berechnung der Inhalte) dafür aufweisen muss. Ein Beispiel, wie so ein Entscheidungsprozess ablaufen kann, wird in weiterer Folge im Kapitel 5.1.2 bis 5.2.1 angeführt werden, da es dort um die konkrete Auswahl eines HMD zur Realisierung einer bestimmten AR-Applikation mit definierten Funktionen und Anforderungen geht.

⁹⁸ Die Frage nach der Berechnungseinheit und die in den Brillen verbauten Peripheriegeräte wird an dieser Stelle außenvorgelassen. Es wird also angenommen, dass die Berechnung der Inhalte geregelt ist und alle nötigen Peripheriegeräte vorhanden sind.

3.4.1.3 Projektionsbasiertes AR⁹⁹

Der letzte Block, bevor die drei verschiedenen und gängigsten Möglichkeiten zur Darstellung von AR-Inhalten abschließend besprochen und verglichen werden, widmet sich nun Projektionsgeräten (ugs. Beamer), welche Inhalte in die reale Welt projizieren. Einer, wenn nicht der wichtigste Unterscheidungspunkt zu den beiden anderen Arten, wie AR erlebt werden kann, ist hierbei, dass sich anders als bei diesen die Darstellungen nicht auf einem oder mehreren Displays abspielen und dabei, wenn nur ein Gerät zur Visualisierung vorhanden ist, nur eine Person¹⁰⁰ die AR-Inhalte zu einem bestimmten Zeitpunkt erleben kann. Genau durch diese Besonderheit und einigen sich daraus ergebenden negativen Aspekten dieser Methode, welche später genauer besprochen werden (an dieser Stelle sei nur der Punkt der Notwendigkeit (halb-)dunkler Umgebung genannt), erschließt sich auch, dass dieser Darstellungsweise auch ein eigener AR-Typ, das Projection-Based AR gewidmet ist.

Bei projektionsbasiertem AR wird in weiterer Folge so vorgegangen, dass Projektionseinheit – in diesem Fall Teil der externen Peripherie – und Komponenten der Zentraleinheit plus weitere für die Funktionsfähigkeit nötige Komponenten voneinander getrennt sind. Die Projektionseinheit ist dabei, ähnlich wie ein HMD ohne Berechnungseinheit, für die Darstellung der Inhalte zuständig, welche ihr, zumeist von Seitens eines Laptops oder Computers übermittelt werden.

Grundsätzlich werden projektionsbasierte AR-Darstellungen noch in sogenannten statischen AR-Projektionen¹⁰¹, und dynamische AR-Projektion¹⁰² unterteilt. Bei ersterer werden Flächen mittels Beamer beleuchtet und visuelle Effekte eingeblendet, wobei aber keine User-Interaktion stattfindet. Ein einfaches Beispiel hierfür wäre etwa eine künstlerische Installation, bei welcher nur die Außengrenzen eines Mandalas an die Wand gemalt sind und der Beamer dann das Mandala per Projektion „ausmalt“ und visuelle Effekte einbaut¹⁰³. Statische Projektionen müssen dabei nicht zwangsläufig eine flache Projektionsebene¹⁰⁴ aufweisen, sondern können auch uneben sein beziehungsweise mehrere Ebenen haben. Logischerweise erhöht dies die Komplexität Inhalte darzustellen, weshalb hierzu auch weitere Geräte zur Distanzmessung und Erkennung beziehungsweise Interpretation der Ebene(n) nötig sind¹⁰⁵. Weiters können zusätzlich auch noch Tracking-Systeme, welche

⁹⁹ Im Englischen auch auffindbar unter den Namen „Projected AR“, „Projection-based AR“ oder „Spatial Augmented-Reality“.

¹⁰⁰ Dies gilt nur in beschränktem Ausmaß für Smartphones, aber auch hier ist es schwierig, dass mehrere Personen denselben Inhalt am selben Display betrachten.

¹⁰¹ ... aus dem Englischen „Static AR Projection“.

¹⁰² ... aus dem Englischen „Dynamic AR Projection“.

¹⁰³ Zur Visualisierung siehe: (Lightform, 2018)

¹⁰⁴ Die Projektionsebene ist die Fläche, auf der die vom Beamer ausgestrahlten Lichtstrahlen auftreffen.

¹⁰⁵ In der einfachsten Form kann hierzu etwa eine Kamerasystem, welches Depth-Recognition und ggf. Pattern-Recognition (um geeignete Projektionsflächen und deren Distanz zu erkennen) unterstützt – wie es bei vielen

beispielsweise die Kopfbewegungen des/ der Nutzer*in¹⁰⁶ verfolgen, eingesetzt werden, in dessen Folge es durch Veränderung des Bilds je nach Position und Blickwinkel des/ der Betrachter*in zur Projektionsfläche möglich wird, einen dreidimensionalen Eindruck zu erzeugen. Gerade die beiden nun erwähnten Punkte (Anm. der Distanzmessung/ Erkennung von Ebenen und dem Tracking) werden oftmals bei der zweiten Art von projektionsbasiertem AR, der sogenannten dynamischen AR-Projektion, eingesetzt, wobei sich diese durch die Möglichkeit der Interaktion mit dem projizierten Inhalt vom statischen AR unterscheidet. Ein in den Geowissenschaften sehr berühmtes Beispiel hierfür ist die sogenannte „*Augmented Reality Sandbox*“ der Universität von Kalifornien (UCLA) (UCLA, 2015), bei der es möglich ist, in einer Sandbox, Sand aufzutürmen oder Löcher zu graben und mittels Projektionshardware Isochronen¹⁰⁷ anzeigen zu lassen¹⁰⁸. (Davis, kein Datum; Dumbre, 2023)

3.4.1.4 Vergleich der Kommunikationsmöglichkeiten für AR-Inhalte

Nachdem nun die gängigsten Möglichkeiten wie AR-Inhalte erlebt werden können besprochen wurden, soll nun ein abschließender Vergleich – im Hinblick auf Vor-/ und Nachteile – mit Ausblick auf die Nutzungsarten und -möglichkeiten, samt Besprechung der Limitation der einzelnen Realisierungsarten gemacht werden.

	SMARTPHONE-AR	HMD-AR	PROJECTION-BASED AR
USABILITY	+	o	+ / -
KOMMUNIKATIONSFORMEN	Visuell, Auditiv, Haptisch	Visuell, Auditiv, (Haptisch)	Visuell, Auditiv, (Haptisch)
VISUELLE DARSTELLUNG	2D	2D / 3D	2D bzw. Abhängig von der Projektionsfläche
ANWENDUNGSBEREICHE	Freizeit, Marketing, Navigation	Industrie, Schulungen, Freizeit	Kunst, Marketing und Lehre
PREIS	Mittel	Mittel bis Hoch	-

Abbildung 11: Vergleichsmatrix ausgewählter AR-Systeme

Usability:

Die höchste Usability, also Grad der Einfachheit in der Benutzung, hat logischerweise das Smartphone, da beinahe jeder Mensch ein solches Gerät besitzt und zumindest die grundlegenden Funktionen bedienen kann. Die Installation der Applikation (sofern das Smartphone diese unterstützt) geht einfach von statten und es ist eine in sich geschlossene

Smartphones der Fall ist –, verwendet werden. Zu einem kurzen Überblick über Depth-Recognition siehe: (Kumar, 2024)

¹⁰⁶ Hier besteht rein technisch die Einschränkung der Verfolgung und Nutzung für eine Person, weshalb an dieser Stelle auch das Singular verwendet wird.

¹⁰⁷ Einfärbung der Höhenstufen mittels verschiedener Farben.

¹⁰⁸ Zur Visualisierung siehe: (UCLA, 2015)

Lösung – also ein All-in-One Gerät, da Berechnung, Visualisierung und ggf. Sensorik in einem Gerät kombiniert ist –, was bedeutet, dass keine weiteren Geräte zur Realisierung des AR-Erlebnisses, außer vielleicht ein Ladegerät, benötigt werden. Ebenso sind Smartphones sehr mobil und relativ stabil, was darüber hinaus die Anwendungsgebiete beinahe unendlich macht. Die Bedienung der einzelnen in der AR-Applikationen umgesetzten Inhalte gestaltet sich dann, wie bei Smartphones bekannt, über den Touchscreen, wodurch auch an dieser Stelle auf die Nutzer*innen keine Hindernisse zukommen.

Bezüglich HMDs gestaltet sich die reine Anwendung ebenso einfach, wobei es an dieser Stelle jedoch zu bedenken gilt, dass sich das initiale Setup¹⁰⁹ je nach HMD (auch hinsichtlich der Komplexität) etwas unterscheiden kann und natürlich die Frage zu stellen ist, ob ein extra Gerät – beispielsweise ein Smartphone – zur Berechnung der Inhalte benötigt wird. Diese Frage nach der Notwendigkeit eines anderen Geräts zur Berechnung führt dann unweigerlich zur Folgefrage, wie die Brille an das Gerät zur Berechnung angebunden ist, also wie mobil man mit dieser Kombination sein kann und welche möglichen Komfort- und Bewegungsfreiheitseinbußen damit einhergehen. Betrachtet man die beiden häufigsten Kombinationen eines HMD mit einer externen Berechnungseinheiten, so ist es so, dass die Mobilität von HMD in Kombination mit einem Computer zumeist relativ stark eingeschränkt, da zumeist kabelgebunden ist. Bei einer Kombination aus Smartphone und HMD hingegen ist die Mobilität beinahe, bis auf ein Kabel, welches gegebenenfalls als störend empfunden werden kann, als (beinahe) uneingeschränkt zu bezeichnen.

Kommen wir nun zum Schluss noch zum Punkt der Bedienung einer mithilfe einer AR-Brille realisierten Applikation und deren Inhalte am Beispiel der Kombination HMD und Smartphone. Hierbei gibt es zwei Optionen: entweder die Bedienung über das Smartphone, welches in diesem Fall wie ein Controller funktioniert, der dann auch zumeist einen „Strahl“ in den virtuellen Raum sendet mit welchem dann mit den dort befindlichen Objekten und den auf dem Controller vorhandenen Knöpfen/ Eingabeflächen interagiert werden kann, oder direkt über auf dem HMD angebrachten Knöpfe beziehungsweise Gestenerkennung, falls die Brille über Kameras verfügt. Diese Interaktionsmethoden sind zwar alle nicht als hoch komplex zu beschreiben, im Hinblick auf jene einfache Interaktion, wenn die Applikation nur am Smartphone läuft, sind sie jedoch als komplexer und vor allem ungewohnt/ gewöhnungsbedürftig zu deklarieren.

Noch differenzierter gestaltete sich die Betrachtung der Usability von Projection-Based AR, da hier eine dezidierte Unterscheidung zwischen der reinen Nutzung und der Herstellung der Nutzungsbereitschaft gemacht werden muss. Begründet ist dies zum einen in der Grundidee von Projection-Based AR, welche AR-Inhalte im Vergleich zu den anderen beiden Arten einer Vielzahl an Menschen gleichzeitig und nicht nur immer einer einzigen

¹⁰⁹ Bei der Verwendung eines Samsung Galaxy S21 in Kombination mit einer XREAL Air2 ist es beispielsweise so, dass zusätzlich zu der auszuführenden Applikation deren eigene Software „Nebula“ installiert sein muss, um eine Vielzahl an Apps korrekt auszuführen. Ist diese nicht installiert und eine App wurde trotzdem für eine Kombination aus Smartphone und XREAL HMD entwickelt, so ist es an dieser Stelle dann gegebenenfalls nötig, den Anzeigemodus der Brille manuell mittels einer versteckten Tastenkombination umzustellen.

Person¹¹⁰ zur Verfügung stellen soll. Insgesamt ist die reine Nutzung somit, mit der Einschränkung eines (halb-)dunklen Raumes und des Schattenwurfs, je nach Positionierung des Projektors als einfach anzusehen – wobei dies natürlich immer im Hinblick auf die jeweilige Applikation zu betrachten ist. Die Herstellung der Nutzungsbereitschaft hingegen ist aufgrund der Kalibrierungsschritte und der Positionierung, um Schattenwürfe möglichst zu vermeiden, als eher komplex einzustufen ist. Hinsichtlich der Bedienung können an dieser Stelle keine Aussagen zur Bedienung der Applikation an sich gemacht werden, da keine Quellen hierzu existieren. Zur Interaktion mit den durch die Applikation bereitgestellten Inhalten kann jedoch anhand aufgefundener und bereits angemerakter Beispiele gesagt werden, dass hierzu oftmals – da es sich um ein Medium für eine Vielzahl an Personen handelt – eine Anleitung durch eine mit der AR-Realisierung vertraute Person geschieht, oder aber textliche Informationen bereitstehen und die Bedienung aus dieser Sicht somit als unproblematisch zu bezeichnen ist. Interagiert wird bei Projection-Based AR dabei eigentlich immer mit den Händen, indem etwa auf Gesten zurückgegriffen wird, Teile der Projektionsfläche berührt oder Objekte der physischen Umwelt verschoben werden.

Kommunikation von Inhalten

Betrachtet man nun als nächstes, wie Inhalte vermittelt werden können, so unterscheidet sich, wie bereits zuvor erwähnt, zwar die Anzahl an gleichzeitig möglichen Nutzer*innen, erheblich, jedoch nicht die zur Verfügung stehenden Kommunikationswege. Alle drei oberhalb angeführten Arten zur Bereitstellung von AR-Applikationen können sich der visuellen und auditiven, sowie bei Smartphone-AR und je nach Umsetzung auch bei Projection-Based-AR und HMD-AR der haptischen Kommunikation bedienen. Aus diesem Grund wird an dieser Stelle weiters nun auch die Betrachtung der einzelnen Arten der Kommunikation an sich und deren Ausformung bei den einzelnen Geräten folgen. Angefangen wird dabei mit der wohl am wenigsten in diesem Konnex klaren Punkt, der Kommunikation von Inhalten in haptischer Form.

Bei AR am Smartphone, wie auch bei AR auf HMDs, welche ein Smartphone als Berechnungseinheit nutzen, können Hinweise etwa per Vibration (beispielsweise eine nötige Richtungsänderung bei einer Navigationsapplikation) vermittelt werden, was an dieser Stelle weiters als haptisches Feedback zu werten ist. Bei Projection-Based AR gestaltet sich dies etwas abstrakter, da ein haptisches Feedback per se nicht über die AR-Inhalte, sondern über tatsächlich reale, in der Welt vorhandene Objekte vermittelt werden, beispielsweise beim Auftürmen von Sand¹¹¹. Hinsichtlich der auditiven Komponente der

¹¹⁰ Natürlich ist es bei Smartphones möglich, dass mehrere Personen gleichzeitig auf das Display schauen und Inhalte betrachten, jedoch stellt dies nicht die Intention von AR-Applikationen auf Smartphones dar.

¹¹¹ Im engeren Sinne handelt es sich hierbei um keine durch AR (bzw. durch den virtuellen Raum) bereitgestellte Information, sollte aber trotzdem Erwähnung finden, da das Erlebte nicht explizit nur durch die Haptik zustande kommt.

Kommunikation von Inhalten sollte es nicht verwundern, dass alle drei der zu besprechenden Präsentationsarten die Möglichkeit zu Audioausgaben über Lautsprecher bieten.

Interessanter wird es hingegen bei der visuellen Komponente der Vermittlung von Inhalten. Hierbei gestaltet es sich so, dass die Anzahl der möglichen darzustellenden Dimensionen beziehungsweise der Eindruck solcher¹¹² immer von der Anzahl an Bildflächen¹¹³, die sich im Sichtfeld der Nutzer*innen befinden¹¹⁴ oder aber der Ausgestaltung der Projektionsfläche¹¹⁵ abhängen. So können etwa ein Smartphone, wie auch ein monokulares HMD und ein Projektor, welcher auf eine ebene Projektionsfläche projiziert, nur einen 2D Eindruck vermitteln (mit Ausnahme von Verfahren wie etwa Anaglyphenbetrachtung bei Smartphone und Projektor), während binokulare HMDs mittels der Prinzipien der stereoskopischen Bildbetrachtung durch zwei Displays eine 3D-Darstellung realisieren können. Projektoren wiederum können zwar auf dreidimensionale Flächen, also Ebenen mit verschiedenem Abstand zum Brennpunkt, projizieren und somit eine dreidimensionale Darstellung realisieren, jedoch handelt es sich dabei im eigentlichen Sinne nur um eine beliebig große Anzahl an 2D-Projektionsflächen. Außerdem ergibt sich zusätzlich das Problem der Abschattungsbereiche, wodurch keine gänzliche 3D-Darstellung entsteht, durch welche man sich, wie etwa bei HMDs, hindurchbewegen (und beispielsweise hinter einer Wand liegende Bereiche sehen) kann.

Preis

Hinsichtlich des Preises ist es, wie oft in stark technischen Bereichen, der Fall, dass Aussagen zu diesen rasch hinfällig sind¹¹⁶, jedoch soll hier eine kurze Einordnung geschehen. Den günstigsten Einstieg in die AR-Welt stellen sicherlich Smartphones dar, wobei hier je nach Anwendungswunsch recherchiert werden muss, ob das jeweilige Smartphone mit der Applikation kompatibel ist. Als nächst teurere Option stellen sich dann Smart-Glasses und AR/VR-Brillen dar, wobei die Preise hier von einigen hundert Euro bis zu einigen tausend für spezifische Industrieanwendungen reichen. Bei HMDs darf an dieser Stelle darüber hinaus nicht vergessen werden, dass je nach HMD gegebenenfalls auch noch ein extra Gerät benötigt wird, welches die Berechnungen übernimmt. Bezüglich der preislichen Gestaltung von Projektionsgeräten besteht das Problem, dass der ehemalige Marktführer (Anm. LIGHTFORM) in diesem Bereich seit 2022 die Produktion seiner Geräte eingestellt hat. Somit bliebe zur Realisierung von Projection-Based AR (neben sehr

¹¹² In anderen Worten: Der räumliche Gesamteindruck

¹¹³ Siehe hierzu auch: Stereoskopische Bildbetrachtung in Fußnote 91

¹¹⁴ Im Fall von HMDs sind die Bildflächen die projizierten Display-Inhalte.

¹¹⁵ Gemeint ist an dieser Stelle, dass eine flache Wand nur einen zweidimensionalen Eindruck (mit Ausnahme etwa vom Anaglyphenverfahren) vermitteln kann, während mehrere im Raum verschachtelte Projektionsflächen, da tatsächlich Tiefe vorhanden ist ein dreidimensionalen vermitteln.

¹¹⁶ Gründe hierfür sind etwa die Veröffentlichung von neuen Geräten, Fluktuationen im Preis aufgrund von Nachfrage/ Herstellungskosten, Materialkosten und vieles weitere.

kleinen Nischenlösungen für spezifische Anwendungen) nur noch der Weg einer Individuallösung (d.h. Beamer und Kamerasystem) mit Individualsoftware, was preislich nur schwer zu beziffern ist. Hinzu kommt, dass auch hier, ähnlich wie bei manchen HMDs zusätzlich noch ein Gerät zur Berechnung der Darstellung benötigt wird.

Anwendungsfelder

Als Abschluss dieses Unterkapitels soll nun noch ein kurzer Überblick über die möglichen Anwendungsfelder für AR-Darstellungsmöglichkeiten gegeben werden, um sich in weiterer Folge ein genaueres Bild zur Zweckmäßigkeit der einzelnen vorliegenden Optionen zu machen. Beginnen wir dabei nun mit der wie bereits zuvor angesprochenen relativ schwierig zu realisierenden Lösung von Projection-Based-AR mit Projektoren. Hier liegt das Anwendungsfeld hauptsächlich in künstlerischen Umsetzungen und Marketing, aber auch zu kleineren Teilen im Lehrumfeld. Als konkreten Anwendungsfall sei hierzu nochmal auf die beiden bereits zuvor erwähnten Beispiele des Wandgemäldes¹⁰³ und der Sandbox¹⁰⁸ verwiesen.

Betrachtet man als nächstes die AR-Applikationen, welche für Smartphones existieren, so wird klar, dass sich die meisten Anwendungen im Feld der Freizeitgestaltung, des Marketings oder in letzter Zeit auch der Navigation wiederfinden, wobei zu ersterem Bereich an dieser Stelle beispielsweise Spiele wie Pokemon GO und Apps (wie etwa Instagram), die es ermöglichen Live-Filter über Personen zu legen – Stichwort: Frisuren ausprobieren, Schnurrbart hinzufügen, Hundeohren, etc. – zählen. Zur Anwendung im Marketing sei an dieser Stelle direkt auf die Applikation von Ikea, mithilfe derer man Möbelstücke innerhalb der eigenen Wohnung platzieren kann, verwiesen und hinsichtlich der Navigation, die für die vorliegende Arbeit als sehr relevant zu betrachtenden Applikationen wie etwa Google Maps LiveView oder Sygic Real View Navigation erwähnt.

Der Anwendungsbereich von HMDs findet sich momentan noch am stärksten (aber nicht exklusiv) im Militär, der Medizin, der Industrie, bei Schulungen und zuletzt aufgrund der sinkenden Preise und dem Hype rundum Smart-Glasses auch vermehrt im Bereich der Freizeitgestaltung. Als Anwendungsbeispiel hierfür kann beim Militär, dem HMDs grundsätzlich entstammen¹¹⁷, etwa die Markierung feindlicher Objekte festgestellt werden¹¹⁸, während AR-Applikationen mit HMD im medizinischen Bereich etwa bei Operationen¹¹⁹ zur Anwendung kommen. In der Industrie finden AR-HMDs auch noch Anwendung zur Wartung technischer Anlagen oder der Qualitätskontrolle, während in der Freizeitgestaltung etwa Anwendungen zum Spazierenführen von Tieren¹²⁰ oder etwas

¹¹⁷ Die ersten HMDs waren in Helmen von Kampfflugzeugpiloten integriert.

¹¹⁸ Für weitere Informationen hierzu siehe (Morozov, kein Datum; Verma, 2022)

¹¹⁹ Siehe hierzu etwa: (Grobman, 2023)

¹²⁰ Siehe: (konden, 2022)

praktischer der Anzeige von Live-Untertitel¹²¹ gesprochener Sprache festgestellt werden können.

3.4.2 Interaktionsmöglichkeiten in und mit AR-Systemen

Nachdem nun bereits auf die häufigsten Arten, wie AR erlebt werden kann, eingegangen wurde und diese besprochen wurden, folgt nun eine etwas weiterführende Betrachtung der Interaktionsmöglichkeiten mit einer AR-Applikation beziehungsweise mit den durch diese dargestellten Inhalte. Hierzu soll an dieser Stelle das Modell der Kommunikation im Rahmen eines AR-Systems von Ghazwani und Smith angesprochen werden, welches eine Dreiteilung des Kommunikationssystems in **User**, **User-Interface** und **Virtual Content** vornimmt. Der User-Teil wird hierbei nach zwei Kategorien, wie auch bereits oberhalb kurz angesprochen, in **Single-User** (dt. einzelne/r Nutzer*in) beziehungsweise **Multi-User**¹²² (dt. mehrere Nutzer*innen) und weiteres nach deren Nutzungsform in aktive oder passive (eng. **Passive/ Active User**) Nutzung untergliedert. Zweiteres bestimmt hierbei die Art der Kommunikation¹²³, also ob diese bidirektional – AR-System kommuniziert mit Nutzer*in und Nutzer*in kommuniziert mit AR-System – oder nur unidirektional – AR-System kommuniziert mit Nutzer*in – stattfindet. Auf der rechten Seite befindet sich dann weiters, wenn man an dieser Stelle das User Interface überspringt, die Untergliederung in die diversen Arten des Virtual Contents (dt. virtueller Inhalt) danach, ob es sich um offline oder online Inhalte (eng. **offline/ online content**) handelt. Dies bestimmt dabei unter anderem, ob überhaupt die technische Möglichkeit gegeben ist, dass man ein AR-System kollaborativ nutzen kann¹²². Der Grund liegt hierbei darin, dass Inhalte, die nur offline auf einem Gerät vorliegen dementsprechend auch nur von diesem Gerät und dabei zumeist nur einem/r Nutzer*in genutzt werden können: Eine Ausnahme stellt Projection-Based AR dar. Hieran schließen dann natürlich auch noch weitere Punkte, wie die Updatebarkeit von Inhalten und

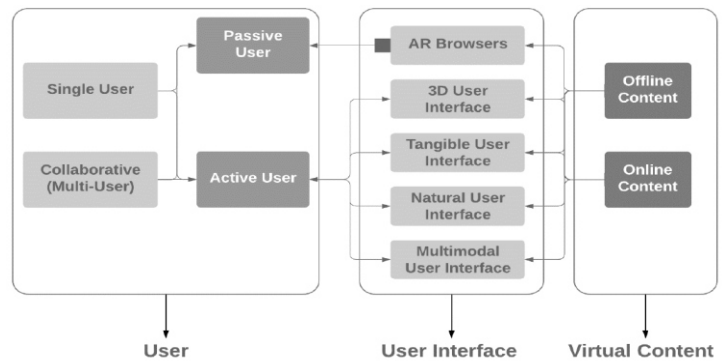


Abbildung 12: Das Kommunikationssystem zwischen User und AR-Inhalten nach (Ghazwani & Smith, 2020)

¹²¹ Siehe: (XRAI, 2022)

¹²² Generell gibt es neben der oberhalb angesprochenen Art der Nutzung virtueller Inhalte von mehreren Personen mittels projektorbasierter AR natürlich auch Möglichkeiten zur multiplen Nutzung virtueller Inhalte, wenn jede Person ihr eigenes Endgerät (HMD, Smartphone) nützt. Hier stellt sich jedoch dann die Frage nach der Übertragung der durch eine/n Nutzer*in im AR-Raum (veränderten) Inhalten mit den jeweils anderen.

¹²³ Hierzu sei erwähnt, dass eine räumliche Veränderung der Position des/r Nutzer*in in diesem Modell nicht als Kommunikation verstanden wird.

ähnliches an, was an dieser Stelle aber keiner weiteren Ausführung bedarf, da es als selbsterklärend angesehen werden kann. (Ghazwani & Smith, 2020)

Im Mittelteil des Modells, welcher nun zu guter Letzt noch betrachtet werden soll finden sich die verschiedenen Typen der Anzeige virtueller Inhalte wieder¹²⁴, wobei die Autoren hier fünf Typen unterscheiden. Es handelt sich dabei um vier Typen mit bidirektionaler Kommunikation – **3D-**, **Tangible-**, **Natural-**, **Multimodal-User Interface** – und einem, namentlich dem Typ des **AR-Browsers** mit unidirektionalem Interface. Der **AR-Browser** Typ stellt sich dabei als einfachste (nicht interaktive) Form des User-Interfaces, wobei nur Inhalte dem Nutzer präsentiert werden, jedoch nicht mit diesen interagiert werden kann dar¹²³. Hierdurch ergeben sich jedoch Problematiken hinsichtlich einer gelungenen funktionalen Realisierung, wie etwa die Größe der anzuzeigenden Inhalte, die mögliche Überfüllung der Sicht und so weiter. Als einfachstes Beispiel zu diesem Typ seien AR-Navigationsapplikationen wie Googles Live View oder Sygic Real View Navigation¹²⁵ genannt. Von einem **3D-UI** wiederum wird gesprochen, wenn sich das User Interface selbst im 3D-Raum befindet und ein eigenes Gerät zur Ansteuerung desselben und der Veränderung¹²⁶ der Objekte im 3D-Raum verwendet wird: Hier wird von drei Typen der Veränderung, nämlich der Auswahl, Manipulation und Navigation gesprochen. Eine etwaige Möglichkeit hierzu stellen eine Vielzahl diverser Eingabegeräte, die am Ende dieses Kapitels angeführt werden, dar. **Tangible-UI**, erstmals von Kato et al. in deren Arbeit zu „*Virtual object manipulation on a table-top AR environment*“ (Kato, et al., 2000) aufgezeigt basieren darauf, dass virtuelle Objekte auf physischen Objekten angezeigt werden. Etwaige Beispiele hierzu sind die Verkaufsautomaten von LEGO, bei welchem das fertig zusammengebaute Objekt direkt auf der LEGO-Verpackung angezeigt wird und sich je nach Bewegung des Kartons auf diesem mitbewegt¹²⁷. **Natural-UIs** wiederum setzen auf das Prinzip des Trackings von Bewegungen, um mit den virtuellen Inhalten zu interagieren: Hierzu wird unterhalb ebenso wie zu Eingabegeräten noch ein eigener Punkt angeführt werden. Der letzte Typ, der **Multimodal-UIs**, weist, wie der Name bereits vermuten lässt, schlussendlich verschiedene Eingabearten in verschiedenartiger Kombination auf. (Ghazwani & Smith, 2020)

Eingabegeräte: Die einfachste und bekannteste Form der Interaktion stellt an dieser Stelle, wie bei einer jeden Software, sicherlich die Interaktion über ein Eingabegerät dar: etwa ein Touchpad, ein Touchscreen, eine Maus, eine Tastatur, ein Controller, ein Smartphone oder ein eigens hierfür entwickeltes Gerät. Da die ersten Arten der Eingabegeräte ohnehin

¹²⁴ Hinsichtlich dieses Teils des Modells sollte (in kritischer Weise) darauf hingewiesen werden, dass bis zu einem gewissen Grad eine Vermischung der Teile des User-Interfaces, welche der Bedienung und jenen, welche der Präsentation von Inhalten dienen, geschieht.

¹²⁵ Für Google Live View siehe etwa: (Cantisano, 2022; Hall, 2023; Kasanmascheff, 2021) und für Sygic Real View Navigation: (Sygic, 2024; Sygic a.s., 2017)

¹²⁶ Unter Veränderung werden in diesem Sinne die Prozesse der Auswahl oder Veränderung eines Objekts und die Navigation innerhalb des 3D-Raums verstanden. (Bowman, et al., 2004)

¹²⁷ Siehe hierzu: (Klaß, 2010)

bekannt sein sollten, sich aber die Leserschaft gegebenenfalls fragt, was mit den beiden letzteren Arten gemeint sein soll, sei hierzu eine kurze Erläuterung gegeben: Dadurch, dass Smartphones oftmals in Kombination mit HMD zur Realisierung eines AR-Systems eingesetzt werden und die Position des Geräts hierbei in Relation zum HMD bekannt ist, wird es möglich, beim Smartphone neben dem Touchscreen auch weitere virtuell hinzugefügte Funktionen, wie beispielsweise eine Pointerfunktion¹²⁸ zu implementieren, welche bei Berührung des Pointers mit einem Objekt im virtuellen Raum (beispielsweise auf Knopfdruck am Touchscreen oder auch automatisch) eine gewisse Interaktion an selbigem durchführt. Zum zweiten Punkt, der soeben kryptisch als „eigens hierfür entwickelte Geräte“ bezeichnet wurde, sei an dieser Stelle darauf hingewiesen, dass es gerade zu Beginn der 2010er Jahre (etwa erste Versuche mit haptischen Gegenständen¹²⁹), aber insbesondere auch heute Geräte entwickelt werden, die dezidiert der Eingabe und gleichzeitig der haptischen Kommunikation dienen. Das bekannteste Beispiel hierzu stellen an dieser Stelle sicherlich verschiedene Arten von Handschuhen¹³⁰, welche zumeist bei VR-Systemen zum Einsatz kommen, die nichtsdestotrotz aber auch für die Zwecke von AR genutzt werden könnten. Das Prinzip dahinter ist das gleiche wie bei der Nutzung des Smartphones als „Schaltzentrale“ zur Bedienung, da es hierbei rein darum geht, dass die jeweilige Applikation das Gerät insofern richtig „interpretieren“ kann, als dass es die Position des selbigen kennt und die einzelnen Kommandos (bspw. auf etwas zeigen, eine Faust machen, ...) richtig interpretiert.

Sprachbefehle: Die zweite Möglichkeit zur Interaktion mit einer AR-Applikation bezieht sich auf Sprachbefehle, wie sie mittlerweile bei fast allen Smartphones und Autos zur Verwendung kommen. Voraussetzung hierfür ist an dieser Stelle nur, dass das Gerät, welches zur Realisierung des AR-Systems eingesetzt wird, über ein Mikrofon verfügt und die Applikation die Befehle versteht. Im Vergleich zur Nutzung von Eingabegeräten handelt es sich hierbei jedoch um eine hinsichtlich der Funktionen limitierte Interaktionsform. So wäre es beispielsweise möglich, per Sprachbefehl noch möglich ein einziges im virtuellen Raum befindliches Objekt zu platzieren und zu verändern. Existieren jedoch eine Vielzahl an ähnlichen oder gleichen Objekten, gestaltet es sich relativ rasch schwierig, das gewünschte Objekt auszuwählen, da diese dann nur schwer beschreib- und unterscheidbar wären.

Tracking von Gesten/ Augen: Die dritte und an dieser Stelle sicherlich neuartigste und sehr interessante Art zur Interaktion ist das Tracking von Gesten beziehungsweise das Tracking von Kopf- und Augenbewegungen. Zu ersterem gibt es an dieser Stelle grundsätzlich zwei Ansätze, wovon es sich bei der ersten Methode einfach um die Verwendung von eigens dafür verwendeten Eingabegeräten handelt, etwa in Form von Handschuhen zum Tracking

¹²⁸ Bei der Pointerfunktion wird mittig vom Smartphone ausgehend ein Strahl in den virtuellen Raum gesendet.

¹²⁹ Als Beispiel hierzu sei etwa auf die Anwendung von *Phantom*, einem Produkt bestehend aus HMD, Stift mit haptischem Feedback und Basisstation von *Magic Lab*, welches in der Arbeit von Rampolla und Kipper erwähnt wird, hingewiesen. (Rampolla & Kipper, 2012, pp. 46-47)

¹³⁰ Eine kurze Übersicht zu solchen Geräten ist beispielsweise auf der Website von „Unbound XR“ unter dem Titel „Top 6 Virtual Reality (VR) Gloves“ zu finden. Siehe hierzu: (Unbound XR, kein Datum)

der Position und Bewegung der Hände. Die zweite Methode, welche auch beim Tracking von Kopf- und Augenbewegungen angewendet werden kann, ist das Tracking mithilfe einer Kamera/ eines Kamerasystems. Um an dieser Stelle technisch nicht zu sehr auszuschweifen, sei zu dieser Technik nur kurz erwähnt, dass das dahinterliegende Prinzip zur Realisierung eines solchen Trackings auf Algorithmen basiert, welche Hände und Finger beziehungsweise beim Augentracking die Augen und beim Kopftracking den Kopf erkennen und deren Bewegungen nachverfolgen und interpretieren können. Als einfaches Beispiel kann sich an dieser Stelle etwa eine Wischbewegung mit der Hand in der Luft vorgestellt werden, mithilfe derer man ein Buch im virtuellen Raum umblättert. Generell bietet diese Form der Kommunikation sehr großes Potential, da es hierdurch möglich wird, einfach ohne jegliche Eingabegeräte mit Inhalten in der virtuellen Welt potentiell wie in der realen Welt zu interagieren. Als Beispiel hierzu sei sich hier etwa eine Einrichtungsapp, realisiert auf einem HMD vorgestellt, bei welchem ein platzierter Stuhl einfach „angefasst“ und verschoben werden kann: etwa indem beide Hände, wenn man sich vor dem Stuhl befindet, zur Faust geballt werden.

3.5 Entwicklung von AR-Applikationen

Nachdem nun in den vorangegangenen Kapiteln die Möglichkeiten besprochen wurden, wie AR-Systeme aufgebaut sind, wie sie unterteilt werden können sowie in welcher Form deren Inhalt dargestellt und mit ihnen interagiert werden kann, folgt an dieser Stelle nun noch, aufgrund der Fassbarmachung und Abschätzbarkeit des Aufwands zur Erstellung einer AR-Applikation für Geoinformatiker*innen und der in Kapitel 5 anstehenden eigenen praktischen Umsetzung einer AR-Applikation für die Navigation im Zustellungsbereich, ein kurzer genereller Blick auf die Entwicklung von AR-Apps. Bezüglich der Entwicklung einer (AR-)Applikation, egal für welchen Aufgabenbereich, ergibt sich eine Fülle an Fragen, welche von den Programmierkenntnissen hin zur zu verwendenden Software und den Zielgeräten, das heißt der Hard- und auch Software, reichen. Um diese Fülle an Fragen herunterzubrechen ist, es in erster Linie einmal nötig, ein Konzept auszuarbeiten, welches sich um die Frage dreht, was das Ziel der jeweiligen Applikation und der gewünschte Funktionsumfang sein soll, sowie welche Geräte und Software hierfür verwendet werden können/ müssen.

Meist gelangt man an dieser Stelle bereits zu einem Prozess des Abwägens, ob etwas und wenn ja, wie und mit welchem Aufwand es umsetzbar ist. Dieser Prozess wird dabei vor allem von den Faktoren des eigenen Wissensstandes (hinsichtlich Programmierung, Umgang mit Programmen, ...), den zur Verfügung stehenden Geräten und Software beeinflusst: einerseits in Hinblick auf die Erstellung der Applikation, andererseits in Hinblick auf die Ausführung der Applikation. Bei den allermeisten Umsetzungen ist das Ziel natürlich, eine im Rahmen desselben in Bezug auf die Funktionen möglichst umfangreiche und auf einer möglichst breiten Palette an Geräten ausführbare Applikation zu entwickeln. Gerade letzterer Punkt jedoch ist dabei oftmals durchwegs komplex, da

Interdependenzen¹³¹ stark zunehmen, insbesondere wenn mehrere verschiedene Geräte – wie beispielsweise Smartphone und HMD – zusammenspielen. Hier schließt dann direkt auch der Punkt an, wie für welche(s) Gerät(e) (mit welchen Betriebssystem(en)) auf welcher Plattform entwickelt werden soll und ob für die zu verwendenden Geräte auf der für die zur Entwicklung geplanten Plattform (plus in dieser verwendete Programmiersprache) SDKs existieren, da hiervon natürlich der Grad der Komplexität abhängt. Hinzu kommt dann noch, dass, falls SDKs für Geräte existieren, diese oftmals nur für eine Plattform angeboten werden und diese dadurch wiederum zum Teil (mit-)bestimmen. Denkt man dies alles an dieser Stelle am Beispiel der Entwicklung einer einfachen mobile-AR App (etwa für das Überlagern eines Gesichts mit Gesichtsfilt­ern) für Smartphones mithilfe von Unity durch, so zeigt sich, dass sowohl ein SDK für iOS, als auch Android zur Verfügung stehen, diese eine gute Dokumentation aufweisen und es somit mit etwas Expertise ohne größere Probleme möglich ist eine auf beiden lauffähige Applikation zu entwickeln.

Problematisch wird es hingegen, wenn es darum geht, weitere Geräte, wie etwa HMDs miteinzubinden. Hier stellt sich als erstes einmal die Frage, ob überhaupt SDKs für die HMDs zur Entwicklung zur Verfügung stehen und wie es bezüglich Interdependenzen zwischen Smartphone und HMD aussieht – das heißt, ob das HMD überhaupt mit Smartphone X kompatibel ist. Insgesamt ist es im Optimalfall so, dass sich über diese Punkte bereits im Vorfeld zur praktischen Umsetzung Gedanken gemacht wurde und man bereits auf der korrekten Plattform und mit (miteinander kompatiblen) Geräten arbeitet, für welche SDKs zur Verfügung stehen. Der Punkt der miteinander kompatiblen Geräte ist an dieser Stelle leider, selbst wenn SDKs zur Verfügung stehen nicht immer direkt einfach erfüllbar, da es oftmals so ist, dass nur eine unzureichende Dokumentation seitens der Gerätehersteller bezüglich Kompatibilitäten zur Verfügung gestellt wird. Der Grund dafür liegt darin, dass die Kompatibilität zwischen Geräten zumeist nicht nur vom Betriebssystem abhängt und somit die Testung, welche Kombinationen dann tatsächlich funktionieren, rasch sehr komplex, wie auch zeit- und kostenintensiv wird.

Abschließend zur Entwicklung von Applikationen sei noch erwähnt, dass es natürlich auch möglich wäre, eine Applikation ohne jegliche Zuhilfenahme einer Plattform für die Entwicklung und SDKs rein mithilfe programmatischer Kenntnisse zu erstellen – solange es sich nicht um proprietäre Systeme handelt. Jedoch wäre dies mit einem enormen Aufwand verbunden (da alle Grundlagenfunktionen selbst erstellt werden müssten), weshalb dies in der Regel nicht gemacht wird.

Nachdem also nun bereits kurz über die generelle Entwicklung von Applikationen und in den vorangegangenen Kapiteln über die meist bei AR verwendeten Geräte wie auch die verschiedenen AR-Typen gesprochen wurde, folgt nun die Betrachtung eines exemplarischen Workflows – anhand des Workflows¹³² von Gerasymov im Artikel „9-Step

¹³¹ Mit Interdependenzen ist dabei gemeint, ob Gerät A mit Gerät B (bzw. selbiges für Softwarekomponenten) kompatibel ist.

¹³² An dieser Stelle sei angemerkt, dass ein Workflow selten bis gar nie linear, wie in der Abbildung archetypisch dargestellt, abläuft, sondern immer wieder Feedbackschleifen enthält, die es nötig machen, vorher bereits definierte

Augmented Reality App Development Guide“ (Gerasymov, 2024) – für die Erstellung einer AR-Applikation.

Bei der Erstellung ist es in einem ersten Schritt einmal nötig, die generellen Projektanforderungen zu definieren und sich grundlegend über mögliche für die Umsetzung heranziehbare Geräte und Software zu informieren. Im nächsten Schritt sollten sich Gedanken zum zu wählenden Entwicklungszugang und den Technologien (eng. Development approach and technologies), wie von Gerasymov genannt, gemacht werden, wobei diese ihrerseits wieder die möglichen zu verwendenden Geräte mitbestimmen.

Für AR-Applikationen stellen sich hier insgesamt vier wählbare Zugänge dar:

Der **Native**, **Cross-platform**, **Hybrid** und **web-AR** Zugang. Der **Native** Zugang besagt dabei, dass die Entwicklung direkt für spezifische Geräte und deren Funktionen ohne Zuhilfenahme einer Plattform für die Entwicklung oder SDKs rein über Code stattfindet: Dies bedeutet ein mögliches Plus im Hinblick auf Performanz, aber ein großes Minus im Aspekt Aufwand. Der **Cross-platform** Ansatz wiederum ermöglicht es, direkt für eine Vielzahl verschiedener Geräte zu entwickeln, indem eine Software als Plattform mit verschiedenen SDKs zur Erstellung genutzt wird – im Bereich von AR ist dies oftmals Unity mit den Basis-SDKs für iOS (ARKit)/Android (ARCore) und vielen weiteren. Dies bringt ein Plus in Hinblick auf die Anzahl an Geräten, für welche relativ einfach entwickelt werden kann, aber ein mögliches Minus im Aspekt der Performanz. Beim **web-AR** Ansatz wiederum wird die Applikation direkt fürs Internet, also den Browser entwickelt – was ein Plus in Hinblick auf die zu nutzenden Geräte, solange der Browser unterstützt wird, aber ein Minus im Aspekt der Performanz und der offline-Nutzung mit sich bringt. Beim **Hybrid** Zugang schlussendlich handelt es sich um eine Mischung aus web-AR und Cross-platform Zugang, in dem Sinne, dass eine Applikation fürs Internet entwickelt und darauf aufbauend eine App, welche sich auf die Webapplikation verbindet (mit allen zuvor genannten Vor- und Nachteilen) erstellt wird. (Gerasymov, 2024)

9-STEP AR APP DEVELOPMENT GUIDE

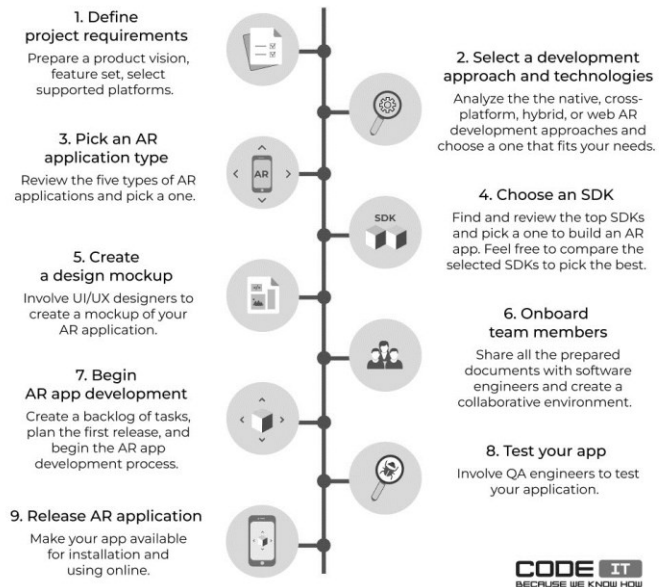


Abbildung 13: Ablauf der Erstellung einer AR-Applikation (verändert nach (Gerasymov, 2024))

Sachverhalte anzupassen. Ebenso ist es möglich, dass manche Schritte bei diversen Umsetzungen umstrukturiert werden müssen oder gar nicht vorkommen.

Nachdem diese essentiellen Schritte erledigt wurden, folgt dann logischerweise noch die Frage¹³³, um welche Art¹³⁴ von AR-Applikation, es sich handeln soll und die Auswahl der weiters hierfür benötigten SDKs, falls dies nicht ohnehin durch die Auswahl der zu verwendenden Geräte/ die zu verwendende Plattform vorgegeben ist¹³⁵. Schlussendlich folgen dann noch designtechnische Überlegungen, die Aufstellung des Teams zur Umsetzung, die eigentliche Entwicklung und das Testing – wobei es bei letzteren beiden immer wieder zu Korrekturschleifen kommen kann und vorangegangene Schritte nochmals überdacht werden müssen. Zu guter Letzt erfolgt dann die Veröffentlichung der Applikation.

Nachdem also nun einige Worte hinsichtlich der Entwicklung einer AR-Applikation verloren wurden, sollen im Anschluss einige Softwarepakete, welche die Entwicklung von ebensolchen ermöglichen, besprochen werden. Der Blick wird an dieser Stelle auf den Cross-platform-Zugang beschränkt werden, da dieser oftmals den einfachsten, schnellsten und vermutlich auch beliebtesten Weg zur Erstellung einer AR-Applikation darstellt. Grundsätzlich gestaltet sich das Feld der verwendbaren Software, welche als Plattform zur Umsetzung und Einbindung von SDKs genutzt werden kann, als relativ überschaubar, wobei die Optionen je nach Anwendungsgebiet von kostenfreien online-Applikationen, welche keine Programmierkenntnisse erfordern und die gängigsten SDKs bereits eingebunden haben, bis hin zu kostenpflichtigen offline, stärker auf Programmierkenntnisse fußenden Softwarepaketen reichen, wo benötigte SDKs selbst hinzugefügt werden müssen.

Um an dieser Stelle nicht unnötig viele verschiedene solcher, auch immer wieder neu auf den Markt kommender Softwarepakete aufzuzählen seien an dieser Stelle Unity als Platzhirsch, aber auch Unreal als vollumfängliche offline Lösungen auf der einen Seite und Lösungen wie mywebar oder EasyAR als einfache online Lösungen auf der anderen Seite erwähnt. Natürlich gibt es darüber hinaus noch Lösungen mit teils sehr spezifischen Fokussen, wie etwa Vossle für webbasierte AR-Applikationen, ScopeAR für Arbeitskraftschulungen, ViewAR für Location-based AR ohne Programmierkenntnisse und eine Vielzahl weiterer¹³⁶. An dieser Stelle soll diese Aufzählung auch enden, da wie erwähnt ohnehin immer neue Lösungen auf den Markt kommen und keine unnötig lange Liste spezifischer (Insel-)Lösungen angeführt werden soll¹³⁷. Viel eher soll an dieser Stelle noch kurz auf einige grundlegende Konzepte, die in weiterer Folge zum Verständnis der weiteren

¹³³ Insofern dies nicht direkt zu Beginn durch das zu erreichende Ziel vorgegeben wurde.

¹³⁴ Siehe hierzu: Kapitel 3.3.1

¹³⁵ Natürlich kann auch der umgekehrte Weg erfolgen: Gemeint ist hiermit, dass man zuerst eine für die Projektidee passendes SDK sucht und demnach das/ die Gerät(e) zur Umsetzung auswählt, welche eben jene unterstützen.

¹³⁶ Für die einzelnen erwähnten Lösungen siehe deren Websites: (Entertainment LLC, 2023; VisionStar Information Technology, 2024; Vossle, 2024; ScopeAR, 2024; ViewAR, 2024)

¹³⁷ Für eine etwas umfangreichere Auflistung an Augmented-Reality Software sei an dieser Stelle noch auf jene von Paulo Gardini Miguel in seinem Artikel „An Expert’s Guide To The 12 Best Augmented Reality Software Of 2024“ (Gardini Miguel, 2024) verwiesen, wo er detailliert auf die Möglichkeiten einzelner Softwarepakete, als auch deren Vor-/ und Nachteile und Preise eingeht.

Arbeit nötig sein werden, besprochen werden und auf Unity – aufgrund dessen Führungsposition im Bereich der Entwicklung von AR-Applikationen –, wie auch SDKs, welche zum Standardrepertoire im Hinblick auf die Entwicklung von AR-Applikationen zählen, und auf drei spezifische, im Rahmen der weiters durchzuführenden praktischen Arbeit essentielle SDKs eingegangen werden. Ebenso soll in kurzer, möglichst einfacher Weise das Grundprinzip, wie eine AR-Applikation funktioniert, dargelegt werden, um die Brücke zwischen weniger mit Realtime-Engines versierten Menschen und Menschen, welche über ein weiterführendes Verständnis für diese besitzen, zu schlagen.

3.5.1 Grundlegendes zu Unity und ausgewählten (AR-)SDKs/ Erweiterungen

Bei Unity handelt es sich, wie bereits erwähnt, um eine Realtime- oder auch Game-/ 3D-Engine, welche der Realisierung (interaktiver) zweidimensionaler oder dreidimensionaler Inhalte¹³⁸ dient, wobei sich Unity genau an dieser Stelle zum einen durch den Fakt, dass die gerenderten Bilder ohne Verzögerung angezeigt werden und durch die Möglichkeit der Interaktivität während der Laufzeit von 3D-Software wie Blender abgrenzt: bei letzterem kann immer nur eine gewisse Szene mit Verzögerung gerendert¹³⁹ werden, wobei das im Renderprozess entstehende Bild/ Video, da nicht interagierbar, als statisch anzusehen ist. Im Rahmen dieser Arbeit werden Softwarepakete wie Unity und Unreal als Realtime-Engines angesprochen, da sich der eigenen Recherche und dem eigenen Wissen folgend in den letzten Jahren eine starke Tendenz hin zur Nutzung dieses Begriffs, weg vom Begriff der Game-Engines, in viele wissenschaftliche wie auch wirtschaftliche Bereiche vollzogen hat. Der Begriff 3D-Engine ist überdies als etwas unglücklich zu bezeichnen, da das Alleinstellungsmerkmal dieser Softwarepakete die direkte Anzeige von erstellten Inhalten ohne Wartezeit (eben in „realtime“) darstellt und der Begriff der 3D-Engine schnell in einen Topf mit 3D-Software/ 3D-Grafiksoftware wie etwa Blender geworfen werden könnte.

Die Funktionsweise einer Realtime-Engine kann man sich dabei vereinfacht gesagt, wie jene einer üblichen 3D-Software wie Blender vorstellen: mit dem Unterschied, dass eine Vielzahl von Bildern und das in extrem kurzer Zeit live berechnet und ausgespielt werden, sodass ein flüssiges Bild entsteht. Man spricht in diesem Kontext von Frames per Second (dt. Bilder pro Sekunde) wobei für ein flüssiges Bild in unterschiedlichen Kontexten von unterschiedlichen Werten zumindest aber von 24, wie etwa bei Filmen (30-60 bei Computerspielen) gesprochen wird.

¹³⁸ Die anzuzeigenden Inhalte werden dabei entweder auf einer Ebene, auf der die Blickrichtung der Kamera (im Zentrum) im 90° Winkel steht, beziehungsweise in einem dreidimensionalen Raum, wo die Kamera frei positioniert werden kann, angeordnet.

¹³⁹ Rendern bedeutet in diesem Kontext, dass ein Bild aus den im Programm erstellten dreidimensionalen Raum berechnet und ausgegeben wird.

Grundlegendes zu Unity

Während oberhalb nun stark vereinfacht und schemenhaft auf RE eingegangen wurde, ist es aufgrund der Fokussierung und der später stattfindenden Besprechung der eigenen praktischen Arbeit nötig einige Begrifflichkeiten und damit verbundene Konzepte kurz anzusprechen. An dieser Stelle sei mit der Besprechung einer Szene begonnen. Eine **Szene** beschreibt im vereinfachten Sinn ein Behältnis, einen zwei-/ dreidimensionalen Raum in welche Inhalte (sogenannte **Objekte** (eng. Game-Objects), wie bspw. 3D-/2D-Modelle; oder **Komponenten** (eng. Components), wie bspw. zu Objekten hinzugefügte Skripte) eingefügt und dargestellt werden können. Ein Spiel kann dabei aus einer Vielzahl aus Szenen bestehen, welche sich zwar Grundfunktionen teilen können, einander aber nicht zwangsläufig beeinflussen. Sollen sich verschiedene Szenen jedoch gegenseitig in irgendeiner Weise beeinflussen (im Sinne: Wenn in Szene A Situation X auftritt, dann soll in Szene B gewechselt werden), so müssen diese miteinander verbunden werden. Ein aus der Game-Branche relativ berühmtes Beispiel ist das Verbinden zweier Szenen, oder auch Levels mittels einer eigenen einfachen Szene¹⁴⁰, sodass dem/r Nutzer*in kein Ladebildschirm beim Übergang von einer in die andere Szene angezeigt werden muss.

Jede Szene muss überdies über mindestens eine¹⁴¹ **Kamera**¹⁴² verfügen, welche in diesem Sinne, wie eine (Video-)Kamera in der echten Welt einen Ausschnitt derselben, aufnimmt: Inhalte einer Szene werden aufgenommen und für den/die Nutzer*in angezeigt, wobei sich zum einen die Inhalte der Szene verändern können und die Kamera dies abbildet, oder aber sich die Kamera selbst etwa in Bezug auf ihre Blickrichtung und ihre Position verändern kann.

Generell spricht man in Hinblick auf eine Szene und den zur Verortung von Objekten verwendeten Zahlenwerten in X, Y, Z Richtung vom **Unity-Space** beziehungsweise **Unity-Koordinaten(-system)**, wobei dieser, auch bei einer AR-Applikation eigentlich nie (mit Ausnahme der Einbindung der ArcGIS for Unity SDK) einem Real-World Koordinatensystem, etwa WGS84 oder UTM entspricht. Dies ist grundsätzlich nicht auf technische Limitationen in diesem Bereich, sondern unter anderem auf das inkrementelle Laden und die Unsinnigkeit, für die meisten Anwendungen direkt zu Beginn eine unnötig groß angelegte Szene zu erstellen, zurückzuführen¹⁴³.

¹⁴⁰ Ein Beispiel hierfür ist, dass beim Betreten eines Gebäudes von der Straße aus (Szene 1) ein Aufzug („Lade“-Szene) genutzt werden muss um in einen Raum (Szene 2) zu gelangen.

¹⁴¹ Die Anzahl der Kameras in einer Szene ist rein theoretisch unbegrenzt und ermöglicht somit unterschiedliche Blickwinkel, Anzeige von komplett unterschiedlichen Inhalten und so weiter.

¹⁴² Eine Kamera verfügt über ein virtuelles „Ohr“, über welche Tonausgaben, welche von Objekten in einer spezifischen Szene ausgehen, an den/die Nutzer*in ausgegeben werden. Auch dies verhält sich demnach wie bei einer Videokamera mit Mikrofon.

¹⁴³ An dieser Stelle stellt sich für Geograph*innen natürlich die erwähnte ArcGIS SDK (esri, 2023a) als besonders interessant dar.

Blicken wir nun noch etwas genauer auf die Realisierung der Umsetzung der Anzeige in Echtzeit berechneter und interagierbarer Inhalte, so ist es nötig, sich mit der sogenannten „**Execution-Order**“ (dt. Ausführungsreihenfolge) auseinanderzusetzen, also in welcher Reihenfolge bestimmte Dinge – in diesem Sinne Abschnitte von Code – erledigt werden. Ohne an dieser Stelle detailliert auf diese einzugehen¹⁴⁴, sei erwähnt, dass sich diese in drei große (Haupt-)Teile teilt: in das, was während des Starts, während der Laufzeit und bei Beendigung durchgeführt wird. Die wichtigsten hierfür bereitgestellten Funktionen sind in Unity die „Start“ Funktion, worin Code vorzufinden ist, welcher wenn ein Skript aufgerufen ausgeführt wird, und die „Update“ Funktion, in welcher sich Code befindet, welcher während jedes Frames durchgeführt wird. Die „OnDisable“ bzw. „OnDestroy“ Funktion hingegen wird nur selten genutzt.

Sprechen wir nun noch kurz über die in Unity vorhandenen Bestandteilen zur Befüllung einer Szene, so muss man an dieser Stelle einmal definiert werden, welche Bestandteile eigentlich existieren. Insgesamt gibt es zwei Arten von Bestandteilen, mit welchen eine Szene befüllt werden kann: **Objekte** und **Komponenten**, welche Objekten zugewiesen werden und bestimmte Funktionen ausführen. **Objekte** können hierbei alles vom simplen Würfel, welcher bei der Erstellung keine spezifische Komponente¹⁴⁵ aufweist, hin zu sogenannten „**Prefabs**“ (dt. vorgefertigtes Objekt) darstellen, welche komplexere für einen spezifischen Zweck erstellte Objekte – beispielsweise ein komplettes Menü mit allen für die Funktionsfähigkeit zugewiesenen Skripten. **Komponenten** sind wiederum veränderbare Bestandteile (bspw. zur Änderung der Farbe, Größe, ...) eines Objekts. Hierzu zählen auch sämtliche Objekten zugewiesenen **Skripte** – in Unity C# Code –, wobei diese gemäß der *execution order* ausgeführt werden, wenn das Objekt, welchem das Skript zugewiesen ist, aktiviert wird.

Ausgewählte SDKs/ Erweiterungen

Gehen wir an dieser Stelle nun dazu über, einige (für die Arbeit/ für AR im Allgemeinen) essentielle Erweiterungen/ SDKs zu besprechen, so kann man hinsichtlich der vorangegangenen Ausführung zu Objekten, Prefabs und Komponenten an dieser Stelle perfekt mit der **TextMeshPro** (kurz: TMP oder TMPPro) Erweiterung beginnen. Hierbei handelt es sich um eine Erweiterung, welche eine Vielzahl an Prefabs, die sich mit der visuellen textlichen Kommunikation zum/r Nutzer*in beschäftigen, wie etwa Textfelder zur Anzeige/ Eingabe, Dropdown Menüs, Buttons, etc bereitstellt. Zur Funktionsweise muss angemerkt werden, dass die Grundfunktionen von Objekten, wie die Abfrage, ob ein Knopf gedrückt wurde, nicht selbst geschrieben werden muss, sondern direkt implementiert ist.

¹⁴⁴ Für die detaillierte Abfolge der *execution order* von Code in Unity siehe: (Unity, 2024)

¹⁴⁵ Grundsätzlich besitzt jedes Objekt immer Komponenten (etwa zur Zuweisung der Farbe, Abfrage der Größe oder der Bounding Box). Diese sind zumeist aber so weit als essentiell zu bezeichnen, als dass das Objekt ohne diesen nicht existieren würde. Darum wird in diesem Fall von keinen „spezifischen Komponenten“ gesprochen. Unter „spezifischen Komponenten“ wird in diesem Sinne verstanden, dass es sich beispielsweise um Skripte handelt, welche eine gewisse Funktion abseits der Grundfunktionen zur Verfügung stellen.

Dabei ist es dann, um beim Beispiel des gedrückten Buttons¹⁴⁶ zu bleiben, auch direkt möglich, auf einen eigens geschriebenen Codeblock zu verweisen, welcher beim Buttondruck ausgeführt werden soll.

Für die Entwicklung von AR-Applikationen als besonders relevant und darum auch im Unity Template¹⁴⁷ für die Entwicklung einer AR-Applikation enthalten sind zudem **ARCore**, **ARKit**, **Vuforia** und **ARToolkit** anzusehen. Die ersten beiden erwähnten dienen dazu, diverse Funktionen, wie etwa die Handkamera, GPS & IMU und vieles mehr, von Smartphones mit Android Betriebssystem bei ARCore und iOS Betriebssystem bei ARKit nutzbar zu machen und stellen darüber hinaus teils fertige Funktionen für die Bewegung durch den AR-Raum etwa mittels Nutzung der GPS Antenne und der IMU zur Verfügung. Der Unterschied zwischen diesen beiden SDKs ist dabei, dass diese im Gegensatz zu den letzteren beiden direkt von den Softwareentwicklern, respektive Google und Apple, jeweils für deren Betriebssystem bereitgestellt werden, wohingegen es sich bei Vuforia und dem ARToolkit um von dritter Seite entwickelte SDKs für eine Vielzahl von Plattformen handelt: eben auch Android und iOS mit einer großen Bandbreite an Funktionen. Welche Erweiterung weiters für die Umsetzung eines Projekts herangezogen werden soll, hängt dabei zum einen von der Frage des Betriebssystems, für welches eine spezifische Funktion umgesetzt werden soll, ab und zum anderen, ob das jeweilige SDK die dafür benötigten Funktionen bereithält, oder ob diese ohnehin selbst erstellt werden müssen. Weiters spielt natürlich oft auch das eigene Wissen im Umgang mit gewissen SDKs eine nicht unwesentliche Rolle.

Da nun bereits einige grundlegende AR-SDKs für Unity in aller Kürze angesprochen wurden, sollen hier nun noch kurz für die vorliegende Arbeit relevante und für die praktische Umsetzung ausgewählte SDKs erwähnt werden. Insgesamt handelt es sich an der Stelle neben der ARCore Erweiterung um drei zusätzliche SDKs. Erstere, die kostenfreie Erweiterung „**NRSDK**“ (von XREAL) (XREAL, kein Datum) dient dabei der Nutzbarmachung der Funktionen aller von XREAL hergestellter und auch jener im Rahmen des Projekts genutzten AR-Brille. Zweites und drittes SDK sind dann die sogenannte kostenfreie „**Maps SDK for Unity**“ von Mapbox¹⁴⁸ (Mapbox, 2019) und die kostenpflichtige „**AR + GPS Location**“ Erweiterung (kurz: ARGPS) von Daniel Fortes (Fortes, 2020). Die beiden letzteren sind dabei eng miteinander verknüpft, da eine Vielzahl der ARGPS Funktionen auf der Mapbox SDK aufbauen und Funktionen eben dieser genutzt werden. Insgesamt ist es bei der NRSDK- Erweiterung, wie auch bei der ARCore-Erweiterung, so, dass diese hauptsächlich der Nutzbarmachung der Funktionen eines Geräts, in diesem Fall der AR-Brille dienen und nebenher noch einige Prefabs für die leichtere Umsetzung von Projekten bereitgestellt werden. Die Mapbox-SDK wiederum dient ihrerseits hauptsächlich

¹⁴⁶ Die Funktionsweise ist bei anderen Objekten hier natürlich ähnlich.

¹⁴⁷ Ein Template ist in diesem Sinne ein Projekt, welches mit gewissen installierten Erweiterungen und definierten Parametern erstellt wird.

¹⁴⁸ Die mapbox-SDK kann insofern gratis genutzt werden, als dass eine gewisse Anzahl an Anfragen bezüglich des Routings pro Monat kostenfrei sind und erst nach Überschreitung dieses Limits die Nutzung kostenpflichtig wird.

dazu die API von Mapbox hinsichtlich Routinganfragen nutzbar zu machen, aber auch deren Kartenservice in das eigene in Unity erstellte Projekt einbinden zu können. Betrachtet man nun zu guter Letzt noch die ARGPS-Erweiterung, so stellt diese zum einen Codes, wie auch Prefabs und ein fertiges Beispielsprojekt zur Verfügung, mit welchen es relativ einfach möglich wird, eine AR-Navigationsapp für Smartphones, ähnlich jener von Google, zu entwickeln.

3.5.2 Räumlichkeit in einer AR-Applikation (am Beispiel einer künstlerischen AR-Installation & AR-Navigationsapplikation):

Abschließend für dieses Kapitel bleibt nun nur noch zu klären, wie eine AR-Applikation in abstrahierter, vereinfachter Weise eigentlich funktioniert: Den ersten Punkt stellt dabei das Starten der AR-Funktionalität in der Applikation an sich dar, wobei an dieser Stelle ein virtueller Raum erstellt wird, in welchem dann die anzuzeigenden Inhalte platziert werden, mit Nullpunkt an der Startposition. Das bedeutet, dass der Koordinatenursprung des virtuellen Raumes sich immer an dem Punkt, wo die AR-Applikation in der realen Welt gestartet wurde, befindet. Damit diese Verknüpfung jedoch erhalten bleibt und man sich durch den virtuellen Raum bei Bewegung durch den realen Raum manövrieren kann, gibt es mehrere Lösungen, wobei eine davon die Verknüpfung des Koordinatenursprungs des virtuellen Raums mit realen Weltkoordinaten (GNSS oder mittels IMU) ist – wobei nicht zwangsläufig eine Kamera, jedoch Hardware zur Positionsüberwachung erforderlich ist. Zudem gibt es die Möglichkeit über die Erkennung von Strukturen/ Muster, also etwa Kanten und der Ableitung von Räumen oder ähnlichem: Hier ist zwangsläufig eine Kamera vonnöten^{149,150}.

Zum Verhältnis zwischen realem und virtuellem Raum muss an dieser Stelle noch hinzugefügt werden, dass es zum einen sein kann, dass der virtuelle Raum mehr oder minder so groß ist wie der reale Raum, oder aber auch, dass der virtuelle Raum nur einen kleinen Bereich des realen Raums überlagert und man sich nicht beziehungsweise nur in minimalem Ausmaß¹⁵¹ durch diesen hindurchbewegen kann. Um dies etwas konkreter zu gestalten seien zwei einfache Beispiele angeführt: Handelt es sich etwa um eine AR-Applikation, welche mithilfe von einzuscannenden an künstlerischen Werken angebrachten QR-Codes die

¹⁴⁹ Einfacher gestaltet es sich, wenn die Bewegung durch den virtuellen Raum nicht mit der Bewegung im realen Raum verknüpft sein soll. Dabei ist es dann nämlich möglich, sich einfach mittels Eingabe am Eingabegerät durch diesen zu bewegen.

¹⁵⁰ An dieser Stelle soll auch noch auf die Realisierung von Google bei deren Live View Navigation hingewiesen werden, wobei zum einen auf GPS zur Verknüpfung und zum anderen die Verbesserung der Positionsgenauigkeit mithilfe von Bilderkennung der von Google gemachten Street-View Aufnahmen gesetzt wird.

¹⁵¹ Man denke bei der minimalen Bewegung an eine ganz einfache Realisierung einer AR-App, bei welcher ein QR-Code eingescannt wird und daraufhin ein Gemälde eingeblendet wird. Man kann sich diesem mithilfe der APP insofern nähern und davon entfernen, solange der QR-Code für die App zur Gänze lesbar bleibt. Der Grad an Genauigkeit, also wie viele Details des Bildes man erkennen kann, hängt dann weiters von der räumlichen Nähe zum QR-Code ab.

künstlerischen Werke selbst erkennen und verändern kann, dann findet sich der virtuelle Raum nur im Bereich des künstlerischen Werkes wieder. Wendet man sich aber davon ab, ist kein virtueller Raum mehr, da auch nicht benötigt, existent. Dies bedeutet aber nicht, dass Bewegung in diesem virtuellen Raum zur Gänze unmöglich sind. So kann man sich der Realisierung nähern beziehungsweise sich weiter von dieser entfernen, um andere Blickwinkel zu erreichen, wobei sich der virtuelle Raum aber immer nur rundum den Bereich des künstlerischen Werkes befindet. Ein Beispiel für einen großen, unter Umständen beinahe so großen virtuellen Raum wie die reale Welt wäre hingegen bei einer Navigationsapplikation zu suchen.

Legt man das in diesem und den vorangegangenen Kapiteln akkumulierte Wissen nun auf die eigene geplante Umsetzung (AR-Applikation zur Navigation im Bereich von Zustelldienstleistungen) um, so stellen sich in weiterer Folge die folgenden technisch-funktionalen Aspekte, welche bei der Umsetzung bedacht werden müssen, beziehungsweise wie diese umgesetzt werden können, dar:

Erstellung eines virtuellen Raums, der dem realen Raumausschnitt, in dem die Navigation stattfindet, entspricht

- → Virtueller Raum wird automatisch in Unity erstellt und mit realem Raum verknüpft. Nur Inhalte müssen selbst bestimmt und korrekt positioniert werden
- → Funktion zum Bewegen durch den virtuellen Raum ist in Unity standardmäßig vorhanden

Ermittlung der aktuellen Position und Blickrichtung

- → am einfachsten zu realisieren über GNSS-Ortung, War-Ist Vergleich der GNSS-Position und IMU
- → Funktionalität standardmäßig in Unity vorhanden

Erstellung einer Route im virtuellen Raum mit Navigationshinweisen

- → Funktionen hierzu bspw. über die Mapbox SDK oder ARGPS

Erstellung von Anzeigen für Zustellhinweisen

- → am einfachsten zu realisieren über die Verknüpfung der realen und virtuellen Welt und die Platzierung von Hinweisen an den virtuellen Unity-Koordinatenentsprechungen der tatsächlichen WGS84-Koordinaten, wo Zustellhinweise (beispielsweise an der Eingangstür) angezeigt werden sollen

4 Augmented-Reality Navigationssysteme

Nachdem nun alle für diese Arbeit benötigten Grundlagen und Konzepte in den vorangegangenen Kapiteln in angemessenem Ausmaß beleuchtet wurden, folgt nun, bevor zur eigenen Umsetzung eines AR-Navigationssystems für den Zustellbereich übergegangen werden kann, noch eine kurze Betrachtung von bereits existierenden AR-Navigationssystemen sowie eine Besprechung dieser in Hinblick auf die daraus abzuleitenden Rückschlüsse für die eigene Umsetzung. An dieser Stelle muss darüber hinaus erwähnt werden, dass einige Arbeiten mit Blick auf AR-Navigationssysteme mit deren spezifischen Fokussen, wie etwa dem generellen Aufbau einer AR-Navigationsapplikation, dem Erlernen von Routen oder dem Vergleich anderer Navigationsmethoden mit AR-Navigationsmethode existieren: Dennoch gibt es auch einen blinden Fleck im Hinblick auf wissenschaftliche Arbeiten rundum die Verwendung, Potentiale und ableitbaren Weiterentwicklungsmöglichkeiten bereits existierender AR-Navigationsapplikationen, wie etwa Google Live View, Sygic Real View Navigation oder diverse Lösungsansätze von Automobilherstellern oder für Flugzeuge. Genau an dieser Stelle soll in weiterer Folge die Arbeit den Spagat schaffen, einerseits die wissenschaftlichen Arbeiten zu beleuchten, auf denen in logischer Konsequenz natürlich hinsichtlich der Ideen und Herangehensweisen auch die vorzufindenden praktischen Umsetzungen zum Teil fußen, andererseits aber auch die praktischen Arbeiten und die wenigen diese besprechenden (News-)Artikel zu diskutieren. Ziel soll dabei, neben einer kritischen Betrachtung der Arbeiten per se das Ableiten von „Learnings“ für die eigene geplante Umsetzung eines AR-Navigationssystems sein.

4.1 Beispielhafte Umsetzungen aus der Wissenschaft

Bevor in weiterer Folge eine Reihe an Beispielen, die einem relativ breiten Publikum bekannt sein dürften, besprochen werden, folgt nun zuerst einmal die Beleuchtung der wissenschaftlichen Arbeiten zu AR-Navigation. Hierzu sei erwähnt, dass sich in weiterer Folge auf rezentere Arbeiten konzentriert werden wird, das heißt auf Publikationen ab Beginn der 2010er Jahre – auch wenn natürlich bereits davor diverse Umsetzungen stattfanden. Das liegt darin begründet, dass sich seit Beginn der 2010er Jahre die Geräte, welche für die Realisierung eines AR-Systems eingesetzt werden können, wie auch die Arten der Präsentation der Navigationsinhalte – von der Anzeige von POIs hin zu tatsächlichen Navigationshinweisen – stark gewandelt haben. Mehr oder minder stellt es sich bei der Entwicklung der Geräte für AR-Navigation weiters so dar, dass die ersten AR-Navigationssysteme ohnehin nicht auf Mobiltelefonen, PDAs oder ähnlichem – da nicht verfügbar beziehungsweise zu leistungsschwach –, sondern auf HMDs mit einer Vielzahl an hierzu benötigten Geräten für Berechnung, Verortung und ähnlichem, wie etwa bei dem von Höllerer et al. in ihrer mehr oder minder als Pionierarbeit zu bezeichnenden Ausarbeitung mit dem Titel *„Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system“* (Höllerer, et al., 1999) vorgestellten MARS-Prototypen fußten. Aufgrund diverser Umstände, wie etwa der Umständlichkeit und

Komplexität dieser Systeme, aber auch der Entwicklung leistungsstärkerer Mobiltelefone und schlussendlich Smartphones fokussierte sich der Bereich rundum AR-Navigationssysteme in den darauffolgenden Jahren nach einer kurzen Phase, in der beinahe keine Forschung zum Thema Navigation mit AR beobachtet werden konnte, auf diese. In letzter Zeit wiederum ist darüber hinaus, unter anderem aufgrund der diversen Einschränkungen hinsichtlich der Anwendung von *handheld-devices* (dt. Handgerät) zur Navigation und dem vermehrten Aufkommen von HMDs und Smart-Glasses wiederum ein gestiegenes Interesse an der Verwendung selbiger für die Navigation zu erkennen, wobei zum momentanen Zeitpunkt die Forschungswelt in diesem Bereich noch etwa hinterherhinkt. Insgesamt könnte man den Forschungsbereich somit in etwa als kreisartig beschreiben, da direkt zu Beginn mit HMDs aufgrund deren bester konzeptioneller Eignung experimentiert wurde¹⁵², anschließend hin zu den im Aufkommen begriffenen Smartphones¹⁵³ gegangen und zu guter Letzt, aufgrund der sich geänderten technischen Limits, wieder zurück zu HMDs geschwenkt wurde. Nach diesen einleitenden Worten folgt nun in weitere Folge, untergliedert in die Hauptaspekte der vorzufindenden wissenschaftlichen Arbeiten – Applikationsdesign & Kommunikation der Inhalte, Nutzbarkeit, Navigationsqualität und Spatial Knowledge Acquisition, AR mittels HUD in Fahrzeugen, AR mittels HMD – deren nähere Betrachtung.

4.1.1 Applikationsdesign und Kommunikation von Inhalten

Eine relativ frühe und sehr interessante Arbeit im Hinblick auf AR-Navigation, deren Art der Umsetzung und Kommunikation der Inhalte bis heute in einigen Aspekten durchwegs von Relevanz ist, ist jene von Mata und Claramunt mit dem Titel „*Augmented Navigation in Outdoor Environments*“ (Mata & Claramunt, 2013). Die Autoren beschäftigen sich in der vorliegenden Arbeit dabei mit dem Aufbau eines AR-Systems zur Navigation auf Smartphones im historischen Zentrum von Mexiko-Stadt. Hierbei mussten, dem damaligen Stand der Technik geschuldet, zwar einige Einschränkungen in Hinblick auf die auswählbaren Ziele und der Visualisierung von Sachverhalten gemacht werden, der Ansatz zur Realisierung hingegen hat sich im Hinblick auf die Verwendung von GPS und Kompass – heutzutage mit IMUs ergänzt – und der Verwendung von Image-Recognition zur Positionsverbesserung bis heute nicht grundlegend verändert, wie anhand von Google Live View oder Sygic Real View gesehen werden kann. Den Grundüberlegungen für die Hinzunahme von Image-Recognition wurde in der Arbeit damals mit der bis heute in Teilen gültigen Ungenauigkeit von GPS-Positionierung begründet, wobei die Autoren in deren Conclusio zusätzlich erwähnen, dass auch Image-Recognition allein keine zuverlässigen und zufriedenstellenden Ergebnisse hätte liefern können und somit eine Kombination unabdingbar sei.

¹⁵² ... aber die technischen Voraussetzungen es für ein breiteres Publikum sinnvoll nutzbar zu machen noch nicht erfüllt waren.

¹⁵³ ... aufgrund der Leistungsfähigkeit und der Nutzbarkeit für ein breites Publikum

Ebenso interessant zu erwähnen, wenn auch nicht wie bei heutigen AR-Navigationsapplikationen gelöst ist an dieser Stelle die Aufbereitung des UIs. Als Navigationshinweise wurden dabei einfach Pop-Ups¹⁵⁴ in Form von Schildern mit einem Richtungspfeil, Ortsbeschreibung und Distanz für die Navigation (siehe Abb. 15) eingeblendet. Weiters existierte noch eine zweite Funktion, mithilfe derer Symbole für spezifische Orte in deren tatsächlich befindlicher Himmelsrichtung angezeigt werden konnten, wobei hier nur der Name und die Distanz zum jeweiligen Ort angezeigt (siehe Abb. 14) wurde. Leider wird an dieser Stelle nicht weiter erläutert, wie die Richtungspfeile und die Entfernung definiert wurden und ob es sich um eine reine Angabe der horizontalen Distanz und einen Richtungspfeil basierend auf einer Richtungsangabe (Kompass) handelt oder ob es sich um ein tatsächliches Routing handelt¹⁵⁵. Was ein weiteres Problem der Applikation darstellte, war der Punkt, dass immer alle Objekte eines durch den User bestimmten Typs in einem bestimmten Radius rundum die Nutzerposition angezeigt wurden, wodurch man zu dem Problem der Überlagerung von mehreren Pop-Ups kam, welche wiederum Probleme im Hinblick auf die Navigation bedingte. Ebenso als suboptimal, zumindest aus heutiger Sicht zu beschreiben ist der Umstand, dass nicht an den jeweiligen Kreuzungspunkten des Weges Informationen zum besten weiteren Vorgehen bei der Navigation, sondern nur in ganz genereller Form eine Richtungsangabe und Distanz zum finalen Ort angegeben wurden. Nichtsdestotrotz war das Fazit eines Feldversuchs von einer Probandengruppe mit 40 Personen, entsprechend der aufwendigen und technisch gut durchdachten Umsetzung und der fortschrittlichen Herangehensweise, durchwegs positiv, auch wenn sich eine Vielzahl an Teilnehmer*innen eine Sprachausgabe wünschte. (Mata & Claramunt, 2013)



Abbildung 15: Anzeige von Navigationsanweisungen in der App von Mata & Claramunt (Mata & Claramunt, 2013)



Abbildung 14: Anzeige von POIs in der App von Mata & Claramunt (Mata & Claramunt, 2013)

¹⁵⁴ Bei Klick auf die Pop-Ups wurden Zusatzinformationen und teils auch Bilder zum jeweilig zugehörigen Objekt angezeigt.

¹⁵⁵ An dieser Stelle kann unter anderem aufgrund eines Hinweises in einem Artikel von Chu et al. (Chu, et al., 2017) und Anderen zu dieser Zeit umgesetzten Arbeiten mit relativ großer Wahrscheinlichkeit davon ausgegangen werden, dass es sich um eine horizontale Distanz und eine Richtungsangabe aufgrund einer Kompassrichtung handelte.

Unter anderem auf der vorangegangenen Arbeit aufbauend beschäftigten sich dann weiters Chu et al. in deren Arbeit zur AR-Navigation auf Smartphones tiefergehend mit zwei in diversen, unter anderem in der vorangegangenen Arbeit vorgefundenen Problembereichen: dem UI-Design, also insbesondere Überlappungen und der Geschwindigkeit hinsichtlich der Positionierung und des Abrufens korrigierter Routinginformationen (Distanz & Richtung) bei Positionsänderungen. Zudem erarbeiteten sie einen umfangreichen Vergleich von AR-Navigationsapplikationen (der ihrerseits erstellen, wie auch weiteren) mit einer 2D-Karte in Kombination mit textlichen Anweisungen zur Navigation. Hinsichtlich der Verbesserung des UIs wählten die Autoren den Ansatz anstelle von Anweisungen in Form von Pop-Ups kleine anklickbare, auf Metainformationen verweisende Symbole, welche je nach Entfernung zum/r Nutzer*in eine unterschiedliche Größe aufweisen, zu verwenden und integrierten eine Logik hinsichtlich der Verdrängung (Gewichtung nach Wichtigkeit) von Symbolen, sodass keine Überlappungen geschehen. Um Informationen hinsichtlich der Entfernung zu einem ausgewählten Objekt bereitstellen zu können, integrierten sie dann noch zusätzlich einen Pfeil mit einer Distanzangabe (siehe Abb. 16 rechts unten), welcher bei Klick auf ein Symbol erschien. Natürlich weist auch dieser Ansatz einige Schwachstellen im Hinblick auf die Nutzerfreundlichkeit auf, wie etwa, dass die Symbole universell verständlich sein müssen, um korrekt verwendet werden zu können und außerdem Informationen zum Objekt (etwa der Name und die Adresse) nicht direkt angezeigt werden können. Im Hinblick auf die Untersuchung der Nutzbarkeit der AR-Navigationsapplikation im Vergleich zu einer leeren beziehungsweise mit Symbolen versehenen 2D-Karte in Kombination mit Text schließen die Autoren, neben einem positiven Fazit bei der Userbefragung zur Nutzungserfahrung ihrer AR-Applikation, damit, dass die verschiedenen getesteten AR-Navigationsapplikationen einen signifikanten Vorteil hinsichtlich der Orientierung in einer räumlich unbekannten Umgebung bieten und sich der/die Nutzer*in mit einer AR-Applikation schneller zurecht findet und ein vorgegebenes Ziel in kürzerer Zeit erreichen kann. (Chu, et al., 2017)



Abbildung 16: UI der AR-Applikation von Chu et al. (Chu, et al., 2017)

4.1.2 Nutzbarkeit, Navigationsqualität und Spatial Knowledge Acquisition

Neben der Thematik rundum die Art der Darstellung und Vermittlung von Inhalten lassen sich auch einige Arbeiten finden, welche sich mit dem räumlichen Wissenserwerb beschäftigen. Hierzu zählt etwa jene Arbeit von Haosheng et al., welche sich in komparativer Form verschiedenen Arten der Navigationshilfen widmet: etwa in Form von auditiven Hinweisen, einer 2D-Karte mit eingezeichneter Route am Smartphone oder eben einer AR-Navigationsapplikation mit tatsächlicher Routenanzeige – im Gegensatz zu jenen,

wo nur POIs gezeigt werden. Die Qualität hinsichtlich der Wissensgenerierung wurde in diesem Fall, im Unterschied zu anderen Arbeiten, nicht rein anhand der Zeit, welche Proband*innen zur Absolvierung einer Route brauchten, und der Anzahl an Fehlern bei der Navigation gemessen, sondern unter Bezugnahme auf das LRS-Modell von Siegel und White¹⁵⁶. Insgesamt kamen die Autoren dabei zu dem Schluss, dass mithilfe des LRS-Modells ein dezidierter Unterschied zwischen Personen, welchen die räumliche Umgebung bereits bekannt war, und Personen, welchen diese unbekannt war, festmachen lässt, sich hinsichtlich des räumlichen Wissenserwerbs von Personen ohne vorherige räumliche Kenntnis bei Verwendung verschiedener Hilfsmittel jedoch keine signifikanten Unterschiede feststellen lassen. Dies wird dann mit den bereits in vorangegangenen Arbeiten gemachten Beobachtung hinsichtlich der Tatsache, des „*active encoding principles*“¹⁵⁷ gestützt und führt insgesamt zur Frage, ob der Aufbau von räumlichem Wissen bei der Routenfindung mithilfe von Navigationshilfsmitteln nötig/ gewünscht ist oder es rein um die Erleichterung der Routenfindung gehen sollte. Abschließend wirft diese Forschung somit die Frage nach dem Wie, also wie räumliches Wissen mittels der vorgestellten Mittel besser vermittelt werden kann, auf, sodass es in die mental-map übergeht. (Huang, et al., 2012)

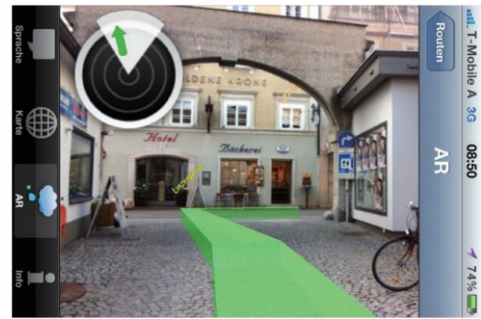


Abbildung 17: Umsetzung der AR-Navigationsapplikation bei Huang et al. (Huang, et al., 2012)

Betrachtet man das Feld der AR-Navigation nun ab der Wissensvermittlung eher vom Standpunkt der Nutzbarkeit solcher Applikationen, beginnt man auch hier am besten bei den für die vorliegende Arbeit definierten „Anfängen“. Hierzu bietet sich etwa die Arbeit von Dünser et al. zu AR-Navigation mit *Handheld-devices* im Jahr 2012 an, auch wenn direkt zu Beginn gesagt werden muss, dass einige Aspekte, wie etwa die Umsetzung einer AR-Navigationsapplikation nur mithilfe von POIs ohne Routenanzeige mehr als suboptimal und veraltet bezeichnet werden muss¹⁵⁸. Andere Aspekte hingegen haben bis heute durchwegs Relevanz. So widmen sich Dünser et al. in ihrer Arbeit ähnlich jener von Haosheng et al.

¹⁵⁶ Im Hinblick auf das Erlernen der räumlichen Umgebung haben Siegel und White das sogenannte LRS-Modell erstellt, welches sich in die folgenden drei Punkte einteilt:

LRS-Modell (Siegel & White, 1975):

- *Landmark knowledge: the visual appearance of prominent objects and cues*
- *Route knowledge: a point-by-point sequence of actions needed to travel a route*
- *Survey knowledge: the relationship between landmarks and routes*

Insgesamt ermöglicht dieses System Forschenden die Qualität von räumlicher Wissensvermittlung mit verschiedenen Mittel der Navigation zu vergleichen und der Frage, wie gut eine Person in einem gewissen Gebiet, wo diese zuvor mit Navigationshilfsmitteln unterwegs war, bei einem Ausfall selbiger mithilfe der eigenen Mental-Map navigieren kann. Für genauere Informationen hierzu siehe: (Siegel & White, 1975)

¹⁵⁷ Das „*active encoding principle*“ besagt, dass sich nur initial und aktiv bei der Routenfindung gemachte Beobachtungen von Probanden gemerkt werden und zum Aufbau einer mental-map führen. (Münzer, et al., 2006)

¹⁵⁸ Selbiges trifft natürlich auch auf die ersten beiden im Rahmen dieses Kapitels besprochenen Arbeiten zu.

dem Vergleich von AR-Navigation, AR+Karten-Navigation und reiner Kartennavigation auf *Handheld-devices*. Hierbei kommen sie zu dem Schluss – welcher auch in anderen Arbeiten zur Navigation vorzufinden ist – dass Männer bei der Navigation tendenziell besser als Frauen performen, wobei der Unterschied bei AR-Navigation in einem geringeren Ausmaß zum Tragen kommt als bei digitalen und analogen Karten. Zum anderen konkludieren sie, dass bei den verschiedenen verglichenen Hilfsmitteln keine signifikanten Unterschiede hinsichtlich des zurückgelegten Wegs bei der Navigation und der benötigten Zeit zur Absolvierung des Wegs festgestellt werden kann. Interessant ist die Arbeit an dieser Stelle weiters deshalb, weil eine umfassende User-Befragung, welche sich mit Aspekten des User-Interfaces beschäftigte, und eine Nutzungsverhaltensaufzeichnung durchgeführt wurde. Bei ersterer kam heraus, dass sich die Proband*innen mit der Kombination aus AR+Karte am wohlsten und sichersten in Hinblick auf Bedienung und Orientierung fühlten und diese auch den anderen vorziehen würden. Die Beobachtung des Nutzungsverhalten förderte darüber hinaus zu Tage, dass die jeweiligen Navigationshilfsmittel zumeist während des Gehens genutzt wurden und die Betrachtungszeit, hierbei bei der Kombination aus AR+Karte, gefolgt von Karte und dann AR allein am höchsten war. (Dünser, et al., 2012)

4.1.3 AR mithilfe von HUD in Fahrzeugen

Ein weiterer Punkt rundum AR-Navigation, zu dem wissenschaftliche Arbeiten vorgefunden werden können, bezieht sich auf den Einsatz von AR (mithilfe von HUD) in Autos. Hierbei ist etwa die Arbeit von Yount et al., welche stellvertretend für eine Vielzahl anderer im Bereich von AR-HUD in Autos steht, besonders interessant, da sie zum einen die erste im Rahmen dieser Arbeit angeführte Arbeit ist, welche sich mit AR-Navigationshilfsmitteln bei einer anderen Fortbewegungsart als zu Fuß beschäftigt, und zum anderen im Gegensatz zu einigen oberhalb besprochenen Arbeiten wie jene von Huang et al. nicht nur auf eine Anzeige der Ziele als POI setzt, sondern tatsächlich an den jeweiligen Kreuzungen und auch zwischen Kreuzungspunkten Informationen zum weiteren Verlauf der Route (basierend auf einem Routing) anzeigt.

Direkt zu Beginn sollte bei der Besprechung dieser Arbeit aber noch einschränkend erwähnt werden, dass der Testaufbau am Computer und nicht in der realen Umwelt, also nicht im Straßenverkehr stattfand, was sich einerseits auf technologischen Problemen bei der Umsetzung, aber auch der Sicherheit beim Testaufbau zurückführen lässt. Denn gerade dies, nämlich die Fahrsicherheit, also der Ablenkung während der Nutzung der zu testenden Navigationshilfsmittels – AR-Anzeige, digitale Karte, analoge Karte – und die Spatial Knowledge Acquisition bei der Verwendung stellten zwei der Hauptgesichtspunkte der Arbeit dar und sollten getestet werden. Zu zweiterem Punkt ergab sich dabei, wie, wenn andere Arbeiten zu dem Thema bekannt sind zu erwarten war, dass hinsichtlich der Spatial Knowledge Acquisition eine (analoge) 2D-Karte, selbst wenn die Route auf dieser nicht selbst gesucht werden muss, den größten Lernerfolg bietet. Ebenso waren die Ergebnisse in Hinblick auf die Unterschiede des Lernerfolgs zwischen Nutzung eines AR-Systems im Vergleich zu einer digitalen Karte im Rahmen des Versuchsaufbau insignifikant. Signifikant waren hingegen die Ergebnisse im Hinblick auf die Fahrsicherheit, wobei die 2D-Karten am

schlechtesten abschnitten und bei Nutzer*innen des AR-Systems nicht nur ein gleichmäßigerer Fahrstil festzustellen war, was auf die geringere Ablenkung während des von der Fahrbahn abgewendeten Blick auf die digitale, auf einem im Auto montierten Display präsentierte 2D-Karte geschoben werden kann, sondern auch der Richtungswechsel an Kreuzungen mit größerer Sicherheit geschieht. Mit diesem generell positiven Fazit zu AR in Autos – in den allermeisten Fällen realisiert über ein HUD – reiht sich die Arbeit damit in eine Reihe an Arbeiten, welche einem AR-HUD ein sehr positives Zeugnis ausstellen, ein: etwa bei der Gefahrenerkennung älterer Lenker*innen, bei schlechter Witterung und einer damit folgenden Reduktion von Stress. Komplementierend sollte an dieser Stelle jedoch noch erwähnt werden, dass trotz aller positiver Beobachtungen bei freier Fahrt eine Abnahme der Genauigkeit bei Spurwechselmanövern in Hinblick auf die korrekte Verwendung des Blinkers beobachtet werden konnte, wobei an dieser Stelle über die Gründe hierfür (noch) keine genauen Aussagen gemacht werden konnten. (Yount, et al., 2022)

4.1.4 Navigation mittels Head-Mounted-Displays

Wie bereits eingangs zu diesem Kapitel erwähnt, ist es gerade zu Beginn teils schwer, konkrete wissenschaftliche Arbeiten zum Thema AR-Navigation mit bestimmten Fokussen zu finden. Dies liegt darin, dass es sich zwar um ein Themenfeld handelt, welches von einer Vielzahl an wissenschaftlichen Bereichen¹⁵⁹ erforscht wird, es jedoch wie bei Technologien nicht unüblich, so ist, dass zwar Grundlagenforschung, auf welcher dann Firmen ein praktisches Produkt aufbauen, betrieben wird, danach die Forschung aber in gewisser Weise, sobald Produkte veröffentlicht werden hinterherhinkt oder der Bereich aus dem Interessensfeld der Wissenschaft verschwunden ist.

Dieses Problem tritt auch insbesondere zu Tage, wenn es darum geht, rezente Arbeiten¹⁶⁰ zur AR-Navigation mit Fokus auf HMDs zu finden. Hier finden sich im ersten Anlauf so gut wie keine rezenten Arbeiten, bis man zur Suche rundum die im Jahr 2023 eingestellten Google Glass¹⁶¹ kommt, wo dann zumindest vereinzelt Arbeiten, die aber auch nur in geringem Ausmaß von Relevanz für andere HMDs zum Einsatz der Navigation sind, vorzufinden sind. Diese Einschränkung der Relevanz beruht dabei in der Funktionsweise von HMDs. Rufen wir uns an dieser Stelle also nochmals kurz die Funktionsweise von HMDs in Erinnerung: HMDs können monokular oder binokular aufgebaut sein und demnach entweder (wie bei der Google Glass) nur 2D-Inhalte (also etwa normale 2D-

¹⁵⁹ Bspw. Psychologie für das Erlernen räumlicher Information, Geoinformation in Bezug auf die Berechnung und Bereitstellung von Route, Informatik, ...

¹⁶⁰ Wie gezeigt, gibt es bereits 1999 beinahe als historisch zu bezeichnende Arbeiten, wie etwa jene von Höllerer et al. (Höllerer, et al., 1999)

¹⁶¹ Bei der Google Glass handelt es sich um eine sogenannte Look-Around-HMD, welches 2012 erstmals veröffentlicht wurde.

Karten; Richtungspfeile o.ä.), wie auch Smartphones¹⁶² darstellen oder tatsächlich (wie etwa bei der XREAL Air) 3D-Inhalte aufbereitet in einem virtuellen Raum darstellen.

Betrachten seien nun also einige vorzufindende Arbeiten zur Google-Glass und der „AR“¹⁶³ Navigation mit dieser, vor allem aufgrund deren Wichtigkeit als Einstieg zu HMDs in diesem Bereich und der Nutzbarkeit von HMDs, nicht jedoch aufgrund deren Realisierung/„AR“¹⁶³ Umsetzung. Als zu betrachtende Arbeiten stellen sich hier etwa der Erfahrungsbericht von Jackson (Jackson, 2013), welcher die nativ implementierte Google Maps Kartenanzeige während des Autofahrens testete, der Showcase von Google selbst (Google Glass, 2013), oder auch die Arbeit von Ang et al. (Ang, et al., 2018) hinsichtlich der Implementierung einer eigenen 2D-kartenbasierten Navigationsanzeige für einen Universitätscampus auf einer Google Glass dar. Diese schließen dabei gesammelt mit einem durchwegs positiven Fazit hinsichtlich der Nutzbarkeit, wobei vor allem auch der Vorteil der nicht benötigten Abwendung des Blicks vom navigierten Weg, wie etwa bei Smartphones oder Navigationsgeräten positiv hervorgehoben wird und keine der Arbeiten auf etwaige Probleme bei der Nutzung per se, also der Empfindung eines HMDs als störend, hinweisen.

Möchte man nun Arbeiten, welche sich dezidiert mit einer tatsächlichem AR-Navigation auf HMDs beschäftigen¹⁶⁴, ist es nahezu unmöglich, solche zu finden: Tatsächlich konnte nur eine solche Arbeit gefunden werden, wobei sich diese mit der Indoor-Navigation und dem Vergleich eines *handheld-devices* und Google Glasses beschäftigt – diese wurde unter dem Titel „*Augmented Reality-based Indoor Navigation: A Comparative Analysis of Handheld Devices vs. Google Glass*“ von Rehman und Shi (Rehman & Shi, 2017) publiziert. Für die Standortbestimmung wird im Rahmen dieser Arbeit nicht auf GPS-Verortung – da diese in Innenräumen ohnehin nur sehr eingeschränkt möglich ist – sondern die Standortermittlung mittels drahtloser Übertragungsmethoden (etwa WLAN, UWB, RFID)¹⁶⁵, wie etwa auch bei Ang et al. für die Innenraumorientierung angewandt zurückgegriffen. Die Platzierung der Navigationshinweise im virtuellen Raum wird dabei über Pattern-Recognition in dem Sinne realisiert, als dass das Gebäude an für die Navigation relevanten Orten (Kreuzungspunkten) zuvor 3D-gescannt wurde, sodass die für die Navigation erstellte Applikation den Ort, welcher mittels der in Google Glass integrierten Kamera erfasst wird, erkennt und daraufhin den korrekten Navigationshinweis präsentieren kann. Die aus dieser Arbeit abzuleitenden und für die hier vorliegende Forschung wichtigen Schlusspunkte sind dabei, dass AR-Brille zur Navigation im Vergleich zu Smartphones und einer Papierkarte einen Vorteil im

¹⁶² Der Grund hierfür liegt in deren Aufbau, also, dass es sich eigentlich um nichts anderes als um ein einfaches, vor den Augen getragenes Display handelt.

¹⁶³ AR wird an dieser Stelle in Anführungszeichen gesetzt, da sich darüber diskutieren lässt, ob es sich bei den jeweiligen Umsetzungen um eine AR-Anwendung handelt oder nicht, da kein virtueller Raum aufgebaut wird, welcher anzuzeigende Inhalte enthält.

¹⁶⁴ Das heißt die darzustellenden Inhalte liegen in einem virtuellen dreidimensionalen Raum vor und dieser muss wiederum mit der realen Welt vermischt werden können.

¹⁶⁵ Siehe zu den verschiedenen Arten von drahtlosen Übertragungsmethoden (eng. Wirells Communication/ Transmission Methods) etwa: (deepaman07ad, 2022)

Hinblick auf die Genauigkeit und Ablenkung bieten, jedoch gleichzeitig mit dem Smartphone den schlechtesten Lernerfolg bieten. Darüber hinaus wird als negativer Punkt noch die Akkulaufzeit angesprochen, welche es nicht ermöglichte, längere Routen zu testen. (Rehman & Shi, 2017)

4.2 Beispielhafte Umsetzungen aus der Praxis

Der anschließende Abschnitt widmet sich nun einige aktuelle Beispiele aus dem „alltäglichen Leben“, welche einem nicht unbeträchtlichen Teil von Personen bekannt sind oder welche diese selbst bereits erlebt und/oder genutzt haben. Hierbei handelt es sich vor allem um drei Bereiche – AR zur Navigation, AR in Autos, AR in Flugzeugen –, in denen AR-Systeme eingesetzt werden und nun kurz angesprochen werden sollen. Bei ersterem handelt es sich um **Navigationshilfen** bei fußläufiger Fortbewegung, wie etwa Google Live View oder Sygic Real View. Bei Google Live View¹⁶⁶ handelt es sich um eine Erweiterung von Google Maps für Smartphones, welches erstmals 2018 auf Googles jährlicher Entwicklerkonferenz vorgestellt und im Laufe des Jahres 2019 ausgeliefert und seither stetig verbessert, um

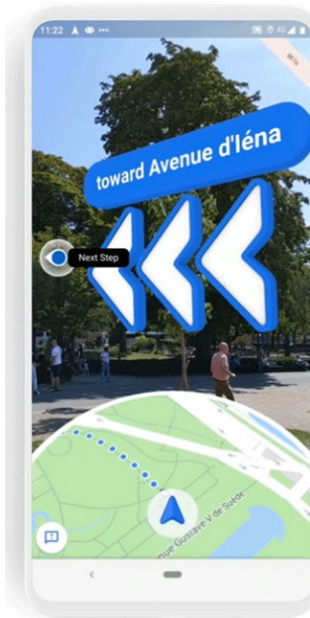


Abbildung 18: Google Live View Navigation, verändert nach (I., 2019)

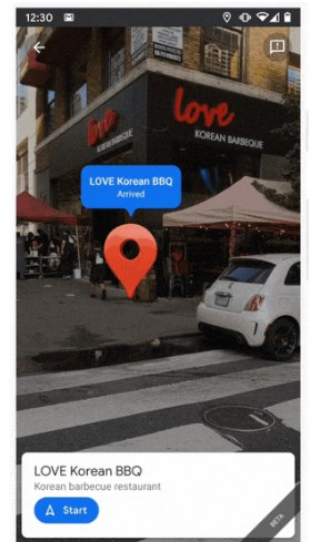


Abbildung 19: Google Live View Navigation Ziellanzeige, verändert nach (Ranieri, 2020)

Funktionen ergänzt und weiterentwickelt wurde. Insgesamt ermöglicht Live View – sofern das Smartphone die zur Ausführung benötigten SDKs (ARCore bzw. ARKit) unterstützt – es sich Navigationsanweisungen in Form einer Linie am Boden, Pfeilen und Hinweisschildern und seit neuerem auch Informationen zur in der Umgebung liegenden Orten anzeigen zu lassen. Bei ersterer Verwendungsart (Abb. 18, 19) kann das Display des Smartphones dabei in einen Bereich, wo eine 2D-Karte mit normaler Routenanzeige und einen Bereich, wo AR-Inhalt angezeigt werden unterteilt werden oder nur die AR-Ansicht Verwendung finden. Hinsichtlich der generellen Verwendung ist darüber hinaus anzumerken, dass diese a) nur für fußläufige Bewegung eingesetzt werden kann und b) die App erkennt, sobald sich der/die Nutzer*in bewegt und den Bildschirm aus Sicherheitsgründen abschaltet. Zum Funktionsprinzip lässt sich an dieser Stelle sagen, dass, wie eigentlich alle Navigationsapplikationen am Smartphone GPS, IMU und Kompass, zur

¹⁶⁶ Der Grund, weshalb an dieser Stelle Googles Live View und nicht etwa Sygic Real View näher betrachtet wird, liegt dabei in den Gründen, dass für Googles Realisierung mehr Informationen zur Verfügung stehen und darüber hinaus die Umsetzung der beiden Applikationen bis auf den Aspekt der Standortverbesserung mithilfe von Googles Street-View (bei Google Live-View) und Abweichungen im Styling der angezeigten Inhalte (den verfügbaren Informationen zufolge) als ident anzusehen sind.

Bestimmung der aktuellen Position und ein Wegenetz für das Routing eingesetzt wird. Hinsichtlich der Routendarstellung im AR-Raum und dem dahinterliegenden Prinzip findet man an dieser Stelle leider keine weiteren Informationen – abseits von diversen Aussagen bezüglich der Notwendigkeit einer guten Google Street-View Abdeckung (Google, kein Datum) –, wobei es bei genauerer Analyse der App so wirkt, als ob Google einfach nur an koordinativ verorteten Kreuzungspunkten (bezogen aus den Informationen des zum Routing an sich verwendeten Wegenetzes einen Pfeil und ein Schild¹⁶⁷ platziert, welche die neue Richtung anzeigen¹⁶⁸. (Cantisano, 2022; Google, kein Datum; Hall, 2023; Phillips, 2022)

Interessant ist an dieser Stelle dann auch noch die zweite angesprochenen und relativ rezente, genau gesagt 2022 für einige Städte¹⁶⁹ veröffentlichte Funktion (Abb. 20), welche es ermöglicht, sich innerhalb des AR-Raums Informationen zu den im Umkreis des aktuellen Standorts liegenden Objekten¹⁷⁰ anzeigen zu lassen. In andere Worte gefasst stellt dies nichts anderes als eine Suche von Objekten nach bestimmten Gesichtspunkten (Typ des gesuchten Objekts) und anschließender Repräsentation von Informationen zu diesen im AR-Raum dar. Die Funktionalität gestaltet sich hierbei so, dass der/ die Nutzer*in den AR-Raum startet und dann die Kamera schwenkt, bis Live View fertig initialisiert ist und Informationen zu Orten anzeigt. Abermals ist es an dieser Stelle so, dass Google keine Informationen zur genauen Funktionsweise angibt – abseits der abermaligen Notwendigkeit einer guten Street-View Abdeckung –, wobei hinsichtlich der Funktionsweise auf eine Kombination aus herkömmlicher Verortung des Smartphones per GPS und IMU sowie Image-Recognition in Bezug auf die umliegenden Objekte durchgeführt wird. Aufbauend auf diese genaue Verortung werden die Informationen dann aller Wahrscheinlichkeit nach entweder an koordinativen Orten oder an abermals mittels Image-Recognition erkannten Gebäudeflächen angezeigt. (Cantisano, 2022; Google, kein Datum; Hall, 2023; Phillips, 2022)

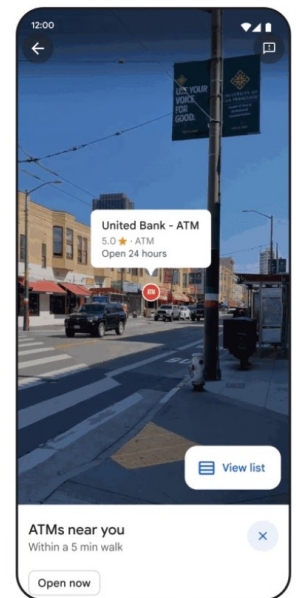


Abbildung 20: Google Live View Search, verändert nach (Phillips, 2022)

¹⁶⁷ Das Schild ist bereits eine Weile bevor an den Kreuzungspunkt gelangt, wird sichtbar und zeigt die verbleibende Distanz, während der Pfeil erst kurz davor angezeigt wird. Eine Navigationslinie ist hingegen die ganze Zeit über sichtbar.

¹⁶⁸ Hierzu sei auf die eigene Ausarbeitung verwiesen, in der eben jenes Prinzip auch angewandt wird. Fraglich ist an dieser Stelle nur, weshalb Google deren Realisierung auf Regionen mit guter Street-View Abdeckung einschränkt, da dies zumindest, soweit ersichtlich, nur der Verbesserung der Positionsgenauigkeit dient.

¹⁶⁹ Die Städte, die diese Funktion unterstützen sind: London, Los Angeles, New York, Paris, San Francisco und Tokio.

¹⁷⁰ Hierbei sind die auch ansonsten in Google Maps zur Verfügung stehenden Objekte gemeint: Restaurants, Sehenswürdigkeiten, Geschäfte, ...

Der zweite durchwegs bekannte Bereich, in welchem AR-System, oftmals unter dem Radar der bewussten Wahrnehmung zum Einsatz kommen, ist der Automobilbereich mit einer Vielzahl an verschiedenen Anwendungsfällen und Ausgestaltungen. Die verschiedenen **AR-Systeme in Autos** reichen hierbei von einfachen Einparkhilfen, die den Fahrtweg interaktiv je nach Lenkeinschlag, wobei die Umgebung mittels Kamera aufgenommen und die Inhalte auf einem im Auto installierten Display angezeigt werden¹⁷¹ bis hin zur Projektion von navigationsrelevanten – oder diversen weiteren – Informationen an die Frontscheibe, mittels sogenannter HUDs, für dessen Betrachtung der Blick nicht von der Straße abgewandt werden muss (siehe Abb. 21 – 23). An sich muss es sich bei den auf einem HUD angezeigten Inhalten dabei zwar nicht zwangsläufig um Augmented-Reality Realisierungen handeln, kann jedoch der Fall sein. Darüber hinaus gilt zu erwähnen, dass erste Realisierungen von HUDs, etwa zur Anzeige der Geschwindigkeit bereits in den 2000er Jahren zu beobachten waren.

Zwei der bekannteren und rezentesten Beispiele für AR-Systeme mittels HUD oder auch Displays sind sicherlich jenes von Mercedes mit dem Namen Mbus und jenes von VW in ihrer ID-Fahrzeugreihe¹⁷². Abermals ist auch bei diesen Umsetzungen keine Dokumentation aufzufinden, die erklären würde, wie die Applikationen umgesetzt wurden, jedoch stellt sich bei der Betrachtung der Navigationsapplikation die Funktion so dar, dass zum einen, wenn der Inhalt über ein Display präsentiert wird, jedenfalls eine Live-Kameraübertragung zum Einsatz kommt. Diese wird dann mit Inhalten zur Navigation überlagert – je nach Anwendung nur an Kreuzungen/ bei Spurwechseln/ Abbiegungen und/ oder die gesamte Zeit in Form einer Linie auf der Straße: aller Wahrscheinlichkeit nach wird dies mithilfe von

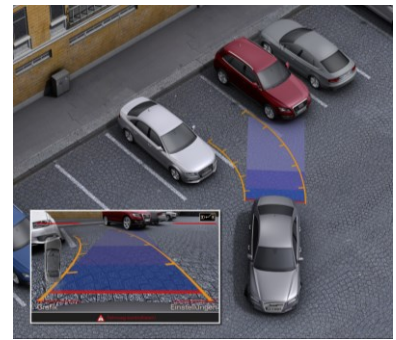


Abbildung 21: Darstellung des Parkassistenten, verändert nach (Audi AG, 2011)



Abbildung 22: AR-Navigation auf einem Display im Auto am Beispiel eines Mercedes GLC verändert nach (Eddy, 2022)



Abbildung 23: schematische Darstellung eines AR-HUD mit Navigationsanweisungen, verändert nach (Volkswagen, kein Datum a)

¹⁷¹ Als Beispiel hierzu sei an dieser Stelle, repräsentativ für andere Hersteller, die Realisierung von Audi angeführt (Audi AG, 2011). Hinsichtlich der Funktionsweise gestaltet es sich dabei in einfachster Form so, dass eine Kamera die Umgebung in Fahrtrichtung aufnimmt und basierend auf dem Lenkwinkel, Radstand und Längen (Gesamtlänge bzw. Überhänge) des Fahrzeuges Kreisbögen, welche die weitere Richtung angeben angezeigt werden. An dieser Stelle sich natürlich ohne weiterführende Informationen abermals vorzüglich darüber diskutieren, ob es sich denn hierbei tatsächlich um AR handelt oder nicht.

¹⁷² Siehe zur AR-Navigation mittels HUDs in Autos etwa: (Dirscherl, 2019; DPA, 2022; Mercedes-Benz USA, 2021; Volkswagen, kein Datum a; Volkswagen, kein Datum b)

Daten eines zuvor durchgeführten Routings und einer Vielzahl an diese aufbauenden Passpunkten realisiert. Hinzu kommt an dieser Stelle der Faktor, dass eine sogenannte Plane-Detection durchgeführt werden muss, falls Inhalte direkt die Straße überlagernd dargestellt werden sollen. Bei der Darstellung der Navigationsinhalte mittels eines HUDs ist es hingegen so, dass die anzuzeigenden Inhalte, ähnlich wie bei der Verwendung eines See-Through-HMDs, nur die Inhalte des virtuellen Raums, also die für die Navigation verwendeten Pfeile, Linien, Schilder oder Ähnliches, auf die Frontscheibe projiziert werden müssen, wobei es sich hier, da nur ein Blickwinkel dem menschlichen Auge zugeführt wird rein um die Anzeige von 2D angezeigte Inhalte handelt.

Beim dritten bekannten Anwendungsfall, den zwar höchstwahrscheinlich die allerwenigsten Personen je selbst erlebt haben, der aber durch Film und Fernsehen – zuletzt durch die Neuauflage des Films Top Gun – einen gewissen Grad an Bekanntheit genießt, handelt es sich um **AR-Systeme in Flugzeugen**. Die Technologie rundum HUDs und HMDs kommt hierbei, wie relativ oft in der Entwicklung zeitgenössischer Technologien, von militärischer Seite und wird in Kampfflugzeugen der vierten (ab ca. 1970) und fünften (ab ca. 2000) Generation standartmäßig verbaut. Handelte es sich bei der Ausführung in den Flugzeugen der vierten Generation noch um HUD, so wurde bei der fünften Generation aufgrund des Vorteils der Anzeige von Inhalten, selbst wenn der Pilot den Kopf bewegt auf Displays in den Helmen, gesetzt. Einer der Hauptgründe für die Implementierung von AR in Kampfflugzeugen war dabei der Vorteil, dass Pilot*innen den Blick nicht auf die Instrumente richten müssen, und natürlich im militärischen Bereich auch die Zielerfassung. Im Bereich von kommerziellen Flügen kommt AR bisher zwar nicht bei Pilot*innen zum Einsatz, dafür aber, wie bereits bei HMDs generell erwähnt, in der Wartung/ im Bau von Flugzeugen und der Schulung von Personal. Ebenso wenig kann momentan eine Verwendung von aus dem militärischen Bereich stammenden Umsetzungen im privaten Bereich festgestellt werden, wobei dies, vor allem dem Kostenfaktor (und auch Kosten-Nutzenfaktor) geschuldet sein dürfte. Insgesamt zeigen die militärischen Umsetzungen also, was technisch alles möglich ist, wobei sich mittlerweile auch diverse Firmen, wie etwa Aero Glass, darauf fokussieren, diesen Bereich für den kommerziellen Sektor weiterzuentwickeln und leistbar Geräte für Pilot*innen, um ihnen so das Leben zu erleichtern leistbarer zu machen. Angezeigt werden sollen hierbei flugzeugrelevanten Informationen, aber auch Navigationsanweisungen. (Maróy, 2017; Velichko, kein Datum; Verma, 2022)

4.3 „Learnings“ aus den Grundlagen und vorangegangenen Arbeiten

Nachdem nun die Grundlagen, wie auch eine genauere Besprechung von Augmented-Reality und ein Blick in Richtung für die vorliegende Arbeit relevante wissenschaftlichen, wie auch praktischen Arbeiten geworfen wurde, stellt sich nun, bevor an dieser Stelle zur Bearbeitung des praktischen Use-Cases übergegangen werden kann, schlussendlich noch die Frage, was sich aus den Grundlagen zu AR und den zuvor dargelegten wissenschaftlichen und praktischen Arbeiten für die Umsetzung eines eigenen AR-Systems für Zustelltätigkeiten ableiten lässt.

Erstens zeigt sich, dass generell die Frage nach dem Zweck eines Navigationshilfsmittel gestellt werden muss, da wie unter anderem einige oberhalb dargelegte wissenschaftliche Arbeiten ausreichend beleuchtet haben, die Spatial Knowledge Acquisiton bei Zuhilfenahme von jeglichen digitalen Hilfsmitteln, wie etwa digitalen Navigationskarten am Handy, durchwegs schlechter ist als bei der reinen Zuhilfenahme einer analogen Karte (mit oder ohne eingezeichnete(r) Route). Daraus lässt sich in einem ersten Schritt ableiten, dass es grundsätzlich, wenn ein Gebiet kennengelernt und darin regelmäßig navigiert werden soll, von Vorteil wäre, das Wissen über das Gebiet ohne Hilfsmittel aufzubauen, sprich die Navigation ohne Navigationshilfsmittel durchzuführen. An dieser Stelle spricht aber oftmals zum einen der Zeit- und Kostenfaktor dagegen und zum anderen gerade auch die Schwierigkeit, sich eine Vielzahl an Zwischenstopps beim Navigieren entlang einer ganz spezifischen Route, wie im Falle von Zustelltätigkeiten benötigt, zu merken. Dies macht es beinahe unmöglich, in diesem Bereich auf digitale Hilfsmittel oder Einschulungen zur Gänze verzichten.

Hinsichtlich der Einschulung entlang einer bestimmten Route, ist die 1:1 Einschulung¹⁷³ rein vom Lernerfolg her sicherlich auch gegenüber digitalen Hilfsmitteln zu präferieren, jedoch stellt sich auch hier relativ schnell die Frage nach dem Zeit- und Kostenfaktor, wobei die Problematik darüber hinaus in einer Branche, wo es ohnehin zu wenig Arbeitnehmer*innen gibt, noch dadurch verschärft wird, dass die Frage ist, ob überhaupt Personen verfügbar sind/ entbehrt werden können, die eine Einschulung vornehmen können.

Zusätzlich kommt dann noch der Punkt hinzu, dass es wie erwähnt, sein kann, dass selbst wenn man bereits im Zustellbereich mit fixer Route tätig ist neue Gebiete bedienen muss, etwa wegen Ausfall eines Arbeitskollegen, und dieses nicht bekannt ist.

4.3.1 Technische Aspekte

Im Hinblick auf die für AR-Systeme einsetzbaren Geräte zur Visualisierung, stellen sich darüber hinaus zum Zweck der Navigation, wie besprochen vier gängige Möglichkeiten dar: Smartphones, HUDs, Displays in Fahrzeugen und HMDs. Hinsichtlich der Anwendbarkeit und den daraus für die AR-Navigation ableitbaren „*Learnings*“ können dabei die in weiterer Folge zu besprechenden Punkte ausgemacht werden (unter Ausklammerung der Ausgestaltung der App, das heißt angezeigte Inhalte, deren Visualisierung, etc.):

- die generelle Usability
- die technische Umsetzbarkeit
- die multimodale Anwendbarkeit bei der Fortbewegung

¹⁷³ Hiermit ist gemeint, dass immer eine Person eine andere, neue Person einschult. Bei der Post AG wird dieses Prinzip eine Woche lang gemacht, bis die neu eingeschulte Person selbstständig die Tätigkeit verrichten muss. Hierzu ist anzumerken, dass je nach einzuschulender Person zumeist eine Trennung der Haupttätigkeit in dieser ersten Woche vollzogen wird. Dies bedeutet, dass die einzuschulende Person die erste Wochenhälfte den geringeren Teil der Arbeit übernimmt und erst im zweiten Teil der Woche die Haupttätigkeit von der neuen Person übernommen werden muss.

- der veränderbare Raumausschnitt bzw. der Anteil des Sichtfelds
- der Grad an Ablenkung

Wie gezeigt wurde, weisen Smartphones im Vergleich zu HMDs und auch in gewissem, wenn auch geringem Maße zu HUDs/ Displays in Fahrzeugen die beste Usability, aufgrund deren ohnehin großer Verbreitung und dem Wissen des Umgangs mit selbigen auf. Dies stellt sich darüber hinaus nicht nur aus Sicht eines/ einer Nutzer*in, sondern auch aus Sicht eines/ einer Entwickler*in¹⁷⁴ im Hinblick auf die Entwicklung und Verbreitung der eigenen App so dar. Dies liegt darin begründet, da es sich, wie dargelegt, bei einem Smartphone um eine All-in-One Lösung handelt und das Entwickeln hierfür relativ einfach ist. Im Hinblick auf die Usability von HUDs/ Displays in Fahrzeugen und HMDs muss an dieser Stelle hinzugefügt werden, dass HUDs und Displays in Fahrzeugen, wie auch Smartphones, eine sehr gute Usability für Nutzer*innen aufweisen, da von Autoherstellern eine möglichst verständliche, einfache Bedienung und Übermittlung von Informationen forciert wird, wohingegen die Usability sowohl aus Nutzer- als auch Entwicklersicht bei HMDs aufgrund der oftmals existenten Trennung von HMD und Berechnungseinheit (und ungewohnter Bedienungsmethoden für Nutzer*innen) tendenziell schlechter abschneidet.

Ein weiterer Punkt, welcher direkt an die Usability anschließt, ist dann die Multimodalität in der Nutzbarkeit der Geräte, das heißt es geht um die Frage, ob ein Gerät nur fußläufig oder nur in Autos verwendet werden kann. Dies ist einfach beantwortbar, da HUDs/ Displays in Fahrzeugen immer in einem Fahrzeug montiert sind und somit nur bei der Fortbewegung mit diesen genutzt werden können. Bei Smartphones und HMDs¹⁷⁵ hingegen gestaltet es sich so, dass diese bei allen Arten der Fortbewegung, also fußläufig, in Fahrzeugen etc. Verwendung finden können.

Betrachtet man als nächstes den für eine AR-Applikation nutzbaren Raumausschnitt und den damit in Verbindung stehenden Teil des Blickfeldes einer Person, welcher verändert werden kann¹⁷⁶, so muss zu Beginn zuerst einmal dargelegt werden, worum es sich bei diesen beiden Dingen eigentlich handelt. Mit einem für eine AR-Applikation nutzbaren Raumausschnitt ist an dieser Stelle der Bereich gemeint, welcher von der Applikation „aufgenommen“ und verändert werden kann, wohingegen es sich beim Blickfeld einer Person um den Bereich des Raumes handelt, worauf gerade die Augen gerichtet sind, die diesen somit aufnehmen. Anhand eines Beispiels, etwa einem Smartphone gestaltet es sich demnach so, dass das Blickfeld eines/ einer Nutzer*in nicht zwangsläufig dem Raumausschnitt, welcher von der AR-Applikation aufgenommen und verändert wird,

¹⁷⁴ Aller Wahrscheinlichkeit spielt dieser Punkt auch eine Rolle im Bezug auf das Fehlen von wissenschaftlichen Auseinandersetzungen mit HMDs zur AR-Navigation.

¹⁷⁵ Zu HMDs sollte an dieser Stelle einschränkend erwähnt werden, dass, wenn diese ein externes Gerät zur Berechnung der Inhalte besitzen, dieses auch mobil sein muss und die Verbindung zu diesem – etwa eine kabelgebundene Verbindung zwischen HMD und Smartphone – Bewegungsabläufe nicht in größerem Ausmaß stören darf.

¹⁷⁶ Hierzu sei gesagt, dass Veränderungen des AR-Raums theoretisch auch außerhalb des Blickfelds geschehen könn(t)en, es jedoch nicht als notwendig anzusehen ist, diese durchzuführen/ anzuzeigen solange der Blick des/der Nutzer*in nicht darauf gerichtet ist.

entspricht¹⁷⁷, da das Smartphone sich a) nicht automatisch mit den Kopfbewegungen mitbewegt und b) nicht das gesamte Blickfeld abdeckt und somit daran vorbeigeschaut werden kann. Diese banal wirkende Diskrepanz der wahrgenommenen Raumausschnitte zwischen Mensch und Gerät (zur Visualisierung von AR-Inhalten) ist von größter Wichtigkeit: Die Positionierung des AR-Geräts und der von diesem aufgenommene Bereich des Raums und *nicht* das Blickfeld des/ der Nutzer*in bestimmt den mittels virtueller Inhalten ergänzbaren Raumausschnitt, was im Umkehrschluss bedeutet, dass das Gerät entweder so vor den Augen platziert werden muss, dass ein (möglichst großer) Teil des momentanen Blickfelds abgedeckt wird oder der Blick bewusst auf das AR-Gerät gewandt werden muss. Daraus ergeben sich in weiterer Folge drei Punkte, welche allesamt im Anschluss besprochen werden sollen: Dazu zählen a) die Ablenkung (etwa bei Abwendung des Blicks) und Sicherheit bei der Nutzung, b) die Art wie ein Gerät getragen wird und c) der Anteil des Blickfeldes, welcher mithilfe eines AR-Geräts grundsätzlich überhaupt verändert werden kann.

Bevor auf die anderen Punkte eingegangen wird sollte kurz der Anteil des Blickfeldes, welcher von verschiedenen Geräten abgedeckt werden kann angesprochen werden. Hierbei ist es so, dass einzig mit HMDs¹⁷⁸ beinahe das gesamte Blickfeld, wie auch bei einem Sehbehelf, beeinflusst, soll heißen mit virtuellen Inhalten überlagert werden kann. Bei HUDs hingegen kann immer nur ein fixer (durch die Montage des HUDs, aber auch ggf. der Kamera im Auto determinierter) Bereich verändert werden und bei Smartphones gestaltete es sich so, dass, selbst wenn diese direkt vor die Augen gehalten werden, sich das Blickfeld in dem Sinne verändert, so dass der Öffnungswinkel der Kamera nicht zwangsläufig dem Blickfeld entspricht.

Hieran schließt auch direkt die Positionierung der Geräte zur Visualisierung relativ zum Blickwinkel an, was wiederum den veränderbaren Raumausschnitt mitdefiniert. Wie bereits angemerkt, sind HUDs/ Displays in Fahrzeugen fix montierte Geräte, welche sich somit nicht mit der Veränderung der Blickrichtung mitbewegen können, während dies bei Smartphones und HMDs der Fall ist. Bei den letzten Beiden ist es so, dass bei HMDs eine Veränderung der Ausrichtung ohne „manuelles“ Zutun tun, da sie direkt vor den Augen wie eine normale Brille getragen werden und somit jegliche Bewegung des Kopfs mitmachen, geschieht während bei Smartphones immer eine „manuelle“ Korrektur, soll heißen ein Bewegen des Smartphones in den aktuellen Sichtbereich gemacht werden muss (bzw. zuerst das Smartphone auf den zu Verändern gewünschten Bereich gerichtet und der Blick angepasst werden muss). Wie an dieser Stelle dann bereits an Smartphones mit der manuellen Anpassung der Blickrichtung desselben gesehen werden kann, handelt es sich demnach zumindest bei der Nutzung von Smartphones für AR mitnichten um einen Prozess, welcher gänzlich ohne aktives Zutun passiert, weshalb somit auch ein gewisser Grad an Ablenkung passiert. Dies mag zwar auf den ersten Blick marginal wirken, kann jedoch bei

¹⁷⁷ Ein Beispiel hierfür ist etwa das Tragen des Smartphones in der Hand mit Kamerablickrichtung Richtung Boden und das Entlanggehen eines/ einer Nutzer*in auf einem Gehsteig mit Blickrichtung horizontal nach vorne.

¹⁷⁸ Mit Ausnahme von Look-Around-HMDs, bei welchen auch nur ein kleiner Teil des Blickfeldes mit virtuellen Inhalten überlagert werden kann.

Nutzung im Straßenverkehr – egal ob als Fußgänger*in oder mit Fahrzeugen daran teilnehmend – ein erhebliches Sicherheitsrisiko darstellen. Ein wirkliches Problem ist die Ablenkung¹⁷⁹ dabei eigentlich nur bei der Nutzung von Smartphones und in Fahrzeugen montierten Displays, da nur hier der Blick von anderen wichtigen Bereichen abgewandt beziehungsweise die Position des Geräts angepasst werden muss. Bei HUDs und HMDs ist es nämlich so, dass HUDs ohnehin so positioniert sein sollten, dass sie (bei Ausnahme des Abbiegens) immer im relevanten Blickbereich liegen und HMDs sich sowieso, da am Kopf getragen, immer der Blickrichtung mitanpassen. Erwähnt werden sollte im Hinblick auf die Sicherheit bei der Nutzung dann noch die Problematik eines Ausfalls der bei einigen AR-Systemen zum Einsatz kommenden Kamera(s). Ein wirkliches Problem stellt dies im Bezug auf die Sicherheit eigentlich nur bei Non-See-Through-HMDs, da dann plötzlich überhaupt nichts mehr gesehen wird, da, da bei allen anderen HMD-Formen und HUDs dann einfach die Einblendungen ausbleiben. Bei Smartphones und in Fahrzeugen montierten Displays gestaltet sich ein Ausfall der Kamera bei Nutzung eines AR-Systems ebenso als relativ unproblematisch, da dieses (zumeist) ohnehin nicht den gesamten Sichtbereich einnimmt und von diesen aufgeblickt werden kann.

Insgesamt kann somit, für die Verwendung für AR-Navigationsapplikationen (im Zustellbereich), geschlossen werden, dass See-Through-HMDs trotz der gegebenenfalls existierenden Problematik rundum Geräte zur Berechnung der darzustellenden Inhalte und der damit verbundenen schwierigeren Entwicklung einer Applikation, die beste Lösung zur Abdeckung eines großen Bereichs eines sich verändernden Sichtfelds¹⁸⁰ darstellen, da es im Vergleich dazu HUDs/ fix montierten Displays überhaupt nicht möglich ist, sich mit dem Sichtfeld mitzubewegen und bei Smartphones der Nachteil der manuellen Ausrichtung des Smartphones auf die Blickrichtung (oder ggf. das Abwenden des Blicks zur Nutzung) zum Tragen kommen. Außerdem bieten HMDs wie auch Smartphones den Vorteil, multimodal, soll heißen im Auto, auf dem Moped oder zu Fuß, eingesetzt werden zu können und die Ablenkung mit selbigen und das somit entstehende Risiko bei Teilnahme am Straßenverkehr relativ gering ist. Darüber hinaus bieten HMDs vor allem bei fußläufiger Fortbewegung einen weiteren großen Vorteil: die Hände sind frei, was bei einem Zustellvorgang bedeutet, dass Briefe, Pakete oder Ähnliches ohne Probleme transportiert werden können, was etwa bei Smartphones nicht der Fall wäre.

4.3.2 Gestalterische Aspekte

Anschließend an die Besprechung der rein auf technischen Aspekten basierenden Anwendbarkeit verschiedener Geräte zur Umsetzung von AR-Systemen soll nun zur gestalterischen Realisierung von AR-Navigationsapplikationen übergegangen werden,

¹⁷⁹ An dieser Stelle wird bewusst eine Ausklammerung der Gestaltung der Applikation per se, auch wenn diese natürlich auch einen großen Einfluss auf den Grad der Ablenkung abseits der zur Visualisierung verwendeten Geräte hat, vorgenommen.

¹⁸⁰ Soll in diesem Fall heißen, dass sich die Blickrichtungen und Blickwinkel immer wieder ändern.

wobei die vorangegangenen Arbeiten in einem ersten Schritt gezeigt haben, dass es zwei Zugänge zu Navigationshilfestellungen gibt: Die Bereitstellung einer simplen Richtung und Distanz zum gewünschten Zielort und die Bereitstellung einer kompletten Route mit allen zur Navigation relevanten Informationen. Bei ersterer Art wird der Zielort – in diesem Fall als POI zu betiteln – zumeist nur mittels eines Symbols und gegebenenfalls dem Namen in der befindlichen Richtung und einer bestimmten Größe, welche im Optimalfall von der Distanz zu diesem abhängt, angezeigt. Als Hilfestellung zur Navigation werden darüber hinaus dann nur die Richtung und die Distanz zum POI angezeigt (siehe Abb.14-16). Bei der zweiten rezenten und komplexeren Art hingegen wird zu jeder Zeit der nächste durchzuführende Schritt angezeigt. Dies bedeutet, dass immer Informationen zum weiteren Vorgehen, also ob einem Weg weiterhin (und wie weit noch) gefolgt, ob abgelenkt oder ob umgedreht werden soll angezeigt wird. Realisiert wird dies durch ein Routing und eine Vielzahl an Zwischenschritten entlang der Route und auf der Darstellungsseite dadurch, dass zumindest an jedem Ort, wo eine Richtungsänderung vorliegt (Kreuzungspunkt), ein Pfeil (und Schild), welche(r) über das weitere Vorgehen informiert angezeigt wird. Bei der Beibehaltung des Weges wiederum wird oftmals in Wegrichtung zusätzlich eine Linie am Boden, welche den weiteren Verlauf vorgibt angezeigt.

Zusammenfassend heißt dies also, dass bei der ersten Umsetzung die Route zum Zielort zu einem hohen Grad selbst gesucht werden muss, während sie bei der zweiten Art der Umsetzung fix und fertig präsentiert wird. Daraus lassen sich in weiterer Folge wiederum zwei Dinge, die bereits einleitend erwähnt wurde, folgern: Die Spatial-Route Acquisition wird bei ersterer Form besser sein¹⁸¹, jedoch wird es im wirtschaftlichen Umfeld, aufgrund der zu Beginn dargelegten Punkte, so sein, dass die zweite Realisierung als zu präferieren gilt. Darüber hinaus hatten diverse Arbeiten gezeigt, dass sich Nutzer*innen mit Navigationshilfsmitteln, welche ihnen den aktiven Prozess der Wegfindung abnehmen, wohler fühlen als mit solchen, wo dieser selbst bewerkstelligt werden muss und natürlich mit zweiterer auch die Anzahl an Fehlnavigationen geringer und die Zeit zur Absolvierung kürzer ist.

Aufgrund dieser Überlegung und jener der Ablenkung ist es schlüssig, nun noch kurz die vorzufindenden Visualisierungsoptionen der im AR-Raum einzublendenden Informationen (und generell Optionen zur Übermittlung von Informationen zur Navigation) zu beleuchten. Hierbei stellen sich vor allem die folgenden Punkte als relevant dar: Die Menge (wie viel?) an eingeblendeten Informationen, wo die Einblendung passiert, als auch die Frage wie (Farbe, Transparenz, Größe, Position, ...) diese eingeblendet werden. Hierzu lässt sich dabei, wie sehr oft, die sprichwörtliche, aber bewährte Faustformel „so wenig wie möglich, so viel wie nötig“ finden, wobei zusätzlich von Mulloni et al. (Mulloni, et al., 2011) gezeigt wurde, dass Nutzer*innen deren Weg bei Nutzung von Navigationshilfsmittel zumeist vor und nach Wegkreuzungen – während des Gehens – kontrollieren und dementsprechend eine Anzeige von Informationen genau im Bereich von Kreuzungen einer besonderen Relevanz

¹⁸¹ Es sei ergänzend erwähnt, dass hierzu keine Forschungen vorliegen, dies sich jedoch als logischer Schluss darstellt, da es sich bei ersterer Form um einen aktiven Prozess der Wegfindung und bei zweiterer um einen passiven Prozess handelt – Stichwort „*active encoding principle*“.

zukommt. Zusätzlich wurde etwa in der Arbeit von Dünser et al. (Dünser, et al., 2012), für die AR mit einer digitalen 2D-Karte kombiniert wurde und jener von Ang et al. (Ang, et al., 2018), für die ausschließlich AR verwendet wurde, gezeigt, dass sich Nutzer*innen von AR-Navigationsapplikationen in starkem Ausmaße Multimedialität wünschen: soll heißen etwa die Bereitstellung einer zusätzlichen, durchwegs bekannten digitalen 2D-Karte oder aber akustischen Anweisungen zur Navigation.

Zuletzt soll nun noch auf zwei weitere, für die Nutzung von HMDs relevante Punkte eingegangen werden: Die Frage nach den rechtlichen Rahmenbedingungen und der Verträglichkeit von AR-Brillen. Hinsichtlich des ersten Punkts ist es in Deutschland so, dass die Nutzung jeglicher Videobrillen im Straßenverkehr verboten ist¹⁸², wohingegen in Österreich hierzu nicht klar Stellung genommen wird und Geräte, wenn diese zuvor genehmigt wurden, oder zu Testzwecken unter bestimmten Auflagen¹⁸³ dienen, verwendet werden dürfen. Interessant an dieser Stelle im Hinblick auf das Verbot in Deutschland ist jedoch, dass etwa BMW trotz dieser Einschränkung gerade dabei ist eine solche Implementierung einer AR-Brille¹⁸⁴ in Fahrzeugen vorzunehmen, wodurch es im Umkehrschluss interessant zu sehen sein wird, wie sich die rechtliche Lage entwickelt.

Bevor zum zweiten Punkt der Verträglichkeit von AR-Brillen übergegangen wird, muss an dieser Stelle kurz erwähnt werden, dass es für Brillenträger*innen mit optischer Sehkraft bei den meisten HMDs die Option gibt, sich eigene Gläser mit optischer Sehkraft für das jeweilige HMD anfertigen zu lassen oder direkt am Gerät einen gewissen Grad an Sehkraftkorrektur vorzunehmen. Zum Punkt der Verträglichkeit von AR-Brillen – insbesondere, wenn es sich nicht um See-Through-Modelle handelt – sei an dieser Stelle erwähnt, dass immer wieder von einem geringen Teil der Nutzer*innen von AR-/VR Brillen von Übelkeit, Schwindel und in verstärkten Ermüdungserscheinungen gehört oder gelesen werden kann, darüber hinaus aber keine nennenswerten Probleme bekannt sind.

¹⁸² Siehe hierzu: (Burhoff, 2018)

¹⁸³ Siehe hierzu: (RIS, 2023)

¹⁸⁴ Siehe hierzu: (DerStandard, 2024)

5 Use-Case: Erstellung einer AR-Applikation für Zustelltätigkeiten

An dieser Stelle sollen nun, basierend auf den vorangegangenen Kapiteln und den sich daraus ergebenden Schlussfolgerungen und Wissen, die konkreten Anforderungen, welche an eine AR-Navigationsapplikation im Zustellbereich gestellt werden Betrachtung finden: Hierzu wird in Kapitel 5.1 damit begonnen, die benötigten Funktionen und die Möglichkeiten der Bereitstellung von visuellen Navigationshinweisen darzustellen, bevor anschließend auf die für den konkreten Anwendungsfall als optimal zu klassifizierenden Geräte eingegangen wird. Im Kapitel 5.2 wird dann auf den an dieser Stelle noch auf theoretischer Ebene befindlichen Prozess der Erstellung einer AR-Navigationsapplikation für den Zustellbereich fokussiert, wobei hierzu in einem ersten Schritt die grundlegenden Design-Ideen sowohl in abstrakter Form in Bezug auf die einzusetzende Hard- und Software, aber auch in konkreter Form in Hinblick auf die Gestaltung der anzuzeigenden Inhalte (+ der Interaktion mit diesen) besprochen werden. Ebenso soll in diesem Kapitel eine kurze Einbettung der theoretisch geplanten Umsetzung ins XR-Feld geschehen, bevor anschließend im nächsten Kapitel (Kapitel 5.3) die Betrachtung der konkreten praktischen Ausarbeitung folgt. Hierbei wird in erster Linie der idealtypische Verlauf der Erstellung der eigenen Ausarbeitung dargestellt um in weiterer Folge dann auf das Endprodukt und „Learnings“, wie auch Probleme bei der Umsetzung einzugehen.

5.1 Funktionsanforderungen

5.1.1 Funktionen

Bei der Frage nach dem Funktionsumfang einer (AR-)Navigationsapplikation im Zustellbereich stellt sich als erste essentiellste Funktion natürlich jene der Navigation dar. Hierbei ist jedoch zu bedenken, ob es sich bei der Navigation um eine Navigation entlang einer fix vordefinierten Route („Tour“) handeln soll, oder ob es einfach darum geht einen Zielort auszuwählen und zu diesem geleitet zu werden¹⁸⁵. In weiterer Folge kommt dann, gerade bei Zustelltätigkeiten wie bei der Post AG, noch die Verknüpfung der Applikation mit Daten zu den zuzustellenden Lieferungen und den dabei zu beachtenden Zustellbedingungen hinzu. Insgesamt lassen sich also drei essenzielle Punkte für eine AR-Navigationsapplikation im Zustellungsbereich ausmachen, welche unterhalb nochmals zusammengefasst wurden:

¹⁸⁵ Bei letzterem kann es sich, wie bei Navigationsapplikationen üblich, um die schnellste, kürzeste, ... Route handeln. Bei fixen Touren können hingegen andere Aspekte, wie etwa im Zustellbereich die Reihenfolge der anzusteuernenden Adressen eine wichtige Rolle spielen.

1. Auswahl der Tour/ des Zielorts
2. Navigation entlang einer entsprechenden Route
3. Bereitstellung von Infos zur Zustellung einzelner Sendungen

5.1.2 Art der Bereitstellung von Navigationshinweisen

Im Hinblick auf die Anforderungen der Darstellung und die prinzipiell möglichen Darstellungsarten sollte angemerkt werden, dass bei Zustelltätigkeiten jeglicher Art die Hände zumindest auf den letzten Metern des Zustellweges nicht frei sind: beispielsweise, weil Pakete, Briefe oder ähnliches, welche übergeben werden müssen, in den Händen gehalten werden. Dies bedeutet, dass eine Darstellungsmethode, bei welcher die Hände nicht mit einem Gerät belegt sind, als präferabel gilt: Das Gerät sollte also kein *handheld-device* sein. Als optimal stellen sich demnach HMDs dar, da diese die ganze Zeit über wie eine normale Brille getragen und somit genutzt werden können. Das Problem, das bei vielen HMDs im Moment jedoch noch existiert, ist jenes der physischen Anbindung des HMD mittels Kabel zu einem Gerät, meistens einem Smartphone, welches die Berechnungen der Daten und der Darstellung übernimmt, da dies gegebenenfalls als störend empfunden werden kann.

Eine weitere Möglichkeit der Darstellung von Navigationshinweisen wäre darüber hinaus natürlich noch die Darstellung der Informationen über ein Display oder ein Heads-Up Display in einem Fahrzeug. Hier bestehen jedoch eine Vielzahl an Einschränkungen, angefangen dabei, dass nicht bei allen Zustelltätigkeiten Fahrzeuge verwendet werden, hin zu der Tatsache, dass auf den letzten Metern bei der Zustellung keine Informationen bereitgestellt werden könnten. Zudem stellt sich die Frage nach der generellen Möglichkeit der Implementierung einer solchen Software in das Autosystem – soll heißen die Installation eigener Softwarepakete im Auto-Infotainment.

Als letzte, eigentlich direkt zu Beginn aufgrund der Belegung der Hände als suboptimal zu deklarierende Option wäre dann noch jene der Verwendung von Smartphones zur Präsentation der Informationen. Diese Methode würde zwar alle verschiedenen Fortbewegungsmöglichkeiten abdecken können, weist aber eben den Nachteil der Belegung der Hände und weiters das Problem, dass es bei Verwendung zu einer stärkeren Ablenkung als bei den beiden anderen Arten führt, da der Blick immer zum Smartphone und weg von der Umwelt gelenkt werden muss, auf.

Ein weiterer wichtiger, bereits kurz nebenher angemerakter Punkt ist dann noch jener der Ablenkung während der Nutzung einer AR-Applikation: Da die potentielle Nutzergruppe immer in irgendeiner Weise am öffentlichen Straßenverkehr oder zumindest am Fußverkehr teilnimmt und somit bei einem zu großen Störfaktor¹⁸⁶ Gefahr läuft, in irgendeiner Weise selbst Schaden zu nehmen oder Schaden zu verursachen, ist es weiters wichtig, die bei so

¹⁸⁶ Hierin begründet sich auch die zuvor durchgeführte Besprechung des Grads der Ablenkung bei Nutzung verschiedener Geräte.

einer Applikation dargestellten Informationen auf ein notwendiges und somit am wenigsten ablenkendes Minimum zu reduzieren. Genau für solche Anwendungsfälle bietet sich im Umfeld der XR-Visualisierungen eben am besten AR an, da diese Form von XR die reale Welt „durchlässt“ und nur gewisse zusätzlichen Informationen als Einblendungen ergänzt und somit einen möglichst geringen Ablenkungsgrad aufweist. Von einem Ausfall des gesamten Systems bei einer VR-Brille sei an dieser Stelle noch gar nicht gesprochen.

Aufgrund all dieser nun vorangegangenen Überlegungen und dem zur Verfügung stehen eines HMD-Smartphone Systems, wie auch der generell momentan sehr hohen Relevanz und dem wissenschaftlich wie wirtschaftlichen Interesse an HMDs, wird im Rahmen der Arbeit in weiterer Folge auch auf die Kombination aus HMD und Smartphone für die Umsetzung der AR-Applikation gesetzt werden.

Zusammengefasst kristallisieren sich also im Bereich der Darstellung die folgenden zwei Punkte heraus

1. Die zu präferierende Darstellungsform bei Zustelltätigkeiten sind HMDs, gefolgt von der Anzeige auf Displays in Autos und auf Smartphones.
2. Im Feld der XR-Darstellungsmöglichkeiten bietet AR die besten Möglichkeiten zur Darstellung von relevanter Zusatzinformation in der realen Welt, wobei die dargestellten Informationen auf ein notwendiges möglichst wenig ablenkendes Minimum reduziert werden sollten.

5.1.3 AR-HMD Systeme

Hinsichtlich der für eine AR-Applikation im Rahmen von Zustelltätigkeiten einzusetzenden Geräten verhält es sich, wie bereits oberhalb angemerkt, so, dass diese einen möglichst geringen Einfluss auf die Bewegungsfreiheit des/der Nutzer*in haben und hinsichtlich der Fortbewegungsart multimodal einsetzbar sein sollen. Darüber hinaus sollen diese natürlich auch nur in einem möglichst geringen Maße ablenken. Wie bereits erwähnt, stellen sich hierbei (See-Through-)HMDs als optimal dar: aufgrund der Tatsache, dass die Hände frei bleiben und dass durch See-Through-HMDs auch bei einem möglichen Ausfall der Applikation oder Hardware weiterhin ein Blick auf die Umgebung garantiert werden kann. Einschränkend muss zu HMDs an dieser Stelle zwar gesagt werden, dass etwaige Kabel zur Verbindung mit der Berechnungseinheit, etwa einem Handy, als störend empfunden werden können, dies jedoch im Vergleich zu allen anderen Möglichkeiten nur einen geringen negativen Punkt darstellt.

Um nun weiters valide Aussagen und ein gesamtheitliches Bild garantieren zu können, sollen an dieser Stelle noch die aktuell verfügbaren (nicht proprietären) HMD-Lösungen hinsichtlich deren Verfügbar- als auch Leistbarkeit betrachtet werden. Einschränkend sei zu diesen Punkten jedoch erwähnt, dass diesen im Rahmen der Arbeit eine andere Gewichtung zukommt, als wenn eine Anschaffung solcher Geräte in großen Stückzahlen von einer Firma gemacht werden würde, da es bei Abnahme großer Stückzahlen oftmals so ist, dass sich preislich ein anderes Bild ergibt. Darüber hinaus eröffnet sich eher die Möglichkeit, auch

Hersteller, die proprietären Lösungen anbieten, heranzuziehen, da diese etwa, wie bei proprietären Lösungen nicht unüblich, die mögliche Beschränkung der Bereitstellung eigener Applikationen für deren Hardware aufheben können beziehungsweise direkt die Entwicklung und Bereitstellung der gewünschten Applikation durchführen können¹⁸⁷. Geht man nun aber dazu über, die aktuell am Markt verfügbaren, bekannten Lösungen, bei welchen sich es um Smart-Glasses mit echten AR-Funktionen¹⁸⁸ oder einfach nur AR-Glasses handelt, so stellt sich an dieser Stelle eigentlich nur noch die Wahl zwischen den beiden Herstellern Rokid oder XREAL (vormals NReal). Stand Mitte April 2024 stellt es sich so dar, dass Rokid nur ein Modell, die sogenannte Rokid MAX und Zubehör für diese anbietet und sich der Straßenpreis auf etwa 450€ beläuft. Im Gegensatz hierzu bietet XREAL eine große Palette an Brillen¹⁸⁹, die sogenannte XREAL Air und deren Weiterentwicklung die XREAL Air 2 (Pro/ Ultra)¹⁹⁰ an. Die Preise dieser Modelle belaufen sich hierbei, ähnlich wie bei der Rokid MAX, für die Air 2 auf 399\$ (459€), Air2 Pro 449\$ (515€) und für die Air 2 Ultra auf 699\$ (Stand 07. Juni 2024).

Grundsätzlich sind sich die Rokid Max und XREAL Air 2 (Pro) aus technischer Sicht, soweit Informationen hierzu zur Verfügung stehen, relativ ähnlich: Es handelt sich um See-Through-Modelle mit einem FOV $\sim 50^\circ$, einer Auflösung von 1920x1080 pro Auge, sowie Lautsprechern und der Notwendigkeit der Verbindung zu einem Berechnungsgerät mittels USB-C Kabel. Alleinstellungsmerkmale gibt es hingegen hinsichtlich der Möglichkeit der Einstellung einer Dioptrienstärke für Personen mit Sehbehelf (Rokid MAX) oder dem Dimmen der Brillengläser (XREAL Air 2 Pro). Benötigt man nun jedoch weiterführende Informationen, treten an dieser Stelle jedoch schon erste Schwierigkeit auf: Wie oberhalb angemerkt bekommt man relativ einfach Informationen zu Gewicht, Auflösung, Frame-Rate, Field of View und anderen für 0815-Nutzer*innen interessante Fakten. Möchte man

¹⁸⁷ Aus eben jenen Gründen und dem zumeist höheren Preis werden in weiterer Folge keine proprietären Lösungen betrachtet werden, auch wenn diese zum Teil den Vorteil bieten, dass keine externe Berechnungseinheit benötigt wird.

¹⁸⁸ Hierzu sei erwähnt, dass etwa auch Smart-Glasses, wie etwa die „Ray-Ban Meta Smart Glasses“ existieren, die nur im sehr beschränkten Maße zu AR gehörig zu definieren sind, da diese „nur“ über eine Kamera zur Aufnahme von Sachverhalten und einem Mikrofon und Lautsprechern zur Interaktion mit dem Meta-Netzwerk dienen, aber keine visuellen Inhalte anzeigen können.

¹⁸⁹ Eine weitere AR-Brille von XREAL war darüber hinaus die XREAL Light, welche ihrerseits den Vorgänger der XREAL Air2 Ultra darstellte. Diese ist aber seit geraumer Zeit nicht mehr lieferbar, beziehungsweise wurde diese zuletzt auch aus dem Sortiment entfernt.

¹⁹⁰ Die Modelle der zweiten Generation unterscheiden sich hinsichtlich ihrer Features, wobei der Unterschied zwischen der normalen Air2 und Air2 Pro darin liegt, dass die Air2 Pro die Möglichkeit bietet die Brillengläser zu dimmen. Die Ultra-Version wiederum bietet dann zusätzlich zur Dimmbarkeit der Gläser noch eine integrierte Kamera.

Zur Ultra Version sei an dieser Stelle zusätzlich erwähnt, dass diese um einiges später auf den Markt kam als die anderen beiden Air 2 Modelle (Ende 2023 für Air2 (Pro) im Vergleich zu Juni 2024) und deswegen nicht in die Auswahl der für das AR-System einzusetzenden Hardware miteinbezogen werden konnte. Grundsätzlich wäre es aber, wenn diese bei Beginn der Arbeit bereits verfügbar gewesen wäre, so gewesen, dass die Wahl auf dieses Modell gefallen wäre. Der Grund hierfür liegt im Vorhandensein der Kameras im HMD, der sich daraus ergebenden Vereinfachung der Umsetzung der eigenen Applikation und der Möglichkeiten der Nutzung von Pattern-Recognition mit diesem Modell.

jedoch etwa Informationen zur IMU, gestaltet sich dies bereits sehr kompliziert, da dies zwar für Entwickler, aber nicht für 0815-Nutzer*innen von Relevanz ist und demnach nicht explizit bei den Produktinformationen (und teils auch nur sehr versteckt in den Dokumentationen für Entwickler) aufgeführt wird. Ebenso komplex ist es, wenn es um Informationen und Dokumentationen rundum die Entwicklung für/ mit den AR-Glasses geht. Zwar stellen beide Hersteller SDKs zur Entwicklung, ebenso wie Dokumentationen zu den SDKs, bereit, jedoch ist die Dokumentation gerade bei Rokid als durchwegs ausbaufähig (in Hinblick auf Anwendungsbeispiele, Details zu Funktionen, ... und der Aktualität) zu bezeichnen, während XREAL hier eine relativ gute Figur macht¹⁹¹.

Hinzu kommt an dieser Stelle bei oberhalb besprochenen AR-Glasses noch die Notwendigkeit eines Smartphones als Berechnungseinheit¹⁹² für die später darzustellenden Inhalte. Hierbei ist es jedoch zum einen wichtig, dass das Smartphone mit den AR-Glasses kompatibel ist und weiters das Smartphone von den später für die Entwicklung der App benötigten SDKs unterstützt wird. Demnach ist es schwierig, eine konkrete Aussage über die möglichen zu verwendenden Geräte und deren Preise zu machen, jedoch kann bei Neugeräten von einem Preis, welcher ab 500€ startet, ausgegangen werden, da (offiziell) zumeist nur die (neuesten) Topmodelle der jeweiligen Hersteller unterstützt werden. Hierzu muss an dieser Stelle ergänzend erwähnt werden, dass AR-Glasses Produzenten aufgrund deren geringer Firmengröße¹⁹³ relativ wenige Geräte testen und es darum sein kann, dass auch Geräte abseits deren offizieller Statements funktionieren. Als jedenfalls notwendig zur Verwendung von AR-Glasses ist jedoch, wie bei jedem visuellen Ausgabegerät, welches an ein Smartphone angeschlossen wird, dass das Smartphone über USB-C mit DP-ALT Mode verfügt, da über simples USB-C allein keine Bildinformationen übertragen werden können¹⁹⁴. Einige Geräte, die von den oberhalb erwähnten AR-Glasses unterstützt werden, sind dabei etwa die Samsung Galaxy S21/ S22/ S23 Reihe und die OPPO X3 Serie neben vielen anderen Geräten¹⁹⁵. Noch komplexer wird es, wenn man selbst für die AR-Glasses entwickeln möchte, da hier eine weitere Ebene, nämlich jene, dass Geräte zum einen von den AR-Glasses unterstützt werden und gleichzeitig auch alle für die App-Entwicklung benötigten SDKs funktionieren müssen, hinzukommt, was aber nicht zwangsläufig der Fall ist. Insgesamt gestaltet sich der Prozess rundum die Auswahl des richtigen Smartphones zur

¹⁹¹ Hinsichtlich der SDK-Dokumentationen siehe: (Rokid, kein Datum a; XREAL, kein Datum c)

¹⁹² Natürlich könnte an dieser Stelle theoretisch auch eine Spielekonsole, Tablet oder einen Laptop genutzt werden, wobei sich im Feld der Zustellungen jedoch ein Smartphone als beste Lösung (Mobilität) darstellt.

¹⁹³ ... im Vergleich zu anderen Firmen im IKT-Bereich

¹⁹⁴ Siehe bezüglich USB-C und DP-ALT Mode auch: (uniaccessories, 2023)

¹⁹⁵ Ergänzend hierzu siehe die Quellen der jeweiligen Hersteller: (Rokid, kein Datum b; XREAL, kein Datum a; XREAL, kein Datum b)

Der Blick lohnt an dieser Stelle auch auf die offiziellen Foren der Hersteller, etwa auf Reddit oder Discord, da die Nutzer*innen-Community dort selbst damit angefangen hat, Listen mit unterstützten Geräten, deren Firmware und Softwareversion (da bereits Abweichungen in letzteren beiden, teils abhängig von der Region, in der das Gerät ausgeliefert wird zu Problemen führen kann) zu erstellen.

Verwendung mit AR-Glasses also wie der Hasenbau in Alice im Wunderland, welcher immer tiefer und tiefer geht und in dem Orientierung und Navigation (d.h. die Auswahl) ohne konkrete Zielsetzung und guter Vorabinformation nicht möglich ist.

5.2 Design-Idee

Nachdem nun eingangs die grundlegenden Funktionen dargestellt und die für eine AR-Applikation für Zustelldienstleistungen anwendbaren, und im Rahmen dieser Arbeit (d.h. außerhalb eines Unternehmens) zur Verfügung stehenden Geräte beleuchtet wurden, ist es nun in weiterer Folge nötig, das konkrete Projekt genauer zu definieren und zu planen. Das Projekt und die in dessen Folge erstellte Applikation soll unter dem Namen „**DelivAR**“ firmieren und vor allem auf Zustelltätigkeiten im Bereich der postalischen Zustellung und Paketzustellung zugeschnitten werden – aber nicht ausschließlich, da die allermeisten Zustelldienstleistungen, wie gezeigt wurde, ähnliche Anforderungen aufweisen. Als ein erster initialer Planungsschritt wurden bereits die im Bereich der Hardware zur Verfügung stehenden Optionen an AR-Brillen mit Blick auf die für diese benötigten Berechnungseinheiten, als auch die Möglichkeiten zur Entwicklung von eigenen Applikationen dafür und die sich daraus ergebende Bedingungen (bspw. Bindung an Softwarepakete o.ä.) kurz betrachtet. Anschließend daran soll es nun darum gehen, die konkrete, für die Umsetzung benötigte Hard- und Software auf der einen Seite und die Gestaltung der Applikation in dem Sinne, was die Nutzer*innen sehen (und die Applikation bedienen) sollen, auf der anderen Seite zu definieren, bevor zur eigentlichen Erstellung der Applikation übergegangen werden kann. Ebenso soll eine kurze Einordnung der eigens entwickelten Applikation hinsichtlich der theoretischen Gliederung von AR stattfinden.

5.2.1 Auswahl der Hard- und Software

Hinsichtlich der Auswahl an Hardware gestaltete es sich im Rahmen des Projekts so, dass, wie oberhalb bereits erwähnt, grundsätzlich nur zwei beziehungsweise drei AR-Brillen für das geplante Projekt in die engere Auswahl kamen: die Rokid Max und die XREAL Air 2 (Pro)¹⁹⁶. Die Wahl fiel in diesem Fall dann, obwohl der Faktor der Korrektur der Fehlsichtigkeit bei der Rokid Max für Personen wie mich, die an Fehlsichtigkeit leiden, attraktiv ist, auf die



Abbildung 24: Zum Einsatz kommende Hardware (XREAL AIR 2 & Samsung Galaxy S21 5G)

¹⁹⁶ Die Auswahl hinsichtlich der AR-Brille wäre aller Wahrscheinlichkeit nach anders ausgefallen, wäre die XREAL Air 2 Ultra zum Zeitpunkt der Bearbeitung des Projekts bereits oder die XREAL Light noch verfügbar gewesen. Der Grund hierfür liegt in der Möglichkeit der Verwendung der Kameras in der Brille zum Zwecke der Pattern-Recognition und dem einfacheren Aufbau des Projekts in Unity.

XREAL Air 2¹⁹⁷ aufgrund der besseren Dokumentation hinsichtlich der Entwicklung eigener Apps, deren Aktivitäten hinsichtlich der Fehlerbehebung, Updatebereitstellung und Weiterentwicklung der eigenen Produkte und zu guter Letzt aufgrund der Größe der Entwickler*innen-Community und der Aktivität derselbigen, welche bei Problemen gegebenenfalls Hilfestellung leisten könne. Die Wahl des Geräts zur Berechnung der Inhalte fiel weiters nach mehrstündigen Analysen hinsichtlich der Unterstützung durch die XREAL Air 2 und die Lauffähigkeit der für die Umsetzung zu verwenden geplanten SDKs auf ein Samsung Galaxy S21 5G da es zu diesem Zeitpunkt günstig gebraucht (ca. 200€) zu erstehen war.

Bezogen auf die softwareseitigen Überlegungen war ebenso relativ rasch, direkt nach Betrachtung der zur Verfügung stehenden AR-Glasses klar, dass als Entwicklungsumgebung auf Unity gesetzt werden würde, da nur für diese von den beiden Herstellern Rokid¹⁹⁸ und XREAL¹⁹⁹ SDKs herausgegeben werden. Hiernach war es dann nötig sich zu informieren, ob etwaige Limitationen bei der Entwicklung von Applikationen für Smartphone-HMD Kombination existieren, wobei, wie bereits erwähnt, solche im Rahmen der Unterstützung des Smartphones der zur Verwendung geplanten SDKs vorgefunden wurden. Konkret ist es, um eine AR-Applikation mittels Unity für Smartphones entwickeln zu können, so, dass die Geräte, auf welchem die App schlussendlich ausgeführt wird, zumindest durch Googles ARCore (für iOS ARKit) unterstützt werden muss. Anschließend daran ging es um die Frage nach dem Aufbau der Navigationsabfragen in der Applikation, wobei hierzu hinsichtlich Erweiterungen für Unity, die eben dies ermöglichen, recherchiert wurde. Hierbei wurden zwei mögliche Lösungen, nämlich die eingestellte Erweiterung von Google²⁰⁰ und die nicht gewartete, aber zumindest noch offiziell zur Verfügung stehende, großteils funktionierende und kostenfreie²⁰¹ Erweiterung von Mapbox²⁰² aufgefunden und letztere dann auch verwendet.

In einem dritten und letzten Schritt ging es dann nochmals, wie bereits in der Einarbeitung bezüglich der praktischen Umsetzung darum, sich hinsichtlich bereits bestehender praktischer AR-Navigationsprojekte zu informieren. Hierbei wurde bereits zu Beginn die von Daniel Fortes veröffentlichte und sehr gut dokumentierte, aber kostenpflichtige Unity Erweiterung Namens „AR + GPS Location“²⁰³ gefunden, welche es ermöglicht ein Google-

¹⁹⁷ Wie erwähnt unterscheidet sich die XREAL Air 2 Pro nur im Hinblick auf die Dimmbarkeit der Gläser von der normalen Air 2.

¹⁹⁸ Siehe: (Rokid, kein Datum a)

¹⁹⁹ Siehe: (XREAL, 2024)

²⁰⁰ Siehe hierzu: (Google, 2024)

²⁰¹ Dies gilt, solange eine bestimmte Anzahl an Routingabfragen pro Monat nicht überschritten wird.

²⁰² Siehe: (Mapbox, 2019)

²⁰³ Siehe hinsichtlich dieses Projekts die folgenden Quellen: (Fortes, 2020; Fortes, 2022a; Fortes, 2022b; Fortes, 2023). Weiters sei zu dieser Erweiterung angemerkt, dass diese zum Teil auf Funktionen von Mapbox hinsichtlich des Routings und der Kartenfunktionen aufbaut. Dies bedeutet weiters, dass die ARGPS Erweiterung bereits die

Live View ähnliches Projekt in Unity für Smartphones umzusetzen. Da sich durch diese ein großer Teil an Arbeit und Zeit einsparen ließ²⁰⁴, wurde in weiterer Folge beschlossen, diese Erweiterung zu erwerben. Dadurch mussten die grundlegenden Funktionen des Routings²⁰⁵ und der Anzeige von Inhalten, welche ihrerseits in der ARGPS-Erweiterung wieder Teils auf der Mapbox Erweiterung basierten für die eigene Umsetzung nicht komplett von Grund auf entwickelt, sondern nur umgebaut werden. Umgebaut werden mussten diese dabei in dem Gesichtspunkt, dass die Inhalte anstelle über das Smartphone Display auf der AR-Brille angezeigt wurden und die Applikation insgesamt ohne Kamera funktioniert.

5.2.2 Überlegungen bezüglich des Aussehens der Applikation

Hinsichtlich der Gestaltung der Applikation muss in einem ersten Schritt einmal erwähnt werden, dass dieser Punkt in zwei Teile zu unterteilen ist: So geht es zum einen um die Benutzeroberfläche am Smartphone, mithilfe derer man die Applikation startet, Einstellungen trifft, Funktionen überwacht und vieles mehr, zum anderen um die Gestaltung der eigentlichen anzuzeigenden Navigationshinweise mittels AR-Brille.

Gestaltung der Benutzeroberfläche

Da die Benutzeroberfläche eine Vielzahl an Optionen²⁰⁶ aufweisen musste wurde beschlossen die einzelnen Einstellungen in sogenannte Tabs zu unterteilen. Demnach sollten neben einem Button für den Start/ Stop der Applikation, einem nur zeitweise aktivierten Button zur Bestätigung von Zustellhinweisen, einem Trackpad zur Eingabe von Inhalten im virtuellen Raum und einem Feld für Hinweise²⁰⁷

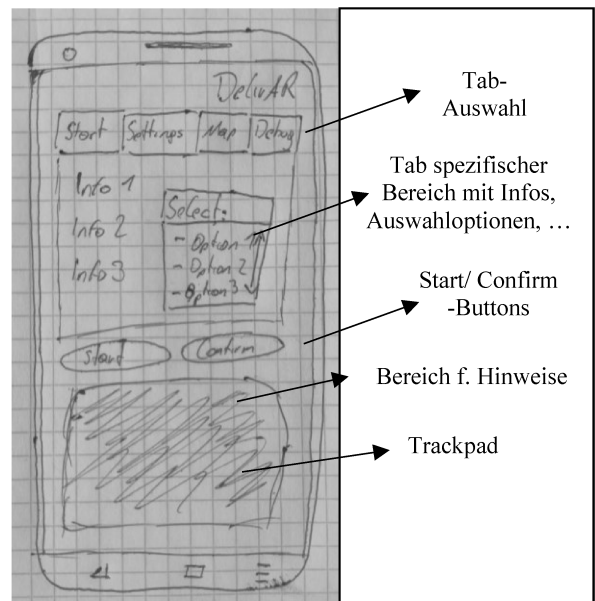


Abbildung 25: Skizze der Benutzeroberfläche

Mapbox SDK (zumindest in Teilen) beinhaltet und letztere somit nicht zwangsläufig extra (außer es treten Fehler, welche auf fehlende Imports zurückzuführen sind auf) eingebunden werden muss.

²⁰⁴ Dadurch mussten etwa keine Überlegungen und Codeabschnitte zu Themen wie...

- Wie bringt man Routen in die Applikation?
 - Wie müssen diese verarbeitet werden?
 - Wie ist die grundlegende Funktionsweise der Navigation auszugestalten?
- ... bearbeitet werden.

²⁰⁵ Hiermit sind die beiden Funktionen des Routings entlang einer bestimmten Route und das Routing vom aktuellen zu einem gesuchten Standort gemeint.

²⁰⁶ Ein Teil davon war zugegebenermaßen nur in der Entwicklungsversion enthalten, da diese nur für die Entwicklung aber nicht für den/ die Nutzer*in relevant waren.

²⁰⁷ Das Textfeld für Hinweise wird in weiterer Folge dem Trackpad überlagert dargestellt werden.

vier Tabs erstellt werden. Diese Tabs und deren Funktionen können dabei der angefügten Skizze²⁰⁸ (Abb. 25) und der nachfolgenden kurzen Beschreibung entnommen werden:

Start-Tab:

Nach dem Start der Applikation gelangt man zur eigentlichen Benutzeroberfläche der Applikation, wobei der Start-Tab vorausgewählt ist. Dieser soll dem/ der Nutzer*in die Möglichkeit bieten, die gewünschte Funktion mittels Dropdown-Menü auszuwählen: a) Navigation entlang einer **vorab definierten Route**; b) Navigation zu einem **gesuchten selbst ausgewählten Ort**; c) **Platzierung eines Objekts** an einem koordinativ verorteten Punkt/ oder vorab definierten Punkt²⁰⁹. Weiters finden sich dann abhängig von der ausgewählten Funktion weitere Einstellungen, die getroffen werden müssen, um danach die jeweilige Funktion starten zu können vor: Bei vorab definierten Routen bezieht sich dies auf das Ein- / Ausschalten von Zustellhinweisen; bei freier Navigation auf die Eingabe des konkreten Zielorts; bei Positionierung von Objekten auf die Eingabe von Koordinaten/ die Auswahl eines vorab definierten Orts.

Settings-Tab:

Unter den Einstellungen ergeben sich Möglichkeiten zur gegebenenfalls notwendigen Korrektur der Anzeige²¹⁰ zur Anzeigehöhe der Navigationselemente²¹¹, zum Smoothing²¹², zur Distanz der Anzeige von Zustellhinweisen²¹³ und zum visuellen Aussehen der Navigationshinweise. Hinsichtlich letzterem Punkt soll dann nochmals die Auswahlmöglichkeit bestehen, ob nur eine Route, nur POIs oder beides gemeinsam angezeigt wird, sowie die Möglichkeit, die Farbe der Navigationshinweise und deren Größe zu definieren.

²⁰⁸ Die Skizze wurde absichtlich sehr grob gehalten, da von vornherein klar war, dass sich die in den einzelnen Tabs befindlichen Bestandteile während der Entwicklung hinsichtlich Aussehens und Anordnung mehrmals ändern würden.

²⁰⁹ Dieser letzte Punkt der Positionierung eines Objekts wurde rein zum Testen eingeführt, aber aus eigenem Interesse über die eigentliche Arbeit hinaus noch etwas ausgebaut. Ziel war es, (geplante) Gebäude in Lebensgröße an bestimmten Orten platzieren und somit begehbar machen zu können. Etwaige Einsatzszenarien stellen sich hierfür etwa in der Architektur oder am Bau in der Baufortschrittsüberwachung mithilfe von BIM dar. Insgesamt funktionierte dies auch durchwegs gut, wobei die Ansicht jedoch etwas wackelig (aufgrund fehlender „Smoothing-Funktionen“) war und ein leichtes Offset hinsichtlich der Drehachse festzustellen war. Da dies alles aber nicht im Rahmen und Ziel der Arbeit inkludiert war, sei an dieser Stelle mit der Besprechung an dieser Stelle geendet.

²¹⁰ ...für den Fall, dass eine der drei Drehrichtungen Probleme machen sollte.

²¹¹ ...für unterschiedlich groß gewachsene Menschen, da ansonsten die Elemente ggf. zu hoch/ zu niedrig angezeigt werden.

²¹² Dies beschreibt, wie „wackelig“ die angezeigten Inhalte sind. In anderen Worten bezieht sich dies darauf, wie schnell die Aktualisierungsrate der Elemente ist und ob die Bewegung von Elementen von A nach B sprunghaft ist oder ob eine Bewegung zwischen den Orten interpoliert wird.

²¹³ In anderen Worten: In welchem Radius zum tatsächlichen Zustellort (etwa die Haustüre eines Hauses) über eine zuzustellende Sendung informiert werden soll.

Map-Tab:

Der dritte Reiter der Applikation ist der Map-Reiter, in welchem eine 2D-Karte angezeigt werden soll, um weiters auch auf eine gewohnte Präsentationsform für die Navigation zurückgreifen zu können. Auf der Karte soll dabei zum einen nicht nur die aktuelle Position, sondern auch die hierfür verwendete Route angezeigt werden können, wenn eine Navigationsaufgabe gestartet wurde. Ebenso soll in diesem Fenster die aktuelle Position als Koordinatenpaar (in WGS84-Koordinaten) angezeigt werden.

Debug-Tab:

Hierbei handelt es sich um einen Reiter, welcher nur zum Zwecke der Entwicklung der eigentlichen Applikation und für die Überwachung der korrekten Funktion selbiger erstellt werden soll. Hierzu sollen etwa Orientierungsparameter und andere relevante Inhalte angezeigt werden.

Gestaltung der Navigationshinweise

Hinsichtlich der Gestaltung der Navigationshinweise an sich soll sich an bereits umgesetzten AR-Navigationsapplikationen für Smartphones orientiert werden, was im Umkehrschluss bedeutet, dass es unabdingbar ist, visuelle Navigationshinweise an den Kreuzungspunkten eines Weges anzuzeigen, gegebenenfalls ergänzt um weitere Informationen wie etwa der Distanz bis zur nächsten Wegkreuzung. Hinzu kommt, wie etwa bei Googles Live View, die Anzeige des Weges von einem Kreuzungspunkt zum nächsten, wobei an dieser Stelle erwähnt werden muss, dass hierfür zum einen entweder eine sehr genaue Route mit einer großen Anzahl an Wegpunkten, damit die Route korrekt verläuft, benötigt werden dürfte, oder aber Pattern Recognition, welche wie erwähnt nicht möglich ist, da die zu verwendende AR-Brille keine Kamera aufweist zu verwenden wäre²¹⁴. Hinzu kommt hinsichtlich der Navigationsanweisungen noch, wie auch in den zuvor erwähnten Arbeiten gezeigt wurde, dass es als optimal anzusehen wäre, die Navigationshinweise nicht nur visuell, sondern auch mittels Sprachausgabe darzulegen, wobei an dieser Stelle der Aufwand nicht abschätzbar ist und zu sehen sein wird, inwiefern dies umsetzbar ist. Ebenso wurde in den vorangegangenen Arbeiten gezeigt, dass Nutzer*innen zwar nicht signifikant besser performen, wenn sie zusätzlich zu AR-Inhalten auch eine 2D-Karte zur Verfügung haben, jedoch das Nutzerempfinden positiver ist, weshalb aus diesem Grund eine solche ebenso, wie es bei der Sprachausgabe der Fall ist, als durchwegs positiv anzusehen ist, falls diese eingebunden werden kann.

Ein weiterer wichtiger Punkt im Hinblick auf die Anzeige von Navigationshinweisen ist dann noch jener, dass die einzelnen Hinweise sich nicht überlagern und den Bildschirm

²¹⁴ Der Grund hierfür liegt darin, dass eine Linie per se immer nur gerade von einem Punkt zum nächsten dargestellt werden kann und Kurven somit, falls es sich um den normalen Straßenverlauf handeln würde, nicht dargestellt werden könnten. Dies kann entweder, wie in den GISc, ermöglicht werden, indem mit einer Vielzahl an Punkten (Vertices), die die Kurve definieren, gearbeitet wird oder etwa über Pattern-Recognition, wobei die Straße erkannt wird und die App „versteht“, dass sie den Weg trotzdem weiterhin auf der Straße anzeigen muss.

„zumüllen“. Dies kann nämlich zur Folge haben, dass die einzelnen Inhalte nicht mehr lesbar sind und zum anderen das Blickfeld so überladen wird, dass die Betrachter*innen zu stark von der Umgebung (etwa dem Straßenverkehr) abgelenkt werden, und es somit gefährlich wird die Applikation zu nutzen. Auf der anderen Seite dürfen die Hinweiseinblendungen auch nicht zu klein sein, da diese ansonsten übersehen werden könnten. Ein weitere Funktion, welche zusätzlich zu den bereits bei Smartphone AR-Navigationsapplikationen umgesetzten Funktionen hinzukommt, ist dann noch jene der Einblendung von Zustellhinweisen, wobei diese zum einen nur wenn man sich in einem gewissen räumlichen Gebiet (bspw. 5m rundum die Zustelladresse) befindet, angezeigt werden sollen und zusätzlich bestätigbar sein müssen, sodass die Zusteller*innen angeben können, die Hinweise bezüglich der Zustellung gelesen und zur Kenntnis genommen zu haben.

Hinsichtlich der Gestaltung der Einblendung ist in einem ersten Schritt einmal geplant, mit Hinweistafeln und Pfeilen in blauer Farbe, wie etwa bei Google Live View (und bereits in ARGPS implementiert), zu arbeiten, während als Wegmarkierungen eine Linie an blauen Punkten angezeigt werden soll. Der Hinweis bezüglich der Zustellanmerkung soll überdies einfach als Einblendung am Rande des Sichtfeldes platziert werden. In weiterer Folge soll es dann für die Nutzer*innen auch möglich sein, aus einer definierten Palette an Farben zu wählen, genauso wie zu bestimmen, ob sie die Einblendung des Wegs aktiviert haben möchten oder einfach nur die Anzeige der Navigationshinweise an Knotenpunkten präferieren.

5.2.3 Theoretische Einordnung der Applikation

Hinsichtlich der theoretischen Einordnung der geplanten Umsetzung gestaltet es sich so, dass es sich ohne Frage um eine AR-Applikation handelt, da diese definitiv die drei Punkte der Verknüpfung virtueller und realer Informationen, Interaktivität in Echtzeit und Registrierung in einer 3D-Umgebung erfüllt. Weiters ist die geplante Applikation eine mobile-AR Applikation, da sie nicht an einen fixen Ort gebunden ist, und ist dem Typus der „*Creation of an artificial environment*“ einzuordnen, da in der Umwelt in dieser Form nichtexistierende Elemente angezeigt werden. Hinsichtlich des Typs an AR-Applikation findet sich die geplante Applikation weiters im Bereich einer Location-Based AR-Applikation wieder, da, wie bereits angemerkt und weiters zu sehen sein wird, die gesamten Navigationsaufgaben auf GNSS-Abfragen und der daraus abgeleiteten Positionierung von Inhalten, ohne jegliche Anwendung eines Kamerasystems, basiert.

5.3 Erstellung der Applikation^{215 216}

Nachdem nun bereits die, der eigentlichen Applikationserstellung vorangegangenen Überlegungen hinsichtlich notwendiger Funktionen, der Art der Realisierung und der weiterführenden Planungsschritte im Hinblick auf die zu verwendende Hard- und Software und erste Überlegungen bezüglich der Gestaltung der einzublendenden Inhalte angestellt wurden, soll es nun in weiterer Folge um die Applikationserstellung an sich gehen. Hierfür soll an dieser Stelle auch insbesondere auf das dargestellte Diagramm, welches den Ablauf der Erstellung schematisch darlegt, hingewiesen werden, bevor einige Worte zu den einzelnen Schritten verloren werden:

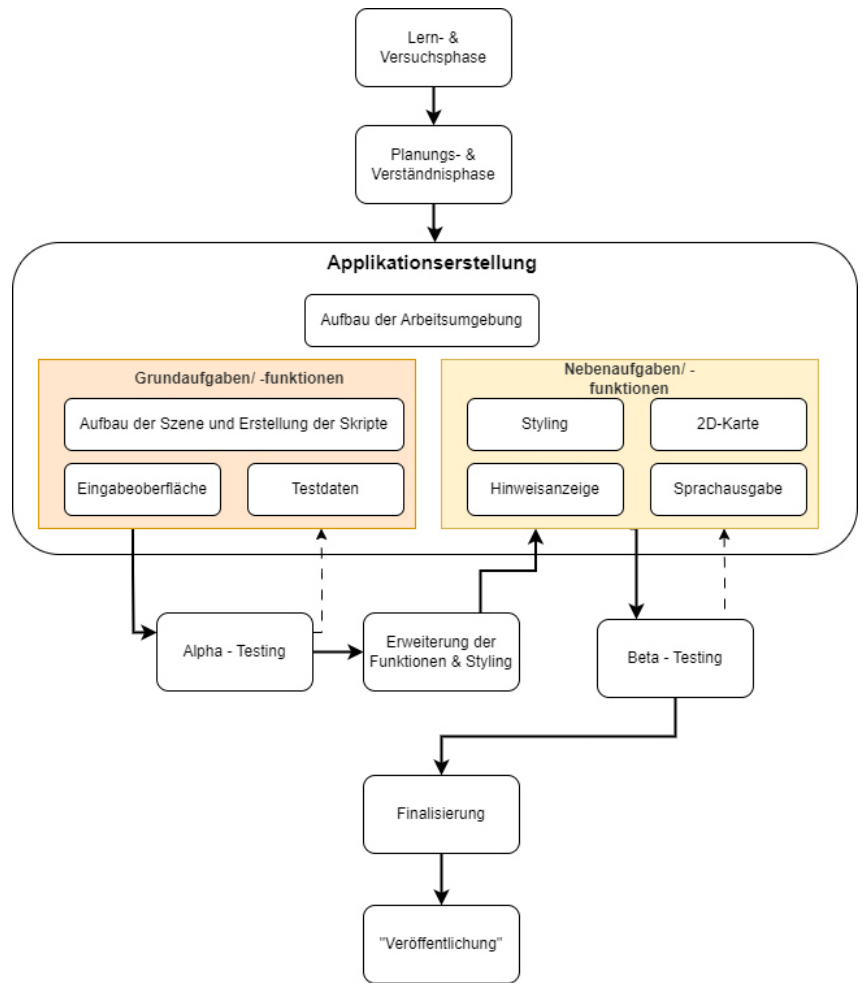


Abbildung 26: Ablaufdiagramm zur Erstellung der Applikation

5.3.1 Lern- und Versuchsphase

In der ersten Phase der Ausarbeitung ging es erstmal darum, sich mit Unity per se weitergehend²¹⁷ auseinanderzusetzen und die Funktionalitäten kennenzulernen. Hierzu

²¹⁵ Wenn in weiterer Folge von der „Anzeige im virtuellen Raum“/ „Anzeige im AR-Raum“ gesprochen wird, ist an dieser Stelle die Anzeige auf der AR-Brille gemeint. Technisch gesehen handelt es sich hierbei nämlich um einen dreidimensionalen Raum, welcher ausgehend vom Startpunkt rundum die Brille erstellt und dann mit den gewünschten Objekten/ Inhalten befüllt wird. Über Trackingfunktionen (GPS + IMU) und die Umrechnung von Welt- in Unitykoordinaten wird es eben dann möglich, sich durch diesen Raum zu bewegen.

²¹⁶ An dieser Stelle sei für Interessierte Personen auf die kommentierte Darstellung eines kleinen Teils des Codes für die 2D-Karte auf der Bedienoberfläche und der kommentierten Darlegung der die Unity Szene aufbauenden Objekte im Appendix verwiesen.

²¹⁷ Grundkenntnisse waren dank eigener Einarbeitung einige Zeit zuvor bereits vorhanden, da andernfalls ein Projekt in diesem Umfang vermutlich (in einem beschränkten Zeitraum) kaum zu realisieren gewesen wäre.

wurden die „Unity Learn“ Inhalte, allen voran der „Unity Essentials“, „Junior Programmer“ und „Mobile AR Development“ Pathway genutzt²¹⁸. Anschließend an das Erlernen der Grundfunktionalitäten von Unity ging es dann daran, sich mit den jeweiligen für die Umsetzung ausfindig gemachten Erweiterungen auseinander zu setzen und sich mit diesen bekannt zu machen. Hierzu wurde so vorgegangen, dass zum einen die Dokumentation der jeweiligen Erweiterungen durchgelesen wurde und hinsichtlich deren Funktionalitäten, welche in weiterer Folge von Relevanz sein würden, notiert wurden, als auch die glücklicherweise vorhandenen Tutorials erarbeitet wurden²¹⁹, wobei hier auch direkt der nächste Schritt anschließt: die Phase des Planens und Verstehens.

5.3.2 Planen & Verstehen²²⁰

In diesem Schritt ging es darum, basierend auf der vorangegangenen Lern- und Versuchsphase zu verstehen, wie die einzelnen SDKs genutzt werden können, wie sie funktionieren und weiters für die eigenen Zwecke umgearbeitet werden können. Hierauf aufbauend wurde dann eine Übersicht erstellt, welche Funktionen vorhanden waren, wie diese falls nötig angepasst werden mussten und ob komplett neue Funktionen nötig sein würden und falls ja, wie diese aufgebaut werden könnten. Als weiterer Schritt, nachdem dieses Wissen akkumuliert worden war, ging es dann noch darum, die Must-Have Funktionen und die Nice-To-Have Funktionen zu definieren: Hierbei wurden die 2D-Kartenfunktion, Sprachausgabe und die Gestaltungsfunktionen (Farbe, Größe, ...) der Navigationshinweise als Nice-To-Have Funktionen und die gesamten reinen Navigationsfunktionen und die Anzeige von Zustellhinweisen als Must-Have Funktionen definiert.

Bei dieser Analyse kam zum Vorschein, dass grundsätzlich hinsichtlich der **Funktionalitäten der Navigation** zu weiten Teilen auf die ARGPS-Erweiterung von Daniel Fortes zurückgegriffen werden konnte. Diese muss jedoch insofern angepasst werden, als dass zum einen die Funktionen anstelle über das Kamerasystem²²¹ des Smartphones über das Kamerasystem der AR-Brille funktionieren und zum anderen eine Lösung für das

²¹⁸ Zu den Inhalten siehe: (Unity, 2024)

²¹⁹ Siehe hierzu: (Fortes, 2020; Mapbox, 2019; XREAL, 2024)

²²⁰ Der Konjunktiv und die Verwendung der Zukunftsform im Rahmen dieses Kapitels bezieht sich darauf, dass es sich in dieser Phase um die Planung handelte und erst im Anschluss an die Planungsphase die praktische Umsetzung der geplanten der Änderung bedürftenden Codeabschnitte gemacht wurde.

²²¹ Ein Kamerasystem stellt im Sinne von Unity keine tatsächliche Kamera dar. Viel mehr definiert eine Kamera im Kontext von Unity oder genereller 3D-Grafikprogrammen und Realtime-Engines den Ausschnitt, welcher aufgenommen wird und auf welchem Ausgabegerät in welcher Weise (also etwa mit gewissen (Farb-)Filtern) dieser angezeigt wird. Im Falle des vorliegenden Projekts existierten simplifiziert – simplifiziert, da alleine die XREAL Air 2 im Projekt aus drei Kameras besteht und auch für die Anzeige am Smartphone mehrere Kameras Verwendung fanden – gesagt aus zwei Kamera(s)(systemen): Eine für die Inhalte, welche auf der AR Brille angezeigt werden sollten, und eine für die Inhalte, welche am Smartphone angezeigt werden sollten.

korrekte Abspeichern und Abrufen von vorab definierten Routen²²² und Navigationshinweisdaten selbst aufgebaut werden muss.

Ebenso wird der Code dahingehend angepasst werden müssen, als dass zwar weiterhin auf die koordinative Verortung vom Smartphone, da die AR-Brille selbst kein GPS-Modul verbaut hat, zugegriffen wird, jedoch die IMU-Werte hinsichtlich der Drehung der Brille und dem Offset zum Smartphone von der Brille abgeholt beziehungsweise berechnet werden.

Hieran schließt auch direkt die **Verknüpfung des Smartphones und der AR-Brille** an, welcher zwar zum Teil durch die “NRSDK“ Erweiterung von XREAL vereinfacht wird, aber etwaige Probleme, wie etwa die Aktualisierung der Position im Zusammenspiel mit der Verwendung der „AR + GPS Location“ Erweiterung, nicht lösen kann. Insgesamt muss hierzu demnach ein eigener Code gebaut werden, der dies löst. Hierzu kann auch direkt erwähnt werden, dass dies einen der aufwändigsten Teile dieser Arbeit darstellte, da es nicht ausreichte einen Code zu schreiben, der die Erweiterungen miteinander in den relevanten Aspekten verknüpft, sondern auch komplette Zuweisungen und Codeabschnitte in der Erweiterung von Fortes angepasst werden mussten.

Hinsichtlich des **Stylings** der auf der AR-Brille anzuzeigenden Inhalte sollen in einem ersten Schritt die von Fortes erstellten Hinweistafeln, Richtungspfeile und Wegmarkierungen weitestgehend weiterverwendet werden, da diese ohnehin als „state-of-the-art“²²³ bezeichnet werden können, sich so Arbeit erspart werden kann und diese in einem abschließenden Schritt, wo die Applikation überarbeitet wird, ohnehin nochmals „aufgehübscht“ werden können. Ähnlich gestaltet es sich darüber hinaus beim Styling der Benutzeroberfläche am Smartphone, wobei hier der grundlegenden Style angelehnt an den in der XREAL Erweiterung enthaltenen Smartphone-Controller Vorlage hinsichtlich der Farbgebung, der Buttons und des Eingabefelds Verwendung finden soll. Hinsichtlich des Aufbaus der Tabs muss jedoch ein eigenes Konzept, welches zu den vorherigen Komponenten passt, erstellt werden.

Für die **Anzeige von Zustellhinweisen** konnte bei der Analyse der zur Verwendung geplanten Erweiterungen keine Lösung gefunden werden, wobei demnach selbst eine solche implementiert werden muss. Dies sollte sich aber als relativ einfach gestalten, da hierzu einfach nur die Distanz zwischen aktueller Position und einem fix koordinativ verorteten Punkt, an welchem der Zustellhinweis auftreten soll, berechnet werden und bei Unterschreitung einer gewissen Minimumdistanz angezeigt werden muss.

Für die Realisierung einer **2D-Karte** wiederum konnte eine Lösung basierend auf der Umsetzung im Beispielsprojekt²²⁴ der „AR + GPS Location“ von Fortes – basierend auf der Mapbox SDK – gefunden werden. Zur korrekten eigenen Verwendung wird es darüber

²²² Diese wurden nach dem Vorbild eines Beispielsprojekts der ARGPS Erweiterung erstellt. Siehe hierzu: (Fortes, 2020)

²²³ ...wenn man diese etwa mit Googles Live View, oder Sygic Real View Navigation vergleicht.

²²⁴ Siehe hierzu: (Fortes, 2022a)

hinaus so sein, dass es sich hier um einen der wenigen Fälle handeln wird, bei welchem beinahe keine Änderungen am Code, abseits des Entferns nicht benötigter Abschnitte und dem neu zuweisen von Werten nötig sein wird.

Komplizierter gestaltete es sich hingegen bei der Suche nach einer Lösung hinsichtlich der **Sprachausgabe** von Navigationshinweisen, wozu sich keine der bisher zur Verwendung geplanten Erweiterungen anboten. Jedoch konnten Erweiterungen zur Lösung im Sinne von Text zu Sprache (eng. *text to speech*) gefunden werden²²⁵. Die Gründe, weshalb dieser Punkt der Erstellung einer Sprachausgabe für Navigationshinweise dann an dieser Stelle nicht weiterverfolgt und demnach in der Applikation nicht implementiert wurde, waren zum einen der bereits durchwegs als groß zu beschreibende bisher geplante Arbeitsaufwand und der Punkt, dass die allermeisten „*text to speech*“ Pakete für Unity kostenpflichtig sind.

5.3.3 Applikationserstellung - Aufbau der Arbeitsumgebung²²⁶

Der Aufbau der grundlegenden Arbeitsumgebung stellte in weiterer Folge das erste Arbeitspaket und den ersten Schritt zu Entwicklung der eigenen AR-Navigationsapplikation dar. Gearbeitet wurde hierbei auf Windows 10 22H2, auf welchem in weiterer Folge Unity3D in der Version 2021.3.32f mit Modulen zur Entwicklung für Android („*Android Build Support*“), WebGL („*WebGL Build Support*“) und Windows („*Windows Build Support (IL2CPP)*“ & „*Universal Windows Platform Build Support*“) installiert wurden. Weiters wurde Visual Studio 2019, als Entwicklungsplattform für die Codebestandteile der Applikation installiert und verwendet. Anschließend an diese grundlegenden Installationen wurde ein neues AR-Projekt (basierend auf dem von Unity hierfür bereitgestellten Template) angelegt, welches bereits grundlegende Erweiterungen wie etwa ARCore (v4.2.9), ARKit, ARFoundation und viele weitere häufig für AR-Entwicklungszwecke angewandte und teils zwingend benötigte Bibliotheken enthält. In weiterer Folge wurde dann noch die mehr oder minder bei der Entwicklung von Interfaces als 0815 zu bezeichnende Erweiterung TextMeshPro (v3.0.6) eingebunden, welche eine Vielzahl an Optionen hinsichtlich der Gestaltung eines Userinterfaces zur Verfügung stellt, bevor zur Einbindung der für die spezifisch für diese Arbeit benötigten SDKs übergegangen wurde. Hierbei handelte es sich, wie bereits erwähnt, zum einen um die NRSDK (v2.1.0) von XREAL, mit welcher es möglich wurde, die AR-Brille und deren Funktionen anzusprechen, und um die AR + GPS Location Erweiterung (v4.0.1) von Daniel Fortes, welche die Entwicklung von AR-Navigationsapplikationen für Smartphones ermöglicht²²⁷.

²²⁵ Siehe hierzu die Ergebnisse unter dem Schlagwort „*text to speech*“ im Unity Asset Store.

²²⁶ Unter Arbeitsumgebung wird im Rahmen dieser Arbeit die Einrichtung des kompletten Systems (Vorbereitung des Betriebssystems, Installation von Softwarepaketen, SDKs, ...), mit welchem es in weiterer Folge möglich wurde die Applikation zu entwickeln, verstanden.

²²⁷ Hinsichtlich der genauen Schritte und den hierbei zu beachtenden Einstellungen sei an dieser Stelle auf die jeweiligen Dokumentationen der Erweiterungen verwiesen.

Hierbei trat auch direkt ein erstes Problem auf, welches mit der Einbindung multipler nicht aufeinander abgestimmter Erweiterungen zu tun hatte und sich in Form von Problemen beim Initialisieren aller importierten Funktionen beim Start einer neuen leeren Szene²²⁸ und deren Funktionsfähigkeit zeigte. Dementsprechend musste an dieser Stelle ein Schritt zurück gemacht und zuerst einmal das Problem ausfindig gemacht werden. Dabei stellte sich heraus, dass es aus unerfindlichen Gründen bei der AR + GPS Location Erweiterungen zu Kompatibilitätsproblemen kommt, da Grundfunktionen der Mapbox SDK, auf welche diese aufbaut, nicht gefunden werden konnten und die Importreihenfolge dieser beiden Erweiterungen somit essentiell ist. Der Grund hierfür ist, dass die AR + GPS Location Erweiterungen Änderungen an Skripten bzw. Codeabschnitte vornimmt, welche erst existierten, nachdem die Mapbox SDK eingebunden und korrekt konfiguriert war, und somit korrekt überschrieben werden konnten. Nachdem der Import in der korrekten Reihenfolge durchgeführt worden war, funktionierte die Initialisierung dann zwar grundsätzlich, jedoch kam es zu einem weiteren Problem, welches darauf zurückzuführen ist, dass, wie bereits erwähnt, die Mapbox SDK nicht mehr geupdatet wird. Das Problem hierbei war, dass die Mapbox SDK auf Funktionen zuzugreifen versucht, welche in neueren Versionen von ARCore umbenannt wurden und somit Fehlermeldungen und Probleme beim Ausführen einer leeren Szene produzierte. Abermals musste also analysiert werden, woher der Fehler kam, wobei an dieser Stelle ein Dank an die immer noch aktive Community rundum die Mapbox SDK und deren Foreneinträge geht, welche auf das Problem der Umbenennung der Funktion in ARCore verwiesen und auch anmerkten, in welchen Mapbox SDK-Skripten die alte Funktion umbenannt werden musste.

Unity Projekt, basierend auf dem AR-Template

- Beinhaltet essentielle Erweiterungen wie etwa ARCore, ARKit, ARFoundation, ...

TextMeshPro

- Beinhaltet Komponenten (Codes, wie auch (Objekt-)Vorlagen) für die Interaktion (Buttons, Textfelder, Dropdown-Menüs, ...)

NRSDK

- Beinhaltet alle Komponenten zur Nutz­­barmachung der XREAL AR-Brillen und deren Funktionen

AR + GPS Location

- Beinhaltet Komponenten zur Erstellung einer AR-Navigationsapplikation auf Smartphones, basierend auf der Mapbox SDK

Mapbox SDK

- Beinhaltet alle Funktionen zur Abfrage und Berechnung von einer Route A nach B, wie auch der Umwandlung von Orten in Koordinaten über die Mapbox API-Schnittstelle

In-game Debug Console

- Beinhaltet Komponenten zur Überwachung von Fehlern während der Ausführung einer App

Abbildung 27: Darstellung der wichtigsten im Rahmen der Applikation verwendeten Erweiterungen zur reinen Unity-Umgebung

Anschließend an diese bereits eingangs auftretenden Schwierigkeiten wurde dann eine Test-Szene erstellt, aufs Smartphone überspielt und ausgeführt, um zu sehen, ob noch weitere Fehler auftreten oder nicht. An dieser Stelle wurde eine weitere Erweiterung namens „*In-game Debug Console*“ (v1.6.7) für Unity importiert, um etwaige Fehlermeldungen auch außerhalb der Entwicklungsumgebung auf Windows in der laufenden Applikation am Smartphone angezeigt zu bekommen. Anschließend daran, dass alle benötigten

²²⁸ Eine Szene bezeichnet in Unity eine Umgebung, welche spezifische Objekte und Komponenten enthält.

Erweiterungen und SDKs korrekt eingebunden waren und glücklicherweise keine Fehler bei der Ausführung am Smartphone auftraten, wurde eine Sicherungskopie der kompletten Projektordnerstruktur gemacht, um jederzeit auf diesen Datenstand zurückgreifen zu können.

Anschließend an den Aufbau der Arbeitsumgebung wurde dann eine neue Szene erstellt, in welche die durch die SDKs bereitgestellten und weiters benötigten Prefabs²²⁹ für die Eingabeoberfläche am Smartphone und die Verwaltung der AR-Brille (NRSDK) wie auch die zur Navigation an sich benötigten Komponenten (AR + GPS Location) hinzugefügt wurden. Die Abbildung 28 zeigt die hierfür im Hierarchie-Fenster eingebundenen Komponenten, wobei diese in kürzester Form unterhalb skizziert werden sollen:

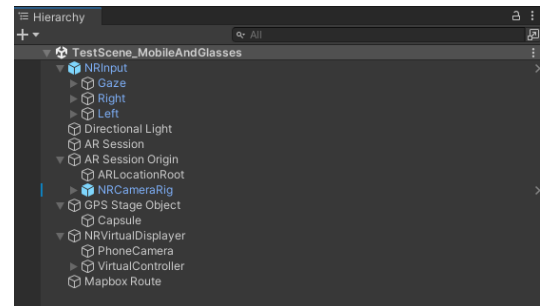


Abbildung 28: Elemente des Aufbaus der initialen Startszene

- NRInput: ein Prefab der NRSDK, welche die Eingaben (Anzeige eines virtuellen Strahls, Abfrage des Auftreffens des Strahls an Objekten, ...) im virtuellen Raum abwickelt
- Directional Light: Standardkomponente einer AR-Szene, welche die darin enthaltenen Komponenten beleuchtet
- AR Session: Standardkomponente einer AR-Szene, die für die Funktionsfähigkeit vonnöten ist
- AR Session Origin: Standardkomponente einer AR-Szene, die für die Funktionsfähigkeit und das Management des virtuellen 3D-Raums und dessen Verknüpfung zum realen Raum nötig ist
 - AR Location Root: ein Prefab der ARGPS SDK, welches sich um die Verortung der Szene und Navigationsinhalte kümmert
 - NRCameraRig: ein Prefab der NRSDK, welches das Kamerasystem der AR-Brille darstellt
- GPS Stage Object: ein Prefab der ARGPS SDK, welche es ermöglicht, ein beliebiges Objekt an einer bestimmten Stelle zu positionieren
- NRVirtualDisplayer: ein Prefab der NRSDK, welches die auf dem Smartphone anzuzeigenden Inhalte (Controller) verwaltet
 - PhoneKamera: Kamera (aus NRSDK) zur Aufnahme der Inhalte im VirtualController Objekt
 - VirtualController: Objekt (aus NRSDK), welches alle auf dem Smartphone anzuzeigenden Objekte und Komponenten enthält
- Mapbox Route: ein Prefab der ARGPS SDK, welches es ermöglicht, ein Routing basierend auf einer textlichen Ortseingabe einer vorab definierten Route von der aktuellen Position aus zu machen

²²⁹ Prefabs bezeichnen vorkonfigurierte Bausteine für die Entwicklung. Ein Prefab kann etwa ein komplettes Hausmodell, wobei jede Tür etc. wiederum ein eigenes Objekt darstellt, sein, oder aber auch Benutzeroberfläche mit Eingabefeldern, Texten, Skripten oder Ähnlichem.

Zur Verbesserung der Verständlichkeit wie in Unity gearbeitet wird sei überdies auf die nachfolgende Abbildung 29, welche das fertige Projekt im Unity-Editor zeigt verwiesen. Darüber hinaus sei zusätzlich erwähnt, dass es nur schwer möglich ist einzelne Komponenten²³⁰, oder Objekte²³¹ für sich allein darzulegen, da diese in sehr engem Maße miteinander verknüpft sind. Sollte jedoch weiterführendes Interesse hierzu bestehen, sei an dieser Stelle auf die versuchte Darstellung und Erläuterung aller das finale Projekt aufbauenden Objekte der Szene und die exemplarische Darstellung eines Teils des Codes (jener für die Anzeige der Karte auf der Nutzeroberfläche) im Appendix verwiesen.

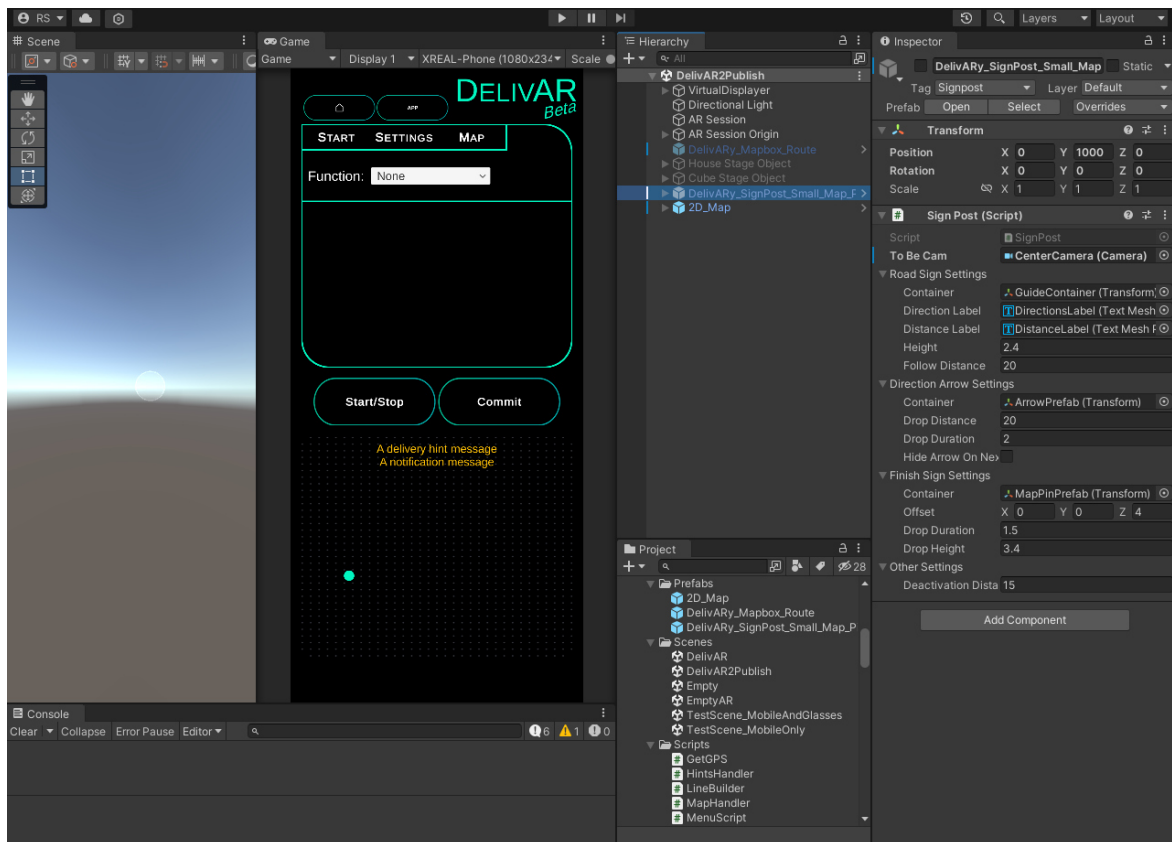


Abbildung 29: Aussehen der Unity-Editor Oberfläche und seiner Teile

Linkes oberes Fenster: 3D-Raum und dessen editierbare Inhalte

- im dargestellten Projekt leer, da erst während der Laufzeit generiert

Zweites Fenster von Links: Anzeige des Inhalts einer ausgewählten Kamera

- im Rahmen des dargestellten Projekts die Nutzeroberfläche, da der 3D-Raum erst während der Laufzeit befüllt und somit erst dann Inhalte angezeigt werden

Drittes oberes Fenster von Links: Hierarchie, welche alle Objekte und Unterobjekte und Komponenten der aktuellen Szene beinhaltet

Fenster unterhalb dem Obigen: Daten-Explorer mit allen zu einem Projekt gehörigen Daten

Rechtes Fenster: Anzeige aller Komponenten und deren Anpassungsmöglichkeiten eines Objekts

Unteres linkes waagrechtes Fenster: Konsole zur Anzeige von Warnungen, ...

²³⁰ Komponenten bezeichnen in Unity alle veränderlichen Teile, wie etwa Skripte, Beleuchtungen, Bounding-Boxen, ...

²³¹ Objekte in Unity bauen eine Szene auf und beherbergen ihrerseits verschiedene Komponenten.

5.3.4 Applikationserstellung Phase I

Nachdem in den vorangegangenen Schritten die grundlegende Planung wie auch der Aufbau der Arbeitsumgebung erledigt wurde, folgte als nächstes die eigentliche Erstellung der Applikation und die Aufteilung der auszuführenden Schritte zur Erreichung einer wie oberhalb dargelegt geplanten Applikation in einzelne Teil-/Arbeitspakete. Der Grund für diese Unterteilung lag dabei neben der besseren Planbarkeit²³² auch darin begründet, dass es so möglich wurde, die einzelnen Funktionen möglichst gekapselt voneinander zu entwickeln, ohne dass sie unnötig aufeinander Einfluss nehmen und sie somit möglichst unabhängig voneinander getestet werden konnten. Außerdem war es dadurch bei einem an dieser Stelle als durchwegs groß zu bezeichnendem Projekt zusätzlich besser möglich, den Überblick über erledigte Funktionen oder Teile, in denen noch Probleme zu beheben waren, zu bewahren.

Die getrennte Entwicklung voneinander/ Separierung der einzelnen Bestandteile darf an dieser Stelle jedoch nicht so verstanden werden, als dass die einzelnen Funktionen nicht teils aufeinander zugreifen und verweisen. So ist es etwa bei der 2D-Karte, als auch der Hinweisanzeige so, dass diese logischerweise auf die grundlegende Navigationsfunktionen an sich zugreifen müssen um korrekte Informationen für deren Funktionalität zu bekommen²³³. Insgesamt wurde die Erstellung schlussendlich in zwei große Blöcke mit sieben Teilpaketen aufgeteilt, welche zum Teil hier in diesem Kapitel und zum Teil in Kapitel 5.3.6 Betrachtung finden werden. Schematisch sah diese Aufteilung dabei wie folgt aus:

- Must-Have Funktionen (Kap. 5.3.4)
 - Aufbau der grundlegenden Funktionalität (Kap. 5.3.4.1 & Kap. 5.3.4.4)
 - Eingabeoberfläche (Kap. 5.3.4.2)
 - Testdaten (Kap. 5.3.4.3)
 - Zustellhinweise (Kap. 5.3.6.1)
- Nice-To-Have Funktionen (Kap. 5.3.6)
 - 2D-Karte (Kap. 5.3.6.2)
 - (Sprachausgabe) (Kap. 5.3.6.3)
 - Styling der Anzeigen (Kap. 5.3.6.4 & Kap. 5.3.6.5)

An dieser Stelle soll, bevor unterhalb auf den Prozess der Erstellung eingegangen wird, noch darauf hingewiesen werden, dass es sich bei der klaren Aufteilung der Arbeitspakete und dem geradlinigen Abarbeiten der einzelnen Pakete um einen idealtypisch dargestellten Prozess handelt, wie er abschließend durchgeführt wurde, nachdem bereits die Applikation

²³² Wann muss etwas erledigt werden?, Welche Voraussetzungen müssen hierfür erfüllt sein?, ...

²³³ Für die Funktionalität der 2D-Karte müssen etwa Informationen zur aktuellen Route und dem aktuellen Standort aus der grundlegenden Navigationsfunktion abgerufen werden um diese Informationen wiederum auf der 2D-Karte anzeigen zu können. Für die Hinweisanzeige wiederum ist es nötig Informationen zum aktuellen Standort und Informationen zu den koordinativ verorteten Adressen von zuzustellenden Lieferungen und die daraus errechnete Distanz abzuholen um darauf basieren die Zustellhinweiseinblendungen zum korrekten Zeitpunkt anzeigen zu können.

mit einigen Fehlversuchen erstellt worden war, um eine „saubere“ Applikation zu haben. Grundsätzlich war es nämlich bis auf die finale App-Erstellung so, dass oftmals an mehreren Bereichen gleichzeitig gearbeitet wurde, da es schlicht und ergreifend nicht möglich war, alle Funktionen und Voraussetzungen/ Notwendigkeiten hierfür von vornherein abzusehen. Unter anderem aus diesem Grund war es auch nötig teils während der Laufzeit Funktionen zu Testen und Werte anzupassen, weshalb während der Entwicklung oftmals Parameter zur Veränderung von Werten auf der Benutzeroberfläche erstellt wurden, um eben live die Veränderungen zu sehen und auszuprobieren. Insgesamt wurde das Projekt dreimal von Grund auf, aufgrund verschiedener Probleme, neu aufgebaut, wobei auf diese Schwierigkeiten, wie auch die daraus abzuleitenden Learnings in Kapitel 5.5 eingegangen werden wird.

5.3.4.1 Aufbau der Skripte/ des Codes für die Grundfunktionen²³⁴

Anschließend an den Aufbau der grundlegenden Arbeitsumgebung und das Hinzufügen der grundlegenden Bausteine in die Szene folgte dann die Aufgabe der Herstellung der Funktionalität bzw. das Erstellen des hierfür benötigten Codes der Applikation²³⁵, also in anderen Worten, die Anpassung und Erstellung von Elementen, sodass die Navigationsinhalte auf der AR-Brille und die Benutzeroberfläche am Smartphone angezeigt werden. Hierzu sei direkt auch gesagt, dass dieser Schritt gemeinsam mit den Überlegungen zur grundlegenden Darstellung der Inhalte im virtuellen Raum (exklusive der Punkte die in Richtung Styling, wie etwa der Veränderung von Farben/ Größenverhältnisse usw. gehen), bei weitem die aufwändigsten Schritte der gesamten Umsetzung waren, die viel Zeit und Nerven geraubt haben. Begründet ist dies darin, dass sich hierfür durch einen großen Teil der durch die AR + GPS Location-Erweiterung zur Verfügung gestellten Codes und deren Funktionen durchgearbeitet werden musste, um zu wissen ...

- a) ... welches Skript und welcher Codeabschnitt wie funktioniert und demnach genutzt werden kann.
- b) ... wie eine Funktion, etwa die Erstellung einer Route von A nach B, grundsätzlich funktioniert und welche Skripte beteiligt sind.
- c) ... worauf (welche anderen Skripte/ Elemente) diese Skripte ihrerseits wiederum zugreifen und welche Auswirkungen ein spezifischer Codeabschnitt und die Änderung eines Parameters im Endeffekt haben.

²³⁴ Zu Kapitel 5.3.4.1 und 5.3.4.4 sei gesagt, dass diese sehr eng miteinander verknüpft sind und deren Entwicklung eigentlich kaum voneinander trennbar war und somit auch als ein Paket gesehen werden kann/sollte.

²³⁵ Das Wissen hierzu basierte (da es sich um einen iterativen Prozess handelte) wie bereits erwähnt teils auf zuvor bei Fehlversuche zur Erstellung einer solchen Applikation gemachten Learnings.

Insgesamt kann man den Aufbau/ Umbau des Codes demnach in die folgenden Schritte untergliedern:

1. Verstehen der Funktionsweisen der bestehenden Skripte
 - a. ...wie auch die Identifikation der Änderung bedürftenden Bestandteile
2. Erstellung eigener Skripte zum Zweck der für die Applikation nützlichen (Weiter-) Verwendung/ Anwendung/ Aktivierung bestehender Funktionen/ Skripte²³⁶
3. Erstellung eigener Skripte für nichtexistierende Funktionen
4. Abänderung bestehender fremder Codes, insofern nicht anders möglich

Hinsichtlich der ersten drei Punkte kann an dieser Stelle gesagt werden, dass dies ein hoch iterativer Prozess war, da oftmals nur über Trial und Error in dem Prozess der Erstellung herausgefunden wurde, dass eine Funktion nicht wie gewünscht funktioniert und entweder eine andere Herangehensweise gesucht oder andere Funktionen angesprochen werden mussten, ein Umbau der Szene für die korrekte Funktionalität nötig wurde oder im schlechtesten Fall Änderungen in fremden Code vorzunehmen waren²³⁷.

Letzterer Punkt der Änderung von bestehendem fremdem Code war jedoch im Hinblick auf die AR + GPS Location-Erweiterung unumgänglich, da ...

- a) ... alle anzuzeigenden Inhalte normalerweise immer auf der Renderkamera²³⁸ des Smartphones ausgespielt wurden, aber auf der Renderkamera der AR-Brille ausgespielt werden sollten.
- b) ... auf die physische Kamera des Smartphones zugegriffen wurde, für die geplante Umsetzung es aber nötig war, die physische Kamera komplett außen vor zu lassen.
- c) ... es nötig war, die Positions- und Rotationsinformationen nicht rein vom Smartphone (Position & Rotation) abzurufen, sondern stattdessen eine Mischform aus Smartphone (GNSS) für die Position und HMD (IMU) für die Rotation zu nutzen.
- d) ... es nötig war die mittels der eigens erstellten Einlesepipeline ins Projekt eingefügten Routen nutzbar zu machen.

²³⁶ Dies kann bedeuten, dass Skripte einfach nur aufgerufen wurden, um eine gewisse Funktion auszuführen, Werte/ Inhalte in anderen Funktionen basierend auf übergebenen Informationen berechnet und zurückgeliefert (und weiterverarbeitet) werden, oder Ähnliches.

²³⁷ Änderungen in fremden Code sind als suboptimal anzusehen, da es nur schwer möglich ist, die Auswirkungen komplett zu überblicken und darüber hinaus die Updatebarkeit eingeschränkt wird – da jedes Update die Änderungen überschreiben würde.

²³⁸ Eine Renderkamera nimmt die Inhalte im virtuellen Raum auf und definiert auf welchem Display diese angezeigt werden sollen.

Beginnen wir anschließend an diese kurze, oberhalb schematisch dargestellte Erläuterung zum Vorgehen bei der Erstellung der Skripte hier nun logisch der Ausführungsreihe nach beim Start der Applikation mit der Erklärung der (Code-)Abläufe der Applikation. Startet man die Applikation (am Smartphone), so werden an dieser Stelle direkt alle in einer Szene vorhandenen Objekte und diese beinhaltenden Komponenten (bspw. Materialien, Skripte) initialisiert und die Skripte der Prozessierungsreihenfolge²³⁹ nach ausgeführt. Im Falle der umgesetzten Applikation bedeutet dies, dass die gesamte Benutzeroberfläche und grundsätzlichen Funktionen zur Verwendung von AR-Brille und Smartphone²⁴⁰ und Navigation²⁴¹, wie auch das eigen Hauptskript (genannt **MenuScript**) gestartet wird. Das MenuScript liest beim Start alle notwendigen Dateien (vordefinierte Routen, Hinweisdateien) aus dem hierfür angelegten Ordner ein und aktiviert beziehungsweise deaktiviert die einzelnen auf der Eingabeoberfläche vorzufindenden Inhalte und initialisiert wichtige Objekte. Anschließend daran kümmert sich dieses Skript bei Interaktion auf der Eingabeoberfläche, etwa bei Klick auf einen Tab – siehe hierzu Kap. 5.2.2 oder Kap. 5.3.4.2 – darum, dass die entsprechenden Objekte aus- beziehungsweise eingeblendet werden. Ein weiterer wichtiger Block in diesem umfangreichen Skript kümmert sich darum, was passieren soll, sobald der Start-Knopf auf der Eingabeoberfläche gedrückt wird. Hierzu zählen etwa das Aktivieren der richtigen Komponenten und das Starten der korrekten (verlinkten) Funktionen (in anderen Skripten – oftmals aus der AR + GPS Location Erweiterung) wie auch das Auslesen der hierfür gesetzten Werte in den Dropdown Menüs, Eingabefeldern und so weiter auf der Nutzeroberfläche. Wenn die Funktionen zur Ausführung der Navigationsanzeige über diesen Code gestartet werden können, ist es an dieser Stelle dann auch nur als logisch anzusehen, diese auch wieder über dieses Skript beenden zu können, was durch einen abermaligen Klick des Start-Buttons und den Aufruf eines hierfür geschriebenen Codeteils passiert. Eine weitere wichtige Funktion, die diesem Skript überdies noch zukommt, ist jene der Bereitstellung einer Funktion, welche sich um die Einblendung von Hinweisen auf der Benutzeroberfläche und deren automatische Deaktivierung kümmert und von allen anderen Skripten aufgerufen werden und somit genutzt werden kann.

Neben dem Hauptskript wurden dann noch einige weitere wichtige Skripte, die teils mittels Verlinkungen vom Hauptcode, per Buttons oder dauerhaft während der Laufzeit der Applikation genutzt werden können, erstellt. Insgesamt handelt es sich hierbei um sechs zur Gänze selbstgeschriebene und ein aus der AR + GPS Location-Erweiterung in modifizierter Weise weitergenutztes Skript, wobei natürlich auch eine Vielzahl weiterer Skripte (etwa aus der ARGPS Erweiterung in Hinblick auf Navigationssachverhalte) angesprochen und genutzt wird: Von einer detaillierten Darlegung der fremden Skripte wird an dieser Stelle aus multiplen Gründen – wie etwa die schiere Menge an Skripten und den hierfür benötigten

²³⁹ Siehe hierzu „*execution-order*“: (Unity, 2024)

²⁴⁰ Hierzu sei erwähnt, dass sich die Funktionen hierfür aus den aus der NRSDK kommenden und in der Szene befindlichen Komponenten speisen.

²⁴¹ Hierzu sei erwähnt, dass sich die Funktionen hierfür aus den aus der AR + GPS Location Erweiterung kommenden und in der Szene befindlichen Komponenten speisen.

Umfang²⁴², das teilweise fehlende Wissen über Details zur Funktionsweise, ... – Abstand genommen. Die eigenen Skripte hingegen sollten an dieser Stelle jedoch zumindest in Kürze²⁴³ Erwähnung finden:

- MenuScript: Hauptskript mit den oberhalb bereits beschriebenen Funktionen
- GetGPS: Dieses Skript kümmert sich während der gesamten Laufzeit der Applikation darum, die GPS-Position des Smartphones abzufragen und auszuliefern.
- HintsHandler: Dieses Skript kümmert sich um alles – abseits der korrekten Ausrichtung der Anzeige der Hinweise auf der AR-Brille – rundum die Zustellhinweise. Die Funktionen reichen hierbei vom Auslesen der Informationen für Zustellhinweise aus einer CSV-Datei bis hin zur Erstellung korrekt im Unity Raum positionierter Objekte mit Zustellhinweisen basierend auf Koordinaten der CSV-Datei. Mithilfe dieser können dann weiters Abfragen zur Entfernung der Nutzer*innen zum Objekt gemacht und weiters bei Unterschreitung eines Schwellenwerts die Anzeige eines Hinweises aktiviert werden. Ebenso kümmert sich das Skript auch um das Bestätigen und somit entfernen dieser Hinweise, wenn die hierfür geschriebene Funktion aufgerufen wird. Siehe hierzu auch Abschnitt 5.3.6.1
- MapHandler: Dieses Skript kümmert sich um alles mit Bezug auf die 2D-Karte. Siehe hierzu auch Abschnitt 5.3.6.2
- PassPosition: Dieses Skript kümmert sich zum einen um die korrekte Berechnung der Positions- und Rotationswerte für die Anzeige der Inhalte und das Ausspielen dieser im Debug Tab: d.h.: 1) Beziehen, Modifizieren und Weitergeben der Unity-Raumkoordinaten vom Smartphone an relevante Objekte. Diese werden ihrerseits selbst wiederum durch das AR-Session Modul basierend auf den Startkoordinaten (0,0,0) und der zurückgelegten Distanz im echten Raum (berechnet über GNSS-Startposition zu aktueller GNSS-Position) berechnet. 2) Beziehen, Modifizieren und Weitergeben der Unity-Rotationswerte von der AR-Brille an relevante Objekte. Diese mussten dabei selbst berechnet werden, da die Smartphone Rotation keinen Einfluss auf die Anzeige haben durfte.
- PlaceCameraOverlayText: Dieses kleine Skript kümmert sich um die korrekte Positionierung (Fixierung) der Hinweiseinblendungen der Zustellhinweise auf der AR-Brille.
- SearchCustomDestination: Dieses Skript basiert auf einem Skript aus der AR + GPS Location-Erweiterung und kümmert sich um die Suche eines bestimmten Ortes, wenn ihm ein Suchtext übermittelt wird, die Anzeige der Suchvorschläge, die

²⁴² Schlussendlich würde man an dieser Stelle, wenn man alle Skripte, auf die ein Skript Bezug nimmt oder aufbaut, sonst irgendwann bei den grundlegenden Programmcodes einer Rechenmaschine enden. Außerdem würde die Darlegung der Codes einige hundert Seiten füllen.

²⁴³ Die Skripte werden an dieser Stelle nicht eingefügt, da diese jeweils rund 500 Zeilen umfassen und somit nicht wirklich sinnvoll abbildbar sind. An dieser Stelle sei, wenn weiteres Interesse besteht auf die Links zu den Projektdaten, auf die exemplarische Besprechung eines Skripts im Anhang und zu guter Letzt noch auf die Möglichkeit der Kontaktaufnahme mit dem Autor dieser Arbeit verwiesen.

Auflösung des Zielorts in Koordinatenform und die Verlinkung zum Objekt für die Routenerstellung und dessen Skripten.

- **StyleSettings:** Dieses Skript kümmert sich um alle stylingtechnischen Belange, welche während der Laufzeit der Navigation geändert werden können, wie etwa die Farbe, Größe oder Art der Navigationshinweise. Siehe hierzu auch Abschnitt 5.3.6.4.

5.3.4.2 Konfiguration/ Erstellung der Eingabeoberfläche

Design:

Hinsichtlich des Aussehens und den Funktionen der Eingabeoberfläche sei an dieser Stelle erwähnt, dass diese, wie in Kapitel 5.2.2 bereits angesprochen, mit verschiedenen Tabs – Start, Settings, Map, Debug – für verschiedene Zwecke, basierend auf der durch die NRSDK SDK bereitgestellten Benutzeroberfläche²⁴⁴, erweitert wurde, welche ihrerseits wieder Schieberegler, Textfelder, Dropdown-Menüs, ein Kartenfeld und weitere Elemente enthielten. Ebenso wurden zwei Buttons erstellt, wovon einer das im vorigen Kapitel besprochene

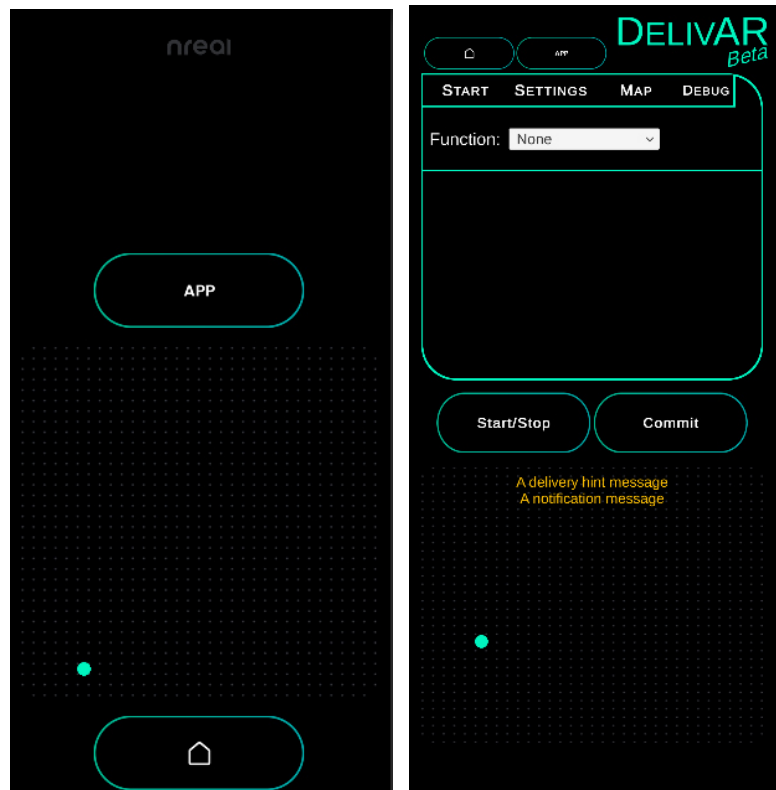


Abbildung 30: Original-Controller Gestaltung im Vergleich zur eigenen Umsetzung

eigentlichen Navigations-/Platzierungsfunktion startet und der andere zur Bestätigung der eingeblendeten Zustellhinweise dient, wie auch ein Textfeld, welches den Nutzer*innen über getätigte Eingaben, aber auch Fehler informiert angelegt. Um weiters ein gewisses Branding nicht außer Acht zu lassen, wurde der Name der App gut erkennbar hinzugefügt und der Home-Button hinsichtlich Position und Größe, nicht aber Funktion²⁴⁵ verändert. Gleiches gilt für das Touchpad, welches verkleinert wurde, aber weiterhin für Eingaben im virtuellen Raum (bspw. der Bestätigung des Dialogs zum Beenden der Applikation) dient.

²⁴⁴ Die Benutzeroberfläche wird in weiterer Folge auch als Controller/ Nutzeroberfläche/ Eingabeoberfläche bezeichnet werden.

²⁴⁵ Klick: Schließen der Applikation über auf der AR-Brille angezeigten Dialog; Halten: Rezentrierung des Bedienungsstrahls, welcher vom Smartphone ausgeht

Code:

Hinsichtlich der Tatsache, dass alle auf der Benutzeroberfläche eingebauten Inhalte auch ihre gewünschten Funktionen erfüllen können, war es natürlich nötig, einiges an Code aufzubauen, wobei an dieser Stelle nur die wichtigsten Punkte kurz umrissen werden sollen. Der wichtigste Punkt war an dieser Stelle sicherlich die Verknüpfung der am Controller enthaltenen Komponenten (Textfelder, Dropdown-Menüs, Schieberegler) und den Skripten, die diesen Informationen zuweisen (beispielsweise welche Funktion gerade gestartet wurde oder ob ein Fehler auftritt) beziehungsweise Informationen aus diesen beziehen sollten (etwa in Bezug darauf, welche Funktion ausgewählt wurde und in weiterer Folge gestartet werden soll), da ansonsten eine Auswahl verschiedener Funktionen, verschiedener Gestaltungsoptionen und so weiter nicht möglich wäre. Insgesamt kann an dieser Stelle also hauptsächlich – neben dem Hauptcodestrang rundum den Start-Button, welcher im vorangegangenen Abschnitt besprochen wurde – von Funktionen, die sich um das Abholen und Ausliefern von Informationen kümmern, gesprochen werden. Der Großteil dieser Funktionen findet sich dabei im „MenuScript“.

5.3.4.3 Erstellung von Testdaten²⁴⁶

Das Aufgabenpaket hinsichtlich der Erstellung von Testdaten gestaltete sich als ein tatsächlich beinahe zur Gänze vom Rest der Entwicklungsschritte losgelöstes, da an dieser Stelle hinsichtlich der Strecken zur Navigation nur der Sachverhalt, dass es sich um eine KML-Datei²⁴⁷ – welche „out of the box“ von der AR + GPS Location-Erweiterung gelesen werden kann – handeln musste. Hierbei musste nur beachtet werden, dass an jeder Stelle, wo eine Richtungsänderung vorgenommen werden soll (Knotenpunkt), ein Punkt gesetzt werden muss und, wie bei einem

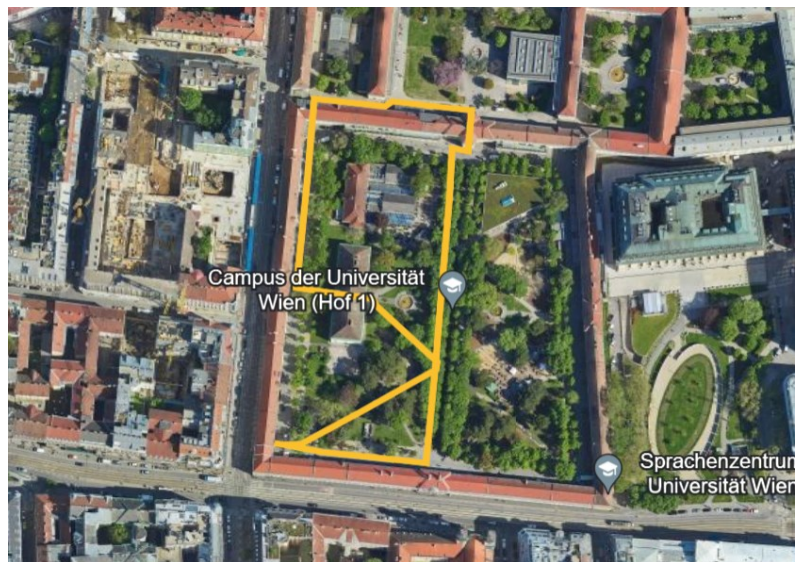


Abbildung 31: Testroute am Campus der Uni Wien, visualisiert mithilfe von Google Earth

²⁴⁶ Im Rahmen der Arbeit wurden nur Testdaten verwendet, da alles andere den ohnehin schon erheblichen Aufwand der Erstellung und Testung der App noch weiter in die Höhe getrieben hätte. Es wurde hierbei jedoch darauf geachtet, die Daten möglichst realitätsnah aufzubauen.

²⁴⁷ Hierbei handelt es sich um ein auf XML aufbauendes Format, welches in allen gängigen GIS oder auch Google Maps/Google Earth verwendet werden kann. Simplifiziert gesagt, werden in diesem Dateiformat rein Informationen zu Wegpunkten (mittels WGS84-Koordinaten) aneinandergereiht gespeichert.

GIS auch, wenn man eine Kurve zeichnen will, diese mit einer Vielzahl an Punkten realisiert werden muss, da die einzelnen Punkte immer nur mit geraden Linien verbunden werden. Ein weiterer Bearbeitungsschritt wurde dann beim Import der KML-Dateien mithilfe der durch Daniel Fortes bereitgestellten SDK nötig, da an dieser Stelle dann Anweisungen zur Navigation, wie etwa „hier rechts abbiegen“, noch an den hierfür relevanten Knotenpunkten eingefügt werden mussten, um diese auf einem Hinweisschild im virtuellen Raum angezeigt zu bekommen.

Die Erstellung der Daten zu den Navigationshinweisen war noch einfacher, da an dieser Stelle das Format selbst, da auch der Code zum Umgang mit selbigen selbst erstellt werden musste, gewählt werden konnte. Hierbei wurde, da es sich ohnehin um (Punkt-)Informationen handelte, welche an einer bestimmten Stelle (Anm. An der Adresse wo etwa ein Paket zugestellt werden soll) angezeigt werden sollten, überlegt, wie diese möglichst einfach automatisierbar (etwa bei Anwendung der Applikation in Zustellbetrieben) aufgebaut und in in C# gelesen und verarbeitet werden können. Hierbei stellte sich, wie bei den meisten Programmiersprachen CSV, als ein gut gangbarer Weg dar, woraufhin eine Datenstruktur, die alle benötigten Informationen aufnehmen konnte aufgebaut wurde²⁴⁸.

	A	B	C	D	E	F	G	H	I	J	K
1	id	lat	lon	adress 1	adress 2	adress 3	receptient	amount	kind	info 1	info 2
2	1	48.213164	16.338339	1160 Vienna	Schellhamnergasse 5		Baklava Süsse Welt		1 EMS	signature	delivery until 12.00
3	2	48.213867	16.338428	1160 Vienna	Veronikagasse 17		Schlicht Vegan		2 parcel	signature	
4	3	48.21397469	16.33737807	1160 Vienna	Payergasse 7	Tür 5	Maier Marie		1 letter	no signature	letterbox applicable
5	4	48.21397469	16.33737807	1160 Vienna	Payergasse 7	Tür 11	Skrepic Matek		1 RSB	signature	personal or family
6	5	48.213166	16.337306	1160 Vienna	Schellhamnergasse 11		Monir Al Maktari		1 parcel	no signature	letterbox applicable
7	6	48.213166	16.337306	1160 Vienna	Schellhamnergasse 11	Tür 7	Mark Piet		1 Hybrid	signature	
8	7	48.213174	16.337674	1160 Vienna	Schellhamnergasse 9		Güzel Ayse		1 RSA	signature	personal

Abbildung 32: Beispielfhafte Liste mit Zustellhinweisen, visualisiert in Excel

5.3.4.4 Anzeige von Inhalten im virtuellen Raum

Design:

Hinsichtlich des, vor allem für technisch weniger versierte/interessierte Leser*innen wahrscheinlich interessantesten Teils des Kapitels 5.3, nämlich der Überlegungen hinsichtlich der Darstellung der Navigationshinweise im virtuellen Raum muss an zuerst betrachtet

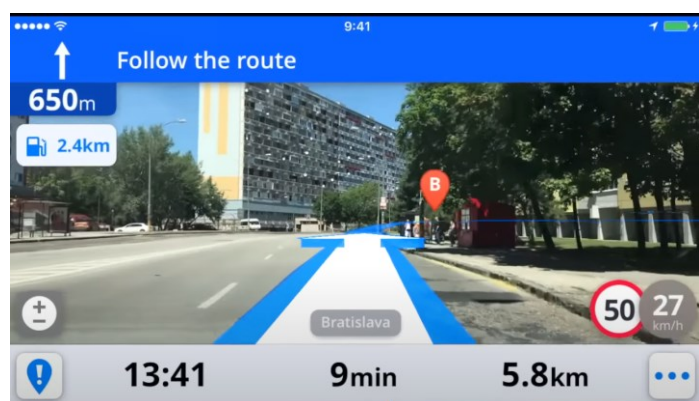


Abbildung 33: Sygic Real View Navigation (historisch) (Sygic a.s., 2017)

²⁴⁸ Insgesamt wurden 11 Spalten definiert: Je 1 Spalte für Identifier, Latitude und Longitude; 3 Spalten für die Adresse in textlicher Form; 1 Spalte für Informationen zur Person und 4 Spalten für Informationen zur zuzustellenden Lieferung

werden, wie zum einen AR-Navigationsapplikation, wie etwa Googles Live View (Abb. 18-20) oder Sygic Real View (Abb. 33, 34) Inhalte präsentieren und zum anderen die zu verwendenden ARGPS Erweiterung (siehe Abb. 35) hierbei vorgeht. Hierbei stellte sich heraus, dass es hier ein beinahe universell ähnliches

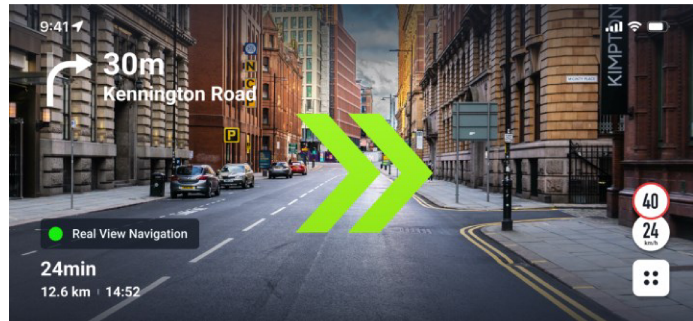


Abbildung 34: Sygic Real View Navigation (Sygic, 2024)

Repertoire an eingesetzten Symbolen gibt. Hierzu zählen Pfeile, welche an Wegkreuzungen positioniert werden und in die entsprechende Richtung, in welche abgezweigt werden muss, zeigen, wie auch Pfeile, Punkte oder Linien, welche sich am Boden befinden, wenn einem Weg einfach nur gefolgt werden soll, und Pinnadeln, die den Zielort markieren. Hinzu kommen dann je nach Applikation und Umsetzung noch textliche Zusatzinformationen, wie beispielsweise der Weg-/Straßenname oder die Distanz, welche direkt über den Richtungspfeilen oder auf den Hinweistafeln an einer Wegkreuzung angezeigt werden. Relative große Einigkeit herrscht auch bei der Farbgebung der einzublendenden Objekte, wenn an dieser Stelle einmal die neueste Version von Sygic Real View Navigation außen vorgelassen wird²⁴⁹, da zumindest alle durchwegs bekannten und hier aufgezeigten Applikationen Symbole mit blauer Umrandung und weißem Inhalt nutz(t)en. Basierend auf diesem Wissen und der Vereinfachung der Erstellung der Applikation – zumindest in einem ersten Schritt, da die Farben und Symbole im Nachhinein ohnehin noch angepasst werden können –, wurde beschlossen, sich in diese stete Einigkeit hinsichtlich der Visualisierung einzureihen. Gründe für diese Entscheidung waren hierbei die folgenden: Zum einen sind Nutzer*innen durch die Einheitlichkeit, die mittels der Symbole übermittelten Informationen geläufig und führen zu wenig bis keine Verwirrungen. Weiters ist die Farbgebung, der eigenen Erfahrung nach hinsichtlich des Grads an Ablenkung, als auch der Sichtbarkeit der Inhalte vor verschiedenfarbigen Hintergründen und bei dunklen als auch hellen Umgebungen als durchwegs gelungen zu bezeichnen.

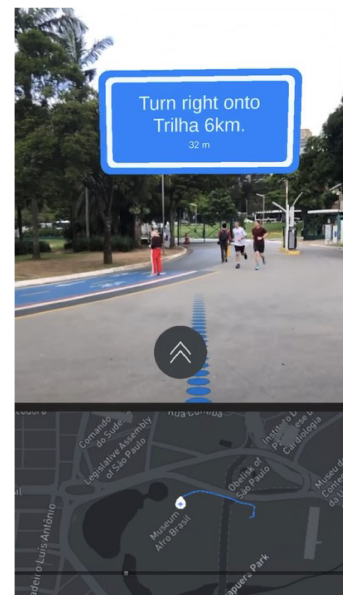


Abbildung 35: ARGPS Beispielsprojekt (Fortes, 2022)

Code:

Hinsichtlich der technischen Realisierung der Anzeige der für die Navigation benötigten Inhalte sei an dieser Stelle kurz, um nicht zu sehr in den technischen Bereich abzutauchen, auf Kapitel 3.5.2 verwiesen, wo die grundlegende Funktionsweise einer AR-Navigationsapplikation kurz erläutert wird. An dieser Stelle sei nur gesagt, dass es sich bei

²⁴⁹ Bei einem der letzten Updates wurde nämlich das gesamte Farbschema von blau auf grün umgestellt.

den einzublendenden Objekten an Kreuzungspunkten (d.h. Pfeile und Hinweisschilder) und Endpunkten (d.h. Pinnadeln), um vorab in einem 3D-Grafikprogramm oder auch direkt in der Realtime-Engine erstellte Objekte handelt, welche ihrerseits dann basierend auf GPS-Koordinaten an den diesen entsprechenden Unity-Koordinaten im virtuellen Raum platziert und angezeigt werden. Hinsichtlich der Verbindungslinien²⁵⁰ zwischen den einzelnen Vertices²⁵¹ ist es dabei so, dass es sich hier im einfachsten Fall einfach nur um gerade Linien zwischen den eine Route definierenden Punkten²⁵² handelt. Im komplexeren, aber im Rahmen dieser Arbeit nur am Rande relevanten Fall – da nicht umgesetzt/ nicht umsetzbar – kann es auch sein, dass Objekterkennung zur Verbesserung der Lage der angezeigten Inhalte verwendet wird. Ein Beispiel hierfür wäre etwa die Erkennung eines Fahrbahnstreifens und die Fixierung der anzuzeigenden Inhalte in der Mitte des erkannten Fahrbahnstreifens, wobei an dieser Stelle dann auch trotzdem zumindest an Kreuzungspunkten Zwischenschritte der Route erstellt werden müssten, da es an diesen Punkten ansonsten trotz alledem zu Navigationsproblemen kommen würde. Aus codetechnischer Sicht läuft im Rahmen der erstellten Applikation die gesamte Positionierung und De-/Aktivierung bzw. (Nicht-)Anzeige der Inhalte über die ARGPS-Erweiterung²⁵³, wobei dieser hierfür nur die relevanten Routeninformationen, also die Route definierenden Zwischenschritte (Vertices) oder direkt per Mapbox-API abgerufene Navigationspunkte, übergeben werden müssen und diese sich selbst die Unitykoordinatenentsprechungen der WGS84 Koordinaten berechnet.

5.3.5 Alpha-Testing

Nachdem die grundlegenden Komponenten²⁵⁴ der Applikation, genauso wie die hierfür nötigen Objekte und Komponenten (Skripte), erstellt beziehungsweise angepasst waren, ging es beim Alpha-Testing darum, zu eruieren, ob alle Grundfunktionen²⁵⁵ korrekt ausgeführt werden oder ob an irgendeiner Stelle Fehler auftreten. Dabei gestaltete sich

²⁵⁰ Diese können unterschiedlich etwa mit Punkten/ Pfeilen in einem bestimmten Abstand oder aber einer durchgehenden Linie visualisiert werden

²⁵¹ Vertices bezeichnen in den GISc die Punkte, aus denen eine Linie/ ein Polygon aufgebaut wird.

²⁵² Eine Route ist wie eine Linie in einem GIS immer aus einer Vielzahl an Punkten aufgebaut, welche ihrerseits immer mit einer Geraden verbunden sind. Um eine Kurve darzustellen, ist es demnach nötig, eine Vielzahl an Punkten in geringem Abstand zu verwenden.

²⁵³ Da es sich um eine nicht selbst erstellte Funktion handelt, sei an dieser Stelle hinsichtlich der genauen Funktion auf die Dokumentation der Erweiterung verwiesen: In dieser Arbeit wird von einer genaueren Beschreibung abgesehen.

²⁵⁴ Als grundlegende Komponenten sind an dieser Stelle die grundlegenden Funktionen zu Navigation, also alles abseits der Möglichkeiten zur Anpassung des Stils der angezeigten Inhalte, den Zustellhinweisen, der 2D-Karte und der Sprachausgabe gemeint.

²⁵⁵ Teils waren in diesem Schritt auch schon gewisse Nebenfunktionen erstellt und eingebunden, wobei diese im Rahmen des Alpha-Testings weitestgehend ignoriert wurden.

während der Entwicklung – vor allem bei den ersten Versuchen zur Entwicklung der Applikation, nicht bei dem finalen Aufbau derselbigen – ein Test-Loop, wobei jenseits der 100 Testiterationen durchgeführt wurden. Das Problem hierbei war zumeist, dass zumeist nicht auf den ersten Blick feststellbar war, weshalb ein gewisser Fehler auftrat, und selbst wenn, es oftmals so war, dass ein gewisser Baustein – Stichwort Kamera in den ARGPS Skripten – nicht nur in einem Skript an einer Stelle sondern an mehreren Stellen in mehreren Skripten angepasst werden musste und durch die Verschachtelung von Skripten, die aufeinander Skripten verweisen, oftmals schwer nachvollziehbar war, wo überall etwas geändert werden musste und es somit zu Folgefehlern kam. Ebenso war es gerade zu Beginn des Testings der Fall, dass Positionen falsch berechnet wurden, was in weiterer Folge an der Verschachtelung der einzelnen Objekte in der Hierarchie und der Zuordnung der Skripte zu den einzelnen Objekten lag. Hierzu sei gesagt, dass jedes Objekt einmal seine eigenen Koordinaten im Verhältnis zu dem übergeordneten Objekt (lokale Koordinaten), aber auch die absoluten Koordinaten in der gesamten Szene (im Unity-Koordinatensystem) aufweist. Dadurch kann es, wenn die falsche Funktion zum Abruf derselbigen ausgeführt wird, dazu kommen, dass man falsche Werte geliefert bekommt, obwohl man auf das korrekte Objekt zugreift. Ein weiterer wichtiger Punkt ist jener der Ablaufstruktur in Unity, also wann welche Komponenten eines Objekts während des Renderprozesses²⁵⁶ aktualisiert werden. Dieser Punkt ist dabei deshalb von Relevanz, da es demnach nicht an jedem beliebigen Punkt in einem Skript immer möglich ist, beispielsweise Koordinaten zu überschreiben oder zu ändern, da man dann in eine Endlosschleife geraten kann, Elemente sich sprunghaft bewegen oder im besten Fall einfach keine Änderung vollzogen wird – aber eben nicht der gewünschte Effekt eintritt. Insgesamt wurde insbesondere in der Planungsphase oftmals auch wieder an das sprichwörtliche Reißbrett zurückgegangen, überlegt, wo und weshalb ein Fehler auftritt und wie dieser gelöst beziehungsweise umgangen werden könne, oder ob es einen gänzlich anderen Weg zur Lösung eines Problems gibt bzw. ob dieser benötigt wird.

5.3.6 Applikationserstellung Phase II

Nachdem im Alpha-Testing der Punkt, an dem alle Grundfunktionalitäten im Großen und Ganzen funktionierten, erreicht wurde, ging es über in die zweite Phase der Entwicklung der Applikation: zum Implementieren weiterer Funktionen, der sogenannten Nice-To-Have Funktionen. Diese umfassten die 2D-Kartenfunktion, die Funktion zur Anzeige von Zustellhinweisen (die Sprachausgabefunktion) und die Funktionen zum Styling der angezeigten Inhalte. Dieser Prozess der Erstellung dieser Bestandteile soll nun in weiterer Folge besprochen werden.

²⁵⁶ Die *execution order* gibt während eines Renderprozesses vor, wann welcher Teil eines Skripts beziehungsweise wann welche Elemente in einer Szene upgedated werden. Siehe hierzu auch *execution order* in Fußnote 144.

5.3.6.1 Anzeige von Zustellhinweisen

Design:

Die Anzeige der Zustellhinweise wurde an dieser Stelle relativ einfach über die fixierte Einblendung einer Hinweistafel mit textlicher Information am oberen linken Rand des Blickfeldes der AR-Brille realisiert²⁵⁷. Gestaltungstechnisch wurde diese Einblendung relativ simpel im Farbschema der anderen Navigationshinweise im virtuellen Raum gehalten, wobei wie bei diesen auch darauf geachtet wurde, dass die Einblendung vor jedem Hintergrund und bei jedem Beleuchtungsverhältnis möglichst gut erkennbar ist.

Code:

Hinsichtlich der codetechnischen Umsetzung handelt es sich hierbei, simplifiziert formuliert, nur um die Abfrage und Berechnung der Distanz der momentanen Position und der über die bereitgestellte CSV-Datei eingelesenen Position des Zustellhinweises²⁵⁸. Wird eine gewisse Distanz unterschritten, wird die Einblendung, deren Position über ein separates Skript geregelt wird, aktiviert: genauso wie ein Button auf der Eingabefläche, mit welchem der Hinweis bestätigt und wieder vom Sichtbereich entfernt werden kann – ebenso über dieses Skript – aktiviert wird. Die ersten besprochenen Funktionen kommen vom „HintsHandler“ und die im zweiten Teil besprochenen vom „PlaceCameraOverlayText“ Skript.

5.3.6.2 Erstellung einer 2D-Karte

Design & Code

Einer der etwas komplexeren Schritte, nicht aufgrund des hierfür zu schreibenden Codes, sondern hinsichtlich des Umgangs mit bestehendem fremdem Code und der Frage, welche Bestandteile für eine korrekte Funktionsweise behalten, verändert und angepasst werden müssen, war die Realisierung einer 2D-Karte auf der Eingabeoberfläche am Smartphone. Hierfür wurde die 2D-Kartenrealisierung basierend auf jener im Beispielsprojekt von Fortes²⁵⁹, welche ihrerseits bei weitergehender Analyse stark auf Tutorials von Mapbox basierte, zurückgegriffen. Um diese jedoch für die eigenen Zwecke in angepasster Form nutzbar zu machen, war es zuerst einmal nötig, herauszufinden aus welchen Bestandteilen (Objekte) sich diese überhaupt zusammensetzte, welche Komponenten (Skripte) involviert waren und wie die Karte korrekt angesprochen werden kann. Insgesamt blieben nach der Analyse der Bestandteile nur noch zwei Objekte mit in diesen befindlichen Komponenten

²⁵⁷ An dieser Stelle sei nochmals auf die Diskussion, ob es sich bei einer derartigen Umsetzung um eine AR-Umsetzung in Kapitel 3.2.2 verwiesen.

²⁵⁸ Die Position des Zustellhinweises entspricht in diesem Sinne im Optimalfall der Position des Wohnorts der die Sendung empfangenden Person.

²⁵⁹ Siehe hierzu: (Fortes, 2022a)

(Skripten) – die Kamera für das Rendering und das Kartenobjekt selbst – übrig beziehungsweise mussten in eigener Form umgebaut werden: etwa hinsichtlich der Kartengröße, der Zoom-Funktion, der Bereinigung unnötiger Codebestandteile, der Erstellung benötigter Codebestandteile zur Interaktion. Der Code bestand dann insgesamt aus mehreren Funktionen, welche sich um folgende Aufgaben kümmerten:

- ... das Zoom-In/ Zoom-Out
- ... die Aktualisierung der Karte (hinsichtlich Kartenausschnitt und Positionspins) bei Bewegungen
- ... das damit verbundene Abfragen des aktuellen Standorts
- ... die über die Eingabeoberfläche getriggerte Funktion des Einblendens der Route, deren Positionierung und Aktualisierung

Es handelt sich hierbei um das „MapHandler“ Skript, welches im Appendix auch in kommentierter Form angehängt ist.

5.3.6.3 Erstellung der Sprachausgabe für Navigationsanweisungen

Dieser Schritt wurde aufgrund des ohnehin bereits sehr umfangreichen Projekts nicht mehr weiterverfolgt, wobei an dieser Stelle einige Ansätze zu Lösung der Problematik der Sprachausgabe unter dem Schlagwort „*text to speech*“ gefunden werden konnten. Weiters wäre es vermutlich ohne größere Umstände möglich gewesen, die ohnehin in der Applikation vorliegenden Navigations- und Zustellhinweisdaten auszulesen und für die Sprachausgabe weiterzuverwenden.

5.3.6.4 Styling der anzuzeigenden Inhalte

Eine weitere umzusetzende Aufgabe war die Erstellung verschiedener auswählbarer Optionen hinsichtlich der Farbgebung, Größe und den anzuzeigenden Objekten als Hinweise zur Navigation: nur Richtungspfeile und Hinweistafeln an Wegkreuzungen, nur Wegmarkierungen, oder beides kombiniert. Die ersten beiden Änderungsoptionen waren hierbei relativ einfach umzusetzen, indem man einfach dem Material²⁶⁰, welchem den für die Navigation verwendeten Objekten (Hinweisschild, Pfeil) zugewiesen ist, einfach eine andere Farbe zuweist. Die Farben können dabei in einem Dropdown ausgewählt werden woraufhin ein Skript je nach ausgewählter Farbe, ebenjene dem Material der Objekte zuweist. Die Größe der anzuzeigenden Inhalte, welche vor allem im Hinblick auf die Frage, ob die Einblendungen als störend empfunden werden, relevant ist, wurde über einen Schieberegler gelöst, mit welchem sich der Skalierungsfaktor (1 = maximale Größe; 0 = minimale Größe) der Objekte einstellen ließ. Der dritte und letzte Punkt war überdies auch nicht sonderlich komplex, da zur Aus-/ Einblendung der jeweiligen Hinweisarten (bspw. der

²⁶⁰ Ein Material ist in einem 3D-Programm eine Zuweisung einer Textur/ Farbe zu einem Metaobjekt, welches dann auf das tatsächlich einzufärbende/ zu texturierende Objekt angewandt wird.

Linie am Boden) einfach die zugehörigen Objekte im virtuellen Raum (d.h. in der Szene) deaktiviert/ aktiviert werden mussten.

5.3.6.5 Styling der Benutzeroberfläche

Ein hinsichtlich der Umsetzung sehr einfacher und deswegen zum Schluss durchgeführter Teil der Arbeit umfasste das Styling der Benutzeroberfläche am Smartphone. War bis zu diesem Punkt die gesamte Benutzeroberfläche rein funktional und für den Entwickler (= Autor dieser Arbeit) verständlich angelegt worden, das heißt oftmals ohne jegliche Erläuterung oder Beschreibung, musste an dieser Stelle der Schritt hin zu einem übersichtlichen Aufbau, wenn nötig mit Beschreibungen und Informationen, gemacht werden, sodass die Funktionen für Nutzer*innen weitestgehend selbsterklärend sind.

Als erster Schritt wurde sich hierzu kurz ins Gedächtnis gerufen, wer die anvisierte Zielgruppe darstellt, um ein besseres Gespür für mögliche auftretende Unklarheiten zu bekommen. Mit diesem Wissen im Hinterkopf wurde dann basierend auf dem Wissen, dass der Umgang mit Smartphones und Browsern und deren verschiedenen Tabs dem Großteil der Zielgruppe geläufig ist, überlegt, einen solchen Aufbau mit festgelegten Fokussen – siehe auch Kapitel 5.2 – zu erstellen, wobei der ausgewählte Tab immer farblich hinterlegt wird. Insgesamt macht dieser Teil 3/5 des Bildschirms aus. Unter diesem wurde noch ein Bereich für Buttons, mit welcher die Navigationsaufgabe gestartet und ein Zustellhinweis bestätigt werden kann, angefügt. Die letzten zirka 2/5 des Bildschirms nimmt dann ein Touchpad ein, mit welchem in weiterer Folge mit den Inhalten im virtuellen Raum mittels eines vom Smartphone in den virtuellen Raum ausgestrahlten Strahls interagiert werden kann.

5.3.7 Beta – Testing

Beim letzten Testing, dem sogenannten Beta-Testing ging es auf der einen Seite darum, die neu implementierten Inhalte hinsichtlich deren Funktion zu testen, was gerade hinsichtlich der Positionierung der anzuzeigenden Zustellhinweise und der korrekten Aktualisierung der 2D-Karte, wenn gezoomt wird, eine Herausforderung darstellte. Auf der anderen Seite ging es beim Beta-Testing noch darum einen ungeübten User insofern zu simulieren, als dass Fehleingaben abgefangen werden. Beispielsweise ist an dieser Stelle gemeint, dass eine Navigationsaufgabe nicht zweimal gestartet werden kann, oder Fehleingaben bei der Ortssuche keine Fehler produzieren und so weiter. Dieser Testdurchlauf gestaltete sich ebenso wie das Alpha-Testing durchwegs aufwändig, wobei an dieser Stelle ergänzt werden muss, dass es sich bei den hier zu testenden Implementierungen um keine noch nie von jemanden in irgendeiner leicht abgewandelten Weise – wie bei der Umsetzung einer AR-Navigationsapplikation mit ARGPS auf Smartphone und XREAL Air 2 – umgesetzten Dingen handelte. Dadurch war es im Gegensatz zum Alpha-Testing auch so, dass nicht das gesamte Projekt zum Scheitern verurteilt gewesen wäre, wenn einzelne Funktionen nicht

funktioniert(en) (hätten). Überdies wäre es dadurch, dass Ähnliches bereits umgesetzt wurde auch möglich gewesen bei Problemen Hilfestellung im Internet zu suchen, was aber glücklicherweise nicht vonnöten war.

5.3.8 Finalisierung

Im letzten Schritt der Finalisierung der Applikation wurden dann nur noch einige kleine Aufgaben, wie etwa die Deaktivierung der Entwicklungsoptionen – allen voran das Debug-Menü –, die Bereinigung und das Kommentieren des Codes, soweit nicht bereits durchgeführt, gemacht. Ebenso wurde der Package-Name, also der Name, den die für ein Smartphone fertige App und deren Dateien tragen, auf den korrekten Namen, die Versionsnummer auf v.01 und jene Werte, die sich während des Beta-Testings bewährt hatten (als optimal empfunden wurden) und sich auf das Aussehen der Objekte im virtuellen Raum bezogen, als Standart-/ Starteinstellungen gesetzt.

5.3.9 „Veröffentlichung“

Abschließend, nachdem die Applikation final erstellt wurde, wurde nochmal ein finaler *build* (dt. Applikationspaket) durchgeführt, welcher für den Rahmen der Arbeit die finale Version darstellt. In weiterer Folge wird für den Rahmen der Arbeit nun diese finale Version noch genauer besprochen und aufbauend darauf ein kleiner Test der erstellten Applikation (in Kap. 6) erfolgen. Weiters wird – was an dieser Stelle nicht für die Arbeit per se von Relevanz ist, aber trotzdem Erwähnung finden sollte – daran gearbeitet, die Applikation künftig im XREAL AR-Lab und im XREAL Forum zu veröffentlichen²⁶¹.

²⁶¹ Da kostenpflichtige Module implementiert sind und bei Routingabfragen Kosten für den Autor anfallen können, werden die Applikation und die Projektdaten nicht frei im Internet zugänglich gestellt. Momentan steht die Applikation unter folgendem Link bis Ende des Jahres 2024 zur Verfügung: https://drive.google.com/drive/folders/19YVfKD2TzkZT_RnJVuSfhaZDENiSQLFB?usp=sharing

Die Projektdaten stehen ebenso unter einem geschützten Link mit Absprache der weiteren Verwendung bis Ende des Jahres 2024 unter folgendem Link zur Verfügung: https://drive.google.com/drive/folders/1o4YlNkRY_uZ_YUtEzayB8VW6fyCRAAWF?usp=sharing

Anschließend an das Jahr 2024 wird um Kontaktaufnahme mit den Autoren bzgl. der Applikation/ Projektdaten gebeten

5.4 Endprodukt

Nachdem nun in Auszügen, um nicht allzu sehr auf technische Details der Umsetzung einzugehen, die Ausarbeitung der Applikation und der einzelnen Bestandteile besprochen wurde, soll nun an dieser Stelle die fertige Applikation genauer betrachtet werden, bevor im nächsten Kapitel ein kleiner (Selbst-)Versuch hinsichtlich der Anwendbarkeit und die Besprechung der dabei zu Tage tretenden Erfahrungen gemacht wird. Zuvor wird aber an dieser Stelle nun als erstes noch der schematische Ablauf der Nutzung der Applikation beleuchtet werden: also des Starts der Applikation über die Wahl der Einstellungen und dem Start der Navigationsaufgabe bis hin zum Abschluss der Navigationsaufgabe und der Beendigung der Applikation. Hierzu wird weiters aufgrund des Aufbaus des AR-Systems, mithilfe dessen die Applikation umgesetzt wurde, zwischen der Betrachtung der auf der AR-Brille und der am Smartphone angezeigten Inhalte²⁶² hin- und hergesprungen werden müssen, wobei dies in weitere Folge als möglichst „natürlicher“ Ablauf²⁶³ dargelegt werden soll.

Begonnen werden soll die Besprechung an dieser Stelle mit dem Start der Applikation. Wie bei jeder anderen App auch, wird diese über Klick auf deren Symbol am Smartphone gestartet, wobei jedoch bedacht werden muss, dass die AR-Brille zuvor mit dem Smartphone verbunden und mittels drei Sekunden langer Betätigung der Lauter-Taste in den korrekten Nutzungsmodus für die Applikation gebracht werden muss. Wenn dies korrekt erledigt wurde²⁶⁴, startet die Applikation und man gelangt zur initialen Benutzeroberfläche am Smartphone (siehe Abb. 36), während auf der AR-Brille noch keine Inhalte angezeigt werden. Bleibt man an dieser Stelle beim Start-Tab kann man per Klick auf das Dropdown-Menü von hier aus drei verschiedene Funktionen starten: 1) Navigation entlang einer vordefinierten Route (predefined Route), 2) eine Routensuche und darauf basierende Navigation, und 3) die Platzierung eines Testwürfels oder Hauses an einer vordefinierten Position oder einem koordinativ verorteten Punkt (siehe hierzu Abb. 37 – 39).

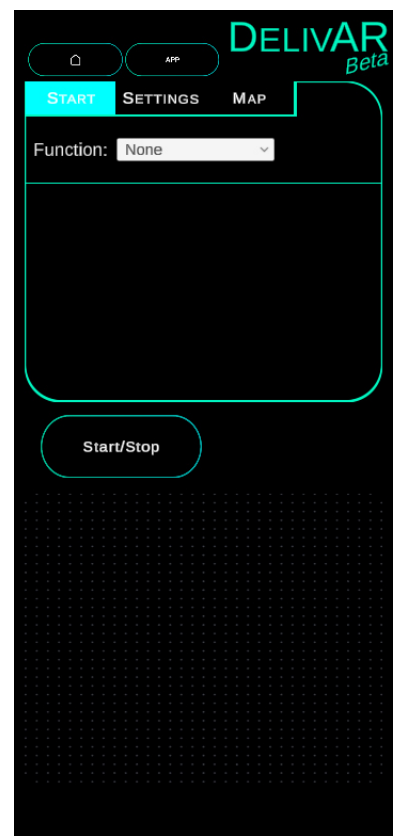


Abbildung 36: Benutzeroberfläche nach dem Start

²⁶² Hinsichtlich der auf der AR-Brille angezeigten Inhalte sei erwähnt, dass es relativ schwierig ist, diese gut aufzunehmen, da die einzige Möglichkeit hierzu das durch ein Brillenglas-hindurch-Fotografieren ist und demnach die Bildqualität nur suboptimal ist. Außerdem erscheinen Objekte versetzt (d.h. liegen teils näher beisammen), da nur ein Teil des stereoskopischen Bildpaares aufgenommen werden kann.

²⁶³ Hiermit ist gemeint: Start der Applikation → Definition der Einstellungen für Navigation → Start der Navigation → Nutzung der Navigation → Beendigung der Navigation

²⁶⁴ Wurde dieser Prozess nicht korrekt erledigt, so erscheint beim Start der Applikation eine Fehlermeldung.



Abbildung 39: Auswahl einer vorab definierten und in der App hinterlegten Route

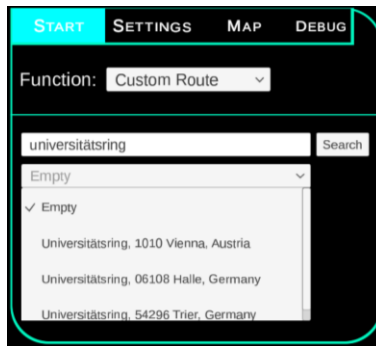


Abbildung 40: Suche eines Zielpunkts und Auswahl des selbigen



Abbildung 38: Auswahl eines an einer bestimmten Position zu platzierenden Objekts

Anschließend an das Auswählen der gewünschten Funktionen existieren zwei Möglichkeiten zum weiteren Vorgehen, wobei entweder direkt a) die gewünschte Funktion per Knopfdruck auf „Start/Stop“ gestartet werden kann, oder b) zuerst noch, etwa wenn die Applikation bereits bekannt ist, Einstellungen im Settings-Tab getroffen werden können. Zu den im Settings-Tab vorzufindenden Einstellungen soll an dieser Stelle ergänzend erwähnt werden, dass die dort vorzufindenden Einstellungen auch während einer bereits laufenden (Navigations-)Aufgabe angepasst werden können. Der Einstellungs-Tab (siehe Abb. 40) ist dabei in zwei Bereiche unterteilt, einmal jenen mit den „Raw Application Settings“ und einmal jenen mit den „Navigational Hints Styling“-Optionen. Zu ersterem sei gesagt, dass sich in diesem nur die Option der Anpassung des Alignments, was im Sinne der Applikation eine Option zur Einstellung eines Offsets der Drehrichtung (Yaw) der AR-Brille zum Smartphone²⁶⁵ darstellt befindet. Die anderen beiden Optionen der Einstellung der „Raw Height“ und des „Calculation Modes“ sind dabei Überbleibsel aus der Entwicklungsphase, die aber aufgrund von weiteren Testzwecken (im Bereich der Platzierung von Häusern) noch nicht entfernt, sondern nur deaktiviert wurden. Der zweite Bereich wiederum widmet sich wie der Name schon sagt den zuvor bereits angesprochenen Einstellungen rundum das Styling der Navigationshinweise. Hierbei steht zum einen der generelle „Route Style“ (nur Pfeile und Hinweistafeln an Wegkreuzungen, nur eine Navigationslinie, beides gemeinsam) wie auch die Farbe („Color“), Größe („Size“) und Positionierungshöhe („Height“) aller Hinweiskomponenten (Pfeile, Tafeln, Linien) zur Auswahl. Darüber hinaus existiert dann noch ein Eingabefeld für den „Hint Radius“, welcher definiert, in welcher Distanz zur Zustelladresse ein Zustellhinweis angezeigt wird, und ein „Smoothing“-Schieberegler, mit welchem sich das „Herumspringen“ von

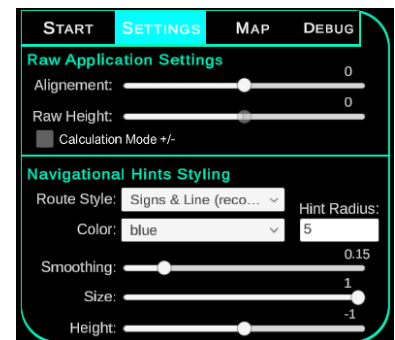


Abbildung 37: Settings Tab und dessen Einstellungsoptionen

²⁶⁵ Dies wird benötigt, wenn die Anzeige bei Start der Navigationsfunktion nicht genau einer Straße folgt, sondern beispielsweise um 45° verdreht auf dieser liegt. Dies kann durch Probleme bei der Initialisierung oder leicht unterschiedlich verbauten Sensoren in HMD und Smartphone der Fall sein.

Hinweiskomponenten (etwa aufgrund der sprunghaften Veränderung der Positionierungsgenauigkeit von GNSS) in gewissem Maße abfedern lässt.

Der dritte und letzte in der zu veröffentlichenden Version enthaltene Tab ist der „Map-Tab“ (siehe Abb. 41), in welchem sich eine 2D-Karte der Umgebung mit unterschiedlichen Zoom-Levels befindet. Wurde bereits die Navigation entlang einer vordefinierten Route gestartet, kann man sich auf der Karte mittels Setzen eines Hakens in einem Kontrollkästchen die jeweilige, gerade aktive Route auf der Karte anzeigen lassen. Selbstverständlich wird darüber hinaus der aktuelle Standort auf der Karte immer mittels eines PINs, welcher sich je nach Standort mitbewegt, angezeigt. Zusätzlich existiert in diesem Tab noch ein Feld, welches die WGS84-Koordinaten des aktuellen Standorts anzeigt.

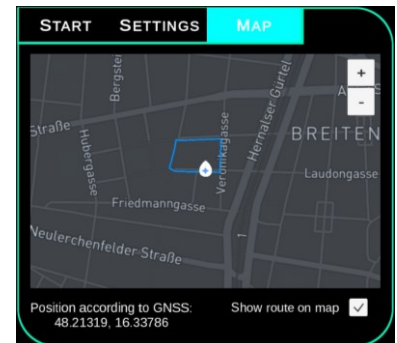


Abbildung 41: Anzeige der Route im Map-Tab

Wird also nun eine ausgewählte Funktion (Predefined Route oder Custom Route) gestartet²⁶⁶, so bleibt die Benutzeroberfläche bis auf eine kurze textliche Zusammenfassung der gestarteten Funktion²⁶⁷ (siehe Abb. 42), welche nach wenigen Sekunden aber verschwindet, unverändert. Einzig im Map-Tab besteht nun, wie bereits erwähnt, die Möglichkeit, sich zusätzlich zur 2D-Karte der Umgebung auch noch die eingezeichnete Route anzeigen zu lassen (siehe Abb. 41). Auf der AR-Brille hingegen erscheinen an diesem Punkt dann die für die Navigation benötigten Hinweise, also je nach ausgewähltem Styling die Route, Wegpfeile und Hinweistafeln. Wie angemerkt, unterscheiden sich die angezeigten Inhalte dabei je nachdem, welche Stylingoptionen getroffen wurden: Als Beispiel für die Standardeinstellungen – vollumfängliche Anzeige von Route und Hinweistafeln in blauer Farbe – sei an dieser Stelle auf die Abbildung 43 und Abbildung 45 links verwiesen. Dies bedeutet also, dass nun die eigentliche Navigation gestartet ist, der Controller, also das Smartphone etwa in der Hosentasche verstaut werden kann, und man nun damit

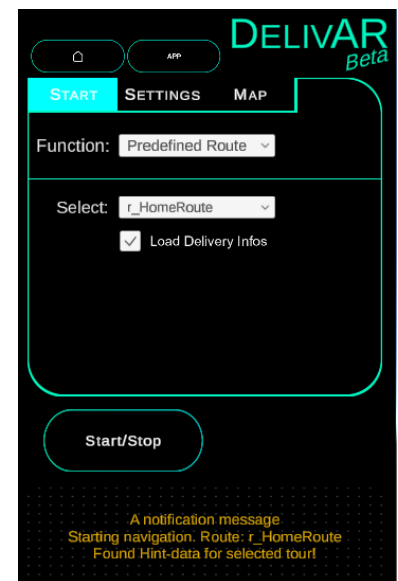


Abbildung 42: Anzeige einer Benachrichtigung bei Start einer Funktion

beginnen kann, sich zu bewegen. Hierbei verändern sich die angezeigten Inhalte dann in dem Sinne, als dass die Inhalte immer passend platziert werden und sich die auf den an Abzweigungen positionierten Hinweistafeln eingeblendete Distanz bei Annäherung verringert. Wird eine Hinweistafel erreicht und dem Navigationshinweis gefolgt, so blendet sich die nächste Hinweistafel ein und die alte wird ausgeblendet. Dies geschieht so lange, bis das Ziel der Navigation, welches mit einem PIN markiert wird erreicht wird.

²⁶⁶ Wie erwähnt müssen im Start-Tab die jeweils korrekten Parameter hierfür ausgewählt sein.

²⁶⁷ ... oder bei einem Fehler eine Fehlermeldung bzw. bei einer fehlenden Eingabe seitens der nutzenden Person ein Hinweis hierzu

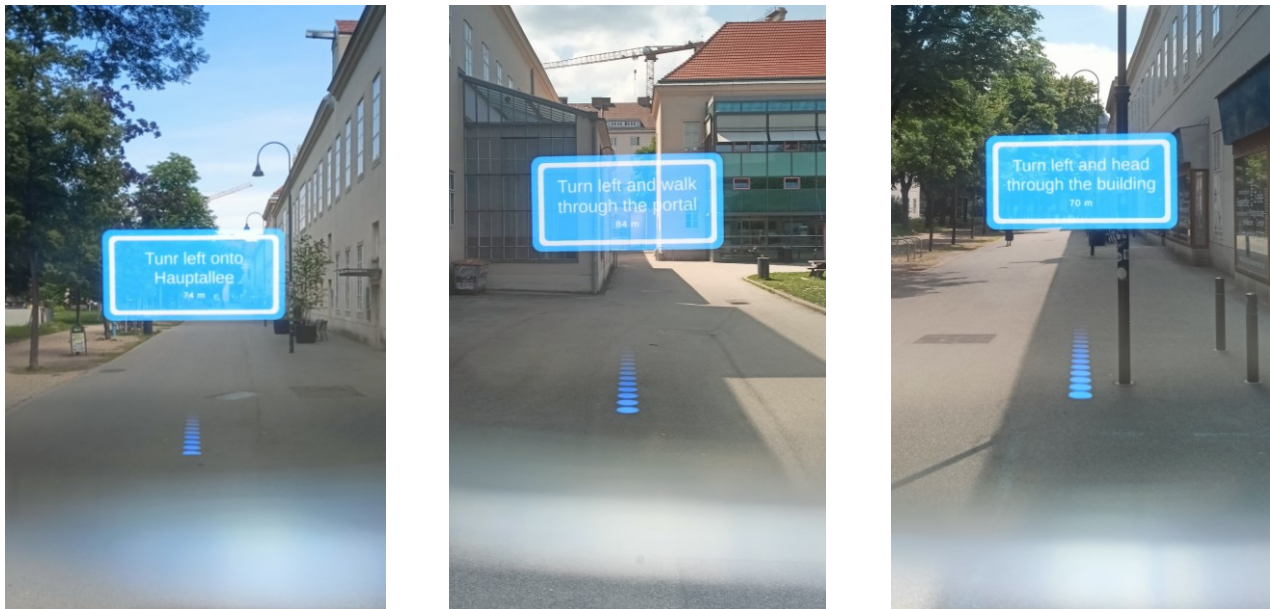


Abbildung 43: Beispiele für die Anzeige von Navigationshinweisen mithilfe der Standardeinstellungen – Fußnote 262 beachten!

Zusätzlich gibt es noch den Fall, dass die Anzeige von Zustellhinweisen (nur bei „Predefined-Routes“ möglich) ausgewählt wurde, wobei es hier dann zur Einblendung eines Zustellhinweises zusätzlich zu den normalen Navigationshinweisen auf der AR-Brille (siehe Abb. 45 Mitte) kommt, sobald man sich in einer vorab definierten Distanz zur Zustelladresse befindet. Diese kann am Smartphone mittels Klick auf den nur bei Aktivität eines Zustellhinweises angezeigten „Commit“-Buttons (siehe Abb. 41) deaktiviert werden. Weiters existiert noch eine letzte weitere Einblendungen (siehe Abb. 45 Rechts) für den Fall, dass sich die anzuzeigende Route außerhalb des momentanen Blickfeldes befindet, was an dieser Stelle mit einem Pfeil in dessen Richtung die Route liegt signalisiert wird.

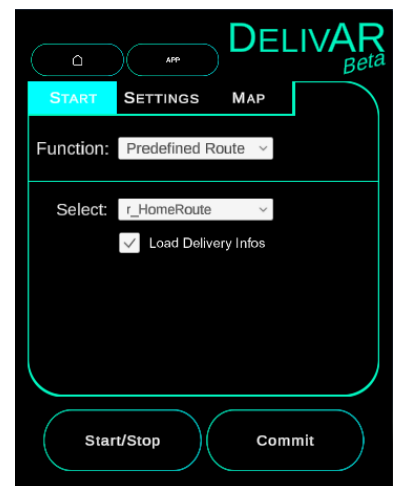


Abbildung 44: Anzeige des Commit Buttons am Controller

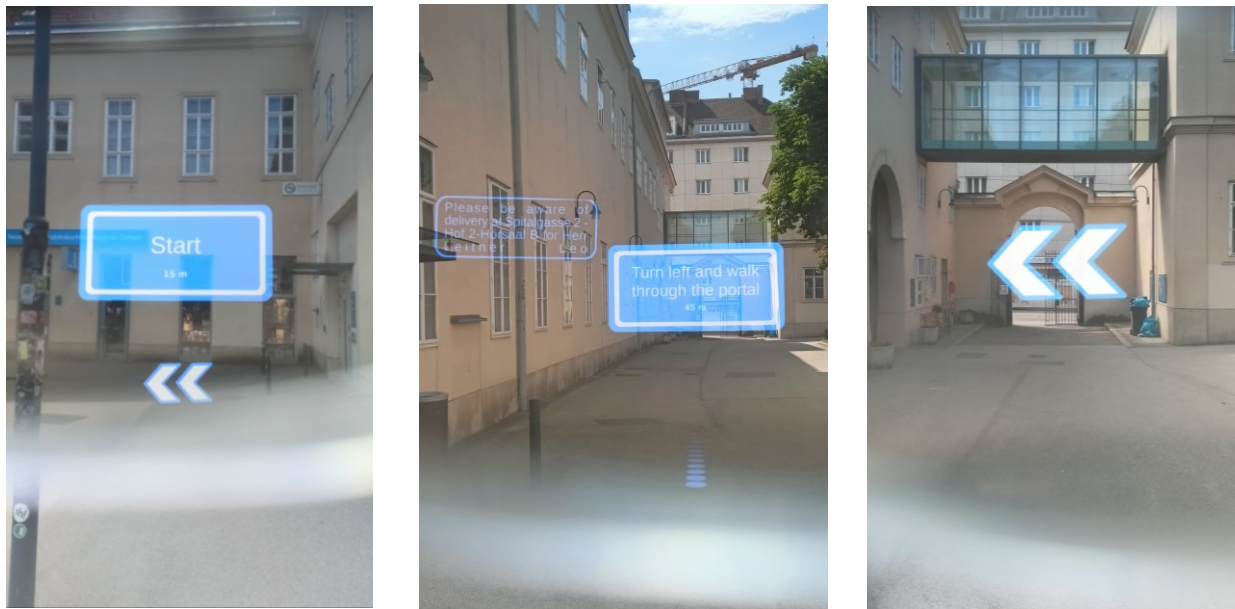


Abbildung 45: Von links nach rechts: Anzeige eines Navigationshinweises direkt an einem Kreuzungspunkt; Navigationsanzeige mit Zustellhinweis; Hinweisanzeige, wenn eine Route außerhalb des Blickfelds liegt – Fußnote 262 beachten!

Soll abschließend die ausgewählte Navigationsaufgabe gestoppt werden, etwa weil man die Navigationshinweise nicht mehr benötigt oder weil man am Ende einer Route angekommen ist, so reicht ein abermaliger Klick auf den „Start/Stop“-Button: Die Aufgabe wird beendet und es wird eine Informationsmitteilung hierzu am Smartphone angezeigt. Nun kann entweder erneut eine beliebige Funktion gestartet oder die Applikation beendet werden.

Um die Applikation endgültig zu schließen, muss nun oben links auf der Benutzeroberfläche auf das Haussymbol gedrückt werden, woraufhin sich auf der AR-Brille ein Fenster mit zwei Buttons öffnet, wobei hier der Exit Now-Button mittels des vom Smartphone ausgestrahlten Pointers anvisiert werden und die Eingabe am Button mittels Klick aufs Touchpad bestätigt werden muss (siehe Abb. 46). Alternativ kann die Applikation natürlich auch über die Bedienbuttons am unteren Ende des Smartphones, wie jede andere App auch geschlossen werden.

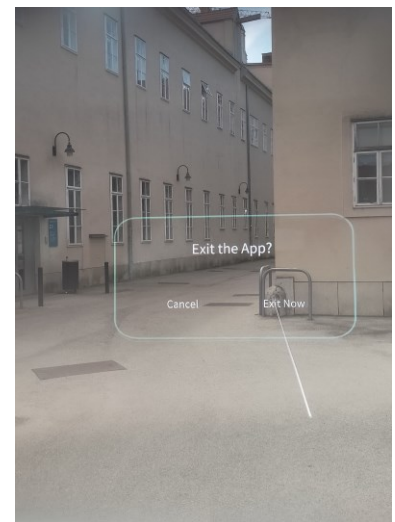


Abbildung 46: Dialog zum Beenden der Applikation

Weiters existiert noch eine versteckte Funktion, welche aber rein dazu dient, den vom Smartphone ausgestrahlten Pointer neu zu zentrieren, falls hier ein Fehler auftritt: Hierzu muss das Haussymbol für einige Sekunden gedrückt gehalten werden, woraufhin sich der Pointer neu zentriert und wieder funktionieren sollte.

5.5 Probleme bei und Learnings aus der Erstellung

Generell ist es hinsichtlich des oberhalb dargelegten Prozesses zur Erstellung der Applikation, wie erwähnt, so, dass es sich hierbei um einen idealtypischen Prozess der Erstellung, wie er beim finalen Anlauf zur Erstellung der finalen Applikation durchgeführt wurde, handelt und darum auch nicht auf Probleme und Learnings, die zuvor gemacht wurden, explizit hingewiesen und eingegangen wurde. Wie ebenfalls bereits angemerkt, wurde mit der Entwicklung der Applikation mehrmals komplett neu von Grund auf gestartet, was zum einen mit dem Lernprozess während der Erstellung zu tun hatte und zum anderen damit, dass sich manches Mal in Sackgassen manövriert wurde, wobei der Fehler, der gemacht wurde, nicht mehr rückverfolgbar war und/ oder Aufwand das Projekt gleich von Grund auf neu aufzubauen gleich groß war wie die Fehlersuche und -korrektur. Ein etwaiger Fall, bei welchem es relativ zu Beginn der Applikationsentwicklung nötig wurde, von neuem zu starten, war etwa das Verwenden von multiplen Szenen für verschiedene Funktionen. Simplifiziert gesagt war es bei der Verwendung multipler Szenen nötig, irgendwo Informationen, welche von einer in die andere Szene übermittelt werden sollten, zu speichern und wieder zu lesen, was zu diversen Problemen hinsichtlich Speicherort, Performanz und Stabilität der Applikation führte. Außerdem war es in diesem Fall nicht möglich dauerhaft eine Benutzeroberfläche, auch während laufender Navigation anzubieten. Diese Probleme wurden dann so gelöst, dass alle Funktionen innerhalb einer Szene realisiert wurden. Bis jedoch der Punkt, an welchem die Probleme augenscheinlich wurden, erreicht worden war, waren bereits mehrere Wochen Entwicklungszeit vergangen, da um die Probleme zu bemerken, ein erstes kleineres Testing der Applikation und deren Funktionen nötig war.

Ein weiteres Learning aus dem Prozess der Erstellung war auch die Versionierung von Entwicklungsschritten. Im Rahmen dieses Projekts wurde nach einiger Weile damit begonnen, an jedem Entwicklungstag die bereits zuvor und neu implementierten Funktionen grob zu testen, eine Sicherungskopie des Projektordners anzulegen und kurz stichwortartig die Änderungen zusammenzufassen. Für die Zukunft würde es sich an dieser Stelle sicherlich lohnen, sich genauer mit github und den Versionierungsoptionen dieser Applikation auseinanderzusetzen, um Entwicklungsschritte tatsächlich Schritt für Schritt rückverfolgbar und analysierbar zu machen, was es im Umkehrschluss ermöglichen würde, festzustellen wo etwaige Fehler liegen und zu einem funktionierenden Letztstand zurückzukönnen.

Ein weiteres, während Ende des Erstellungsprozesses gemachtes Learning war dann noch die Einsicht, dass es höchstwahrscheinlich effizienter und besser gewesen wäre zuerst die zusätzlich selbst zu entwickelnden Funktionen (Anlegen & Auswahl vordefinierter/ freier Routen, Anzeige von Zustellhinweisen, ...) für die Anwendung auf einem Smartphone, wie es mithilfe der AR + GPS Location Erweiterung grundsätzlich gedacht ist, zu entwickeln und erst wenn diese korrekt funktionieren, mit dem Umbau der auf dem Smartphone funktionierenden App für Smartphone plus AR-Brille zu beginnen. Der Grund hierfür liegt in der Tatsache, dass auf diesem Wege getestet hätte werden können, ob die Funktionen per se korrekt funktionieren, ohne eine weitere potenziell mögliche Fehlerquelle in Form des

bereits eingebundenen NRSDK und der damit vollzogenen Verknüpfung von Smartphone und AR-Brille zu haben. Hier kommt jener, bereits oberhalb angesprochene durchwegs komplexe und viel Zeit kostende Faktor der korrekten Abfrage, Berechnung und Zuweisung von Positions- und Rotationsinformationen zur problemlosen gemeinsamen Verwendung eines Smartphones und der AR-Brille für den Zweck einer AR-Navigationsapplikation auf HMDs zu tragen. Wäre wie vorgeschlagen vorgegangen worden – d.h. dass zuallererst alle Funktionen rein für AR auf dem Smartphone entwickelt worden wären – hätten a) alle Funktionen fertig entwickelt und getestet werden können ohne dass man sich um die Kommunikation zwischen AR-Brille und Smartphone kümmern hätte müssen, und b) hätte in einem zweiten Schritt nur mehr die Anzeige vom Handydisplay auf die AR-Brille verschoben, die Bedienfläche überarbeitet und die Kommunikation zwischen Smartphone und AR-Brille (also Position & Rotation der Geräte) geregelt werden müssen. Außerdem hätte in diesem Fall (notfalls) die Möglichkeit bestanden, die Applikation (bei Zeitdruck, oder noch tiefgreifenderen Problemen) zusätzlich auch rein für Smartphones zu veröffentlichen. Dies hätte bedeutet, dass in diesem Fall so auch die Möglichkeit bestände hätte, direkt zwei Applikationen, nämlich eine rein für Smartphones und die andere für Smartphone und AR-Brillen, bereitzustellen.

Ein weiteres Learning wurde dann noch rundum die sprachbedingte Formatierung von Zahlen und Dateien gemacht. Hiermit ist gemeint, dass im Englischen das Kommatrennzeichen ein Punkt und das Tausendertrennzeichen ein Komma (also bspw. 2,568.17) ist, während es im Deutschen genau umgekehrt (also bspw. 2.569,17) ausgedrückt wird. Dies führte etwa bei der Abfrage von Koordinaten und deren Umrechnung zu einem lange nicht identifizierbaren Problem, da bei Testläufen der Applikation am Computer die Zahlen korrekt berechnet wurden, aber am Smartphone Probleme beziehungsweise vice versa auftraten. Der zweite Punkt war ebenso im Bezug auf die Verwendung von Beistrich beziehungsweise Strichpunkt, aber in diesem Fall bei der Verwendung von CSV-Dateien, wobei sich das Problem ebenso wie beim Tausendertrennzeichen gestaltete. An sich stellt sich dieser Punkt, nachdem das Problem einmal identifiziert worden war, nicht als kompliziert zu lösen dar, jedoch dauert es doch einige Zeit bis so ein Fehler identifiziert werden kann. Der Grund für die falschen Kommatrennzeichen kam dabei dadurch, dass bei der Datenabfrage von Sensorwerten nicht immer dasselbe Format ausgegeben wird, sondern dieses von der eingestellten Sprache am Smartphone abhängt. Das Problem der Trennzeichen bei CSV-Dateien war wiederum auf die Spracheinstellungen in Excel, welches zur Erstellung der Dateien benutzt wurde zurückzuführen.

Das nächste im Rahmen der Erstellung der Applikation gemachte Learning bezog sich auf das Dokumentieren der Arbeitsschritte und Codes. Gerade wenn mehrere Tage zwischen der Entwicklung lagen, trat das Problem auf, dass vergessen wurde, was wie gemacht und umgesetzt wurde²⁶⁸. Ein weiteres No-Go ist dann noch, sofern nicht unweigerlich nötig das Editieren von Skripten, welche man nicht selbst erstellt hat: Zum einen verbaut man sich

²⁶⁸ An dieser Stelle sei abermals auf github für solche Zwecke verwiesen, auch wenn sich selbst mit Textdokumenten mit Stichwörtern beholfen wurde.

dadurch das Updaten der SDKs/ Erweiterungen²⁶⁹ und zum anderen verliert man sehr schnell den Überblick über geänderte Passagen. Mindestmaß an dieser Stelle sollte sein, dass sich eine Logik zurechtgelegt wird, wie veränderte Codeteile dokumentiert werden können. Im Falle der vorliegenden Arbeit wurde hier so vorgegangen, dass ein Kommentar am Ende der auskommentierten/ neu erstellten Codezeile beziehungsweise am Ende des auskommentierten/ neu erstellten Codeblocks erstellt wurde, welches mit Namenskürzel in Klammer begann und anschließend daran eine kurze Beschreibung der Änderung und des Grunds der Änderung beinhaltete (siehe Abb. 47).

```
private SignPostEventArgs createSignPostEventArgs(int index)
{
    //var user = Camera.main.transform.position; //(RS) uncommented: to make it work with NREAL
    var user = toBeMainCam.transform.position; //(RS) added: to make it work with NREAL
    var target = s.StepsPlaceAtInstances[index].transform.position;
    var instruction = s.RouteSteps[index].maneuver.instruction;
    var name = s.RouteSteps[index].name;
}

public void clearRoute() //(RS) changed: set the function from private to public to access it from the menu-script
{
    foreach (var signpostInstances in s.SignPostInstances)
    {
        foreach (var instance in signpostInstances)
        {
            Destroy(instance.gameObject);
        }
    }
}
```

Abbildung 47: Zwei Beispiele für Änderungen und deren Dokumentation im fremden Code

Ein weiteres Learning bezieht sich auf die während der Entwicklungszeit neu herauskommenden Updates, sowohl hinsichtlich der selbst verwendeten Erweiterungen und SDKs, der Entwicklungsumgebung, als auch für die ausführenden Geräte: Die Relevanz liegt hierbei daran, dass es ohne eine Sicherungskopie beziehungsweise eine Lösung für einen Rollback und ohne sich bis ins letzte Detail mit den Änderungen auseinanderzusetzen durchwegs riskant ist, ein Update durchzuführen, da im schlimmsten Fall bereits funktionale Funktionen plötzlich nicht mehr funktionieren. Dieses Learning wurde gemacht, da bei einem Update der NRSDK von Version 2.0.1. auf 2.1.0 plötzlich keine der Funktionen hinsichtlich der Kommunikation zwischen Smartphone und AR-Brille mehr korrekt funktionierte und der Fehler nicht nachvollzogen werden konnte. Insgesamt endete dies im Neuaufbau der gesamten Applikation und somit in einem erheblichen Ärgernis mit Frust und einem immensen Zeitaufwand. Natürlich muss hierzu relativierend gesagt werden, dass Updates zumeist Performance-Verbesserungen und Fehlerbehebungen enthalten, es aus Sicht dieser Arbeit während der Entwicklung aber nur dann zu empfehlen ist, ein Update durchzuführen, wenn ein bestimmter Fehler, Probleme in der Performance oder Ähnliches dezidiert auf die alte Version zurückzuführen sind und eine neue Version dies lösen soll, beziehungsweise man sich zu 100 Prozent sicher ist, dass das Update kein Problem darstellt. Ansonsten gilt: „Never change a running system“.

Ein vorletzter Hinweis soll an dieser Stelle nicht außenvorgelassen werden: Hierbei handelt es sich darum, dass es sich bei einem Error, welcher bei der Ausführung einer Applikation

²⁶⁹ Dies stellte im Rahmen der Arbeit bei Änderungen in den Skripten der mapbox / AR + GPS Location-Erweiterung zum Glück kein großes Problem dar, da es sich hier ohnehin um Pakete handelt, wo es so scheint, dass diese nicht mehr gepflegt und upgedatet werden und man sich somit ohnehin selbst behelfen muss.

im Debug-Menü auftritt, nicht zwangsläufig um einen Fehler, welcher auf die Funktion der Applikation Auswirkungen hat, handeln muss. So ist es beispielsweise bei der eigens entwickelten Applikation so, dass bei Testläufen zwar Error-Warnungen ausgegeben wurden, trotz alledem aber alle Funktionen einwandfrei funktionierten. Dies bedeutet nicht, dass die Meldungen einfach ignoriert werden sollten, sondern viel eher, dass diese beizeiten gelöst werden müssen, aber nicht immer direkt ein *make-or-break*-Kriterium sind.

Obwohl nicht direkt auf ein Learning per se bezogen, sei zum Schluss noch angemerkt, dass es mit dem im Rahmen der Ausarbeitung aufgebauten Wissen und der Entwicklung von HMDs in diesem Zeitraum, aufgrund eines geringeren Arbeitsaufwandes in der Umsetzung besser gewesen wäre, auf eine AR-Brille mit Kamerasystem, wie etwa die XREAL Air 2 Ultra, zu setzen. Zwar hätten hiermit auch negative Aspekte, wie etwa Latenzzeiten bei der Anzeige von Inhalten bei Nutzung der Kamera²⁷⁰ auftreten können, jedoch wäre bei einer AR-Brille mit Kamerasystem und Verwendung desselbigen der Arbeitsaufwand um einiges geringer gewesen, da die ARGPS-Erweiterung weniger angepasst hätte werden müssen. Andererseits hätte selbst bei Verwendung von AR-Brille mit Kamerasystem überdies der problematische Punkt der zwangsläufigen Nutzung von IMU- und GNSS-Sensorik von unterschiedlichen Geräten (Smartphone & HMD) nicht gelöst werden können, jedoch wäre auf diese Weise vielleicht Zeit geblieben sich anderen Problematiken, wie der Stabilisierung von Einblendungen und der besseren Platzierung von Inhalten im AR-Raum, dank Pattern Recognition zu widmen. Abschließend soll an dieser Stelle aber noch etwas positives erwähnt werden und zwar, dass aufgrund der Konzeptionierung der im Rahmen der Arbeit entwickelte App (Einsatz von GPS & IMU ohne Kamerasystem), diese sowohl auf AR-Brillen von XREAL mit wie auch ohne Kamerasystem funktioniert, solange diese eine IMU besitzen, was definitiv einen Vorteil der durchgeführten Umsetzung darstellt.

²⁷⁰ Die Nutzung der Kamera wäre Beispielsweise bei der XREAL Air 2 Ultra nicht zwangsläufig nötig gewesen. D.h. der Aufbau der App hätte genauso wie bei der verwendeten XREAL Air 2 stattfinden können. Die Verwendung hätte aber stattfinden können bzw. ergänzend verwendet werden können.

6 Use-Case: Anwendbarkeit der AR-Applikation

Der letzte, im Rahmen dieser Arbeit noch zu besprechende Punkt bezieht sich auf das praktische Testen der zuvor entwickelten Applikation. Hierbei handelt es sich aufgrund des bereits betriebenen Aufwands zur Ausarbeitung tatsächlich nur um einen sehr klein gehaltenen Testaufbau ohne Probandengruppe, Vergleichsmethode oder ähnliches, sondern rein um das eigene Testen der entwickelten Applikation und die Reflexion, ob solch eine Applikation per se (im Bereich von Zustelltätigkeiten sinnvoll einsetzbar) wäre. Falls ja, wären in weiterer Folge ohne Frage natürlich umfangreichere Tests möglich und als durchwegs sinnig zu bezeichnen. Falls nein, soll aufgezeigt werden, welche Schritte im weiteren Verlauf noch gemacht werden müssen, um eine solche Applikation sinnvoll einsetzbar zu machen.

Der Test wurde in zwei Teile geteilt, wobei sich ersterer der fußläufigen Nutzung der Applikation und zweiterer der fahrzeuggestützten Fortbewegung widmete. Als Testgebiet für das erste Szenario wurde dabei der Campus der Universität Wien (Spitalgasse 2, 1090 Wien) gewählt. Der Grund für diese Wahl lag darin, dass man dort in keiner Weise am Straßenverkehr teilnimmt und somit weder sich noch andere gefährdet – also das Risiko möglichst geringhält. Diese Vorsichtsmaßnahme war dabei nötig, da es sich immerhin um die initiale etwas größere Testung einer Applikation, welche Inhalte im Blickfeld darstellt (+ ggf. Fehler bei anzuzeigenden Inhalten und eine komplette Störung des Sichtfelds eintreten könnten) handelt und es somit möglich ist, dass diese ablenkend wirken. Ebenso bietet das Gelände das optimale Umfeld zur Testung einer „städtischen Umgebung“ mit einer Vielzahl nahe beisammen liegender fiktiver Zustelladressen. Die fiktive Testroute war wie auf Abbildung 48 zu sehen gestaltet, hatte etwa 700m Wegstrecke und wies zehn Zustelladressen mit Zustellhinweisen²⁷¹ auf. Weiters wurde beim Aufbau der Route darauf geachtet, dass neben nahe beisammen liegenden Zustelladressen auch sich gegenüberliegende

Zustelladressen existieren, sodass auch diesbezüglich eine möglichst realitätsnahe Situation getestet werden konnte. Ebenso wurde eine Stelle mit einer Wegüberschneidung und mehrere Durchgänge unter Gebäuden eingebaut, um auch bei solchen Situationen zu sehen, wie gut oder schlecht die Applikation funktioniert.

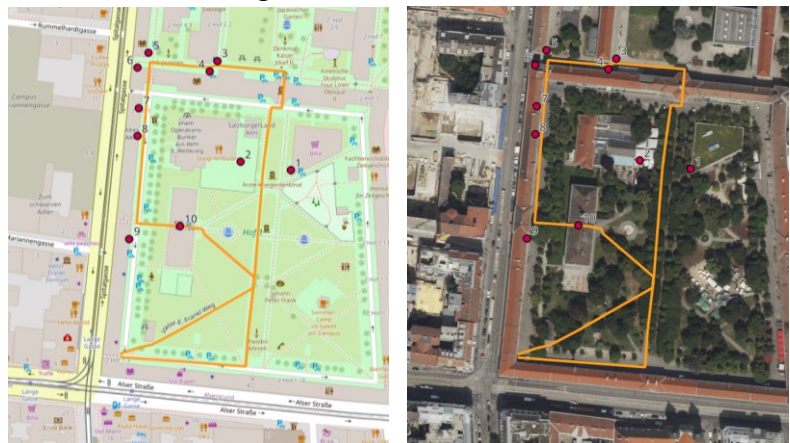


Abbildung 48: Route am Campus der Universität Wien, visualisiert auf Basis von OSM/ BING Maps

²⁷¹ Die Zustellhinweise stellen ebenso wie die Route an sich fiktive Daten dar!

	A	B	C	D	E	F	G	H	I	J	K
1	id	lat	lon	adress 1	adress 2	adress 3	receptient	amount	kind	info 1	info 2
2	1	48.21632467	16.35286802	1090 Vienna	Spitalgasse 2 - Hof 1	Billa	Billa AG		3 parcel	signature	
3	2	48.21637432	16.35240392	1090 Vienna	Spitalgasse 2 - Hof 1	Stiegl Ambulanz	Herr Maier Mario		1 RSB	signature	personal or family
4	3	48.21699757	16.35218895	1090 Vienna	Spitalgasse 2 - Hof 2	Eingang 2.2	Frau Yin Linda		1 letter	no signature	letterbox applicable
5	4	48.2169306	16.3521176	1090 Vienna	Spitalgasse 2 - Hof 2	Hörsaal B	Herr Leitner Leo		1 small parcel		
6	5	48.21704543	16.35155676	1090 Vienna	Spitalgasse 2 - Hof 4	Eingang 4.1	Herr Onur Sedlak		1 RSA	signature	personal
7	6	48.2169556	16.3514529	1090 Vienna	Spitalgasse 2	Portier	Frau Luger Marianne		4 parcel	signature	
8	7	48.2167065	16.3514667	1090 Vienna	Spitalgasse 2 - Hof 1	Seminarraum 1	Frau Meister Brigitte		1 Hybrid	signature	personal or family
9	8	48.2165351	16.3514529	1090 Vienna	Spitalgasse 2 - Hof 1	Seminarraum 2	Herr Walter Sebastian		2 parcel	signature	
10	9	48.2159017	16.3513778	1090 Vienna	Spitalgasse 2 - Hof 1	Alumniverband	Herr Willing Franz		1 Einschreiben	signature	
11	10	48.21597781	16.35184497	1090 Vienna	Spitalgasse 2 - Hof 1	Stöckl-Gebäude 1.17	Frau Mustermeister		1 EMS	signature	delivery until 12.00

Abbildung 49: Liste der Orte mit Zustellhinweisen der Campus Route²⁶⁹

Hinsichtlich der Testung bei fahrzeugbasierter Fortbewegung wurde der Testaufbau dahingehend angepasst, als dass zum einen, um etwaige Risiken auszuschließen und andererseits rechtliche Fragestellungen außen vor lassen zu können, die Applikation nur als Beifahrer genutzt wurde. Als Gebiet zur Testung des zweiten Szenarios wurde dabei bewusst ein ländliches Gebiet in Oberösterreich (Gmeinerhof/ Halmenberg, 4230 Pregarten) gewählt, da dies eine typische ländliche Umgebung mit einer Vielzahl von weit voneinander entfernten Ein-/Mehrfamilienhäusern ist. Die Testroute war wie Abbildung 50 zu entnehmen angelegt: Sie war ca. 3,2 km lang und wies sechs Zustelladressen mit Zustellhinweisen²⁷¹ auf. Bei dieser Route wurde ebenso wie bei der fußläufig zu bewältigenden Strecke darauf geachtet, diese möglichst realitätsnah anzulegen, soll heißen mit Strecken, die hin und retour gefahren werden und auch mit kleineren Schlenkern, wie etwa Hauszufahrten.



Abbildung 50: Route in Oberösterreich, visualisiert auf Basis von OSM/ BING Maps

	A	B	C	D	E	F	G	H	I	J	K
1	id	lat	lon	adress 1	adress 2	adress 3	receptient	amount	kind	info 1	info 2
2	1	48.3649279	14.5626191	4230 Pregarten	Gmeinerhof 13		Herbert Jedermann		1 RSB	signature	personal or family
3	2	48.3615808	14.567617	4230 Pregarten	Gmeinerhof 5		Hans Kunz		1 small parcel		
4	3	48.3585389	14.5712893	4230 Pregarten	Gmeinerhof 9	Kapeller	Josef Kapeller		1 RSA	signature	personal
5	4	48.35229	14.5700122	4230 Pregarten	Halmenberg 2	Franzlhof	Maria Maier		4 parcel	signature	
6	5	48.3522636	14.569736	4230 Pregarten	Halmenberg 3		Edeltraut Werner		1 Hybrid	signature	personal or family
7	6	48.3520478	14.5695305	4230 Pregarten	Halmenberg 4		Jürgen Eberer		2 parcel	signature	

Abbildung 51: Liste der Orte mit Zustellhinweisen der Oberösterreich-Route²⁶⁴

6.1 Testung der Applikation

Bevor im nächsten und übernächsten Abschnitt ein Fazit zur Anwendbarkeit der Applikation bei fußläufigen beziehungsweise fahrzeugbasierten Zustelltätigkeiten gezogen und ein Ausblick auf mögliche Verbesserungen²⁷² gegeben wird, soll hier noch kurz der Ablauf der Testung und aufgetretene Probleme angesprochen werden. Wie bereits erwähnt, wurde das Testen der Applikation zweigeteilt, das heißt einmal für die fußläufige Bewegung in städtischer Umgebung und einmal für fahrzeugbasierte Bewegung in ländlichem Umfeld durchgeführt. Ersterer Teil²⁷³ wurde dabei am 20.05.2024 nachmittags bei gutem und zweiterer am 31.05.2024 abends bei schlechtem Wetter durchgeführt.

6.1.1 Fußläufige Testung:

Der Ablauf der fußläufigen Testung gestaltete sich dabei wie folgt: Als erstes wurde sich in der Nähe des Startpunktes der zu navigierenden Route positioniert und dort die Einsatzbereitschaft des Systems²⁷⁴ hergestellt und die Applikation gestartet. Hierbei konnte auch direkt ein bereits bekanntes Problem, nämlich jenes der inkorrekten Verortung von Smartphone und AR-Brille zueinander beobachtet werden. Dieses Problem tritt auf, wenn beim Start der Navigationsapplikation, im Zuge dessen die AR-Brille und das Smartphone zueinander verortet werden, die beiden Geräte nicht ein und denselben Drehwinkel (entlang der Z-Achse bzw. in Unity ist dies die Y-Achse – „Yaw“) aufweisen. Dies stellt insofern nur ein geringes Problem dar, da dieses bereits bekannt war und eben genau hierfür ein Anpassungsparameter im Settings-Tab („Alignement“) der Applikation eingebunden wurde. War diese initiale Einstellung aber einmal getroffen, funktionierte die Applikation bis auf zwei weitere Punkte ohne Probleme.

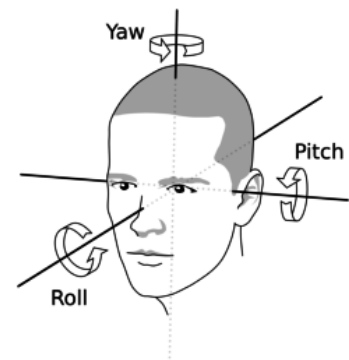


Abbildung 52: Yaw, pitch, roll bei der Kopfbewegung von Menschen nach (Jantunen, 2016)

²⁷² Diese werden nicht mehr im Rahmen der Arbeit durchgeführt werden. Nach momentanem Stand sieht es aber danach aus, als dass die Applikation weiterentwickelt und einige der anzusprechenden Verbesserung umgesetzt werden können und werden.

²⁷³ Insgesamt dauerte der fußläufig Testdurchlauf rund zweieinhalb Stunden. Hierbei wurde die Route mehrmals (mit und ohne Einblendung von Zustellhinweisen; mittels verschiedenen Stylings, ...) abgegangen, um auch Fehler, die nicht direkt beim ersten Durchgang auftraten, gegebenenfalls aufzulösen und weitere Fehler, die zum wiederholten Male auftraten, besser verstehen zu können. Darüber hinaus wurden während der Testung dann auch noch Fotos der Applikation für die Dokumentation dieser Arbeit erstellt. Der Akkuverbrauch beim Smartphone belief sich dabei auf rund 50%, wobei zu sagen ist, dass beim eingesetzten Smartphone Probleme mit dem Akku (ohne Nutzung ca. 20h von 100% auf 0%) bekannt sind.

²⁷⁴ Hiermit ist das Einschalten des Smartphones mit der Navigationsapplikation, das Verbindend der AR-Brille mit dem Smartphone und die Anpassung des AR-Brillen Modus gemeint.

Das erste kleinere Problem bezog sich auf die zumindest in den GISc zur Genüge bekannten Probleme der Positionierungsgenauigkeit und -geschwindigkeit von GNSS (plus IMU). Die hierbei gemachten Beobachtungen und sich daraus ergebenden Probleme waren dabei die folgenden: Zum einen konnte die korrekte Orientierung der Route wie auch der Gehrichtung²⁷⁵ erst nach einigen Meter der Bewegung festgestellt werden und zum anderen war die Aktualisierung des Standorts in geringem Maße sprunghaft – Für eine genauere Besprechung weshalb sei an dieser Stelle auf das Kapitel 6.1.3 verwiesen. Beide Punkte stellten aber bei der fußläufigen Bewegung kein wirkliches Problem dar, da sich zur Lösung des ersten Punkts einfach vor dem Start der Navigation etwas bewegt werden musste und zweiter nicht so tragisch war, als dass er wirklich Probleme bei der Navigation ausgelöst hätte. Ein weiteres Problem, welches aber an dieser Stelle nicht verwundert, war dann noch, dass es zu Aussetzern der Aktualisierung der Route kam, sobald in einem Bereich ohne Abdeckung von GNSS-Empfang, etwa bei Unterführungen, navigiert werden musste, wobei dies, wie in den GISc zur Genüge bekannt ist, mit dem Funktionsprinzip der Datenübermittlung von GNSS und Abschattungsbereichen und nicht mit der Applikation per se zu tun hat.

Ein weiterer sich aus der Ungenauigkeit der Positionierung ergebender Punkt ist folgender: Durch die Ungenauigkeit kommt es dazu, dass bei der Anzeige von Zustellhinweisen ein relativ großer Radius (im Versuchsaufbau 5m) gesetzt werden musste, um Zustellhinweise korrekt anzeigen zu können. Hierbei kam es dann einmal zu dem Fall, dass zwei Hinweise von gegenüberliegende Zustelladressen gleichzeitig angezeigt wurden, sich überlappten und somit unlesbar waren.

Das zweite auftretende Problem hatte ebenso in gewisser Weise mit dem ersten, also der Positionsungenauigkeit der Verortung zu tun, kann aber zukünftig durch Umbau der Applikation in Bezug auf die Funktionsweise der Aus-/ Einblendung von Navigationshinweisen gelöst werden können²⁷⁶. Konkret geht es dabei darum, dass in manchen Fällen, etwa aufgrund von Wegüberschneidungen oder der Positionsungenauigkeit und dem daraus folgenden nicht korrekt funktionierendem Ausblenden von Hinweistafeln, mehrere Hinweistafeln auf einmal eingeblendet wurden (siehe Abb.53).

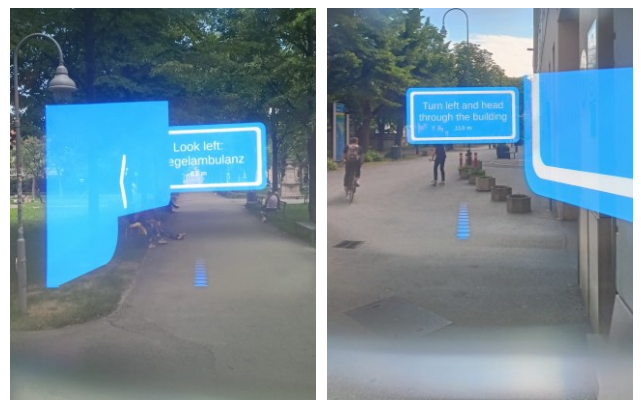


Abbildung 53: Inkorrekte Anzeige multipler Hinweistafeln

Abseits davon machte die Applikation bei fußläufiger Fortbewegung eine gute Figur, wobei einige Beobachtungen hinsichtlich der Gestaltung der für die Navigation bereitgestellten

²⁷⁵ Diese geht auf einer Berechnung des aktuellen Standorts und des Standorts vor x-Sekunden zurück.

²⁷⁶ Etwa mit einer if-Abfrage, wobei definiert wird, dass immer nur eine Einblendung aktiv sein darf und es sich hierbei um die nächstgelegene handeln soll.

Hinweise gemacht wurden, welche aber erst im Kapitel zu möglichen Verbesserungen angesprochen werden sollten.

6.1.2 Fahrzeugbasierte Testung:

Die Testung der Applikation bei fahrzeugbasierter Fortbewegung sollte basierend auf dem Wissen rundum die bereits bei der fußläufigen Testung gemachten Beobachtungen und dem Wissen rundum die potenzielle Problematik rundum die Genauigkeit der Verortung mittels GNSS und der Aktualisierungsrate in zwei Schritten durchgeführt werden. Als erstes sollte eine Testung bei relativ langsamer Geschwindigkeit (bei bis ca. 50km/h) und in einem zweiten Schritt auch bei höheren Geschwindigkeiten gemacht werden. Bevor die eigentliche erste Testfahrt begonnen werden konnte, musste, wie bei der fußläufigen Nutzung auch, die Betriebsbereitschaft hergestellt werden. Anschließend an die Herstellung der Betriebsbereitschaft wurde dann mit der Fahrt begonnen, wobei hier, unter anderem aufgrund der Beschaffenheit der Strecke zu Beginn sehr langsam (ca. 20km/h) gefahren wurde. Bereits zu Beginn traten aber bereits erste Probleme auf, wobei es sich um die Art der Darstellung der auf der Fahrbahn/ dem Weg angezeigten Route handelte. Wie erwähnt, kann die Route immer nur als gerade Linien, welche die einzelnen die Route aufbauenden Punkten verbindet, und nicht als Kreisbogen dargestellt werden. Dadurch kommt es, wenn durch eine Kurve gefahren wird, zu Einblendungen, welche mehr Verwirrung stiften, als sie nützen. Hinzu kommt dann in diesem Aspekt noch jener Punkt, dass die angezeigte Route nicht direkt auf die Straße gelegt werden kann und sie somit, wenn bergab oder bergauf gefahren wird, es so wirkt, als wenn die angezeigte Route von der Straße abhebt beziehungsweise in die Straße hineinführt.

Ein schwerwiegendes Problem trat weiters relativ rasch auf, so bald etwas schneller als ca. 20km/h gefahren wurde. Hierbei kam es zuerst zur sprunghaften Veränderung der Position und schlussendlich zu Verbindungsabbrüchen nach welchen einige Zeit (ca. 30sek) gewartet werden musste, bis die korrekte Position wieder ermittelt werden konnte. Außerdem musste beobachtet werden, dass die Anzeige von Hinweistafeln für die Navigation im Blickfeld erst zu spät geschieht und dann keinen Nutzen mehr hat. Als diese beiden Probleme auftraten und sich auch nach mehreren versuchten Neustarts am Verhalten nichts änderte, wurde beschlossen, den Versuchsaufbau mit Geschwindigkeiten unter 50 km/h abubrechen und jenen mit über 50 km/h gar nicht zu beginnen.

Insgesamt handelt es sich also um die folgenden vier Probleme, welche bei der durchwegs nur kurz währenden Testung auftraten, wovon die beiden letzteren die Applikation abseits fußläufiger Bewegung gänzlich unbrauchbar machen:

1. Anzeige von Navigationslinien nicht als Kreisbögen möglich
2. Anzeige der Navigationslinien nicht auf der Fahrbahn/ Weg fixiert
3. Positionierung funktioniert ab 20km/h nicht mehr (zuverlässig)
4. Anzeige von Hinweistafeln für die Navigation zu spät/ ungenau

Hinsichtlich möglicher Lösungen der dargelegten Problematiken – ausgenommen jener der Anzeige von Kreisbögen und der Fixierung auf der Fahrbahn – gibt es an dieser Stelle zu sagen, dass dies nicht ohne weiteres, also ohne eine weiterführende Analyse, woher die Probleme stammen, möglich ist. So können an dieser Stelle etwa eine Summe an Faktoren, welche von der reinen Performanz der Applikation, die Berechnungsgeschwindigkeit der Werte innerhalb der App, über die Aktualisierungsrate der GNSS-Position bis hin zur generellen Positionierungsgenauigkeit von GNSS-Systemen reichen, mögliche Gründe für die Probleme sein. Insgesamt kann es nach erster Analyse, ohne den Code zur Verortung tiefgreifender analysiert zu haben, demnach folgende Gründe geben, weshalb die Applikation solch große Probleme bei der fahrzeugbasierten Nutzung macht:

1. Problematik im Code (bzw. Berechnung inperformant)
 - a. ...weshalb Werte nicht schnell genug abgerufen und berechnet werden und somit benötigte Informationen nicht schnell genug bereitstehen, um Inhalte anzeigen zu können
2. Systembedingte Limitation(en)
 - a. Aktualisierungsrate/ Positionierungsgenauigkeit von GNSS
 - b. Rechenleistung zu gering

6.1.3 Analyse der Verortungs-Problematik

Bei weiterführender Analyse der Verortungsproblematik wurde an erster Stelle einmal an die im Code für die Verortung zuständigen Abschnitte geblickt, wobei hierfür im Rahmen des Projekts mehrere Bestandteile unabhängig voneinander zuständig sind. So ist es etwa so, dass die Verortung der Einblendungen von Zustellhinweise auf selbst entwickeltem Code zur Verortung und der Berechnung des Radius, in welchem diese angezeigt werden, basiert, während der Code rundum die Positionierung der Navigationslinie, der Hinweistafeln und der Abfrage der aktuellen Position auf Code von Daniel Fortes ARGPS Erweiterung in Kombination mit dem Standardcode der ARFoundation-Erweiterung zur Fortschreibung der Position basiert. In diesem Fall sind in erster Linie demnach die folgenden Objekte/ Komponenten des Projekts genauer zu betrachten:

- ARPoseDriver (ARFoundation)
- ARLocationRoot Objekt mit folgenden Komponenten (ARGPS):
 - ARLocationOrientation
 - ARLocationManager
 - ARLocationProvider

Generell ist es weiters so, dass die Position der im virtuellen Raum platzierten Objekte, also der Route beziehungsweise im Exakten deren Knotenpunkte, aufbauend auf welchen dann die Platzierung der Hinweistafeln und Anzeige der Navigationslinie geschieht, nur einmal direkt zum Start der Applikation (bis auf die Ausnahme eines kompletten Kontaktverlustes der GNSS-Verbindung) durchgeführt und dann weiters nur mehr die Position des

Smartphones im Raum aktualisiert wird, wodurch manche Probleme – hierzu später mehr – verständlicher werden. Ein Überdenken dieses Vorgehens wird demnach in weiterer Folge, wie auch weitere Punkte der Überarbeitung der Verortung, in Zukunft unumgänglich sein, wenn die Applikation möglichst zuverlässig und gut nutzbar werden soll.

Da bei der Betrachtung der ARPoseDriver Komponente und bei Suchanfragen bezüglich von Problemen mit dieser relativ rasch klar wurde, dass diese aller Wahrscheinlichkeit nach, ebenso wie die zwischenzeitlich als mögliches Problem ausgemachte Verbesserung der Standortgenauigkeit, durch eine Internetverbindung ausgeschlossen werden konnte, wurde auf die diversen Komponenten des ARLocationRoot Objekts genauer eingegangen. Hierbei ist es so, dass sich die drei Komponenten der ARLocationOrientation, der ARLocationManager und der ARLocationProvider um die Positionierung und Orientierung des AR-Systems (Smartphone – HMD System) im virtuellen Raum basierend auf (GNSS-) Koordinaten aus dem echten Raum kümmern und die folgenden Kernaspekte bedienen:

- ARLocationManager → verwaltet die Verknüpfung der AR-Session und der ARGPS Komponenten (d.h. ohne diese funktioniert keine der ARGPS Komponenten)
- ARLocationOrientation → Orientierung im Raum (d.h. Blickrichtung bei Bewegung)
- ARLocationProvider → Positionierung im Raum (d.h. Position bei Bewegung)

Dieser Aufgabenverteilung folgend muss das Problem mit der Verortung und den Problemen der Anzeige in einer der beiden letzteren Komponenten liegen. Obwohl an dieser Stelle die Schuldigen für die genannten Problematiken im Endeffekt relativ rasch ausfindig gemacht

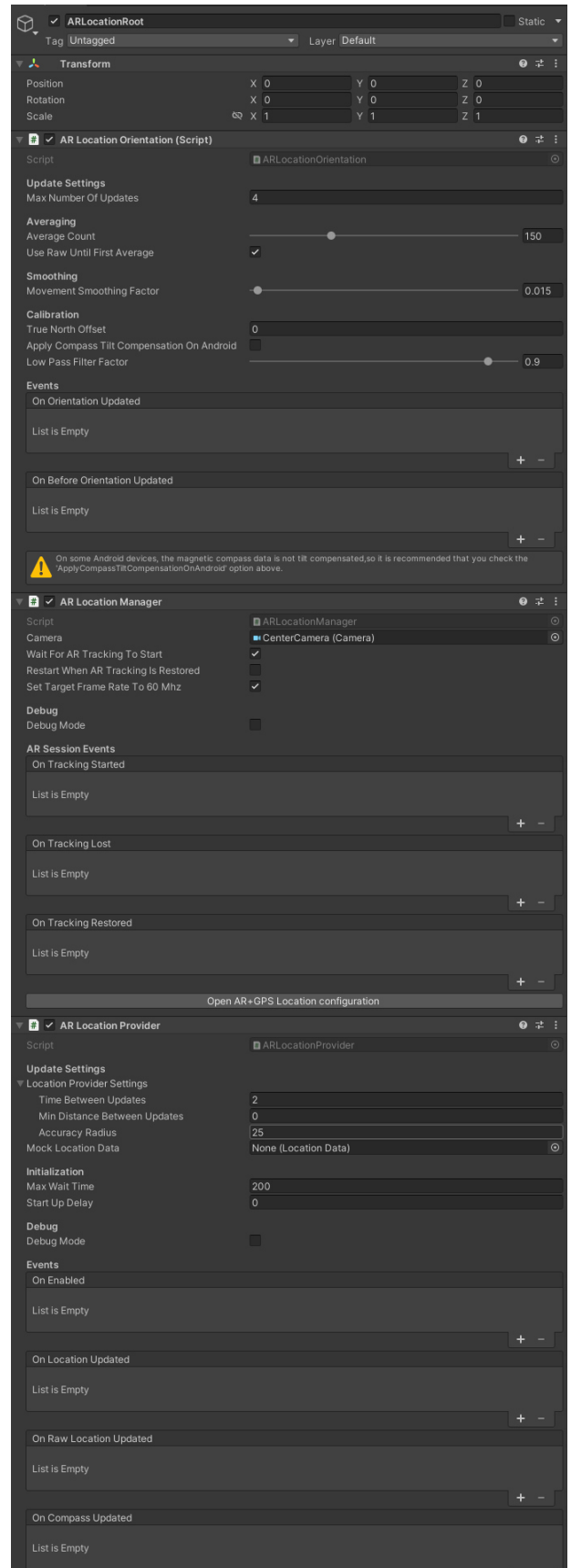


Abbildung 54: ARLocationRoot Objekt mit dessen Komponenten und den hier einstellbaren Parametern

werden konnten, war es im Rahmen der Arbeit nicht (mehr) möglich eine gangbare Lösung im Hinblick auf die Nutz­­barmachung der Applikation bei fahrzeugbasierter (und Verbesserung der Verortung bei fußläufiger) Navigation zu machen. Der Grund hierfür liegt im Aufbau des Codes und der daraus resultierenden Instabilität der angezeigten Inhalte bei Erhöhung des Updateintervalls der Position. Erhöht man etwa die Rate/ Verkleinert den Zeitintervall, in welcher Positionsupdates durchgeführt werden sollen in der ARLocationProvider Komponente, so erhöht sich zwar logischerweise die Aktualisierungsrate der Position, aber es kommt dafür auch öfter zu einem Herumspringen der Position. In anderen Worten: Wenn man auf ein Ziel zugeht, kann es sein, dass dieses bei höherer Abtastrate einmal in einer Entfernung von 150m, im nächsten Moment von 155 und im übernächsten von 145m liegt. Ein weitaus größeres Problem ist dabei aber nicht die Entfernung zum angezeigten Ort, also die Verschiebung der Positionsgenauigkeit in Richtung der Wegstrecke, sondern der Versatz quer zu dieser. An dieser Stelle spielt dann auch die zweite Komponente, die ARLocationOrientation hinein, welche basierend auf einer Mittelung einer Vielzahl an Messwerten basierend auf magnetisch Nord die Bewegungsrichtung und somit auch die Blickrichtung und Anzeige von Inhalten (mit-) bestimmt. Auch hier ist es, wie beim ARLocationProvider so, dass die Parameter, etwa die Anzahl der maximalen Updates der Orientierung oder aber die Anzahl für die Mittelung der Orientierung herangezogenen Messwerte angepasst werden können. Dennoch sinkt auch hier bei einer Verringerung der Anzahl an für die Mittelwertbildung herangezogenen Messwerte die Genauigkeit und bei einer Erhöhung der Anzahl an Updates der Orientierung wird die Anzeige von Inhalten instabiler, ändert sich also öfter sprunghaft.

Warum die verschiedenen Einstellungen bei den genannten Parametern zu Problemen führen, sei an dieser Stelle nochmals durch ein Beispiel für einfachere Verständlichkeit (siehe hierzu auch Abb. 55 bis Abb. 57) erläutert: Man navigiert sich mittig entlang einer Straße zur nächsten Kreuzung, welche 100m entfernt liegt. Die Updaterate ist auf kurze Intervalle (bspw. 0,x Sekunden) gestellt. Dadurch kommt es dazu, dass die eigene Position immer im Intervall von 0,x Sekunden geupdated und angepasst wird²⁷⁷. Bei der Verortung der aktuellen Position kann es dabei, wie bei GNSS-Verortung aufgrund deren Ungenauigkeit nicht unüblich sein, dass die Positionsgenauigkeit je nach Umgebung variiert. So wird man etwa einmal mittig auf der Straße, einmal auf dem linken und einmal auf dem

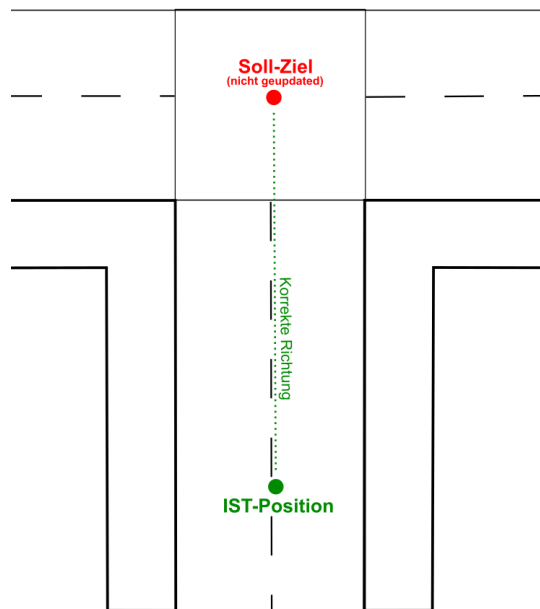


Abbildung 55: Soll-Situation der Navigation bei korrekter Berechnung und Positionierung

²⁷⁷ Die Werte, bei welcher die fußläufige Bewegung relativ gut funktionierte, waren jene, die in Abbildung 54 eingetragen sind.

rechten Gehsteig (gleiches gilt für weiter vorne oder weiter hinten auf der Straße) verortet, obwohl man sich immer noch mittig auf der Straße an derselben Position befindet²⁷⁸. Dadurch kommt es dann dazu, dass immer auch die angezeigten Inhalte basierend auf der neuen Position des Betrachters angepasst werden müssen²⁷⁹. Deshalb ist es so, dass, wenn man mittig auf der Straße verortet wird und die Position in zu kurzen Abständen bestimmt wird, die angezeigten Inhalte zum hin und herspringen bzw. zum Ändern deren Rotation²⁸⁰ (bzgl. der Position auf der AR-Brille) beginnen. Gleiches gilt bei der Erhöhung des Updateintervalls der Orientierung, wobei hier das Problem nur das von links nach rechts beziehungsweise vice versa Verschieben und das Ändern der Rotation der angezeigten Inhalte anbelangt.

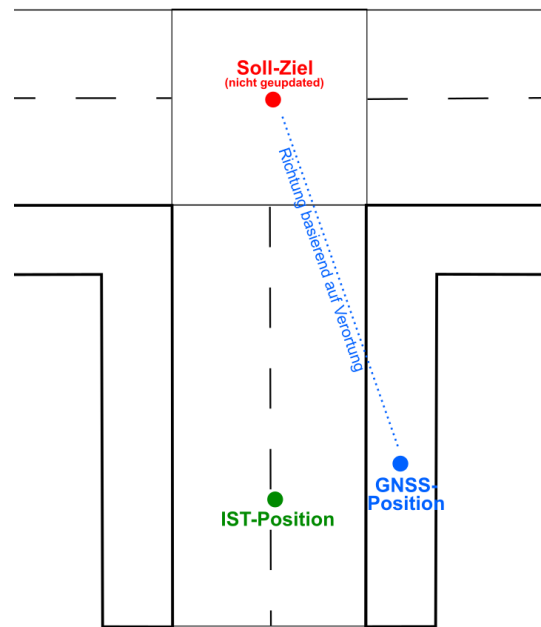


Abbildung 56: Situation bei falscher Verortung

Geht man von dem mittig auf der Straße befindlichen Standort und einer initial korrekten Verortung mittig auf der Straße einerseits, und einer anschließend abweichenden Messung auf dem rechten Gehsteig andererseits aus, so gestaltet es sich bezüglich der Orientierung so, dass korrekterweise (bei geradem Straßenverlauf und korrekter Routeneinzeichnung) die auf der AR-Brille angezeigte Route bei der initialen korrekten Verortung geradeaus führen muss (Abb. 55). Gibt es nun aber Probleme bei der Positionierung und der Orientierung, wird man also etwa auf dem rechten Gehsteig verortet, kommt es dazu, dass die Einblendung verdreht auf der AR-Brille, also schräg nach links verläuft, obwohl man immer noch mittig auf der Straße steht (Abb. 56, 57).

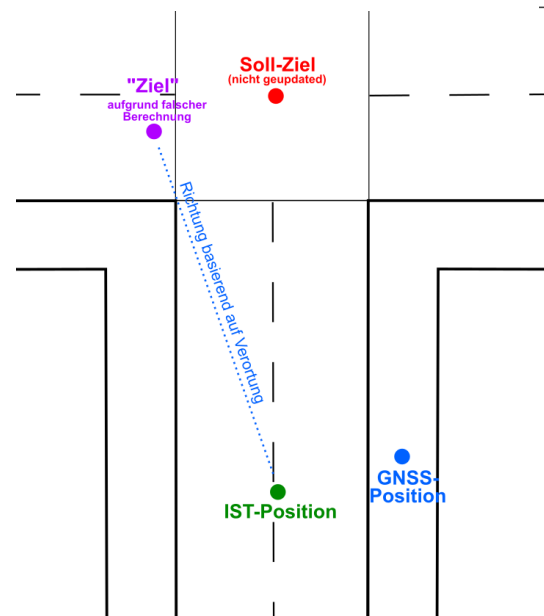


Abbildung 57: Resultierende Situation aus falscher Verortung

²⁷⁸ Vielfach werden diese Probleme abgefangen, indem die neue Position basierend auf Berechnung der letzten und des aktuellen Messwertes bzw. generell mehrerer Messwerte zustande kommt und nicht wie im Fall der Applikation rein aufgrund der aktuellsten Messung.

²⁷⁹ Hierbei ist wichtig zu ergänzen, dass nur die Navigationslinie und die Anzeigerichtung der Navigationshinweise angepasst wird, nicht aber die Lage der Kreuzungspunkte im AR-Raum, an welchen die Informationen verortet sind.

²⁸⁰ Von links nach rechts, aber auch von vorne nach hinten. Genauso ändert sich auch deren Ausrichtung.

Insgesamt lässt sich also sagen: Eine höhere Updaterate der Positionierung wird zumindest momentan und ohne eine Vielzahl an Iterationen der Testung hinsichtlich passender Werte der Aktualisierung für verschiedene Fortbewegungsgeschwindigkeiten teuer mit Problemen bei der Anzeige von Inhalten erkaufte, beziehungsweise macht die Applikation unnutzbar.

Es wird an dieser Stelle nicht gesagt, dass eine Lösung der im Rahmen der Testung gemachten Problematiken nicht möglich ist, es wird sogar so weit gegangen – gerade wenn etwa noch Pattern Recognition zusätzlich Anwendung fände – zu sagen, dass sich aller Wahrscheinlichkeit nach Lösungen hierfür fänden, der Zeitaufwand hierfür jedoch beträchtlich und in keinem für eine Masterarbeit sinnvollen Rahmen möglich sein wird. Der Grund hierfür liegt darin, dass zum einen wie erwähnt entweder eine Vielzahl an Testiterationen mit verschiedenen Werten gemacht werden müsste, oder gleich zur Gänze neuer Code für die Verortung geschrieben werden müsste. Bei einer Besprechung dieser Thematik mit Kollegen wurde einmal der hierzu als ganz passend empfundene Ausspruch geäußert: „Jetzt sind die Gründe, weshalb Google für deren LiveView Umsetzung zwingend eine gute Abdeckung mittels StreetView voraussetzt auch logisch. Diese wird benötigt werden, um die Probleme reiner GNSS-Verortung ausgleichen zu können.“ An dieser Stelle bleibt zusätzlich zu dieser Aussage in Kombination mit dem Wissen, dass wenn die diversen bei der eigenen Ausarbeitung gemachten Beobachtungen hinsichtlich der diversen Probleme einfach zu umgehen wären, dies sicherlich von diversen Unternehmer*innen oder Forscher*innen, die in diesem Feld aktiv sind, bereits (wie auch die Entwicklung eines auf HMDs angezeigten AR-Navigationssystems im Generellen) gemacht worden wäre.

6.2 Mögliche Verbesserungen, Ideen für zukünftige Adaptionen und unlösbare Probleme

Bevor zu guter Letzt ein Fazit hinsichtlich der Applikation gezogen wird und im nächsten Kapitel eine Conclusio der gesamten Arbeit und der erstellten Applikation folgen, sollen an dieser Stelle nun noch einige mögliche Verbesserungsvorschläge und zukünftige Ideen für die erstellte Applikation angesprochen werden²⁸¹.

Insgesamt lassen sich, wie die vorangegangenen Kapitel zeigten, zwei Problemfelder bei der entwickelten Applikation ausmachen: Die Visualisierung von angezeigten Inhalten (Kreislängen, Überlappungen, Linienstile, ...) und der Positionierung und den daraus entstehenden Problemen (das Springen der Inhalte, inkorrekte Orientierung, ...). Diese sollten in weiterer Folge auch unabhängig voneinander angesprochen werden, da es sich beim ersten Punkt um relativ einfach lösbare Probleme handelt, während der zweite Punkt beinahe (zumindest mit dem im Rahmen der Arbeit verwendeten AR-System) unlösbare größere Probleme bündelt. Darüber hinaus soll aufgrund der Verortungsproblematik direkt ein weiterer Punkt an dieser Stelle vorausgeschickt werden: Die nachfolgende Besprechung der Verbesserungspotentiale und zukünftigen Ideen wird exklusive für die fußläufige

²⁸¹ Hierzu soll angemerkt werden, dass die Verbesserungen nicht mehr im Rahmen der Arbeit, sondern wenn, dann nur bei Weiterentwicklung der Applikation außerhalb der vorliegenden Arbeit, durchgeführt werden.

Fortbewegung stattfinden. Der Grund hierfür liegt darin, dass, um die Applikation für fahrzeugbasierte Navigation tauglich zu machen, aufbauend auf dem momentanen Wissen und wie zuvor angesprochen ein sehr hoher Zeitaufwand nötig ist und darüber hinaus nicht mit Sicherheit garantiert werden kann, ob die Bearbeitung dieses komplexen Themas überhaupt mit Erfolg gekrönt sein kann²⁸². Weiters wäre es eine Überlegung wert, noch einmal eine komplett neue Überlegung hinsichtlich geeigneter AR-Systeme, also insbesondere des HMDs angestellt werden. Ziel sollte dabei sein, um möglichst viele negative Einflussfaktoren ausschließen und aus dem Vollen schöpfen zu können, ein All-in-One System in einem HMD – Kamerasystem für Pattern Recognition; GNSS-Empfänger direkt in der Brille, ... – zu verwenden, wobei sich die Problematik der Verortung aber per se nicht ändern würde.

Technische Aspekte

Nach diesen einleitenden und einschränkenden Worten sollen nun aber die technischen Punkte (abseits der Verortungs-Problematik), welche zu einer Verbesserung der Nutzbarkeit bei fußläufiger Fortbewegung beitragen und tatsächlich ohne größere Probleme umgesetzt werden können, erwähnt werden:

1. Verbesserung der Anzeige von Hinweistafeln, sodass keine multiplen, sich überlappenden Einblendungen mehr geschehen
2. Verbesserung der Funktionalität der Applikation im Hinblick auf die Verwendung auf verschiedensprachigen Systemen, sodass etwaige Formatfehler²⁸³ abgefangen werden können
3. Schaffung einer besseren Einlese-Pipeline für Zustellhinweisdaten, da momentan jedes Mal, wenn die Routen und Zustellhinweise getauscht werden sollen, ein neuer *build* der Applikation nötig ist
4. Verbesserung der Stabilisierung von Einblendungen und der sprunghaften Aktualisierung von Inhalten, wobei dies teils auf technische Limitationen im Aufbau des AR-Systems, wie auch GNSS und IMU zurückzuführen ist²⁸⁴

²⁸² An dieser Stelle würde es auch nötig sein, sich nochmals genauer mit GNSS-Verortung auseinander zu setzen und gegebenenfalls neuen Code für die Berechnung der aktuellen Position, welcher die vorherigen Messergebnisse miteinbezieht, aufzusetzen. In anderen Worten kann es nötig sein, hinsichtlich der Verortung zur Gänze von den Komponenten von Daniel Fortes zu eigens entwickelten überzugehen.

²⁸³ Mit Formatfehler sind an dieser Stelle etwa die unterschiedlichen Verwendungsmöglichkeiten von Punkt und Komma bei der Trennung von Kommazahlen (relevant bei Koordinaten) oder aber auch die verschiedenen Formatierungsoptionen hinsichtlich der Trennzeichen bei CSV-Dateien gemeint.

²⁸⁴ Mit Limitationen sind an dieser Stelle die folgenden Punkte gemeint: Zum einen ist es, wie bereits erwähnt, so, dass es durch die Trennung von Berechnungseinheit und HMD der GNSS-Empfänger nicht in der HMD sitzt, die IMU jedoch schon, und es demnach zu Schwierigkeiten in der Abstimmung der Position und Blickrichtung und der Stabilisierung bei Bewegungen kommt. Weiters kommt an dieser Stelle dann noch der Punkt der Aktualisierungsrate des Standorts eines GNSS-Empfängers (ca. eine Aktualisierung pro Sekunde am Smartphone) am Smartphone, der generellen Positionsgenauigkeit (ca. 1-3m) und die Thematik rundum die Abschattung und demnach den fehlenden Empfang bei GNSS-Systemen zu tragen.

- a. → Dies ist nur zum Teil möglich. Es kann aber zumindest ein Abfedern, also dass sich die Einblendung flüssig von A nach B bewegt realisiert werden.
5. Verbesserung der Zeichnung der Navigationslinie im Hinblick auf die Möglichkeit der Anzeige von Kreisbögen anstelle von geraden Linien
6. Implementierung einer Sprachausgabe für Navigationshinweise

Visuelle Aspekte

Ebenso gibt es einige Punkte hinsichtlich der gestalterischen Umsetzung, welche an dieser Stelle angesprochen und verbessert werden sollten beziehungsweise um welche die Applikation erweitert werden sollte:

1. Anstelle der punktierten geraden Verbindungslinie, welche die Richtung zwischen zwei Kreuzungspunkten am Boden anzeigt, wäre es gut, auch andere Arten der Darstellungen, etwa einen in die korrekte Richtung weisenden Pfeil wie bei Sygic Real View als Alternative anzubieten.
2. Die an Kreuzungspunkten angezeigten Hinweistafeln sollten je nach Distanz deren Größe anpassen (desto näher, desto größer) und bei zu weiter Distanz ausgeblendet werden.
3. Direkt an Kreuzungspunkten sollte für eine besser Übersichtlichkeit, gerade bei mehreren kurz aufeinanderfolgenden Abbiegemöglichkeiten in städtischen Gebieten, ein Pfeil an der konkreten Abbiegeposition platziert werden, der angibt, in welche Richtung abgelenkt werden muss.
4. Die Einblendung der Zustellhinweise sollten in ihrer Größe wie auch Position verändert werden können.
5. Die Sprache der ist momentan noch durchwegs auf Englisch gehalten. In Zukunft sollte die Sprache auswählbar sein.

Weitere Aspekte:

Als letztes sei nun noch auf eine weitere potentielle Veränderung verwiesen, welcher sicherlich einen großen Aufwand bedürfen würde, aber hinsichtlich der Auswirkungen der sprunghaften Veränderung der angezeigten Inhalte (technische Verbesserungsmöglichkeiten, Punkt 4) und des Matchens der angezeigten Inhalte auf die Straße/ den Weg (hinsichtlich Höhe, als auch Lage) überaus interessant wäre:

1. Fixierung der angezeigten Navigationshinweise auf der Straße/ dem Weg mittels eines im Hintergrund liegenden Wegenetzes mit Höheninformationen und Fixierung der Route auf diese.

Unlösbare Problematiken:

Insgesamt sind oberhalb zwar nun einige Punkte angeführt, welche in weiterer Zukunft im Hinblick auf die Nutzung der Applikation im fußläufigen Umfeld behandelt werden müssen, wobei jedoch die allermeisten leicht lösbar sind. Ein weiterer zusätzlich zu den bereits in Kapitel 6.1.3 und Eingangs in diesem Unterkapitel angesprochenen unlösbaren Problemen bei der im Rahmen dieser Arbeit umgesetzten Navigationsapplikation kommt jedoch noch

hinzu: Jener der initialen Verortung des HMDs zum Smartphone bei Start der Applikation. Hier ist immer eine manuelle Korrektur nötig, da das Smartphone nie zu 100% gleich mit der AR-Brille ausgerichtet ist. Lösbar wäre dies nur, wenn das HMD, wie eben das Smartphone, seine Position im realen Raum tatsächlich unabhängig vom Smartphone erkennen würde. Dies würde jedoch voraussetzen, dass das HMD über einen GNSS-Empfänger verfügt.

6.3 Fazit zur Anwendung, Potentiale und Limits

Was bleibt nun abschließend zum bisherigen Stand der Applikation und deren Anwendbarkeit für die Zustellbranche zu sagen?

Zum Ersten einmal, dass die Applikation in absehbarer Zeit, mit absehbarem Aufwand und gegebenenfalls ohne komplett neuen Code zur Verortung, einer Neukonzeptionierung und anderen Geräten für fahrzeugbasierte Navigation ungeeignet ist. Dies liegt an der nun bereits zur Genüge angesprochenen Verortungsproblematik und am Kompromiss, welcher zwischen Positionsgenauigkeit, Aktualisierungsrate der Position und Stabilität der angezeigten Inhalte, gemacht werden muss. Für die fußläufige Fortbewegung mag hier möglicherweise ein guter Kompromiss mit dem bestehenden Code zur Verortung zu finden sein, aller Wahrscheinlichkeit nicht jedoch für die fahrzeugbasierte.

Grundsätzlich zeigte die Umsetzung und das Testen der Applikation der eigenen Meinung nach jedoch ein sehr großes Potential von AR-Navigationsapplikation (für die Zustellbranche, gerade im Hinblick auf die Routenfindung und das Erinnern an zuzustellende Sendungen) auf, jedoch ist es zeitgleich so, dass man auch vor einer nicht geringen Anzahl an Problemen steht, insbesondere wenn es um die Nutzung bei höheren Geschwindigkeiten und die Stabilisierung von Einblendungen geht. Konkret gemeint ist hiermit die Problematik der Positionsgenauigkeit, der Geschwindigkeit der Verortung und der daraus folgenden Probleme bei der Nutzbarkeit. Die anderen im Rahmen des Versuchsaufbaus gemachten und im vorangegangenen Abschnitt (technische & visuelle Aspekte) angesprochenen Punkte wiederum sind zwar als unerfreulich zu bezeichnen und die Korrekturen sollten jedenfalls vor einer breiteren Anwendung und Testung der fußläufigen Nutzung umgesetzt werden, stellen aber an sich keine größeren Probleme dar.

Insgesamt ist also an dieser Stelle zu sagen, dass es sich um eine durchwegs interessante und ein großes Potenzial aufzeigende Umsetzung handelt, diese jedoch einige Probleme aufweist und in einem ersten Schritt ohnehin nur für fußläufige Nutzung geeignet sein kann. Anschließend an die Verbesserungen der angesprochenen verbesserbaren Punkte und weiterem Testing noch optimalerer Werte der Positionierung bei fußläufiger Nutzung wiederum wäre es trotzdem sinnvoll, nochmals ein (größer angelegtes) User-Testing durchzuführen, um etwaige weitere Probleme zu erkennen und weiters auch Probleme der Ungenauigkeit der Verortung bei fußläufiger Bewegung noch besser abschätzen zu können. Ebenso gibt es dann auch noch einige andere, im Rahmen dieser Arbeit nicht angesprochene Problematiken, welcher weiterer Behandlung, wie etwa die Einsetzbarkeit eines AR-

Systems bei Schlechtwetter (Regen/ Schneefall/ Sturm)²⁸⁵ oder aber die rechtlichen Rahmenbedingungen einer Nutzung im Straßenverkehr bedürfen.

Eine weiterer in dieser konkreten Arbeit nur am Rande erwähnte, aber für Geoinformatiker*innen durchwegs bekannte Problematik tritt zusätzlich bei der Frage nach Möglichkeiten der Indoor-Nutzung (also der Indoor-Navigation) auf, da diese nur unzureichend mit GNSS realisiert werden kann²⁸⁶. Hier könnte perfekt von zukünftigen Arbeiten angeschlossen werden, um etwaige AR-Navigationsapplikationen für den Zustellbereich nicht nur für Außenräume, sondern auch für Innenräume (etwa Wohnblocks oder Firmenanlagen) anwendbar zu machen.

Abschließend bleibt zur Umsetzung somit zu sagen, dass AR-Navigationsapplikationen, wobei die Inhalte mittels HMDs angezeigt werden, definitiv ein großes Potential für die Zustellbranche und im Allgemeinen die Navigation aufweisen, jedoch auch noch einige Problemfelder im Raum stehen. Demnach sieht es bei der konkreten Ausarbeitung, als auch im Allgemeinen so aus, als ob in diesem Bereich einige Zeit vergehen würde bis von wirklich gut einsetzbaren und nutzbaren Applikationen gesprochen werden kann. Ein erster Schritt in diese Richtung wird im Anschluss an diese Arbeit sicherlich die Bearbeitung der bereits ausgemachten Verbesserungsmöglichkeiten sein, während im Allgemeinen eine stärkere Hinwendung von AR-Brillen als All-in-One-Geräte wünschenswert wäre. Hierdurch würde nämlich nicht nur die Entwicklung von Applikationen vereinfacht werden, sondern auch etwaige Problematiken wie das zueinander Verorten von Geräten – wie auch GNSS-Empfänger (Smartphone) und IMU (HMD) zueinander – und die unter anderem auch in geringem Maße auf diesen Grund zurückführbaren Probleme der Stabilisierung der Veränderung der Position von Smartphone und HMD zueinander zumindest in einem gewissen Maße abgefangen werden können. Weiters wäre es natürlich, wie bei der XREAL Air2 Ultra bereits gemacht, von Vorteil, auf HMDs mit Kamerasystemen zu setzen, da über hierdurch möglich werdendes Pattern-Recognition Einblendungen noch besser im Raum stabilisiert werden könnten. Generell zeigt sich insgesamt ein großes Potential für weitere Forschungen im Bereich der AR-Navigationsapplikationen auf AR-Systemen, welcher sich HMDs bedienen, auf. Es wird zu hoffen sein, dass diesem Bereich zukünftig mehr Aufmerksamkeit von Forschenden, vor allem auch innerhalb der GISc, zukommt und nicht bloß auf Umsetzungen aus der Wirtschaft gewartet wird und diese von Wissenschaftler*innen im besten Fall in wissenschaftlichen Arbeiten Betrachtung finden. Viel eher sollte die Forschung selbst alles daransetzen, Probleme zu definieren, mögliche Lösungen zu finden und diese umzusetzen.

²⁸⁵ Leider finden sich für die näher betrachteten HMDs keine Informationen zur Wasserdichte, den Temperaturen und/ oder Luftfeuchte, in welchen sie eingesetzt werden können, wie es bei Mobiltelefonen der Fall ist. Dadurch können leider auch keine Aussagen zur theoretischen Eignung der Nutzung der Geräte bei Regen, Schneefall oder humiden Klimaten gemacht werden.

²⁸⁶ Dieser Punkt wurde wissentlich im Rahmen der Arbeit zur Gänze außen vorgelassen, da die Fokussierung zur Bearbeitung dieser Thematik eine andere hätte sein müssen und sich zu diesem Thema außerdem gerade in rezentere Zeit eine Vielzahl an Arbeiten finden lassen.

7 Conclusio

Was bleibt nun abschließend zur gesamten hier vorliegenden Arbeit zu sagen? Zum einen, dass der Aufwand, welcher von einem/ einer Geoinformatiker*in betrieben werden muss, um ein XR-Projekt umzusetzen, immens und innerhalb einer Masterarbeit kaum bewältigbar ist, sich zum anderen aber eine riesige Lernkurve in einem Bereich mit einem immensen Potential ergibt. Bevor aber nun geschlossen wird, sollte noch auf einige Punkte genauer eingegangen werden.

Zum einen hat die Arbeit gezeigt, dass es sich beim XR-Bereich um einen sich rasch entwickelnden, aber auch chaotischen Bereich handelt, in dem es oft schwierig ist, sich zurecht zu finden und klare Einteilungsmöglichkeiten innerhalb der einzelnen XR ausmachenden Typen und Klassen zu finden. So problematisch diese Diffusität innerhalb dieses Felds auch manches Mal ist, eröffnet diese generelle Offenheit des XR-Bereichs im Allgemeinen, wie bei den GISc auch, die Möglichkeit, von verschiedenen Seiten her beeinflusst zu werden und diese wiederum in Form einer Wechselwirkung zu beeinflussen. Hierdurch ergeben sich beinahe grenzenlose Anwendungsfelder und Entwicklungsmöglichkeiten, wobei der Kernaspekt des XR-Bereichs aber immer auf der Kommunikation, also der Übermittlung von Inhalten liegt. Eben genau bei diesem Aspekt, der Kommunikation von Inhalten, kommt die Stärke von XR, welche mithilfe von Realtime-Engines, die mittlerweile als Standardwerkzeug zur Visualisierung (dreidimensionaler) interaktiver Echtzeitinhalte in einer Vielzahl an Bereiche betrachtet werden können, zum Tragen. Hierzu muss aus Sicht der GISc jedenfalls erwähnt werden, dass die Betrachtung von Realtime-Engines als Standardwerkzeug für den eigenen Bereich zwar leider noch nicht zutrifft, aber eine starke Hinwendung in letzter Zeit zu beobachten ist – begründet in den Potentialen, die Realtime-Engines bezogen auf XR-Umsetzungen oder einfach nur auf interaktive (3D-)Visualisierungen, bieten. Hinsichtlich XR und Realtime-Engines lässt sich also abschließend sagen, dass es sich um sich schnell entwickelnde Bereiche, welche ein großes Potential – und Lernpotential für die GISc wie man Inhalte am besten darstellt – aufweisen, handelt, auch wenn zugegebenermaßen die Einarbeitung von Personen außerhalb dieser Bereiche einiges an Aufwand verlangt. Für technikversierte Personen lohnt sich dies aber jedenfalls aufgrund der Vielzahl an Möglichkeiten, die diese bieten, und der Vielzahl an Bereichen, in denen diese eingesetzt werden.

Weiters hat die vorliegende Arbeit in Bezug auf XR und die Inhaltsvermittlung mithilfe selbiger, im speziellen AR, gezeigt, dass (gänzlich) neue Möglichkeiten durch die Anzeige von Inhalten eines virtuellen, den realen Raum überlagernden Raums entstehen. Weiters bieten sich im Bereich von XR und der Inhaltsvermittlung eine Vielzahl an Geräte hierfür an, wobei HMDs sicherlich die interessantesten darstellen, wenn auch momentan Smartphones noch das Gerät der Wahl für die meisten Umsetzungen darstellen. Der Grund hierfür führt auch gleich zu den Problematiken der Erstellung (komplexerer) XR-Applikationen in Realtime-Engines: Der Verfügbarkeit und dem Preis von HMD-Systemen, wie auch den Schwierigkeiten der Erstellung von Applikationen (v.a. für bereichsfremde Personen). Insgesamt lässt sich der Punkt der Inhaltsvermittlung mittels XR demnach so zusammenfassen, als dass hierdurch gänzlich neue Möglichkeiten, insbesondere was die

Immersion bei Verwendung von HMDs anbelangt, eröffnet werden, wobei Smartphones aus verständlichen Gründen noch immer zumeist die Geräte der Wahl sind, auch wenn diese in einer Vielzahl an Punkten Limitationen aufweisen.

Zu guter Letzt sei nun noch auf die eigens ausgearbeitete AR-Applikation für Zustellbereiche eingegangen. Die Ausarbeitung, welche leider in ihrer derzeitigen Umsetzung nur in Teilen erfolgreich war, zeigte dabei perfekt die bereits zuvor erwähnten Potentiale hinsichtlich der Vermittlung von Wissen (Navigationshinweise, Zustellhinweise) und die Neuartigkeit eines solchen Umsetzungszugangs auf. Andererseits wurden aber auch die Problematiken, vor allem, aber nicht exklusive, hinsichtlich eines Bereichs aufgezeigt, bei welchem Realtime-Engines und XR nicht zum Standardrepertoire gehören (obwohl sie dazu gehören sollten). Die Problematiken sind hierbei vor allem der Punkt des immensen Aufwands der Einarbeitung und die Probleme in der Umsetzung, teils dem jeweiligen Bereich sogar eigentlich bekannter Thematiken (wie der Geoinformation bspw. GNSS-Verortung), aufgrund von fehlendem Wissen im Umgang mit Realtime-Engines. Wendet man sich nun aber wieder der konkreten Applikation zu, so zeigte diese bei fußläufiger Bewegung die Potentiale, aber auch erste Probleme, bereits auf, bis sie bei fahrzeugsbasierter komplett versagte. Zur konkreten Arbeit bleibt zu sagen, dass diese, wenn sie weiterentwickelt wird, definitiv eine interessante und gut funktionierende Anwendung für fußläufige Navigation im Zustellbereich mit hilfreichen Extras darstellen kann, für fahrzeugsbasierte Navigation aber noch eine große Anzahl an Problemen zu lösen wären. In genereller Form lässt sich zu AR-Navigationsapplikationen weiters sagen, dass wenn in Zukunft in diesem Bereich weiterentwickelt wird, jedenfalls interessante, gut funktionierende und in weitem Teilen anwendbare Applikationen zu sehen sein werden. Im Moment ist es aber aufgrund des Aufwands und der Probleme im Erstellungsprozess durchwegs verständlich, weshalb noch so wenige AR-Applikationen, welche aus dem Bereich der GISc kommen, existieren.

Wie lassen sich nun zum Ende diese Arbeit und die einzelnen Bereiche kurz zusammenfassen?

Generell sind XR (und speziell AR) und RE sehr interessante, ein hohes Potential für künftige Entwicklungen (für die GISc) aufweisende Bereiche, wobei aber momentan noch das Problem besteht, dass zu wenig Wissen über diese (und den Umgang mit den Tools) in den GISc existiert und andererseits die technischen Voraussetzungen im Hinblick auf Soft- und Hardware noch nicht optimal sind – immerhin letztes Problemfeld sollte aber innerhalb der nächsten Jahre gelöst werden können. Insgesamt ist eine Weiterarbeit in und mit XR, AR und RE, wie auch eine stärkere Hinwendung zu diesen, speziell auch aus Sicht der GISc wünschenswert.

8 Appendix

Kommentierter Code des MapHandler-Skripts

```

1  using ARLocation;
2  using ARLocation.MapboxRoutes;
3  using ARLocation.MapboxRoutes.Examples.Search;
4  using Mapbox.Map;
5  using Mapbox.Tokens;
6  using System.Collections;
7  using System.Collections.Generic;
8  using UnityEngine;
9  using UnityEngine.SceneManagement;
10 using static ARLocation.MapboxRoutes.Route;
11
12 namespace ARLocation.MapboxRoutes
13 {
14     public class MapHandler : MonoBehaviour
15     {
16         public Mapbox.Unity.Map.AbstractMap map;
17         public Camera posCamera;
18         public MenuScript menuScript;
19         public MapboxRoute MapboxRoute;
20         private GameObject minimapRouteGo;
21         private int MinimapLayer = 6;
22         private float BaseLineWidth = 2;
23         public Material MinimapLineMaterial;
24
25         // Update is called once per frame
26         void Update()
27         {
28             var location = ARLocationManager.Instance.GetLocationForWorldPosition(posCamera.transform.position);
29             map.SetCenterLatitudeLongitude(new Mapbox.Utills.Vector2d(location.Latitude, location.Longitude));
30             map.UpdateMap();
31         }
32
33         public void ZoomIn()
34         {
35             if (map.Zoom < 20)
36             {
37                 var newZoom = map.Zoom + 1;
38                 map.SetZoom(newZoom);
39                 map.UpdateMap();
40                 loadRoute();
41             }
42         }
43
44         public void ZoomOut()
45         {
46             if (map.Zoom > 5)
47             {
48                 var newZoom = map.Zoom - 1;
49                 map.SetZoom(newZoom);
50                 map.UpdateMap();
51                 loadRoute();
52             }
53         }
54     }
55 }

```

Import zu verwendender Bibliotheken

Definieren eines spezifischen Namensbereichs, welche dann in verschiedenen Skripten die gleiche Aufgabe inne haben

Definieren des Mutterskripts, dessen Hauptfunktionalität geerbt wird (MonoBehaviour --> Standard Skript für die Funktionalität innerhalb von Unity)

Definieren aller Variablen (public = von außen ansprech & im Editor bearbeitbar, private = nicht von außen ansprechbar) und Objekte

Standard Update Funktion, welche den gesamten in ihr befindlichen Code einmal pro Frame durchführt

Abfragen und speichern der aktuellen GNSS-Position des Smartphones;

Zuweisen von berechneten Parametern zum "map" Objekt und dessen Komponenten, wie auch aktualisieren selbiger

Code, welcher von einem Button auf der Nutzeroberfläche aufgerufen wird und die Zoomstufe der Map erhöht und die "loadRoute" Funktion aufruft

Gleiches wie oberhalb nur für das herauszoomen

```

59
60
61 ///<summary>
62 ///DANGER_ZONE
63 /// </summary>
64
65 ///<summary>
66 ///Function references to the steps in a route and passes them to the function which creates our route on the 2d map
67 /// </summary>
68 public void loadRoute()
69 {
70
71     var routeGeo = MapboxRoute.routeGeo;
72     buildMinimapRoute(routeGeo);
73 }
74
75
76
77 /// <summary>
78 /// Function which gets a set of geometries (constructing a route) passed and then creates the
79 /// layer overlayed on the 2d map (Function is a edited copy of Daniel Fortes Sample Project)
80 /// </summary>
81 /// <param name="geo"></param>
82 private void buildMinimapRoute(Route.Geometry geo)
83 {
84
85     var worldPositions = new List<Vector2>();
86
87     foreach (var p in geo.coordinates)
88     {
89         var pos = map.GeoToWorldPosition(new Mapbox.Utls.Vector2d(p.Latitude, p.Longitude), true);
90         worldPositions.Add(new Vector2(pos.x, pos.z));
91     }
92
93     if (minimapRouteGo != null)
94     {
95         minimapRouteGo.Destroy();
96     }
97
98     minimapRouteGo = new GameObject("MinimapRouteGameObject");
99     minimapRouteGo.layer = MinimapLayer;
100
101     var mesh = minimapRouteGo.AddComponent<MeshFilter>().mesh;
102
103     var lineWidth = BaseLineWidth * Mathf.Pow(2.0f, map.Zoom - 18);
104     LineBuilder.BuildLineMesh(worldPositions, mesh, lineWidth);
105
106     var meshRenderer = minimapRouteGo.AddComponent<MeshRenderer>();
107     meshRenderer.sharedMaterial = MinimapLineMaterial;
108 }
109
110 }
111
112

```

Kommentar

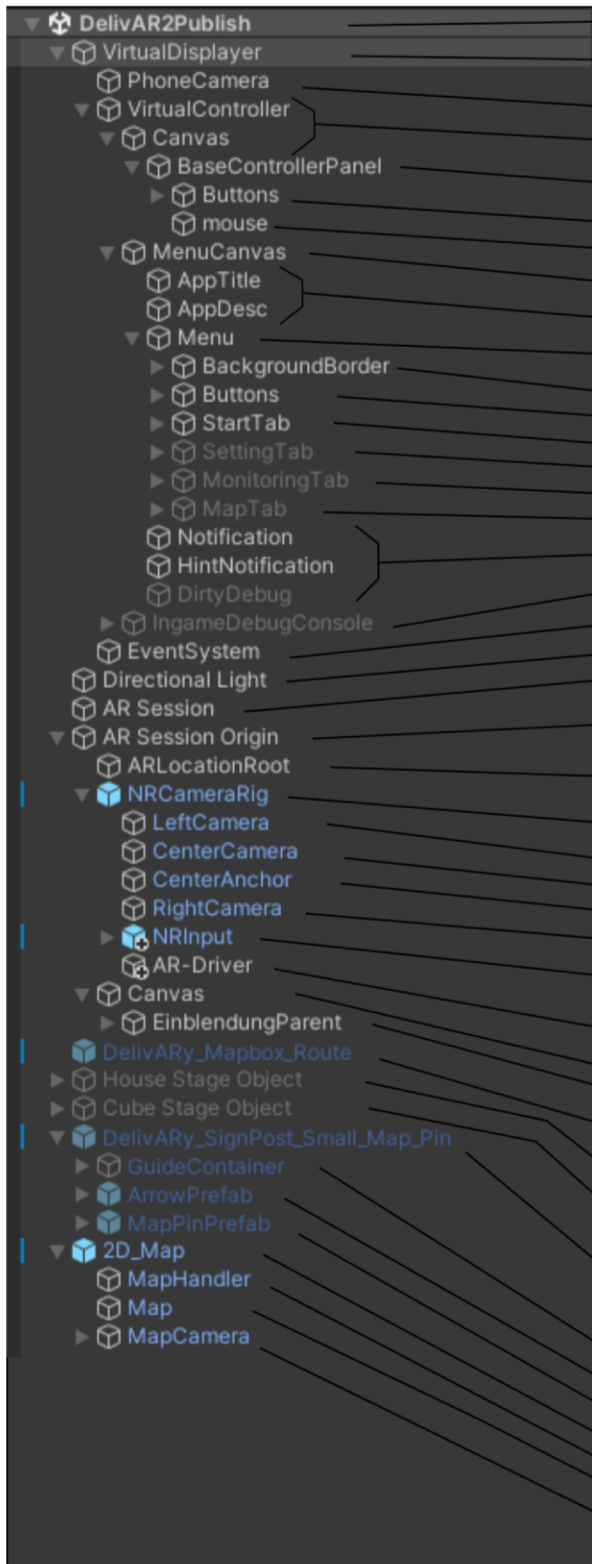
Ruft die aktuell navigierte Route vom "MapboxRoute" Objekt ab und die "buildMinimapRoute" Funktion auf, wobei dieser Routeninformationen übergeben werden

Kommentar

1. Anlegen einer lokalen Variable
2. Iterieren über die der Funktion übergebenen (Koordinaten-) Werte + Umrechnung Echtwelt-koordinaten in Unity-Koordinaten
3. Wenn bereits ein Routenobjekt existiert löschen desselbigen
4. Anlegen eines Objekts, welches die Route darstellt
5. Zuweisen des Objekts als Layer der Karte
6. Anlegen von Variablen und definieren des Aussehens der Route darüber

Kommentierter (Objekt-)Aufbau der DelivAR-Szene

Editor Komponenten



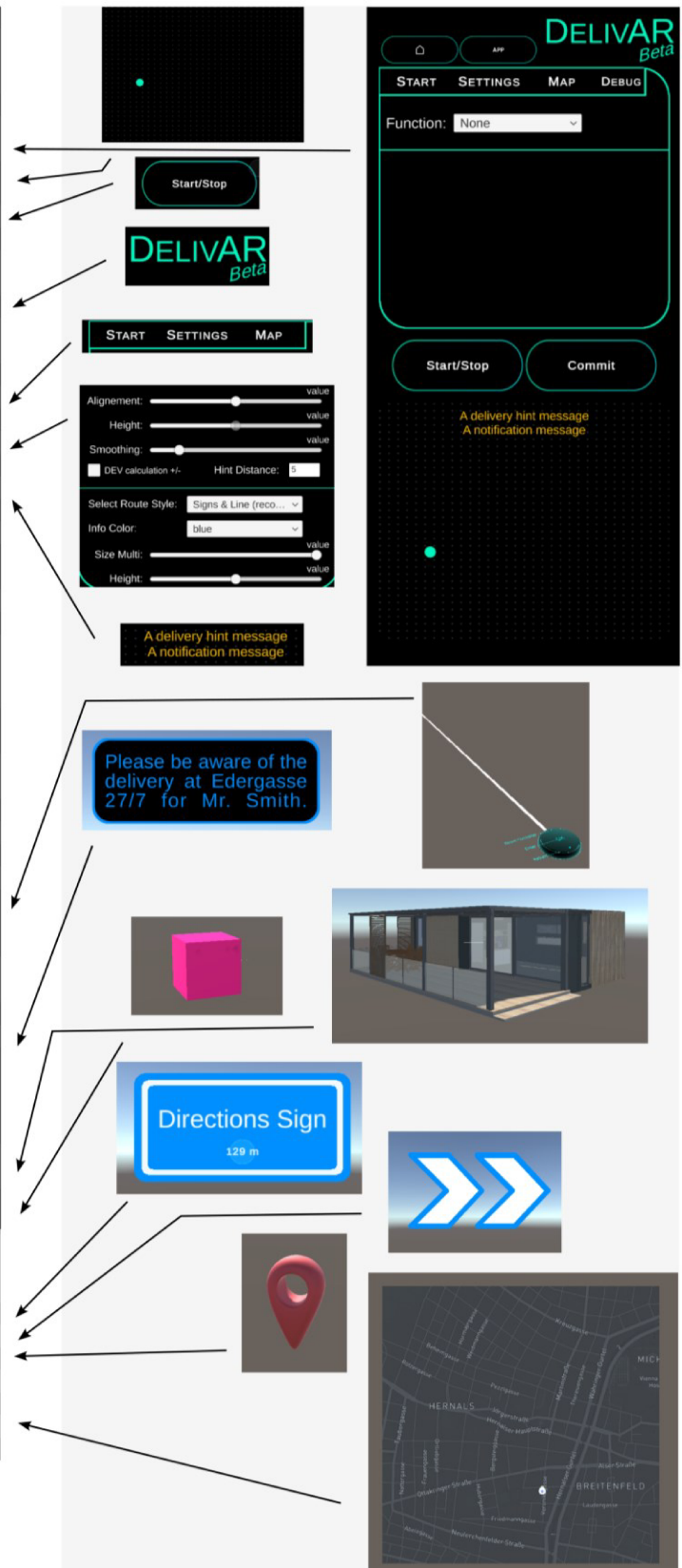
Objektbeschreibung

Aktuell aktive Szene
Gruppe für alle für die Funktionsweise des Smartphones als Bedienoberfläche bzw. Anzeigeoberfläche benötigten Objekte und Komponenten
Renderkamera für die Anzeige am Smartphone
Gruppe & Untergruppe für die Bedienoberfläche ausmachenden Komponenten & Beherbergung des ButtonController-Skript
Gruppe für Button-Objekte , das Trackpads & die Mausanzeige am Trackpad
Gruppe für allgemeine Buttons
Mauszeiger
Gruppe für alle die Bedienoberfläche ausmachenden Objekte (Tabs, Titel, Texte, Buttons, Dropdowns, ...)
Titel + Untertitel
Gruppe für die Tabs ausmachenden Komponenten & Beherbergung des Menü-, HintsHandler-, StyleSettings-Skripts
Hintergrund Objekt und dessen Umrandung
Tab-Buttons
Gruppe für die Objekte des Start Tabs
Gruppe für die Objekte des Settings Tabs
Gruppe für die Objekte des Debug Tabs
Gruppe für die Objekte des Map Tabs
Text-Objekt zur Anzeige von Informationen
Debug-Console für die Entwicklung
Objekt für's Monitoring von Interaktionen auf der Bedienoberfläche
Objekt zur Beleuchtung der gesamten Szene
Objekt zur Erstellung & Management einer AR-Szene
Objekt zur Verortung der AR-Szene im realen Raum, der Gruppierung in diesem befindlichen Objekte (teils erst befüllt während der Laufzeit) & der Beherbergung des PassPosition-, GetGPS-Skripts
AR-Szenen Verortungsobjekt + dazugehörige Skripte (ARLocationOrientation, ARLocationManager)
Gruppe für das XREAL-Kamerasystem zum Management des HMDs und der Skripte hierfür
Linke Kamera (f. linkes Brillenglas) des HMDs
theoretische mittige Kamera des HMDs
Objekt zur Speicherung des Mittelpunkt des HMDs
Rechte Kamera (f. rechtes Brillenglas) des HMDs
Objekt zur Verortung des Smartphones im AR-Raum, der Erstellung eines Strahls ausgehend von diesem & der Abfrage des Auftreffens des Strahls auf einem Objekt im AR-Raum
Objekt zur Abfrage der tatsächlichen Bewegung des Smartphones im Raum
Zeichenbereich für die Anzeige von Zustellhinweisen
Zustellhinweis-Objekt
Objekt welches alle Skripte (MapboxRoute, NextStepRoutePathRenderer, DefaultOnScreenTargetIndicator) die für die Navigation per se und die Anzeige der Navigationshinweise (Pfeile, Schilder, Linie) nötig sind beherbergt
Hausobjekt zur Positionierung im AR-Raum & hierfür benötigtes Skript
Würfel zur Positionierung im AR-Raum & hierfür benötigtes Skript
Prefab-Objekt, welches alle möglichen Objekte die als Navigationshinweise angezeigt werden können (ohne Linie) enthält & hierfür benötigtes Management-Skript (SignPost)
Hinweistafel-Objekt
Pfeil-Objekt
ZielPin-Objekt
Gruppe für Karten-Objekte
Objekt mit Managementskript (MapHandler) für die Karte
Eigentliche Karte & Skript hierfür
Kamera fürs Rendering der Karte

Objektbeschreibung

Aktuell aktive Szene
Gruppe für alle für die Funktionsweise des Smartphones als Bedienoberfläche bzw. Anzeigeoberfläche benötigten Objekte und Komponenten
Renderkamera für die Anzeige am Smartphone
Gruppe & Untergruppe für die Bedienoberfläche ausmachenden Komponenten & Beherbergung des ButtonController-Skript
Gruppe für Button-Objekte , das Trackpads & die Mauseingabe am Trackpad
Gruppe für allgemeine Buttons
Mauszeiger
Gruppe für alle die Bedienoberfläche ausmachenden Objekte (Tabs, Titel, Texte, Buttons, Dropdowns, ...)
Titel + Untertitel
Gruppe für die Tabs ausmachenden Komponenten & Beherbergung des Menü-, HintsHandler-, StyleSettings-Skripts
Hintergrund Objekt und dessen Umrandung
Tab-Buttons
Gruppe für die Objekte des Start Tabs
Gruppe für die Objekte des Settings Tabs
Gruppe für die Objekte des Debug Tabs
Gruppe für die Objekte des Map Tabs
Text-Objekt zur Anzeige von Informationen
Debug-Console für die Entwicklung
Objekt für's Monitoring von Interaktionen auf der Bedienoberfläche
Objekt zur Beleuchtung der gesamten Szene
Objekt zur Erstellung & Management einer AR-Szene
Objekt zur Verortung der AR-Szene im realen Raum, der Gruppierung in diesem befindlichen Objekte (teils erst befüllt während der Laufzeit) & der Beherbergung des PassPosition-, GetGPS-Skripts
AR-Szenen Verortungsobjekt + dazugehörige Skripte (ARLocationOrientation, ARLocationManager)
Gruppe für das XREAL-Kamerasystem zum Management des HMDs und der Skripte hierfür
Linke Kamera (f. linkes Brillenglas) des HMDs
theoretische mittige Kamera des HMDs
Objekt zur Speicherung des Mittelpunkt des HMDs
Rechte Kamera (f. rechtes Brillenglas) des HMDs
Objekt zur Verortung des Smartphones im AR-Raum, der Erstellung eines Strahls ausgehend von diesem & der Abfrage des Auftreffens des Strahls auf einem Objekt im AR-Raum
Objekt zur Abfrage der tatsächlichen Bewegung des Smartphones im Raum
Zeichenbereich für die Anzeige von Zustellhinweisen
Zustellhinweis-Objekt
Objekt welches alle Skripte (MapboxRoute, NextStepRoutePathRenderer, DefaultOnScreenTargetIndicator) die für die Navigation per se und die Anzeige der Navigationshinweise (Pfeile, Schilder, Linie) nötig sind beherbergt
Hausobjekt zur Positionierung im AR-Raum & hierfür benötigtes Skript
Würfel zur Positionierung im AR-Raum & hierfür benötigtes Skript
Prefab-Objekt, welches alle möglichen Objekte die als Navigationshinweise angezeigt werden können (ohne Linie) enthält & hierfür benötigtes Management-Skript (SignPost)
Hinweistafel-Objekt
Pfeil-Objekt
ZielPin-Objekt
Gruppe für Karten-Objekte
Objekt mit Managementskript (MapHandler) für die Karte
Eigentliche Karte & Skript hierfür
Kamera fürs Rendering der Karte

Visualisierung ausgewählter Objekte



9 Literaturverzeichnis

Adams, S., 2023. [Online]
Available at: <https://sabrinaadams.de/kommunikationsmodelle/>
[Zugriff am 12 03 2024].

ALEGER, 2024. *Smart Glasses - Mit Augmented Reality zur Industrie 4.0*. [Online]
Available at: <https://alegerglobal.com/augmented-reality/smart-glasses/#:~:text=Diese%20Informationen%20werden%20in%20die,f%C3%BCr%20Maschinen%20verst%C3%A4ndliche%20Daten%20%C3%BCbersetzen.>
[Zugriff am 17 02 2024].

Ang, K. J. C. et al., 2018. Design and development of Google glass-based campus navigation system. *Journal of Telecommunication, Electronic and Computer Engineering Vol.10 (1-5)*, pp. 75-81.

Antakli, A. et al., 2017. Virtuelle Techniken und Semantic-Web. In: *Web-basierte Anwendungen Virtueller Techniken*. Heidelberg: Springer.

arm, kein Datum *Glossary: Gaming Engines*. [Online]
Available at: <https://www.arm.com/glossary/gaming-engines>
[Zugriff am 04 November 2023].

Audi AG, 2011. *Elektrik/Elektronik - Fahrassistenzsysteme - Parkassistent*. [Online]
Available at: <https://www.audi-technology-portal.de/de/elektrik-elektronik/fahrerassistenzsysteme/parkassistent>
[Zugriff am 16 03 2024].

Awati, R. & Rosencrance, L., 2021. *computer hardware*. [Online]
Available at: <https://www.techtarget.com/searchnetworking/definition/hardware>
[Zugriff am 26 02 2024].

Azuma, R. T., 1997. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, August, pp. 355-385.

Barnard, D., 2023. *Degrees of Freedom (DoF): 3-DoF vs 6-DoF for VR Headset Selection*. [Online]
Available at: <https://virtualspeech.com/blog/degrees-of-freedom-vr>
[Zugriff am 07 03 2023].

Basseville-Lipka, M., 2020. *HD Vision Systems*. [Online]
Available at: <https://www.hdvvisionsystems.com/en/blog/image-recognition-vs-pattern-recognition/>
[Zugriff am 26 04 2024].

Bigelow, S. J. & Rouse, M., 2023. *platform*. [Online]
Available at: <https://www.techtarget.com/searchitoperations/definition/platform>
[Zugriff am 12 03 2024].

Bishop, I. D., 2011. Landscape planning is not a game: Should it be?. *Landscape and Urban Planning*, 30 April, pp. 390-392.

Bishop, I. D. & Stock, C., 2010. Using collaborative virtual environments to plan wind energy installations. *Renewable Energy*, 4 April, pp. 2348-2355.

Blickcheck, kein Datum *Die 3D-Brille – neue visuelle Technologien für Entertainment und Wissenschaft*. [Online]

Available at: <https://www.blickcheck.de/brillen/brillenarten/3d-brille/>
[Zugriff am 30 04 2024].

Böhmermann, J., 2018. *Wir sind Versandsoldaten - Jan Böhmermann & der Chor der Scheinselbstständigen* | *NEO MAGAZIN ROYALE*. [Online]

Available at: https://www.youtube.com/watch?v=pQsS0VqM4aA&t=28s&ab_channel=ZDFMAGAZINROYALE
[Zugriff am 03 06 2024].

Bowman, D. A., Kruijff, E., La Viola, J. J. & Poupyrev, I., 2004. *3D User Interfaces: Theory and Practice*. s.l.:Addison-Wesley.

Bruckner, R. & Gepp, J., 2023. *Wie kann das sein, dass Menschen 17 Stunden täglich zu minimalen Löhnen arbeiten müssen?*. [Online]

Available at: <https://www.derstandard.at/story/2000145302685/wie-kann-das-sein-dass-menschen-17-stunden-taeglich-zu>
[Zugriff am 03 06 2024].

Bruckner, R. & Laufer, N., 2020. *Lange Schlangen bei der Post: Mitarbeiter psychisch und physisch am Limit*. [Online]

Available at: <https://www.derstandard.at/story/2000122597274/team-christkind-und-andere-packer-lzusteller-am-limit>
[Zugriff am 03 06 2024].

Burhoff, D., 2018. *Elektronische Geräte/Mobiltelefon im Straßenverkehr*. [Online]

Available at: https://www.burhoff.de/veroeff/aufsatz/zap_F9_S987.htm#_Toc511496348
[Zugriff am 17 03 2024].

Buyuksalih, I. et al., 2017. 3D MODELLING AND VISUALIZATION BASED ON THE UNITY GAME ENGINE – ADVANTAGES AND CHALLENGES. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 01 11, pp. 161-166.

Cambridge Dictionary, 2014. *routing*. [Online]

Available at: <https://dictionary.cambridge.org/de/worterbuch/englisch/routing>
[Zugriff am 16 03 2024].

Cantisano, T., 2022. *Google Maps' new Live View feature finally gives you a real reason to use AR*. [Online]

Available at: <https://www.xda-developers.com/google-maps-new-features-live-view->

search/

[Zugriff am 04 06 2024].

Carter, R., 2023a. *VR/AR Headsets and Smart Glasses: What's the Difference?*. [Online] Available at: <https://www.xrtoday.com/augmented-reality/vr-ar-headsets-and-smart-glasses-whats-the-difference/>

[Zugriff am 04 06 2024].

Carter, R., 2023b. *Why Are Motorists Switching to AR Navigation?*. [Online] Available at: <https://www.xrtoday.com/augmented-reality/why-are-motorists-switching-to-ar-navigation/>

Carter, R., 2024. *How to Compare Smart AR Glasses in 2024*. [Online] Available at: <https://www.xrtoday.com/augmented-reality/how-to-compare-smart-ar-glasses-in-2024/>

[Zugriff am 18 02 2024].

Chaturvedi, A., 2019. *Google Maps rolls out AR navigation for iOS and Android smartphones*. [Online]

Available at: <https://www.geospatialworld.net/blogs/google-maps-ar-navigation-iphones-android/>

[Zugriff am 14 03 2024].

Chatzopoulos, D., Bermejo, C., Huang, Z. & Hui, P., 2017. Mobile Augmented Reality Survey: From Where We Are to Where We Go. *IEEE Access*, pp. 6917-6950.

Christiansen, T. & Erb, W.-D., 2023. *Lexikon der Geographie: GIS*. [Online] Available at: <https://www.spektrum.de/lexikon/geographie/gis/3079>

[Zugriff am 06 03 2024].

Chu, C., Wang, S. & Tseng, B., 2017. Mobile Navigation Services with Augmented Reality. *IEEJ transactions on electrical and electronic engineering Vol. 12*, pp. 95-103.

Chung, J., Pagnini, F. & Langer, E., 2016. Mindful navigation for pedestrians: Improving engagement with augmented reality. *Technology in society Vol.45*, pp. 29-33.

Dang, A., 2021. *Wie sieht der Arbeitsalltag als Paketzusteller aus?*. [Online] Available at: <https://www.derstandard.at/story/2000131641825/wie-sieht-der-arbeitsalltag-als-paketzusteller-aus>

[Zugriff am 03 06 2024].

Darken, R. P. & Peterson, B., 2001. Spatial Orientation and Wayfinding. In: *Handbook of Virtual Environment Technology*. s.l.:Lawrence Erlbaum Associates, p. 493–518.

Davis, T., kein Datum *Explained! What is Projected Augmented Reality? (with examples)*. [Online]

Available at: <https://essentialpicks.com/projected-augmented-reality/>

[Zugriff am 24 02 2024].

deepaman07ad, 2022. *Different Types of Wireless Communication Media*. [Online]
Available at: <https://www.geeksforgeeks.org/different-types-of-wireless-communication-media/>

[Zugriff am 16 03 2024].

DerStandard, 2024. *Augmented-Reality-Brille und ferngesteuertes Parken*. [Online]
Available at: <https://www.derstandard.at/story/3000000202161/augmented-reality-brille-und-ferngesteuertes-parken>

[Zugriff am 17 03 2024].

Dirscherl, H.-C., 2019. *MBUX im Test: Mercedes A-Klasse gehorcht aufs Wort und AR-Navigation*. [Online]

Available at: <https://www.pcwelt.de/article/1175677/mbux-mercedes-benz-user-experience-navigation-mercedes-me-a-klasse-test.html>

[Zugriff am 16 03 2024].

Downs, R. & Stea, D., 1973. *Image and environment : cognitive mapping and spatial behavior*. New York: Routledge.

DPA, 2022. *Augmented Reality: Mit Head-up-Displays alles im Blick*. [Online]
Available at: <https://www.sueddeutsche.de/wissen/technik-augmented-reality-mit-head-up-displays-alles-im-blick-dpa.urn-newsml-dpa-com-20090101-220307-99-420950>

[Zugriff am 16 03 2024].

Dumbre, S., 2023. *Projection-based AR*. [Online]

Available at: <https://medium.com/@sakshi.dumbre31/projection-based-ar-220240721883>

[Zugriff am 24 02 2024].

Dünser, A. et al., 2012. Exploring the use of handheld AR for outdoor navigation. *Computers & Graphics Vol.36*, pp. 1084-1095.

Eddy, N., 2022. *How BMW, Mercedes use augmented reality to boost navigation*. [Online]
Available at: <https://europe.autonews.com/technology/how-bmw-mercedes-use-augmented-reality-boost-navigation>

[Zugriff am 03 05 2024].

Entertainment LLC, 2023. *Homepage*. [Online]

Available at: <https://mywebar.com/>

[Zugriff am 20 05 2024].

esri, 2023a. *ArcGIS Maps SDK for Unity*. [Online]

Available at: <https://developers.arcgis.com/unity/>

[Zugriff am 03 06 2024].

esri, 2023b. *ArcGIS Maps SDK for Unreal Engine*. [Online]

Available at: <https://developers.arcgis.com/unreal-engine-sdk/>

[Zugriff am 03 06 2024].

esri, 2023. *Esri Support GIS-Wörterbuch - GISc.* [Online]
Available at: <https://support.esri.com/de-de/gis-dictionary/giscience>
[Zugriff am 03 06 2024].

European Commission, The European Union Agency for the Space Programme, Issue 1, 2022. *EUSPA EO and GNSS Market Report*, Luxembourg: Publications Office of the European Union.

Fortes, D., 2020. *Unity AR+GPS Location.* [Online]
Available at: <https://docs.unity-ar-gps-location.com/#main-features>
[Zugriff am 14 04 2024].

Fortes, D., 2022a. *dmbfm/unity-ar-gps-location-mapbox-route-sample.* [Online]
Available at: <https://github.com/dmbfm/unity-ar-gps-location-mapbox-route-sample>
[Zugriff am 15 04 2024].

Fortes, D., 2022b. *Create a Google Maps Live View-like application with Unity3D in under 10 minutes!.* [Online]
Available at: <https://medium.com/@daniel.mbfm/create-a-google-maps-live-view-like-application-in-unity3d-in-10-minutes-3265d6fc6b5d>
[Zugriff am 14 04 2024].

Fortes, D., 2023. *AR + GPS Location.* [Online]
Available at: <https://assetstore.unity.com/packages/tools/integration/ar-gps-location-134882>
[Zugriff am 14 04 2024].

Froehlich, A., 2022. *location-based service (LBS).* [Online]
Available at: <https://www.techtarget.com/searchnetworking/definition/location-based-service-LBS>
[Zugriff am 16 03 2024].

Furness, N. & Hansen, R., 2022. *Introducing ArcGIS Maps SDK for Unreal Engine.* [Online]
Available at: <https://www.esri.com/arcgis-blog/products/unreal-engine/announcements/introducing-arcgis-maps-sdk-for-unreal-engine/>

Gardini Miguel, P., 2024. *An Expert's Guide To The 12 Best Augmented Reality Software Of 2024.* [Online]
Available at: <https://thectoclub.com/tools/best-augmented-reality-software/>
[Zugriff am 03 03 2024].

Gerasymov, O., 2024. *9-STEP AUGMENTED REALITY APP DEVELOPMENT GUIDE.* [Online]
Available at: <https://codeit.us/blog/how-to-create-an-augmented-reality-app>
[Zugriff am 04 06 2024].

Ghazwani, Y. & Smith, S., 2020. Interaction in Augmented Reality: Challenges to Enhance. *ICVARS '20: Proceedings of the 2020 4th International Conference on Virtual and Augmented Reality Simulations*, Februar, pp. 39-44.

Gillis, A., 2020. *Applikation (Anwendung)*. [Online] Available at: <https://www.computerweekly.com/de/definition/Applikation-Anwendung> [Zugriff am 01 03 2024].

GISGeography, 2023. *What is GIScience (Geographic Information Science)?*. [Online] Available at: <https://gisgeography.com/giscience-geographic-information-science/> [Zugriff am 03 06 2024].

Golden, B., 2008. *The vehicle routing problem : latest advances and new challenges*. New York: Springer.

Google Glass, 2013. *Get Directions [through Google Glass]*. [Online] Available at: <https://www.youtube.com/watch?v=V8ofTlynWPo> [Zugriff am 06 06 2024].

Google, 2024. *Deprecations*. [Online] Available at: <https://developers.google.com/maps/deprecations#gaming-svcs-deprecation> [Zugriff am 14 04 2024].

Google, kein Datum *Live View in Google Maps*. [Online] Available at: <https://support.google.com/maps/answer/9332056?hl=de&co=GENIE.Platform%3DAndroid> [Zugriff am 14 03 2024].

Gregory, J., 2014. *Game Engine Architecture*. Boca Raton: CRC Press.

Grobman, A., 2023. *Augmented Reality Is Already Changing Healthcare For The Better*. [Online] Available at: <https://www.forbes.com/sites/forbestechcouncil/2023/06/01/augmented-reality-is-already-changing-healthcare-for-the-better/> [Zugriff am 27 04 2024].

Groves, P., 2013. *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Boston: Artech House.

Haas, H.-D., Neumair, S.-M. & Schlesinger, D., 2018. *Mental Map*. [Online] Available at: <https://wirtschaftslexikon.gabler.de/definition/mental-map-37207/version-260650>

Hall, C., 2023. *What is Google Maps AR navigation and Live View and how do you use it?*. [Online] Available at: <https://www.pocket-lint.com/apps/news/google/147956-what-is-google-maps-ar-navigation-and-how-do-you-use-it/> [Zugriff am 04 06 2024].

Hannah, B., 2021. *Was ist Hardware? Einfach erklärt.* [Online] Available at: https://praxistipps.chip.de/was-ist-hardware-einfach-erklart_41233 [Zugriff am 26 02 2024].

Hansen, R., 2021. *ARCUSER: Bridging the World of 3D GIS and Game Engines.* [Online] Available at: <https://www.esri.com/about/newsroom/arcuser/bridging-the-world-of-3d-gis-and-game-engines/> [Zugriff am 06 03 2024].

Hauger, D., 2006. *Anaglyphen.* [Online] Available at: <http://www.dma.ufg.ac.at/app/link/Grundlagen%3AAllgemeine/module/13889?step=all> [Zugriff am 27 04 2024].

Hayes, G., 2009. *16 TOP AUGMENTED REALITY BUSINESS MODELS.* [Online] Available at: <https://www.personalizemedia.com/16-top-augmented-reality-business-models/> [Zugriff am 11 03 2024].

Herwig, A. & Paar, P., 2002. Game Engines: Tools for Landscape Visualization and Planning?. In: *Trends in GIS and Virtualization in Environmental Planning and Design.* s.l.:Wichmann, pp. 162-171.

Hitti, N., 2019. *Google Maps launches AR walking directions.* [Online] Available at: <https://www.dezeen.com/2019/08/13/google-maps-ar-live-view-tech/> [Zugriff am 14 03 2024].

Hofmann-Wellenhof, B., Legat, K. & Wieser, M., 2003. *Navigation : principles of positioning and guidance.* Wien: Springer.

Höllerer, T. et al., 1999. Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers & Graphics* 23, pp. 779-785.

Huang, H., Schmidt, M. & Gartner, G., 2012. Spatial Knowledge Acquisition with Mobile Maps, Augmented Reality and Voice in the Context of GPS-based Pedestrian Navigation: Results from a Field Test. *Cartography and geographic information science Vol.39 (2)*, pp. 107-116.

Huo, Y. et al., 2021. Efficient visualization of large-scale oblique photogrammetry models in unreal engine. *ISPRS international journal of geo-information vol. 10*, 25 September, p. 643.

I., R., 2019. *Community: Introducing Live View, the new augmented reality feature in Google Maps.* [Online] Available at: <https://support.google.com/maps/thread/11554255/introducing-live-view-the-new-augmented-reality-feature-in-google-maps?hl=en> [Zugriff am 06 06 2024].

- IKEA, 2024. *IKEA Apps*. [Online]
Available at: <https://www.ikea.com/at/de/customer-service/mobile-apps/#56cdd1e0-88da-11ec-9871-d53207cb3be5>
[Zugriff am 04 06 2024].
- Interaction Design Foundation - IxDF, 2022. *What is Virtuality Continuum?*. [Online]
Available at: <https://www.interaction-design.org/literature/topics/virtuality-continuum>
[Zugriff am 03 06 2024].
- Jackson, R., 2013. *Google Glass: Navigation Review*. [Online]
Available at: <https://phandroid.com/2013/05/09/google-glass-navigation-review/>
[Zugriff am 16 03 2024].
- Jantunen, T. & M. J. & P. A. & L. J., 2016. On the rhythm of head movements in Finnish and Swedish Sign Language sentences. *Speech Prosody 2016*, pp. 850-853.
- Kamalanathsharma, R. K., Rakha, H. A. & Zohdy, I. H., 2015. Survey on In-vehicle Technology Use: Results and Findings. *International Journal of Transportation Science and Technology Vol. 4*, pp. 135-149.
- Kasanmascheff, M., 2021. *Google Maps: So aktiviert und nutzt ihr die Live View-Navigation*. [Online]
Available at: <https://www.netzwelt.de/anleitung/191118-google-maps-so-aktiviert-nutzt-live-view-navigation.html>
[Zugriff am 13 03 2024].
- Kato, H. et al., 2000. Virtual object manipulation on a table-top AR environment. *Proceedings - IEEE and ACM International Symposium on Augmented Reality*, pp. 111-119.
- Keil, M., 2023. *7 Kommunikationsmodelle, die Sie kennen sollten*. [Online]
Available at: <https://blog.hubspot.de/sales/kommunikationsmodelle>
[Zugriff am 12 03 2024].
- Kergel, D. & Heidkamp-Kergel, B., 2023. Definition Kommunikation. In: *Beratung und Kommunikation*. Wiesbaden : Springer Fachmedien, pp. 5-6.
- Kim, S. K. et al., 2017. Augmented-reality survey: From concept to application. *KSI Transactions on Internet and Information Systems 11 (2)*, pp. 982-1004.
- Kipper, G. & Rampolla, J., 2012. *Augmented reality : an emerging technologies guide to AR*. Waltham: Syngress.
- Klaß, C., 2010. *Lego setzt auf animierte Verkaufspackungen*. [Online]
Available at: <https://www.golem.de/1004/74585.html>
[Zugriff am 13 03 2024].

konden, 2022. *AnimalWalk for Nreal*. [Online]
Available at: <https://www.xreal.com/de/arlab/detail?id=5&appname=AnimalWalkforNreal>
[Zugriff am 27 04 2024].

Kumar, P., 2024. *What are depth-sensing cameras? How do they work?*. [Online]
Available at: <https://www.e-consystems.com/blog/camera/technology/what-are-depth-sensing-cameras-how-do-they-work/#:~:text=Depth%2Dsensing%20is%20the%20measuring,to%20it%20on%20the%20go.>
[Zugriff am 04 06 2024].

Lackes, R. & Siepermann, M., 2018b. *Systemprogramm*. [Online]
Available at: <https://wirtschaftslexikon.gabler.de/definition/systemprogramm-50325/version-273545>
[Zugriff am 03 06 2024].

Lackes, R. & Siepermann, M., 2018c. *Anwendungsprogramme*. [Online]
Available at: <https://wirtschaftslexikon.gabler.de/definition/anwendungsprogramm-29651/version-253251>
[Zugriff am 29 02 2024].

Lackes, R. & Siepermann, M., 2018d. *Betriebssystem (BS)*. [Online]
Available at: <https://wirtschaftslexikon.gabler.de/definition/betriebssystem-bs-30823/version-254399>
[Zugriff am 22 02 2024].

Lackes, R. & Siepermann, M., 2018a. *Software*. [Online]
Available at: <https://wirtschaftslexikon.gabler.de/definition/software-45585/version-268877>
[Zugriff am 29 02 2024].

Li, A., 2018. *Google I/O 2018: Google Maps' new 'For you' tab introduces personal recommendations, AR camera features*. [Online]
Available at: <https://9to5google.com/2018/05/08/google-maps-new-for-you-tab-introduces-personal-recommendations-ar-camera-features/>

Lightform, 2018. *It's Lit! Make Projected AR Murals with Lightform*. [Online]
Available at: <https://www.youtube.com/watch?v=XTR-NhJ9em8>
[Zugriff am 04 06 2024].

Lo, V. E.-W., Green, P. A. & Franzblau, A., 2011. Where Do People Drive? Navigation System Use by Typical Drivers and Auto Experts. *Journal of navigation Vol. 64*, pp. 357-373.

Luber, S., 2017. *Was ist eine IDE?*. [Online]
Available at: <https://www.dev-insider.de/was-ist-eine-ide-a-399da9ffb32c9a1e2796c436c6c66061/>
[Zugriff am 29 02 2024].

Lütjens, M. C., 2018. *Immersive Virtual Reality Visualisation of the Arctic Clyde Inlet on Baffin Island (Canada) by Combining Bathymetric and Terrestrial Terrain Data*. Hamburg: s.n.

Lütjens, M., Kersten, T. P., Dorschel, B. & Tschirschwitz, F., 2019. Virtual Reality in Cartography: Immersive 3D Visualization of the Arctic Clyde Inlet (Canada) Using Digital Elevation Models and Bathymetric Data. *Multimodal Technologies and Interaction*, 01 03, pp. 9-14.

Maier, G. W., Siepermann, M. & Lackes, R., 2018. *Kommunikation*. [Online] Available at: <https://wirtschaftslexikon.gabler.de/definition/kommunikation-37167/version-260610> [Zugriff am 12 03 2024].

Mapbox, 2019. *Maps SDK for Unity*. [Online] Available at: <https://docs.mapbox.com/unity/maps/guides/> [Zugriff am 07 06 2024].

Markgraf, D., 2018. *Location-based-Services*. [Online] Available at: <https://wirtschaftslexikon.gabler.de/definition/location-based-services-53641/version-276714> [Zugriff am 03 06 2024].

Maróy, Á., 2017. *How Augmented Reality Can Make Aviation Safer and Better*. [Online] Available at: <https://www.uploadvr.com/ar-aviation-safer-better/> [Zugriff am 16 03 2024].

Mata, F. & Claramunt, C., 2013. Augmented navigation in outdoor environments. *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'13)*, November, pp. 524-527.

Mehler-Bicher, A. & Steiger, L., 2014. *Augmented reality : Theorie und Praxis*. München: De Gruyter Oldenbourg.

Mercedes-Benz USA, 2021. *How To: Augmented Reality*. [Online] Available at: <https://www.youtube.com/watch?v=AHaSnYikRpU> [Zugriff am 16 03 2024].

Meta, 2024. *Meta Quest 3*. [Online] Available at: <https://www.meta.com/at/quest/quest-3/> [Zugriff am 18 02 2024].

Milgram, P., Takemura, H., Utsumi, A. & Kishino, F., 1994. Augmented Reality: A Class. *Telemanipulator and Telepresence Technologies*, pp. 282-292.

Mohd, H. M. & Ruzinoor, C. M., 2016. Online 3D terrain visualisation using Unity 3D game engine: A comparison of different contour intervals terrain data draped with UAV images. *Earth and Environmental Science* 37, 01 06.

Morozov, M., kein Datum *Augmented Reality in Military: AR Can Enhance Warfare and Training*. [Online]

Available at: <https://www.jasoren.com/augmented-reality-military/#:~:text=This%20AR%20device%20provides%20soldiers,a%20soldier%20is%20aiming%20at.>

[Zugriff am 27 04 2024].

Mulloni, A., Seichter, H. & Schmalstieg, D., 2011. User experiences with augmented reality aided navigation on phones. *10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 229-230.

Münch, S., 2022. *Begriff erklärt: Was genau ist Hardware?*. [Online]

Available at: <https://www.computerbild.de/artikel/cb-Tipps-PC-Hardware-Was-ist-Hardware-Technik-Fragen-31395727.html>

[Zugriff am 26 02 2024].

Münzer, S. et al., 2006. Computer-assisted navigation and the acquisition of route. *Journal of Environmental Psychology* 26, pp. 300-308.

National Geographic, 2017. *ENCYCLOPEDIA ENTRY: GIS (Geographic Information System)*. [Online]

Available at: <https://education.nationalgeographic.org/resource/geographic-information-system-gis/>

[Zugriff am 21 11 2023].

National Geographic, 2023. *Navigation*. [Online]

Available at: <https://education.nationalgeographic.org/resource/navigation/>

[Zugriff am 03 06 2024].

Nextech3D.ai, 2022. *What Are The Different Types of Augmented Reality?*. [Online]

Available at: <https://www.nextechar.com/blog/what-are-the-different-types-of-augmented-reality>

Niantic, 2024. *Pokemon GO*. [Online]

Available at: <https://pokemongolive.com/?hl=de>

[Zugriff am 04 06 2024].

Painter, Lewis, 20204. *Meta Quest 3 Review*. [Online]

Available at: <https://www.trustedreviews.com/reviews/meta-quest-3>

Phillips, C., 2022. *4 new updates that make Maps look and feel more like the real world*.

[Online]

Available at: <https://blog.google/products/maps/4-new-updates-maps-searchon-2022/>

[Zugriff am 14 03 2024].

Pichler, G., 2023. *Meta Quest 3 im Test: Vielversprechender Sprung in die Mixed Reality*.

[Online]

Available at: <https://www.derstandard.at/story/3000000192065/meta-quest-3-im-test->

vielversprechender-sprung-in-die-mixed-reality
[Zugriff am 04 06 2024].

Pranoto, H. et al., 2023. Augmented reality navigation application to promote tourism to local state attraction “Lawang Sewu”. *Procedia computer science Vol. 216*, pp. 757-764.

Rafiee, A., Van der Male, P., Dias, E. & Scholten, H., 2017. Developing a wind turbine planning platform: Integration of “sound propagation model GIS-game engine” triplet. *Environmental Modelling & Software*, 29 Juni, pp. 326-343.

Rampolla, J. & Kipper, G., 2012. *Augmented Reality : An Emerging Technologies Guide to AR*. Waltham: Syngress.

Ranieri, M., 2020. *A new sense of direction with Live View*. [Online] Available at: <https://blog.google/products/maps/new-sense-direction-live-view/> [Zugriff am 20 04 2024].

Rauschnabel, P., kein Datum *Ist das Realitäts-Virtualitäts-Kontinuum noch zeitgemäß?*. [Online] Available at: <https://www.philipp-rauschnabel.com/augmented-reality-forschung/kritik-realitaets-virtualitaets-kontinuum-milgram/> [Zugriff am 18 02 2024].

Reckmann, T., 2014. *Google Glass - Vorderansicht*. [Online] Available at: https://commons.wikimedia.org/wiki/File:Google_Glass_Front.jpg [Zugriff am 20 05 2024].

Redaktion, 2023. *Logistiker: Paketzusteller soll Mangelberuf werden*. [Online] Available at: <https://orf.at/stories/3324995/> [Zugriff am 03 06 2024].

Rehman, U. & Shi, C., 2017. Augmented-Reality-Based Indoor Navigation: A Comparative Analysis of Handheld Devices Versus Google Glass. *IEEE transactions on human-machine systems Vol.47 (1)*, 02, pp. 140-151.

RIS, 2023. *Bundesrecht konsolidiert: Kraftfahrgesetz 1967 § 102, tagesaktuelle Fassung*. [Online] Available at: <https://www.ris.bka.gv.at/Dokumente/Bundesnormen/NOR40251910/NOR40251910.html> [Zugriff am 17 03 2024].

RokidGlass, 2023. *Rokid Glass Developer Docs and SDK*. [Online] Available at: <https://github.com/RokidGlass> [Zugriff am 14 04 2024].

Rokid, kein Datum a. *Rokid Developer Documentation*. [Online] Available at: <https://rokid.yuque.com/ouziyq/tvgpgk> [Zugriff am 14 04 2024].

Rokid, kein Datum b. *Rokid Max Specs.* [Online]
 Available at: <https://global.rokid.com/pages/rokid-max-specs>
 [Zugriff am 14 04 2024].

Rönisch, M., 2021. *Usability von Augmented Reality bei der Indoor-Navigation : ein Vergleich zwischen einer Augmented Reality-Applikation und einer analogen Karte.* Wien: s.n.

Rouse, M., 2023. *Peripheriegerät.* [Online]
 Available at: <https://www.techopedia.com/de/definition/peripheriegeraet>
 [Zugriff am 01 03 2024].

Ruzinoor, C. M. et al., 2014. *Using game engine for 3D terrain visualisation of GIS data: A review.* Bristol, IOP Publishing, p. 11.

Ruzinoor, C. M. et al., 2015. *WEB BASED 3D TERRAIN VISUALIZATION USING GAME.* Istanbul, s.n., pp. 254-260.

Ruzinoor, C. M. & Mohd, H. M., 2018. Using Game Engine for Online 3D Terrain Visualization with Oil Palm Tree Data. *Journal of Telecommunication, Electronic and Computer Engineering Vol.10*, 21 02, pp. 93-97.

Schmalsteig, D. & Hollerer, T., 2016. *Augmented reality: principles and practice.* s.l., ACM.

ScopeAR, 2024. *Homepage.* [Online]
 Available at: <https://us.scopear.com/>
 [Zugriff am 20 05 2024].

Seydel, P., 2023. *Was sind eigentlich Head-Mounted Displays?.* [Online]
 Available at: <https://triboot.de/head-mounted-displays/>
 [Zugriff am 04 06 2024].

Siegel, A. W. & White, S. H., 1975. The Development of Spatial Representations of Large-Scale Environments. *Advances in Child Development and Behavior Vol. 10*, pp. 9 - 55.

Siegl, R., 2022. *Game-Engines: Das 3D-Pendant zur Darstellung von 2D-Daten mittels Grafiksoftware in der Geoinformation?.* Wien: unpubliziert - Universität Wien.

Spektrum.de, 2000. *Stereoskopisches Sehen.* [Online]
 Available at: <https://www.spektrum.de/lexikon/geowissenschaften/stereoskopisches-sehen/15675>
 [Zugriff am 17 02 2024].

Spektrum.de, 2001. *Lexikon der Geowissenschaften.* [Online]
 Available at: <https://www.spektrum.de/lexikon/geographie/stereoskopische-bildbetrachtung/7655>
 [Zugriff am 17 02 2024].

Stock, C. et al., 2008. SIEVE: Collaborative Decision-making in an. *Cartography and Geographic Information Science*, pp. 133-144.

Sygyic a.s., 2017. *Real View Navigation from Sygyic*. [Online]
Available at: <https://www.youtube.com/watch?app=desktop&v=yzQCCOAEgqY>
[Zugriff am 20 04 2024].

Sygyic, 2024. *Real View Navigation*. [Online]
Available at: <https://www.sygyic.com/de/what-is/real-view-navigation>
[Zugriff am 20 04 2024].

Thöny, M., Billeter, M. & Pajarola, R., 2015. *Vision paper: the future of scientific terrain visualization*. Bellevue, WA, USA, s.n., pp. 1-4.

TOPSIM GmbH, 2024. *In Kürze erklärt: Was sind eigentlich Serious Games?*. [Online]
Available at: <https://www.topsim.com/blog/was-sind-eigentlich-serious-games/>
[Zugriff am 18 05 2024].

Toth, P. & Vigo, D., 2014. *Vehicle routing : problems, methods, and applications*. Philadelphia: SIAM.

Tremosa, L., 2024. *Beyond AR vs. VR: What is the Difference between AR vs. MR vs. VR vs. XR?*. [Online]
Available at: <https://www.interaction-design.org/literature/article/beyond-ar-vs-vr-what-is-the-difference-between-ar-vs-mr-vs-vr-vs-xr>
[Zugriff am 03 06 2024].

UCLA, 2015. *UCLA's Augmented Reality Sandbox*. [Online]
Available at: <https://www.youtube.com/watch?v=CE1B7tdGCw0>

Ulrich, L., 2022. *Augmented Reality Coming to a Windshield Near You > Mass-market players will introduce HUD-enabled driving soon*. [Online]
Available at: https://spectrum.ieee.org/augmented-reality-car-hud?utm_campaign=post-teaser&utm_content=gy0dqv2s

Unbound XR, kein Datum *Top 6 Virtual Reality (VR) Gloves*. [Online]
Available at: <https://unboundxr.co.uk/top-vr-gloves>
[Zugriff am 12 03 2024].

uniaccessories, 2023. *What is DP Alt Mode?*. [Online]
Available at: <https://de.uniaccessories.com/blogs/blog/what-is-dp-alt-mode>
[Zugriff am 14 04 2024].

Unity, 2023. *DIGITAL TWIN APPLICATIONS AND USE CASES*. [Online]
Available at: <https://unity.com/solutions/digital-twin-applications-and-use-cases>
[Zugriff am 04 November 2023].

Unity, 2024. *Learn with Pathways*. [Online]
Available at: <https://learn.unity.com/pathways>
[Zugriff am 15 04 2024].

Unity, 2024. *Order of execution for event functions*. [Online]
Available at: <https://docs.unity3d.com/2021.3/Documentation/Manual/ExecutionOrder.html>
[Zugriff am 20 05 2024].

Unity, kein Datum *ERROR-FREE MAINTENANCE AND ON-THE-JOB TRAINING WITH AR*. [Online]
Available at: <https://unity.com/case-study/taqtile-augmented-reality-training>
[Zugriff am 04 November 2023].

Unity, kein Datum *Game development terms*. [Online]
Available at: <https://unity.com/how-to/beginner/game-development-terms>
[Zugriff am 04 November 2023].

USGS, kein Datum *What is a geographic information system (GIS)?*. [Online]
Available at: <https://www.usgs.gov/faqs/what-geographic-information-system-gis>
[Zugriff am 03 06 2024].

Velichko, M., kein Datum *How Augmented Reality is Used in Commercial Aviation*. [Online]
Available at: <https://www.jasoren.com/augmented-reality-in-commercial-aviation/#:~:text=Pilots%20with%20augmented%20reality%20headgear,Safer%20and%20immersive%20training>
[Zugriff am 16 03 2024].

Verma, P., 2022. *Fighter pilots will don augmented reality helmets for training*. [Online]
Available at: <https://www.washingtonpost.com/technology/2022/08/04/virtual-reality-fighter-pilot-helmet/>
[Zugriff am 16 03 2024].

ViewAR, 2024. *Homepage*. [Online]
Available at: <https://www.viewar.com/>
[Zugriff am 20 05 2024].

VisionStar Information Technology, 2024. *Homepage*. [Online]
Available at: <https://www.easyar.com/>
[Zugriff am 20 05 2024].

Vletter, W., 2019. First Steps in the Use of a Game Engine for Historical Roads and Paths Research. *Journal of Computer Applications in Archaeology*, pp. 3-11.

Vogel, M. A., 1999. *Understanding pattern recognition*. [Online]
Available at: <https://www.vision-systems.com/cameras-accessories/article/16736303/understanding-pattern-recognition>
[Zugriff am 20 05 2024].

Volkswagen, kein Datum a. *AR Head-up-Display: wegweisende Navigation*. [Online] Available at: <https://www.volkswagen.at/elektroauto/id-technologie/id-ar-navigation> [Zugriff am 16 03 2024].

Volkswagen, kein Datum b. *AR-Head-Up-Display*. [Online] Available at: <https://www.volkswagen.at/id7/id7/ar-head-up-display> [Zugriff am 16 03 2024].

Vossle, 2024. *Homepage*. [Online] Available at: <https://vossle.ai/> [Zugriff am 20 05 2024].

XRAI, 2022. *XRAI Glass*. [Online] Available at: <https://www.xreal.com/de/arlab/detail?id=16&appname=XRAIGlass> [Zugriff am 27 04 2024].

XREAL, 2024. *Unity SDK 2.2.0*. [Online] Available at: <https://developer.xreal.com/download> [Zugriff am 14 04 2024].

XREAL, kein Datum a. *Compatibility*. [Online] Available at: <https://www.xreal.com/de/compatibility/> [Zugriff am 14 04 2024].

XREAL, kein Datum b. *Compatibility*. [Online] Available at: <https://xreal.gitbook.io/nrsdk/v/v2.1.0/nrsdk-fundamentals/xreal-devices/compatibility> [Zugriff am 07 06 2024].

XREAL, kein Datum c. *NRSDK Overview*. [Online] Available at: <https://xreal.gitbook.io/nrsdk/v/v2.1.0> [Zugriff am 14 04 2024].

Yasar, K., 2023. *image recognition*. [Online] Available at: <https://www.techtarget.com/searchenterpriseai/definition/image-recognition> [Zugriff am 14 03 2024].

Yasar, K. & Rosencrance, L., 2022. *Software Development Kit (SDK)*. [Online] Available at: <https://www.computerweekly.com/de/definition/Software-Development-Kit-SDK> [Zugriff am 29 02 2024].

Yount, Z. F., Kass, S. J. & Arruda, J. E., 2022. Route learning with augmented reality navigation aids. *Transportation research. Part F, Traffic psychology and behaviour* Vol. 88, pp. 132-140.