

Augmented Reality basierend auf Ort, Ausrichtung und Lage

- ARwizard -

Masterthesis

im Studiengang

MSc Business Information Technology

vorgelegt von

Ralph Büchi

Gehrenholz 9e

8055 Zürich

Matrikel-Nr.: S08-269-147

School of Management and Law

Zürcher Hochschule für Angewandte Wissenschaft (ZHAW)

Winterthur, 29. Mai 2015

Erstprüfer/in:	Dr. Alexandre de Spindler
Zweitprüfer/in:	Dr. Claudia Pedron

Abstract

Augmented Reality (AR) bietet völlig neue Möglichkeiten im Umgang mit der Flut von Informationen, welche heutzutage verfügbar sind. Informationen werden unmittelbar dort präsentiert, wo sie benötigt werden, nämlich im Blickfeld des Anwenders (Ludwig & Reinmann, 2005). Durch das Aufkommen dieser Technologie wurden unterschiedlichste Frameworks und Lösungen von verschiedenen Anbietern entwickelt, mit denen AR-Anwendungen für Mobilgeräte erstellt werden können. Besonders im Bereich standortbezogener AR-Anwendungen setzen die vorhandenen Lösungen jedoch viel Wissen und Können im Bereich der Software-Entwicklung voraus. In dieser Masterthesis wird zur Lösung dieser Problemstellung ein webbasiertes IT-Artefakt namens ARwizard kreiert, mit dem das Entwickeln von standortbezogenen AR-Anwendungen vereinfacht wird. Dazu wird über die Generalisierung der verschiedenen standortbezogenen AR-Anwendungen ein Metamodell entwickelt, dass die drei Szenarien „AR-Navigation“, „AR-Browser mit POIs mit verschiedenen Informationstypen“ und „AR-Browser mit POIs von externen Ressourcen“ miteinander vereint. Es werden die verschiedenen Komponenten solcher Anwendungen herausgearbeitet und in den ARwizard implementiert. Zusammenfassend stellt das IT-Artefakt ARwizard einen solide Prototypen dar, um die Entwicklung standortbezogener AR-Anwendungen auf Basis von webbasierten Technologien und ohne Programmierkenntnisse zu ermöglichen.

Schlagwörter

Augmented Reality, AR-Browser, AR-Navigation
Location-Based-Service

Inhaltsverzeichnis

Abstract.....	II
Inhaltsverzeichnis	III
Abbildungsverzeichnis.....	V
Tabellenverzeichnis	VII
Listingverzeichnis	VII
Abkürzungsverzeichnis	VIII
1 Einleitung	1
1.1 Ausgangslage	1
1.2 Problemstellung	3
1.3 Forschungsfrage	4
1.4 Zielsetzung	4
1.5 Forschungsmethodik und Gestaltungsprozesse	4
1.6 Adressaten und Nutzen der Arbeit	5
1.7 Abgrenzung und verwendete Ressourcen	6
2 Theoretischer Teil.....	7
2.1 Augmented Reality	7
2.2 Tracking Software.....	7
2.3 Standortbezogene Dienste.....	8
3 Implementierung & Design „ARwizard“	10
3.1 Anforderungsanalyse	10
3.1.1 Grundlegende Anforderungen	10
3.1.2 Anforderungsmatrix	10
3.2 Generalisierung	14
3.2.1 Metamodell.....	14
3.2.2 Entity-Relationship-Modell POI	15
3.2.3 Schnittstellen für externe POIs.....	16
3.3 Aufbau der Lösung „ARwizard“	16
3.3.1 Administrationsbereich	16
3.3.2 ARwizard AR-Browser	17
3.4 Komponenten AR-Browser	17
3.4.1 GPS-Sensor	19
3.4.2 Lagesensor.....	20
3.4.3 Berechnungen.....	21
3.4.4 Kamera	25

3.4.5	Navigation	26
3.4.6	Radius.....	27
3.4.7	Radar	27
3.4.8	Listendarstellung	27
3.4.9	Landkarte.....	27
3.4.10	POI-Ansicht.....	27
3.5	Datenbank-Modell	28
4	Vorstellung der Applikation „ARwizard“	30
4.1	Administrationsbereich	30
4.1.1	Dashboard.....	30
4.1.2	AR-Szenario erstellen	30
4.1.3	AR-Szenario löschen.....	34
4.1.4	AR-Szenario bearbeiten	34
4.1.5	POI hinzufügen	35
4.1.6	POI löschen	37
4.1.7	POI bearbeiten.....	37
4.1.8	Externe POI Quelle hinzufügen	38
4.1.9	Externe POI-Quelle löschen.....	39
4.2	AR-Browser	40
4.2.1	AR-Browser starten.....	40
4.2.2	AR-Browser Grunddarstellung	41
4.2.3	POI-Ansicht.....	42
4.2.4	Kartenansicht.....	44
4.2.5	Navigationsmodus	46
5	Flexibilität der Anwendung „ARwizard“	48
5.1	Szenario: Reiseführer „Skulpturen“	48
5.2	Szenario: Denkmalgeschützte Gebäude (externe POIs)	49
5.3	Szenario: AR-Navigation Kurierdienst.....	50
5.4	Szenario: AR-Navigation Eventveranstaltung	51
6	Konklusion	52
6.1	Forschungsergebnis.....	52
6.2	Offene Punkte und Ausblick	54
6.3	Fazit.....	56
	Literaturverzeichnis	57
	Anhang A: CD.....	59
	Wahrheitserklärung	60

Abbildungsverzeichnis

Abbildung 1: Reality-Virtuality-Continuum (Milgram, 1994).....	7
Abbildung 2: LBS im Bereich Navigation & Information (Steiniger, 2006)	9
Abbildung 3: Laya Screenshot	12
Abbildung 4: Mixare Screenshot	12
Abbildung 5: Junaio Screenshot AR-Browser.....	12
Abbildung 6: Junaio Screenshot Karten- & Listenansicht.....	12
Abbildung 7: Wikitude Screenshot AR-Browser	13
Abbildung 8: Wikitude Screenshot POI-Ansicht.....	13
Abbildung 9: Wikitude Navigation (Wikitude.com, 2015)	13
Abbildung 10: Metamodell AR-Anwendung	14
Abbildung 11: Entity-Relationship-Modell POI.....	15
Abbildung 12: Bereiche des ARwizards.....	16
Abbildung 13: AR-Browser Bereich des ARwizards	17
Abbildung 14: Datenobjekt "arwizard"	18
Abbildung 15: Ausrichtung des mobilen Endgeräts (W3c, 2014).....	20
Abbildung 16: Haversine Formel (Sinnott, 1984)	21
Abbildung 17: Geographisches Koordinatensystem (de Lange, 2013)	21
Abbildung 18: Formel für die Berechnung der Peilung (Veness, 2015)	22
Abbildung 19: Darstellung Peilung (Geiger, 2012).....	22
Abbildung 20: Formel für die Berechnung des Höhenwinkels (Geiger, 2012).....	23
Abbildung 21: Darstellung Höhenwinkel (Geiger, 2012)	24
Abbildung 22: Datenbank-Modell ARwizard.....	28
Abbildung 23: Dashboard ARwizard	30
Abbildung 24: Neues AR-Szenario erstellen - Schritt 1	31
Abbildung 25: Neues AR-Szenario erstellen - Schritt 2.....	32
Abbildung 26: Verwaltungsseite AR-Szenario.....	33
Abbildung 27: AR-Szenario löschen	34
Abbildung 28: AR-Szenario löschen bestätigen	34
Abbildung 29: AR-Szenario bearbeiten.....	34
Abbildung 30: Neuer POI hinzufügen [Add new POI]	35
Abbildung 31: Neuer POI erfassen	35
Abbildung 32: POI von Google Maps auswählen	36
Abbildung 33: Verwaltungsseite POI	36
Abbildung 34: POI löschen.....	37
Abbildung 35: POI bearbeiten	37
Abbildung 36: Externe Quellen hinzufügen - Schritt 1	38
Abbildung 37: Externe Quellen hinzufügen - Schritt 2	38
Abbildung 38: Übersicht Externe Quellen.....	38

Abbildung 39: Externe Quelle löschen	39
Abbildung 40: AR-Browser Startprozess	40
Abbildung 41: AR-Browser Darstellung mit Funktionen.....	41
Abbildung 42: POI-Ansicht Uto-Kulm mit Bild	42
Abbildung 43: POI-Ansicht ZHAW mit Video	42
Abbildung 44: POI-Ansicht Paradeplatz mit Audio	43
Abbildung 45: POI-Ansicht Bern nur Text.....	43
Abbildung 46: Kartenansicht starten	44
Abbildung 47: Beispiel Landkarte 1	44
Abbildung 48: Beispiel Landkarte 2	44
Abbildung 49: Kartenansicht beenden.....	45
Abbildung 50: Karte zentrieren	45
Abbildung 51: Button für Navigationsmodus.....	46
Abbildung 52: Navigationsmodus im ARwizard.....	46
Abbildung 53: Kartenmodus mit Navigation.....	47
Abbildung 54: Meldung: Ziel erreicht	47
Abbildung 55: Meldung: Distanz zu klein.....	47

Tabellenverzeichnis

Tabelle 1: 7-Richtlinien nach Hevner et al. (2004).....	5
Tabelle 2: Verwendete Ressourcen und System.....	6
Tabelle 3: Anforderungsmatrix.....	11
Tabelle 4: Entity-Relationship-Modell POI.....	15
Tabelle 5: Reisemodi nach Google Maps Directions (Google, 2015).....	26
Tabelle 6: Datenbanktabelle „ar_poi“.....	28
Tabelle 7: Datenbanktabelle „ar_szenario“	29
Tabelle 8: Funktionenbeschreibung des AR-Browsers	42
Tabelle 9: Evaluation der Richtlinien nach Hevner et al. (2004)	53
Tabelle 10: Ordner-/ Dateistruktur CD.....	59

Listingverzeichnis

Listing 1: JSON Schnittstellenspezifikation für externe Quellen.....	16
Listing 2: Location-Informationen auslesen	19
Listing 3: Compass-Heading berechnen (richt, 2014; W3c, 2014)	21
Listing 4: JavaScript-Code Distanzberechnung.....	22
Listing 5: Berechnung der Peilung (Veness, 2015)	23
Listing 6: Berechnung des Höhenwinkels	24
Listing 7: Zugriff auf Kamera „navigator.getUserMedia“	25

Abkürzungsverzeichnis

AR	Augmented Reality
GPS	Global Positioning System
LBS	Location Based Service / Standortbezogene Dienste
POI	Point of Interest / Ort von Interesse
SDK	Software Development Kit

1 Einleitung

Aufgrund der Flut von Informationen gewinnt ein effizienter Zugriff auf die richtigen Informationen zur richtigen Zeit und am richtigen Ort zunehmend an Relevanz. Klassisch werden derartige Informationen mit Hilfe von Darstellungsformen wie Bücher, Videofilme, Vorträge, etc. wiedergegeben und verbreitet. Augmented Reality bietet eine völlig neue Art und Weise genau dort Informationen zu präsentieren, wo sie benötigt werden und zwar im Blickfeld des Anwenders (Ludwig & Reinmann, 2005).

Um dem Anwender die Informationen direkt im Blickfeld darzustellen, bieten verschiedene Anbieter sogenannte Augmented-Reality-Browser (AR-Browser) für Smartphones an. Dabei ist die Idee hinter diesen AR-Browsern einfach. Der Anwender betrachtet durch die Kamera seines Smartphone die reale Umgebung und online verfügbare Zusatzinformationen werden ins Kamerabild integriert. Die Informationen beziehen sich immer auf einen Point of Interest (POI) und werden durch die Angabe der geographischen Koordination an der richtigen Position im Kamerabild bereitgestellt (Michel, 2011).

1.1 Ausgangslage

Aktuell sind verschiedene Frameworks und Lösungen vorhanden, um gezielt AR-Anwendungen zu entwickeln. Nachfolgend werden die gängigsten Lösungen aufgezeigt, die das Kreieren von AR-Anwendungen für mobile Endgeräte wie Smartphones und Tablets unterstützen. Dabei gibt es verschiedene Arten von AR-Anwendungen, die von den jeweiligen Lösungen unterstützt werden.

DAQRI

DAQRI bietet eine Software an, mit der verschiedene AR-Anwendungen erstellt werden können, wobei keine Programmierkenntnisse erforderlich sind. Mit der Software können vorwiegend AR-Anwendungen erstellt werden, die mit visuellem Tracking arbeiten. Ein Fokus liegt dabei auf der Verarbeitung von 3D-Objekten, welche mit der Software bearbeitet und im Bild platziert werden können (Minock, 2014).

Aurasma

Ähnlich wie DAQRI bietet Aurasma eine Software an, mit der man ohne Programmierkenntnisse AR-Anwendungen erstellen kann. Zentral ist dabei, dass die Software AurasmaStudio direkt im Internet-Browser verwendet werden kann und nicht installiert werden muss. Zusätzlich bietet Aurasma eine Mobile-Applikation an, mit der direkt auf dem mobilen Endgerät eine AR-Anwendung erstellt werden kann. Die erstellte Anwen-

dung kann dann auf einem mobilen Endgerät in einer Bibliothek abgerufen und ausgeführt werden. Diese Applikationen werden auch AR-Browser genannt. Der Vorteil daran ist, dass die erstellten AR-Szenarien vom Anwender nicht jedes Mal auf dem Endgerät installiert werden müssen, sondern direkt mit einer Applikation abgerufen werden können (Minock, 2014).

Metaio

Metaio ist einer der führenden Hersteller von AR-Technologien. Sie bietet eine Software sowie ein funktionsstarkes Software Development Kit (SDK) an. Einfache AR-Anwendungen können ohne grosse Programmierkenntnisse in der Software MetaioCreator erstellt werden. Um die eigene AR-Anwendung dann nutzen zu können, kann die erstellte Anwendung publiziert werden und ähnlich wie bei Aurasma in einer Bibliothek in dem AR-Browser Junaio auf dem mobilen Endgerät aufgerufen und gestartet werden. Die SDK von Metaio unterstützt sowohl nicht-visuelles Tracking, wie auch visuelles Tracking (Metaio, 2014).

Layar

Die Plattform Layar bietet sowohl einen Online-Creator als auch ein leistungsstarkes SDK an. Layar bietet ausserdem ein eigenes GEO SDK an, mit dem Location-Based-AR-Anwendungen erstellt werden können. Hierbei sind Programmierkenntnisse jedoch eine Voraussetzung. Mit einem zweiten SDK namens VISION SDK können AR-Anwendungen gebildet werden, die visuelle Marker erkennen. Die erstellten Anwendungen können wiederum direkt in einer Mobile-Anwendung abgerufen und verwendet werden (Minock, 2014).

Augment Dev

Ein weiteres Werkzeug, um AR-Anwendungen zu erstellen, ist Augment Dev. Benutzer können mit dieser Software mittels Online-Creator oder der Augment Dev-App eine AR-Anwendung erstellen. Es sind keine Programmierkenntnisse erforderlich und es können bereits vorhandene visuelle Marker ausgewählt werden, um dann eigene 3D-Objekte im Kamerabild anzeigen zu lassen (Minock, 2014).

Blippar

Blippar bietet mit dem Blippbuilder eine Software an, mit der interaktive AR-Anwendungen erstellt werden können. Aufgerufen werden die erstellten Szenarien mit der Blipp-App. Das Produkt wird bereits von grossen Unternehmen zu Werbezwecken verwendet und hat sich gut etabliert. Die Anwendungsfälle beschränken sich jedoch auf visuelles Tracking (blippar, 2014).

buildAR

Mit der Plattform buildAR können Location-Based-AR-Anwendungen direkt online erstellt werden. Es können ohne Programmierkenntnisse Interessenpunkte hinzugefügt und mit Aktionen, wie zum Beispiel automatisches Anzeigen von Text-Informationen oder Audiosignalen, verknüpft werden. Die fertige Anwendung kann wie bei Metaio veröffentlicht werden und mit einer Mobile-App in einer Bibliothek abgerufen werden (Minock, 2014).

Hoppala Augmentation

Hoppala Augmentation bietet einen Online-Creator für Location-Based-AR-Anwendungen an. Die Möglichkeiten sind jedoch sehr begrenzt. Man kann bestimmte Positionen auf der Karte markieren und Informationstexte hinzufügen. Abgerufen werden kann das erstellte Szenario mit einer Mobile-Applikation (Minock, 2014).

Wikitude

Eine weitere AR-Software für mobile Endgeräte, welche für standortbezogene Anwendungen eingesetzt werden kann, ist Wikitude. Wikitude bietet dem Entwickler ein SDK an, mit dem verschiedene AR-Anwendungen erstellt werden können. Zudem können Szenarien erstellt werden, die auf der Mobile-Applikation in einer Bibliothek abgerufen werden können. Wikitude bietet zusätzlich einen Online-Creator an, der jedoch ausschließlich für visuelle Tracking Anwendungen verwendet werden kann (Minock, 2014).

1.2 Problemstellung

In der Ausgangslage wurden AR-Softwares und Frameworks aufgezeigt, mit denen man AR-Anwendungen für Mobilgeräte entwickeln kann. Es wurde aufgezeigt, dass einige Software-Anbieter standortbezogene AR-Anwendungen zwar unterstützen, die jedoch viel Wissen und Kenntnisse im Bereich der Software-Entwicklung voraussetzen. Einige Anbieter bieten einen Online-Creator an, dieser ist jedoch in der Funktionalität sehr eingeschränkt. Zudem sind sie an die jeweilige Mobile-Anwendung gebunden, die der Software-Anbieter vorgibt. Das Erstellen einer eigenen standortbezogenen AR-Mobileanwendung ist daher nur mit Programmierkenntnissen möglich. Es kann also gesagt werden, dass die AR-Anbieter im Bereich von standortbezogenen Anwendungen gute Werkzeuge und Frameworks anbieten, diese jedoch nur von erfahrenen Entwicklern angewandt werden können. Es fehlt eine einfach verständliche Plattform, die von Personen ohne Programmierkenntnisse genutzt werden kann, mit der komplexere standortbezogene AR-Anwendungen erstellt und auch veröffentlicht werden können.

1.3 Forschungsfrage

Auf der Grundlage des Wissens aus der Ausgangslage und der Darstellung der Problemstellung wurde die folgende zentrale Forschungsfrage abgeleitet:

- Wie kann die Entwicklung von standortbezogenen AR-Anwendungen für mobile Endgeräte generell unterstützt werden?

Aus dieser Forschungsfrage lassen sich weitere Unterfragen ableiten, deren Beantwortung zur Klärung der Forschungsfrage beitragen sollen:

- Welche Anforderungen müssen erfüllt sein, damit ohne Programmierkenntnissen eine standortbezogene AR-Applikation gebildet werden kann?
- Ist es möglich ausschliesslich mit existierenden Webtechnologien AR-Applikationen zu bilden?
- Welche Anforderungen haben standortbezogene AR-Anwendungen?
- Welche Anforderungen muss eine Plattform haben, mit der man AR-Anwendungen erstellen kann?

1.4 Zielsetzung

Zur Beantwortung der Forschungsfragen soll im Rahmen dieser Masterthesis auf der Basis der Problemstellung ein IT-Artefakt erarbeitet werden. Es soll eine Online-Plattform realisiert werden, mit der standortbezogene AR-Anwendungen generiert und bereitgestellt werden können. Um das Ziel zu erreichen wird eine Generalisierung von standortbezogenen AR-Anwendungen gemacht, um daraus eine modulare Grundlage für das IT-Artefakt zu schaffen. Für dieses IT-Artefakt wird der Name *ARwizard* gewählt.

Das neuartige an *ARwizard* ist, dass ohne Programmierkenntnisse standortbezogene AR-Anwendungen erstellt werden können, die ausschliesslich auf die Lage, Ort und Ausrichtung des mobilen Endgeräts zurückgreifen und somit nur mit nicht-visuellen Tracking auskommt. Zudem sollen der *ARwizard* und die daraus erstellten AR-Anwendungen auf den heutigen Webtechnologien (HTML5, JavaScript, PHP, etc.) basieren. Die Lösung soll daher vollständig webbasiert umgesetzt werden und in einem Internet-Browser gestartet werden können.

1.5 Forschungsmethodik und Gestaltungsprozesse

Da im Rahmen dieser Masterthesis ein IT-Artefakt realisiert wird, wird der Forschungsansatz Design-Science von Hevner et al. (2004) gewählt. Dieser beinhaltet die Beschreibung des Problems sowie die Entwicklung und Evaluation einer Lösung. Er ist darauf ausgelegt ein IT-Artefakt zu kreieren und Probleme praktisch zu lösen. Dabei wird darauf geachtet, dass das entwickelte IT-Artefakt eine Relevanz zur Lösung des

Problems besitzt und evaluiert wird, um damit das Forschungsergebnis bestätigen zu können. Hevner et al. (2004) bietet für die Bestätigung und Evaluierung der Forschungsarbeit folgende sieben Richtlinien an, die erfüllt sein sollten:

Richtlinie	Beschreibung
Design as an Artefact	Es muss ein neuartiges und nützliches Artefakt gebildet werden, mit dem ein Problem in der Realität gelöst werden kann.
Problem Relevance	Das Ziel ist es ein Problem technisch zu lösen.
Design Evaluation	Die Nützlichkeit, Qualität und Effektivität der Forschungsergebnisse sollen methodisch rigoros nachgewiesen sein. Dies kann durch Fallstudien, Analysemethoden, Experimentalmethoden oder Tests geschehen.
Research Contributions	Das Ergebnis der Forschung muss ein eindeutiger und verifizierbarer Beitrag sein, der den Stand der Forschung voranbringt. Das kann in Form eines Artefakts, Beiträgen zur Wissensbasis oder Beiträgen zu Forschungsmethoden sein.
Research Rigor	Durch Methoden soll die wissenschaftliche Strenge nachgewiesen werden. Dies kann durch die Einhaltung von methodischem und strukturiertem Vorgehen erfüllt werden.
Design as a Search Process	Es soll ein wiederkehrender Suchprozess nach der optimalen Lösung durchlaufen werden, wobei mögliche Lösungen vorgeschlagen, evaluiert, verbessert oder verworfen werden. Der Suchprozess soll in Veröffentlichungen publiziert werden.
Communication of Research	Die Ergebnisse der Forschung müssen sowohl technisch- als auch managementorientiert effektiv und zielgruppengerecht kommuniziert werden. Die Neuartigkeit kann zudem explizit erklärt werden.

Tabelle 1: 7-Richtlinien nach Hevner et al. (2004)

Um dem Forschungskonzept nach Hevner et al. (2004) gerecht zu werden, wird diese Masterthesis in zwei Teile aufgeteilt. Zu Beginn wird das Forschungsgebiet Augmented-Reality theoretisch beschrieben. Dann wird im praktischen Teil die Lösung erarbeitet und umgesetzt. Die Lösung wird abschliessend aufgrund von Beispielszenarien evaluiert und bezüglich des Nutzens und der Relevanz überprüft.

1.6 Adressaten und Nutzen der Arbeit

Mit der vorliegenden Arbeit soll der wissenschaftlichen Gemeinschaft ein Mehrwert in der Erstellung und Handhabung von standortbezogenen AR-Anwendungen dargestellt werden. Zudem wird durch die kritische Beurteilung und Reflexion aufgezeigt, in welchen Bereichen weitere Forschung betrieben werden muss. Ein weiterer Nutzen

bietet sich für die Software-Entwickler von mobilen Anwendungen durch die Generalisierung von standortbezogenen AR-Anwendungen.

1.7 Abgrenzung und verwendete Ressourcen

In dieser Arbeit werden zur Positionierungsbestimmung ausschliesslich GPS und Lagesensoren verwendet. Zudem wird das visuelle Tracking Verfahren in dieser Arbeit nicht berücksichtigt. Das Tracking geschieht ausschliesslich mit dem nicht-visuellen Tracking, mit den Komponenten, Kompass, GPS und Trägheitssensoren (Neigung, Beschleunigung). Ergänzend ist der ARwizard nur nutzbar, wenn das mobile Endgerät im Portrait-Modus ausgerichtet ist. Landscape-Modus wird momentan nicht unterstützt.

Nachfolgend werden die verwendeten Ressourcen und Systeme in Tabelle 2 aufgelistet. Diese bilden das Testsystem mit dem der ARwizard entwickelt und kontinuierlich getestet wurde. Das System garantiert einen reibungslosen Betrieb.

Software / System	Beschreibung
Aptana Studio 3	Build 3.6.1.201410201044
JQuery	2.1.3
PHP	5.5.3
HTML	5
CSS	3
MySQL	5.6.11
phpMyAdmin	4.0.4.1
Mobiles Endgerät	Beschreibung
Samsung Galaxy S4 mini (GT-I9195)	Testsystem für den AR-Browser
Android-Version	4.4.2
Google-Chrome-Browser (Mobile)	40.0.2214.89
JavaScript	V8 3.30.33.15

Tabelle 2: Verwendete Ressourcen und System

2 Theoretischer Teil

2.1 Augmented Reality

Eine erste Einordnung des Begriffs Augmented Reality (dt. erweiterte Realität) stammt von Paul Milgram, welcher im Jahre 1994 einen Artikel veröffentlichte, in dem er eine Abgrenzung von verschiedenen Arten der Vermischung zwischen Realität und Virtualität vorgenommen hat. Dabei hatte er ein Reality-Virtuality-Kontinuum gebildet, mit der realen Umgebung auf der einen Seite und der virtuellen Umgebung auf der anderen Seite. Dazwischen befindet sich ein Übergangsbereich, welcher als Mixed-Reality bezeichnet wird (Milgram, 1994).

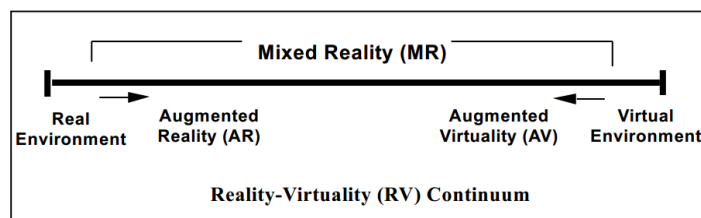


Abbildung 1: Reality-Virtuality-Kontinuum (Milgram, 1994)

Wie in Abbildung 1 ersichtlich, wird Augmented Reality nahe der realen Umgebung positioniert. Dies wird damit begründet, dass bei der Augmented Reality die reale Umgebung mit Zusatzinformationen erweitert wird und nicht eine virtuelle Umgebung geschaffen wird, wie zum Beispiel bei der Virtual Reality, bei der die reale Umgebung grundsätzlich ausgeschaltet wird (Milgram, 1994).

2.2 Tracking Software

Ein wichtiger Teil um AR-Anwendungen zu ermöglichen, ist das Erkennen der realen Umgebung, um anschliessend diese reale Umgebung mit den entsprechenden virtuellen Objekten oder Informationen zu ergänzen. Es wird dabei von einer Tracking-Software oder einem Tracker gesprochen. Die Schwierigkeit liegt darin, dass eine perfekte Erweiterung der realen Umgebung nur dann erzielt werden kann, wenn die Informationen sehr genau in die reale Umgebung platziert werden können. Zudem müssen je nach Anwendungsfall Kompromisse eingegangen werden, so benötigt eine AR-Anwendung für die Chirurgie eine sehr hohe Genauigkeit, während Touristenführer oder sonstigen Anwendungen mit mobilen Endgeräten keine sehr hohe Genauigkeit erfordern. Auch muss der Aspekt berücksichtigt werden, dass eine sehr hohe Genauigkeit nur durch komplexere Algorithmen erreicht wird und diese wiederum eine bessere Hardware benötigen (Mehler-Bicher, 2011).

Gemäss Mehler-Bicher (2011) gibt es im Bereich der Tracker verschiedene Verfahren. Es kann grundsätzlich zwischen dem nicht-visuellen und visuellen Tracking unterschieden werden. Nicht-visuelles Tracking ermöglicht es herauszufinden, wo sich ein Gerät genau befindet. Hierbei können folgende Methoden angewendet werden:

- Kompass
- GPS
- Ultraschallsensoren
- Optische Sensoren
- Trägheitssensoren (Neigung, Beschleunigung)

Visuelles Tracking wird hingegen in der Regel mit einer Videokamera realisiert. Dabei gibt es zwei Kategorien: merkmalsbasierende oder modellbasierende Systeme. Bei Endgeräten mit begrenzter Rechenleistung werden eher merkmalsbasierende Systeme eingesetzt, da der Tracker innerhalb des Videobilds vordefinierte Formen, Bilder oder Objekte erkennt. Diese werden auch Marker genannt und werden im Bereich der AR optisch so optimiert, dass sie von einem Tracker wiedererkannt werden. Bei modellbasierten Systemen ist dem Tracker ein Referenzmodell bekannt, mit dem das Videobild abgeglichen wird, was einen grösseren Rechenaufwand bedeutet (Mehler-Bicher, 2011).

2.3 Standortbezogene Dienste

Standortbezogene Dienste (engl. Location Based Service / LBS) sind Teil der Kommunikationstechnik. Der Grundgedanke dabei ist, dass der Benutzer aufgrund seines aktuellen Standorts mit Informationen versorgt wird, welche im Kontext zu seinem Standort stehen (Elektronik Kompendium, 2014). Gemäss Steiniger (2006) kann es folgende Anwendungsbereiche geben:

- Notfalldienste (Autopannendienst, etc.)
- Navigation (Routenplanung, etc.)
- Information (Reiseführer, Museen, etc.)
- Werbung und Bezahlung (Strassengebühren, Werbeplakate, etc.)
- Tracking (Produkttracking, Taxidienst, etc.)
- Games (Geocaching, etc.)

Je nach Anwendungsfall muss die Genauigkeit der Positionierung höher oder geringer sein. Interessant für diese Arbeit sind die Bereiche Navigation und Information. Bei näherer Betrachtung ist erkennbar, dass die Genauigkeit der Positionierung eher hoch sein muss (Steiniger, 2006). In Abbildung 2 sind verschiedene Anwendungsfälle ersichtlich.

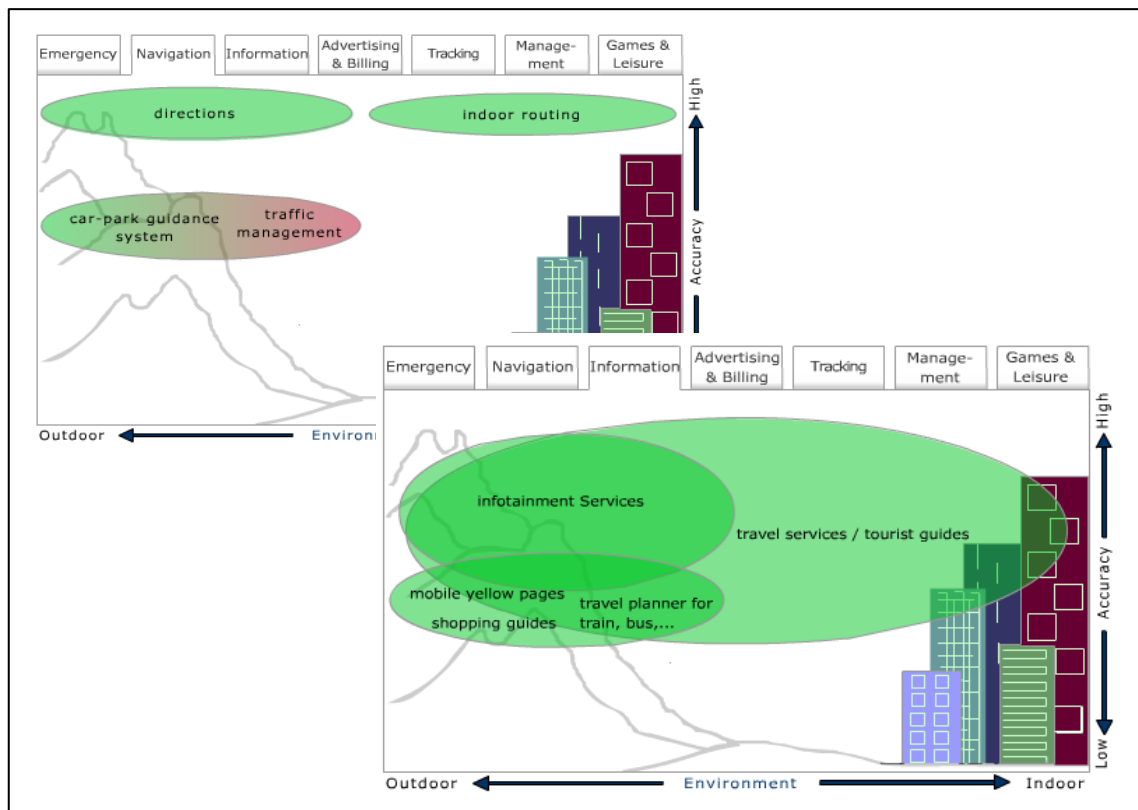


Abbildung 2: LBS im Bereich Navigation & Information (Steiniger, 2006)

3 Implementierung & Design „ARwizard“

In diesem Kapitel werden die Grundlagen für den ARwizard aufgezeigt. Zu Beginn werden mit Hilfe einer die Anforderungsanalyse die wichtigsten Funktionen einer standortbasierten AR-Anwendung definiert. Diese Anforderungsanalyse bildet dabei die Basis für die Erarbeitung einer Generalisierung von standortbezogenen AR-Anwendungen und des ARwizards. Zudem werden in diesem Kapitel die einzelnen Komponenten und der Aufbau vom ARwizard erläutert.

3.1 Anforderungsanalyse

Wie zuvor erläutert, sollen aus der Anforderungsanalyse die wichtigsten Funktionen des ARwizard abgeleitet werden.

3.1.1 Grundlegende Anforderungen

Damit eine standortbezogene AR-Anwendung im Rahmen dieser Arbeit erstellt werden kann, gibt es grundlegende Anforderungen an das Umfeld, die erfüllt sein müssen. Zum einen benötigt das mobile Endgerät eine Kamera, um AR-Anwendungen nutzen zu können. Dabei ist wichtig, dass die Kamera von der Anwendung angesteuert und verwendet werden kann. Zudem muss ein berührungsempfindliches Display vorhanden sein, damit der Anwender mit den virtuellen Informationen im Kamerabild interagieren kann. Der Anwender hat damit die Möglichkeit direkt auf die Szene der erweiterten Realität Einfluss zu nehmen. Eine weitere wesentliche Anforderung betrifft die Datenverbindung. Das mobile Endgerät muss uneingeschränkt eine Verbindung zu den Services aufbauen können, um Grafiken, Informationen oder Routen dynamisch abrufen zu können. Zusätzlich muss das mobile Endgerät bei standortbezogenen AR-Anwendungen die Position vom Gerät und vom Nutzer eindeutig bestimmen können. Dabei muss die Positionsbestimmung eine hohe Genauigkeit (wenige Meter) haben sowie zeit- und ortsunabhängig sein. Eine wichtige Rolle spielt hierbei auch der Lage-sensor und der Kompass, der die Ausrichtung des Geräts bestimmen kann (Stürzekarn, 2010).

3.1.2 Anforderungsmatrix

Aus der Analyse der verschiedenen AR-Anbieter, mit denen standortbezogene AR-Anwendungen erstellt werden können, wurde eine Anforderungsmatrix (Tabelle 3) erstellt. Diese stellt die Funktionen und Spezifikationen dar, die bereits bei existieren-

den AR-Anwendungen implementiert wurden. Aufgrund dieser Matrix können die relevanten Anforderungen und Funktionen für den ARwizard definiert werden.

Anforderungen							ARwizard
	Metaio/ Junaio POI / Excel Plugin	wikitude	Layar	buildar	hoppala	mixare	
POI – Informationen							
Latitude	✓	✓	✓	✓	✓	✓	✓
Longitude	✓	✓	✓	✓	✓	✓	✓
Altitude	✓	✓	✓	✓	✓	✓	✓
Name / Titel	✓	✓		✓	✓	✓	✓
Beschreibung	✓	✓	✓	✓	✓	✓	✓
Video	✓	✓	✓	✓			✓
Audio	✓	✓	✓	✓	✓		✓
Image	✓	✓	✓	✓	✓	✓	✓
Icon / Kategorie	✓	✓	✓	✓			✓
Administrationsbereich							
Keine Programmierung notwendig	✓			✓	✓		✓
POI über-Karte definieren					✓		✓
Extern POI einbinden							✓
Design ohne Programmierung				✓			✓
Webbasiert				✓	✓		✓
Funktionalitäten AR-Browser							
Radar-Umfeld	✓	✓	✓	✓	✓	✓	✓
Radius-Eingrenzung	✓	✓	✓	✓	✓	✓	✓
Landkarten-Modus	✓	✓	✓	✓		✓	✓
Listendarstellung	✓	✓	✓	✓		✓	✓
Navigation		✓					✓
Webbasiert							✓

Tabelle 3: Anforderungsmatrix

In den nachfolgenden Abbildungen (Abbildung 3 bis Abbildung 9) sind verschiedene Screenshots der AR-Anwendungen Layar, Wikitude, Junaio, Mixare ersichtlich, welche auch als Grundlage für die Anforderungsanalyse verwendet wurden. Mit diesen Abbildungen ist zudem ersichtlich, dass die Funktionen, wie zum Beispiel das Radar, bei allen AR-Anwendungen vorhanden sind.



Abbildung 3: Layar Screenshot



Abbildung 4: Mixare Screenshot

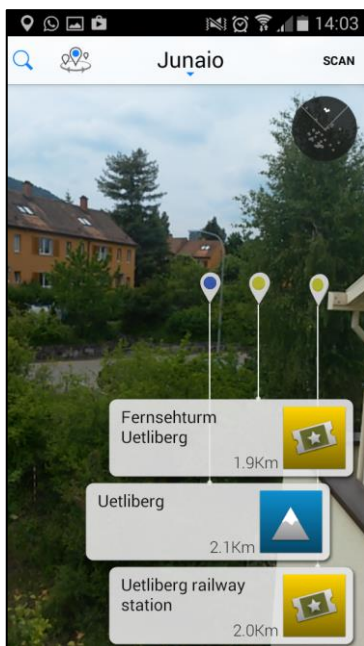


Abbildung 5: Junaio Screenshot AR-Browser

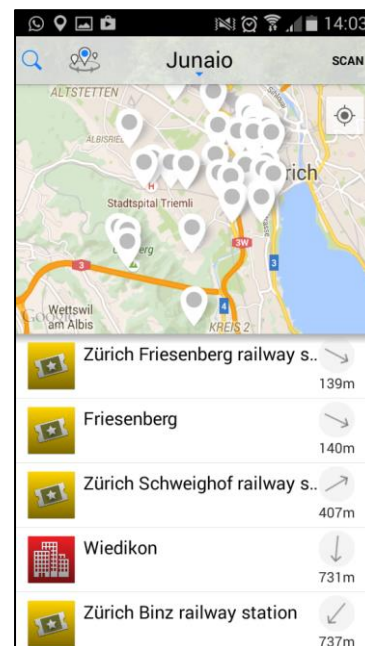


Abbildung 6: Junaio Screenshot Karten- & Listenansicht

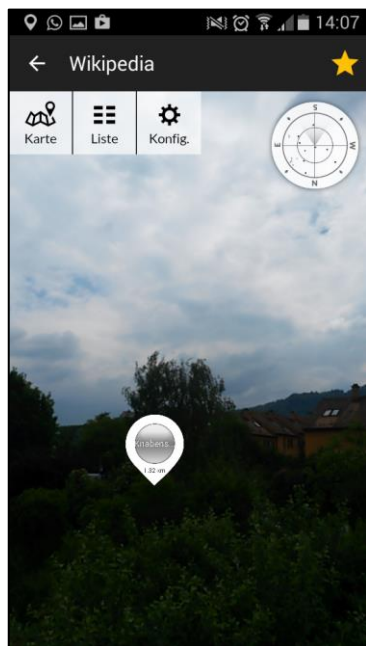


Abbildung 7: Wikitude Screenshot AR-Browser

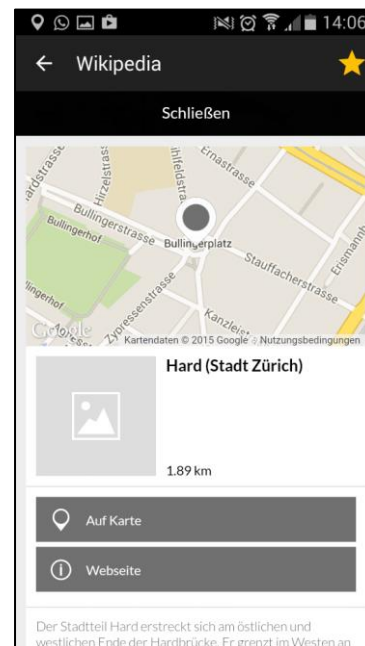


Abbildung 8: Wikitude Screenshot POI-Ansicht



Abbildung 9: Wikitude Navigation (Wikitude.com, 2015)

3.2 Generalisierung

Auf der Grundlage der verschiedenen Arten von AR-Anwendungen wurden die nachfolgenden Modelle zur Konzeptionierung des Prototypen ARwizard gebildet. Dabei sind die drei Typen von standortbezogenen AR-Anwendungen berücksichtigt worden:

- AR-Navigation
- AR-Browser mit POIs mit verschiedenen Informationstypen
- AR-Browser mit POIs von externen Ressourcen

3.2.1 Metamodell

Das Metamodell beschreibt eine übergeordnete Generalisierung der verschiedenen AR-Anwendungstypen, die aus der Anforderungsanalyse stammen.

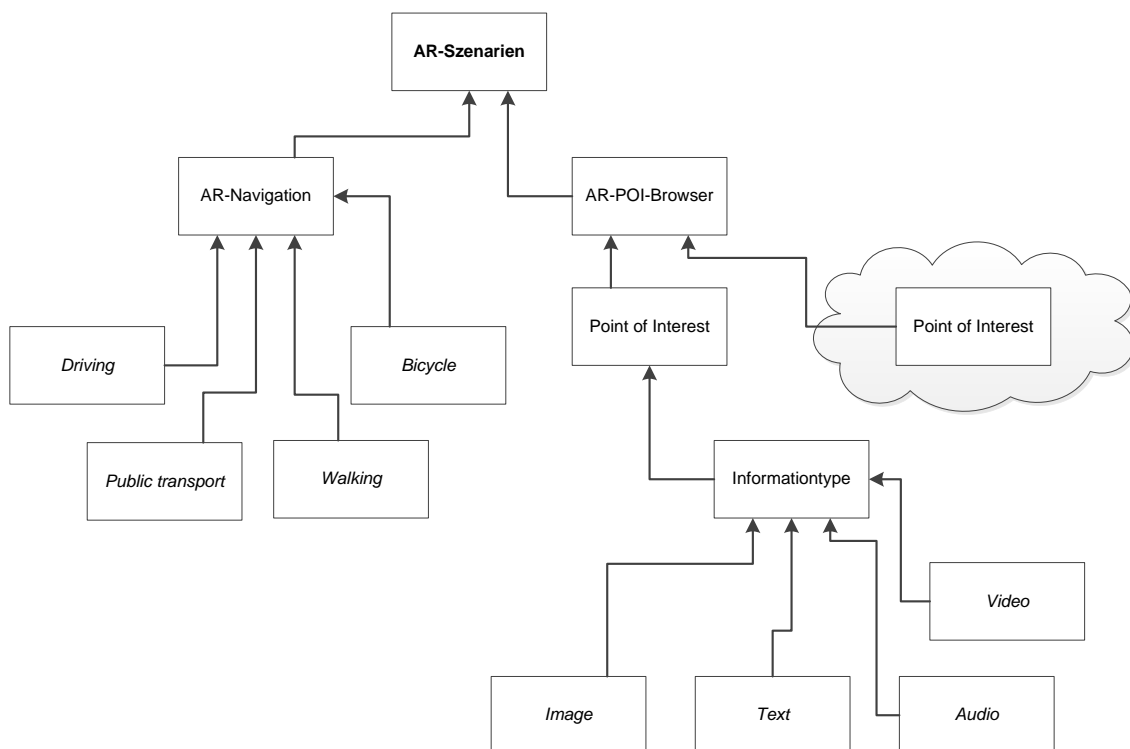


Abbildung 10: Metamodell AR-Anwendung

3.2.2 Entity-Relationship-Modell POI

Weiter wurde folgendes Entity-Relationship-Modell für ein POI gebildet. In diesem Model sind alle nötigen Attribute eines POI ersichtlich.

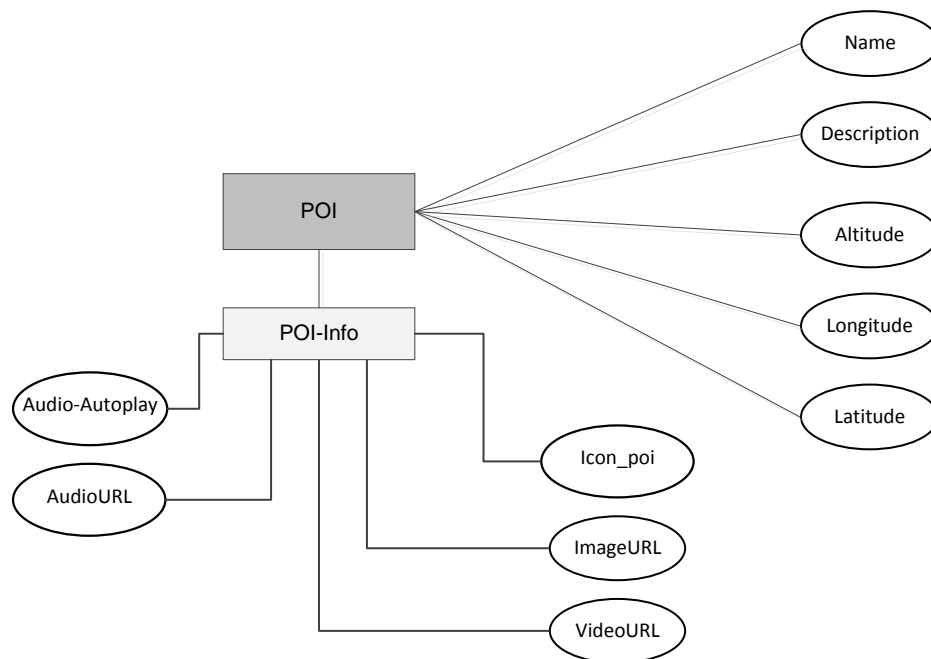


Abbildung 11: Entity-Relationship-Modell POI

Attribute	Description
Name	Der Name bezeichnet den POI. Im AR-Browser wird dieser im Kamerabild angezeigt, sobald der POI im Sichtfeld auftaucht.
Description	Die Beschreibung des POI wird im Description-Feld definiert und kann verschiedene Informationen über den POI stehen.
Altitude	Höher (m. ü. M.) vom POI
Latitude	Geografische Breite des POI (nördliche oder südliche Entfernung vom Äquator im Winkelmass)
Longitude	Geografische Länge des POI (östliche oder westliche Entfernung von Nullmeridian im Winkelmass)
Icon_poi	Das Icon_poi wird in Kombination mit dem Namen im AR-Browser dargestellt. Es erhöht die Übersichtlichkeit für den Anwender, da man verschiedene POIs grafisch kategorisieren kann. In Rahmen dieses Prototyps sind nur Icons von „Font Awesome“ oder „Ionicons“ möglich. Somit gehört in dieses Feld die Klassenbezeichnung des Icons.
Image_URL	Durch den Image_URL kann dem POI ein Bild hinzugefügt werden.
VideoURL	Durch den Video_URL kann dem POI ein Video hinzugefügt werden.
AudioURL	Durch den Audio_URL kann dem POI ein Musiktitel oder eine Tondatei hinzugefügt werden.
Audio-Autoplay	Je nach Auswahl (true oder false) wird beim Öffnen des POIs die hinterlegte Tondatei oder Musiktitel automatisch abgespielt. In der Touristenbranche macht dies durchaus Sinn, da der Anwender so direkt Informationen über den ausgewählten POI erhält.

Tabelle 4: Entity-Relationship-Modell POI

3.2.3 Schnittstellen für externe POIs

Aus dem POI-Modell wurde eine Schnittstellenspezifikation gebildet. Diese beschreibt ein JSON-String, in dem alle Informationen über einen POI enthalten sind.

```
{
  „name“:      „Paradeplatz, Zürich“,
  „description“: „Der Paradeplatz liegt in Zürich. Die Geschichte...“,
  „latitude“:   „47.369717“,
  „longitude“:  „8.538923000000068“,
  „altitude“:   „510“,
  „poi_info“: {
    „icon_poi“: „fa fa-anchor“,
    „video_url“: „http://www.youtube.com/embed/XsRuCIrzs“,
    „img_url“:  „“,
    „audio_url“: „“,
    „audio_autoplay“: „false“
  }
}
```

Listing 1: JSON Schnittstellenspezifikation für externe Quellen

3.3 Aufbau der Lösung „ARwizard“

Der ARwizard wird konzeptionell in zwei Bereiche aufgeteilt. Zum einen in den Administrationsbereich und zum anderen in den AR-Browser. Die Aufteilung ist in der Abbildung 12 ersichtlich.

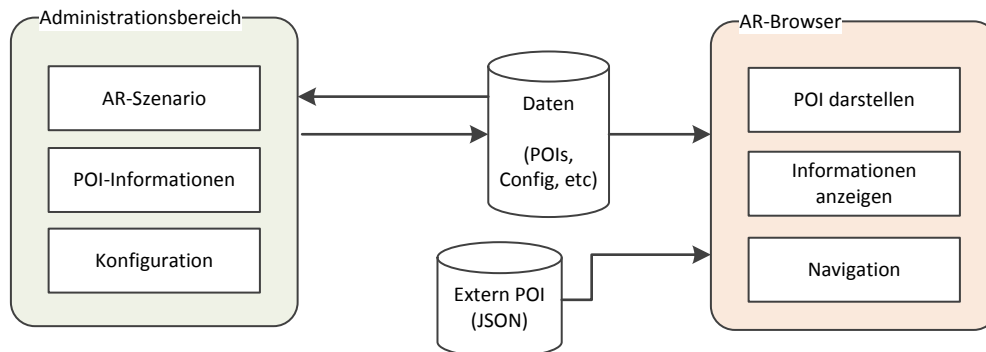


Abbildung 12: Bereiche des ARwizards

3.3.1 Administrationsbereich

Der Administrationsbereich soll die Möglichkeit bieten, sogenannte AR-Szenarien bilden zu können. Dabei soll die Möglichkeit geboten werden, POIs dem AR-Szenario zuzuordnen und bestimmte Funktionalitäten, wie Radar, Navigation usw. zu aktivieren oder zu deaktivieren. Auch soll die Darstellung individuell anpassbar sein.

3.3.2 ARwizard AR-Browser

Im AR-Browser des ARwizards werden die zuvor definierten AR-Szenarien lauffähig. Der AR-Browser muss modular aufgebaut sein, um alle möglichen Konfigurationen, welche im Administrationsbereich definiert werden, darstellen zu können.

Der Hauptbestandteil des AR-Browsers ist die Berechnungslogik. Sie bildet die Grundlage für das Anzeigen der POIs im Kamerabild bzw. der Routenführung im Navigationsmodus. Im folgenden Kapitel wird genauer auf die einzelnen Komponenten des AR-Browser eingegangen.

3.4 Komponenten AR-Browser

Der AR-Browser des ARwizards besteht aus verschiedenen Komponenten, die in diesem Kapitel genauer aufgezeigt werden. Wie in Abbildung 13 gezeigt, gibt es zum einen ein Controller-Bereich, der für die grundlegenden Berechnungen und den Datenaustausch verantwortlich ist und zum anderen den View-Bereich. Der View-Bereich beinhaltet alle Komponenten, die für die Darstellung und den Output im Web-Browser verantwortlich sind.

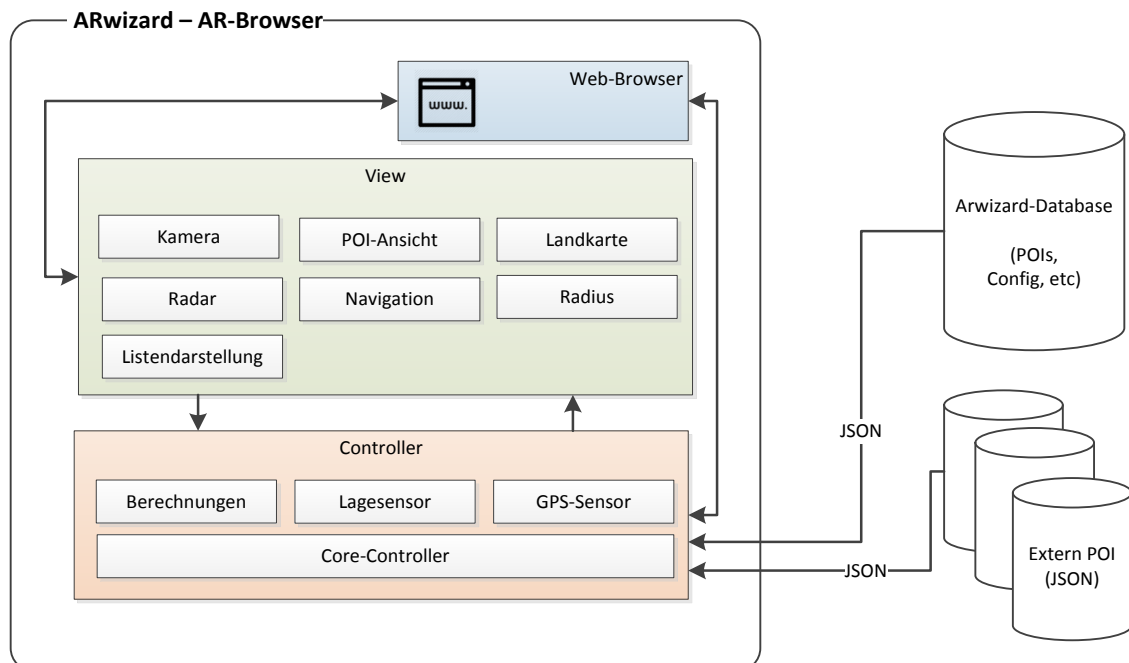


Abbildung 13: AR-Browser Bereich des ARwizards

Eine zentrale Komponente des Controller-Bereichs ist der Core-Controller. Dieser stellt das Zusammenspiel zwischen den einzelnen Komponenten dar. Dabei wird überwacht, dass alle Informationen und Daten für den reibungslosen Betrieb des AR-Browsers vorhanden sind. Um dies zu gewährleisten wurde ein Objekt gebildet, mit dem diese

Daten in der Laufzeit der Applikation verwaltet werden können. Die Zusammenhänge sind im nachfolgenden Objektdiagramm (Abbildung 14) ersichtlich. Dabei bildet das Objekt „arwizard“ die Ausgangslage und wird mit den verschiedenen Daten der einzelnen Komponenten erweitert. In der Laufzeit des AR-Browsers wird auf dieses Objekt zugegriffen.

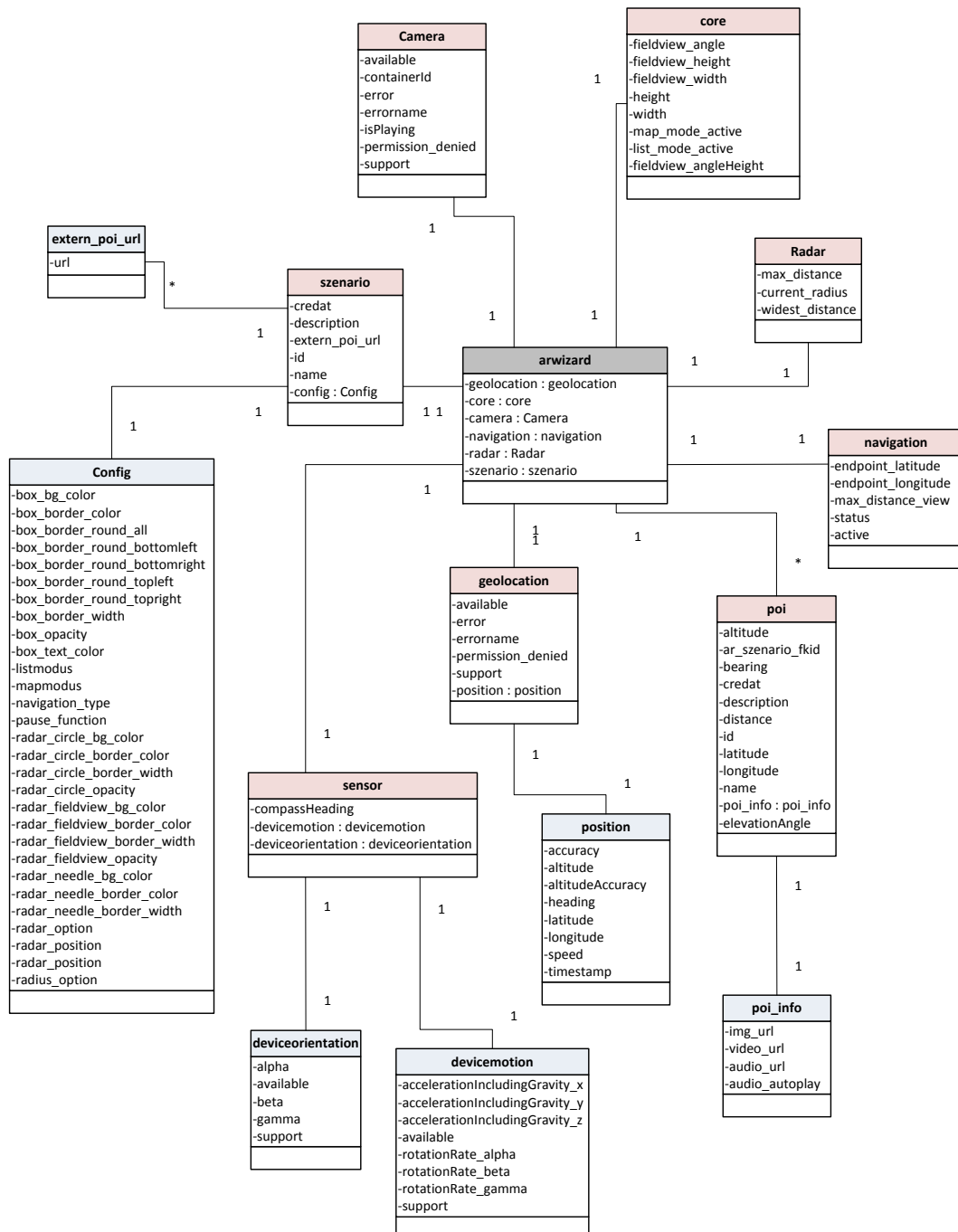


Abbildung 14: Datenobjekt "arwizard"

3.4.1 GPS-Sensor

Mit dem GPS-Sensor wird dem ARwizard der aktuelle Standort übermittelt. Dieser wird verarbeitet und dem Core-Controller weitergegeben. Um auf den Standort des Smartphone zugreifen zu können, wird die Geolocation API benutzt. Diese wurde im Zusammenhang mit HTML5 entwickelt und kann mit JavaScript angewendet werden. (selfhtml-wiki, 2014).

Der Standort wird benutzt um zu einem späteren Zeitpunkt die Distanzen und Peilungen zu den einzelnen POIs berechnen zu können. Massgebend für diese Berechnung ist die Information der geografischen Breite (Latitude) und der geografischen Länge (Longitude), welche durch die nachfolgenden Code-Beispiele ausgelesen werden kann:

```
function getLocation() {  
    var config  
  
    config = {  
        enableHighAccuracy: true,  
        maximumAge: 250,  
        timeout: 2000  
    };  
  
    if (navigator.geolocation) {  
        navigator.geolocation.getCurrentPosition(showPosition);  
        navigator.geolocation.watchPosition(getLocation_watch,  
        watchLocationError, config);  
    } else {  
        //wird nicht unterstützt  
    }  
    return 'ok';  
};  
  
function getLocation_watch(position) {  
    arwizard.geolocation.position = {  
        latitude: position.coords.latitude,  
        longitude: position.coords.longitude,  
        accuracy: position.coords.accuracy,  
        altitude: position.coords.altitude,  
        altitudeAccuracy: position.coords.altitudeAccuracy,  
        heading: position.coords.heading,  
        speed: position.coords.speed,  
        timestamp: position.timestamp  
    };  
  
    console.log("Watch GPS: " + position.coords.accuracy);  
};
```

Listing 2: Location-Informationen auslesen

3.4.2 Lagesensor

Ein weiterer Sensor, der für eine reibungslose Verwendung eines AR-Browser benutzt werden muss, ist der Lagesensor. Durch den Lagesensor wird die Ausrichtung des mobilen Endgeräts ermittelt. Diese wird beim ARwizard benötigt, um die Ausrichtung (Peilung) nach Norden berechnen zu können. Die Lage wird durch drei Winkelangaben beschrieben, Alpha, Beta und Gamma. Mit dem JavaScript-Event „deviceorientation“ kann auf diese zugegriffen werden. Im Normalfall beschreibt dabei der Alpha-Wert die Ausrichtung nach Norden (engl. Compass-Heading), wenn das mobile Endgerät flach auf einem Tisch liegt.

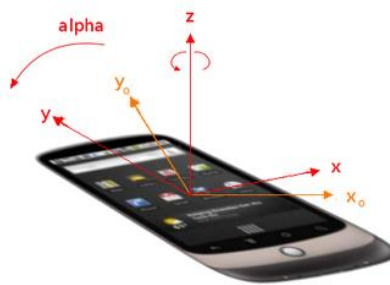


Abbildung 15: Ausrichtung des mobilen Endgeräts (W3c, 2014)

Nachfolgend ist der JavaScript-Code dargestellt, mit dem die Ausrichtung nach Norden berechnet werden kann, mit der Berücksichtigung, dass das mobile Endgerät nicht nur flach auf dem Tisch liegt, sondern auch aufrecht in der Hand gehalten wird (richt, 2014; W3c, 2014).

```
function compassHeading(alpha, beta, gamma) {
    // Convert degrees to radians
    var alphaRad = alpha * (Math.PI / 180);
    var betaRad = beta * (Math.PI / 180);
    var gammaRad = gamma * (Math.PI / 180);

    // Calculate equation components
    var CA = Math.cos(alphaRad);
    var SA = Math.sin(alphaRad);
    var CB = Math.cos(betaRad);
    var SB = Math.sin(betaRad);
    var CG = Math.cos(gammaRad);
    var SG = Math.sin(gammaRad);

    // Calculate A, B, C rotation components
    var rA = - CA * SG - SA * SB * CG;
    var rB = - SA * SG + CA * SB * CG;
    var rC = - CB * CG;

    // Calculate compass heading
    var compassHeading = Math.atan(rA / rB);

    // Convert from half unit circle to whole unit circle
    if(rB < 0) {
        compassHeading += Math.PI;
    } else if(rA < 0) {
        compassHeading += 2 * Math.PI;
    }
}
```

```
// Convert radians to degrees
compassHeading *= 180 / Math.PI;

return compassHeading;
};
```

Listing 3: Compass-Heading berechnen (richt, 2014; W3c, 2014)

3.4.3 Berechnungen

Bei einer standortbezogenen AR-Anwendung sind Berechnungen ein zentraler Punkt. Zum einen muss die Distanz zwischen dem aktuellen Standort des Anwenders zum POI berechnet werden. Zum anderen muss die Peilung (engl. Bearing) berechnet werden, um herauszufinden in welcher geografischen Richtung sich der POI befindet. Diese Informationen werden für eine korrekte Positionierung im Kamerabild benötigt (Geiger, 2012; kowoma.de, 2007).

3.4.3.1 Berechnung der Distanz

Für die Berechnung der Distanz kann gemäss Geiger (2012) die Harversine-Formel verwendet werden. Mit dieser Formel wird die Distanz von zwei Punkten auf der Erdoberfläche berechnet werden (Sinnott, 1984).

$$d = r * \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) * \cos(\varphi_2) * \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

Abbildung 16: Haversine Formel (Sinnott, 1984)

Die Parameter φ beschreibt dabei die geografische Breite und λ die geografische Länge eines Punktes. r beschreibt dabei den Radius der Sphäre (ca. 6371 km). Zu beachten ist dabei, dass die geografische Breite und Länge in Radiant vorliegen müssen (Geiger 2012). Mit der Abbildung 17 wird der Zusammenhang zwischen den Koordinaten veranschaulicht.

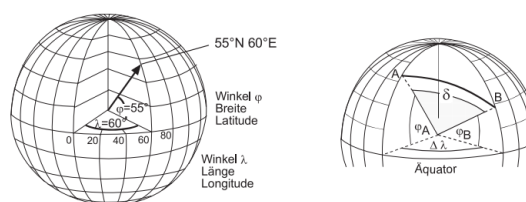


Abbildung 17: Geographisches Koordinatensystem (de Lange, 2013)

```
function getDistanceFromLatLonInKm(lat1,lon1,lat2,lon2) {
  //Haversine-Distance
  var toRadians = Math.PI/180.0;
  var R = 6371; // Radius of the earth in km

  var dLat = toRadians*(lat2-lat1); // deg2rad below
  var dLon = toRadians*(lon2-lon1);
  var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
    Math.cos(deg2rad(lat1)) * Math.cos(deg2rad(lat2)) *
    Math.sin(dLon/2) * Math.sin(dLon/2);

  var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
  var d = R * c; // Distance in km
  d = Math.round(d*1000)/1000;
  return d;
};
```

Listing 4: JavaScript-Code Distanzberechnung

3.4.3.2 Berechnung der Peilung

Um zu bestimmen, ob ein POI im Sichtfeld bzw. in der Richtung des Benutzers liegt, benötigt man die untenstehende Formel (Abbildung 18). Dabei wird die Peilung vom aktuellen Standort zum POI berechnet (Veness, 2015) & (kowoma.de, 2007).

$$bearing = \text{atan2} \left(\begin{array}{c} \sin\left(\frac{\lambda_2 - \lambda_1}{2}\right) * \cos(\varphi_1) , \\ \cos(\varphi_1) * \cos(\varphi_2) - \sin(\varphi_1) * \cos(\varphi_2) * \cos\left(\frac{\lambda_2 - \lambda_1}{2}\right) \end{array} \right)$$

Abbildung 18: Formel für die Berechnung der Peilung (Veness, 2015)

Der Ausgabewert entspricht der Peilung relativ zum Nordpol (engl. Bearing). In Abbildung 19 wird dies veranschaulicht. Wie auch bei der Berechnung der Distanz müssen die geografische Breite φ und Länge λ in Radiant angegeben werden (Veness, 2015) & (kowoma.de, 2007).

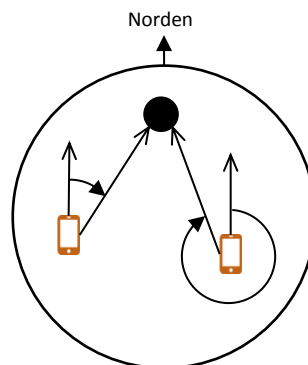


Abbildung 19: Darstellung Peilung (Geiger, 2012)

```
function getBearing(lat1,lon1,lat2,lon2){
// Returns the (initial) bearing from 'this' point to destination
// Bearing: Richtung (Norden = 0) am Compass nach

var toRadians = Math.PI/180.0;
var toDegrees = 180/Math.PI;

var φ1 = lat1 * toRadians;
var λ1 = lon1 * toRadians;
var φ2 = lat2 * toRadians;
var λ2 = lon2 * toRadians;

var y = Math.sin(λ2-λ1) * Math.cos(φ2);
var x = Math.cos(φ1)*Math.sin(φ2) -
      Math.sin(φ1)*Math.cos(φ2)*Math.cos(λ2-λ1);

var bearing = Math.atan2(y, x);
bearing = bearing * toDegrees;
bearing = (bearing+360) % 360;

return bearing;
};
```

Listing 5: Berechnung der Peilung (Veness, 2015)

3.4.3.3 Berechnung des Höhenwinkels

Eine weitere Berechnung dient der Ermittlung des Höhenwinkels (engl. elevation angle). Dadurch lässt sich feststellen, ob sich der POI im Sichtfeld des Benutzers befindet. Dabei geht es um die horizontale Ausrichtung. Man möchte erreichen, dass die Position der POIs im Kamerabild möglichst genau ist. Ohne die Berücksichtigung der horizontalen Ausrichtung würden auch POIs angezeigt, welche sich nicht im Sichtfeld des Benutzers befinden, sondern darüber oder darunter. Um dies zu verhindern, muss der genannte Höhenwinkel berechnet und mit der Neigung des mobilen Endgerätes verglichen werden. Die Berechnung des Höhenwinkels wird mit der untenstehenden Formel in Abbildung 20 durchgeführt.

$$elevationAngle = \alpha = \arctan\left(\frac{altitude_1 - altitude_2}{distance}\right) * \frac{180}{\pi}$$

Abbildung 20: Formel für die Berechnung des Höhenwinkels (Geiger, 2012)

Dabei stellt der Parameter *altitude₁* die Höhe des POI und *altitude₂* die Höhe des aktuellen Standorts des Benutzers dar. Mit diesen beiden Angaben wird der Höhenunterschied berechnet. Zusätzlich wird die Distanz dieser beiden Orte benötigt. Die Distanz stammt aus der Formel der Abbildung 16. In der Abbildung 21 wird dies noch genauer veranschaulicht.

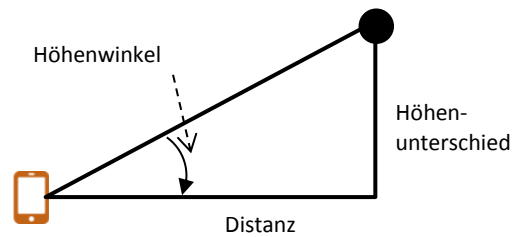


Abbildung 21: Darstellung Höhenwinkel (Geiger, 2012)

```
function getElevationAngle(distanceInKm, startAltitude, endAltitude){  
    var distanceInMeter    = distanceInKm * 1000;  
    var dAltitude          = endAltitude - startAltitude;  
  
    var x                  = (dAltitude) / (distanceInMeter);  
    var elevationAngle     = (180/Math.PI) * Math.atan(x);  
    var elevationAngleRnd  = Math.round(elevationAngle*1000)/1000;  
    return elevationAngle;  
};
```

Listing 6: Berechnung des Höhenwinkels

3.4.4 Kamera

Unter der Komponente Kamera wird das Livebild, das dem Anwender gezeigt wird, verstanden. Im ARwizard wird diese Komponente als Teil der View-Komponente angesehen. Die Aufgabe besteht darin, dass auf die Kamera des mobilen Endgerätes zugegriffen wird und das Bild an die View weitergegeben wird.

Auf die Kamera wird mittels der MediaStream API und der JavaScript-Methode „navigator.getUserMedia“ zugegriffen. Dadurch kann das Kamerabild in einem Video-Element in einem Web-Browser wiedergegeben werden. Das angezeigte Bild wird anschliessend durch die anderen View-Komponenten mit Informationen erweitert.

```
navigator.getUserMedia = navigator.getUserMedia ||
                        navigator.webkitGetUserMedia ||
                        navigator.mozGetUserMedia;

if (navigator.getUserMedia) {
    navigator.getUserMedia({ audio: true,
                           video: { width: 1280,
                                    height: 720 } },
        function(stream) {
            var video = document.querySelector('video');
            video.src = window.URL.createObjectURL(stream);
            video.onloadedmetadata = function() {
                video.play();
            };
        },
        function(err) {
            console.log("The following error occurred: " + err.name);
        }
    );
} else {
    console.log("getUserMedia not supported");
}
```

Listing 7: Zugriff auf Kamera „navigator.getUserMedia“

3.4.5 Navigation

Ein weiterer Bestandteil des AR-Browsers ist die Navigation. Für die Navigationsfunktion im ARwizard wird der Routendienst „Directions“ von der Google Maps JavaScript API V3 verwendet. Der Vorteil dabei ist, dass Google Maps bereits eine vordefinierte Navigations-Engine bereitstellt und diese per JavaScript abgefragt werden kann. Die Route wird dabei in einzelne Schritte aufgeteilt und kann direkt verwendet werden. Zudem bietet Google verschiedene Reisemodi an, die in der Tabelle 5 ersichtlich sind.

Reisemodi	Beschreibung
google.maps.TravelMode.DRIVING	(Standardeinstellung) gibt die Standardroute unter Verwendung des Straßennetzes an
google.maps.TravelMode.BICYCLING	dient zum Anfordern von Fahrradrouten unter Verwendung von Fahrradwegen und bevorzugten Straßen
google.maps.TravelMode.TRANSIT	dient zum Anfordern von Routen unter Nutzung öffentlicher Verkehrsmittel
google.maps.TravelMode.WALKING	dient zum Anfordern von Routen für Fußgänger, die über Fußgängerwege und Bürgersteige führen

Tabelle 5: Reisemodi nach Google Maps Directions (Google, 2015)

Eine Schwierigkeit der Navigationsfunktion in einer AR-Anwendung ist die Darstellung der Route im Kamerabild. Zudem stellt sich die Frage, wie weit die Route angezeigt werden soll. Im ARwizard wurde zur Navigation eine Kombination von Bild- und Textinformationen gewählt. Es werden zum einen Textinformationen der Google Maps Directions API angegeben. Zum anderen wird die Richtung mit einer pfeilartigen Fläche dargestellt. Diese Darstellung beruht auf der Arbeit von Kluge (2009) bezüglich Fussgängernavigation, in der die einzelnen Varianten aufgezeigt werden. Dabei werden jeweils nur die Routendaten für die nächsten 30 Meter der Route im Kamerabild dargestellt. Somit kann sichergestellt werden, dass der Strassenverlauf möglichst realitätsnah und präzise wiedergegeben wird und zum Beispiel eine scharfe Kurve auch als solche im Kamerabild dargestellt wird.

3.4.6 Radius

Die Radius-Funktion wird benutzt, um die Anzahl der anzuzeigenden POIs einzuschränken oder zu erweitern. Durch die Angabe der maximalen Distanz werden nur diejenigen POIs angezeigt, die innerhalb dieser Distanz liegen. Diese Funktionalität kann durchaus wichtig sein, falls ein Szenario mit vielen POIs gebildet wird. Würden alle POIs im Kamerabild angezeigt und mit unzähligen Informationen erweitert, wäre die Übersichtlichkeit nicht mehr gegeben. Dem kann mit dem Radius entgegengewirkt werden, indem nur die im Radius liegenden POIs angezeigt werden.

3.4.7 Radar

Da im eigentlichen Kamerabild nur die POIs angezeigt werden, die sich im Sichtfeld befinden, ist das Radar eine nützliche Funktion, um auch das Umfeld anzeigen zu können. Durch das Radar kann der Anwender erkennen, wo Norden ist und in welche Richtung er sich drehen muss, damit weitere POIs ins Sichtfeld kommen.

3.4.8 Listendarstellung

Mit der Listendarstellung werden die POIs in einer Liste dargestellt. Alle POIs sind dabei nach Entfernung sortiert und der Anwender kann diese direkt auswählen. Der Vorteil ist, dass der Anwender nicht zwingend in die Richtung der POIs schauen muss, um ihn auswählen zu können, da es über die Listendarstellung effizienter geht.

3.4.9 Landkarte

Die Landkarten-Ansicht ermöglicht es, alle POIs auf einer Karte anzuzeigen. Mit dieser Ansicht wird die AR-Anwendung zu einer Karten-App, wie man sie von Google-Maps kennt. Zur Orientierung wird zusätzlich die aktuelle Position des Anwenders auf der Karte angezeigt. Wie schon bei der Navigation wird hierbei auf die Dienste von Google zurückgegriffen.

3.4.10 POI-Ansicht

Die POI-Ansicht sorgt dafür, dass im AR-Browser die POIs angezeigt werden, welche sich im Sichtfeld des Anwenders befinden. Die Darstellung der POIs wird durch den Benutzer im Administrationsbereich vorgegeben. Die Standardanzeige ist ein Rechteck. Damit diese Komponente funktioniert, müssen GPS-Sensor, Lagesensor und Berechnungen funktionieren sowie der Zugriff darauf sichergestellt sein. Denn nur mit diesen Komponenten kann gewährleistet werden, dass die POIs am richtigen Ort im Kamerabild platziert werden.

3.5 Datenbank-Modell

Nachfolgend wird das Datenbankmodell aufgezeigt. Wie in Abbildung 22 erkennbar, ist die Datenbankstruktur einfach gehalten. In der Tabelle „ar_szenario“ werden die einzelnen AR-Szenarien gespeichert. Zu jedem AR-Szenario werden in der Tabelle „ar_poi“ die dazugehörigen POIs gespeichert.

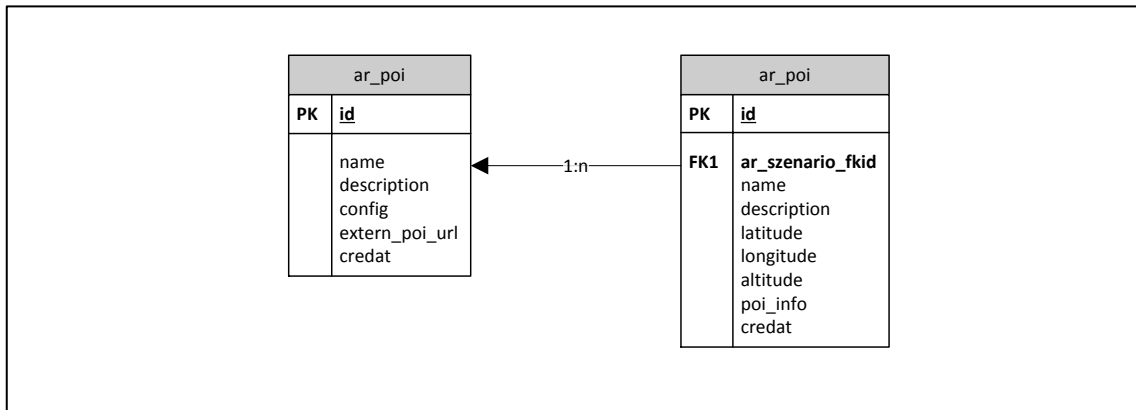


Abbildung 22: Datenbank-Modell ARwizard

3.5.1.1 Tabelle: ar_poi

Mit der Tabelle 6 werden die Spezifikation der Datenbanktabelle „ar_poi“ dargelegt.

Feld	Typ	Description	Beispiel
id	int(11)	Primary Key, AUTO_INCREMENT	5
ar_szenario_fkid	Int(11)	Foreign Key zu ar_szenario.id	1
name	Text	Der Name bezeichnet den POI. Im AR-Browser wird dieser im Kamerabild angezeigt, sobald der POI im Sichtfeld auftaucht.	„Paradeplatz, Zürich“
description	text	Die Beschreibung des POI wird im Description-Feld gemacht. In der Description können verschiedene Informationen über den POI stehen.	„Paradeplatz ist in Zürich ein wichtiger Platz...“
altitude	decimal(11,5)	Höher über Meer vom POI in Meter	500
latitude	decimal(18,15)	Breitengrades des POI (nördliche bzw. südliche Entfernung vom Äquator im Winkelmass)	47.396490013933416
longitude	decimal(18,15)	Längengrad des POI (östliche oder westliche Entfernung von Nullmeridians)	8.513031005859375
poi_info	longtext	JSON-String mit den Angaben zu Image-URL, Audio-URL, Video-URL, Audio-Autoplay-Option und Icon	{ "icon_poi": "fa fa-anchor", "video_url": "", "img_url": "", "audio_url": "", "audio_autoplay": "false" }
Credat	timestamp	CURRENT_TIMESTAMP	2015-04-18 17:40:56

Tabelle 6: Datenbanktabelle „ar_poi“

3.5.1.2 Tabelle: ar_szenario

Die Spezifikation der Tabelle „ar_szenario“ ist in der nachfolgenden Tabelle 7 ersichtlich.

Feld	Typ	Description	Beispiel
ID	int(11)	Primary Key, AUTO_INCREMENT	1
Name	text	Der Name des Szenario	„Reiseführer Zürich“
Description	text	Die Beschreibung des Szenarios	„Diese Szenario ist ein Reiseführer für“
Extern_poi_url	longtext	JSON-String, welches ein Array beschreibt, dass verschiedene Externe-URL beinhaltet.	{ "0":"domain.ch/json-poi", "1":"zuerich.ch/poi" }
Config	Longtext	In diesem Feld wird eine JSON-String gespeichert. Der JSON-String beinhaltet verschiedene Konfigurationen für das Szenario.	{ "navigation_type":"all radar_option":"show radar_position":"topleft radar_radius":"43 radar_circle_bg_color":"rgba(220,253,220,1) radar_circle_border_color":"rgba(131,200,252,1) radar_circle_border_width":"6 radar_circle_opacity":"0.4 radar_fieldview_bg_color":"rgba(64,91,111,1) radar_fieldview_border_color":"rgba(220,253,220,1) radar_fieldview_border_width":"6 radar_fieldview_opacity":"0.6 radar_needle_bg_color":"rgba(220,253,220,1) radar_needle_border_color":"rgba(220,253,220,1) radar_needle_border_width":"5 box_text_color":"rgba(224,17,25,1) box_bg_color":"rgba(220,253,220,1) box_opacity":"0.9 box_border_color":"rgba(224,17,25,1) box_border_width":"7 box_border_round_all":"13 box_border_round_topleft":"13 box_border_round_topright":"13 box_border_round_bottomleft":"13 box_border_round_bottomright":"13 radius_option":"enable mapmodus":"enable listmodus":"enable pause_function":"enable" }
Creat	timestamp	CURRENT_TIMESTAMP	2015-04-18 17:40:56

Tabelle 7: Datenbanktabelle „ar_szenario“

4 Vorstellung der Applikation „ARwizard“

In diesem Abschnitt der Arbeit werden die zuvor konzeptuell dargelegten Funktionalitäten der Applikation ARwizard mit Screenshots dargestellt und erläutert.

4.1 Administrationsbereich

4.1.1 Dashboard

Die Startseite des Administrationsbereiches ist das Dashboard. Über dieses können die bereits erstellten AR-Szenarios verwaltet werden oder ein neues AR-Szenario erstellt werden. Zudem kann man mittels Barcode-Reader bereits vorhandene AR-Szenarios direkt auf dem mobilen Endgerät ausführen und somit den AR-Browser starten.

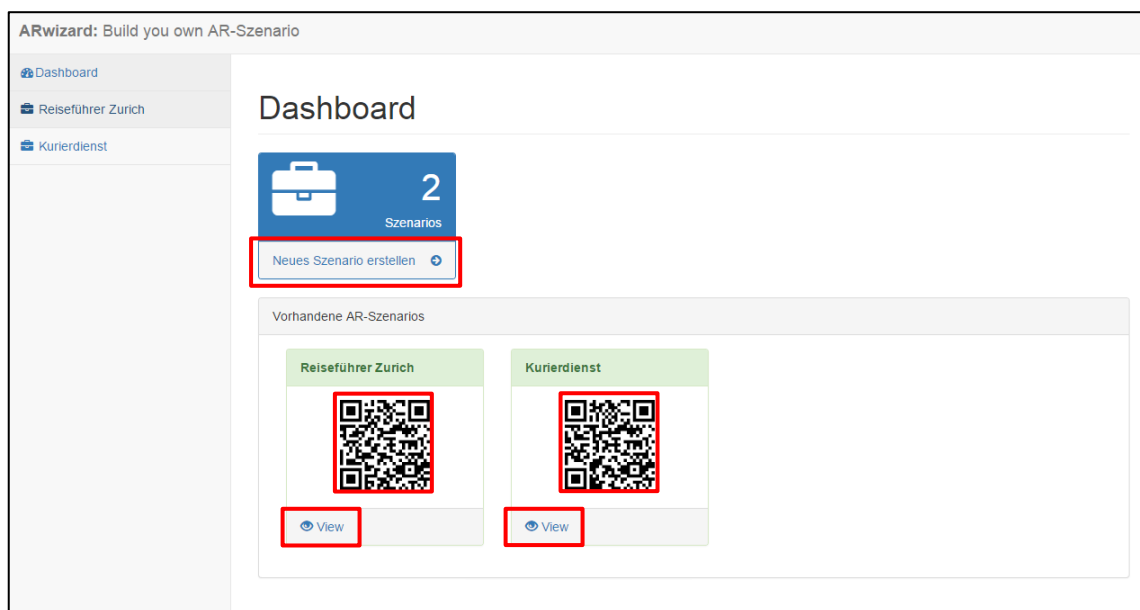


Abbildung 23: Dashboard ARwizard

4.1.2 AR-Szenario erstellen

Mittels dem Menüpunkt [Neues Szenario erstellen] auf dem Dashboard (Abbildung 23) kann man ein neues AR-Szenario erstellen. Es erscheint eine neue Seite mit einem Formular, welches aus zwei Teilschritten besteht. In Abbildung 24 ist der erste Teil ersichtlich, hier kann der Name und die Beschreibung des AR-Szenarios eingegeben werden. Zugleich muss ausgewählt werden, welche Navigationsart in diesem AR-Szenario verwendet werden soll.

The screenshot shows the 'ARwizard: Build you own AR-Szenario' interface. On the left is a sidebar with navigation links: 'Dashboard', 'Reiseführer Zurich', 'Kurierdienst', and 'Edit/Add Szenario' (highlighted in red). The main area is titled 'New AR-Szenario' with a briefcase icon. Below the title, it shows 'AR-Szenario-ID: NEW' and 'AR-Szenario-Name: NEW'. A progress bar indicates 'Step 1' (active) and 'Step 2'. The form contains three fields: 'AR-Szenario-Name' with the value 'Reiseführer Bern', 'Beschreibung' with the text 'Dieses AR-Szenario beschreibt eine AR-Anwendung für Touristen in Bern.', and 'AR-Navigation-Mode' with a dropdown menu set to 'All types allowed'. A blue 'Next' button is highlighted with a red rectangle in the bottom right corner.

Abbildung 24: Neues AR-Szenario erstellen - Schritt 1

Nachdem der Button [Next] gedrückt wird, wird der zweite Teil des Formulars angezeigt. Wie in der Abbildung 25 ersichtlich, können im oberen Bereich die Darstellungsoptionen ausgewählt werden. Über die Slider und Farbauswahlen kann das Radar und die POI-Ansicht individuell angepasst werden. Auch können die einzelnen Funktionalitäten, welche in Kapitel 3.4 beschrieben wurden, in dem AR-Szenario aktiviert oder deaktiviert werden. Bei jeder Änderung der Einstellungen wird im unteren Bereich der Seite eine Vorschau angezeigt, damit der Ersteller direkt erkennt, wie sich die Oberfläche des AR-Browser verändert.

ARwizard: Build you own AR-Szenario

Dashboard
Reiseführer Zurich
Kunerdienst
Stadtplan
Edit/Add Szenario

Edit AR-Szenario

AR-Szenario-ID: 38
AR-Szenario-Name: Stadtplan

Step 1 Step 2

Radar

Show Radar

Radar Position

Top-left

Radar Radius

43

Radar-Style Circle (Grossefläche)

Background-Color

Border-Color

Border-Width

6

Opacity

0.4

Radar-Style Fieldview (Dreieck)

Background-Color

Border-Color

Border-Width

6

Opacity

0.6

Radar-North-Needle

Background-Color

Border-Color

Border-Width

5

Pop-Up-Box-POI

Text-Color

Box-Background-Color

Box-Opacity

0.9

Border-Color

Border-Width

7

Border-Rounding-All

13

Border-Rounding-Top-Left

13

Border-Rounding-Top-Right

13

Border-Rounding-Bottom-Left

13

Border-Rounding-Bottom-Right

13

Radius-Mode

Enable

Map-Mode

Enable

Listmodus

Enable

Pause-function

Allowed

Live View:

AR-Szenario save

Abbildung 25: Neues AR-Szenario erstellen - Schritt 2

Mit dem Button [AR-Szenario save] wird das AR-Szenario gespeichert und die Verwaltungsseite des neuen AR-Szenarios wird geöffnet (Abbildung 26). Auf dieser Seite können nun POI hinzugefügt oder die Einstellungen wieder geändert werden. Dies wird in den nächsten Abschnitten detailliert beschrieben. In dieser Ansicht ist zusätzlich der Link zum AR-Browser ersichtlich, mit dem das AR-Szenario gestartet wird. Um den

Zugang zu erleichtern, kann der Link auch direkt über das Scannen des Barcodes auf dem mobilen Endgerät geöffnet werden. Der Barcode kann zudem als Bild heruntergeladen und bei Bedarf den Anwendern zur Verfügung gestellt werden (Bsp.: Flyer drucken, Twitter, Facebook, etc.).

ARwizard: Build you own AR-Szenario

Dashboard
Reiseführer Zurich
Kurierdienst
Reiseführer Bern

Reiseführer Bern

Szenario #38

config

delete

Name:

Reiseführer Bern

Creation-Date:


2015-05-17 16:06:08

Description:

Dieses AR-Szenario beschreibt eine AR-Anwendung für Touristen in Bern.

Link to AR-Szenario:

<http://localhost/master/arwizard/webapp/?sid=38>



Download

Point of Interest

0 results

Show

10

entries

Search:

ID	Name	Latitude	Longitude	Actions
No data available in table				

Showing 0 to 0 of 0 entries

Previous

Next

Add new POI

Extern Source with POI

Add Source

Do you want to include an extern Source with POI?! Put your url from the json subject here...

Abbildung 26: Verwaltungsseite AR-Szenario

4.1.3 AR-Szenario löschen

Durch die Auswahl von [delete] in der Verwaltungsseite eines AR-Szenarios wird der Löschvorgang gestartet (Abbildung 27) und muss gemäss Abbildung 28 zusätzlich bestätigt werden.

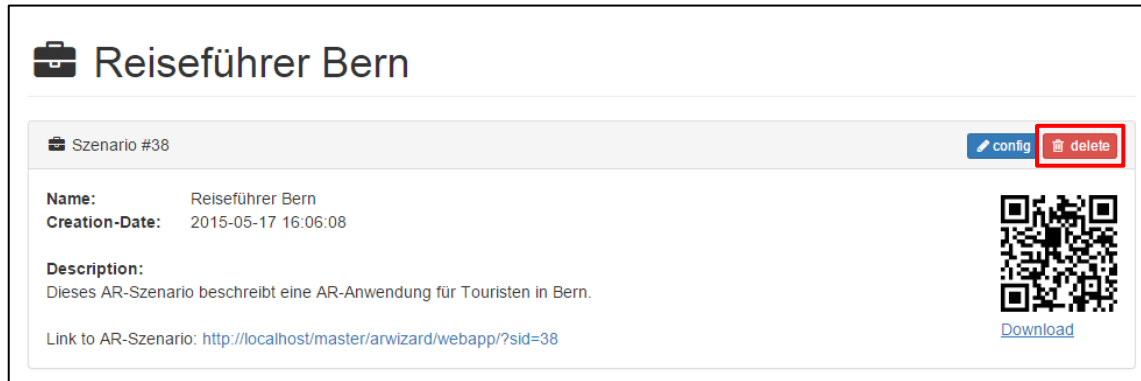


Abbildung 27: AR-Szenario löschen



Abbildung 28: AR-Szenario löschen bestätigen

4.1.4 AR-Szenario bearbeiten

Mit dem Button [config] wird das Formular für die Bearbeitung geladen. Es handelt sich um dasselbe Formular wie bei der Erstellung eines neues AR-Szenarios (vgl. 4.1.2).

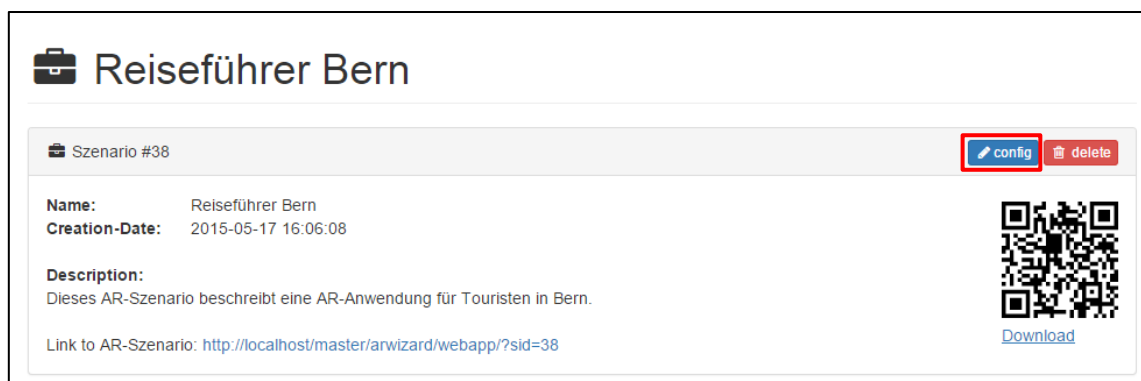
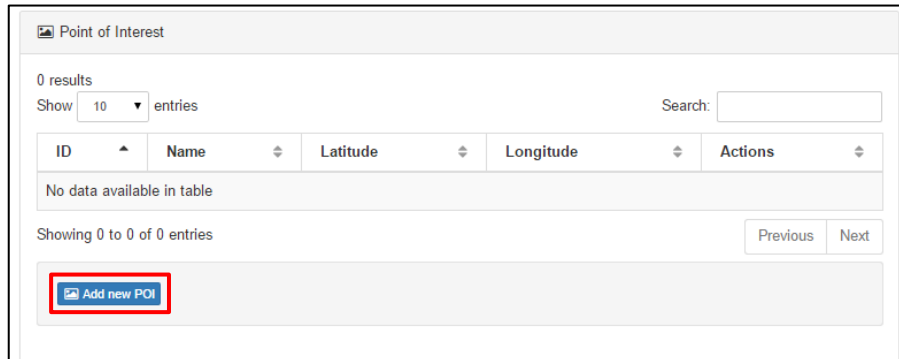


Abbildung 29: AR-Szenario bearbeiten

4.1.5 POI hinzufügen

Um einen POI zu einem AR-Szenario hinzufügen zu können, muss bereits ein AR-Szenario erstellt worden sein. Die Ausgangslage dabei ist die Verwaltungsseite des AR-Szenarios.



Point of Interest

0 results

Show 10 entries

Search:

ID	Name	Latitude	Longitude	Actions
No data available in table				

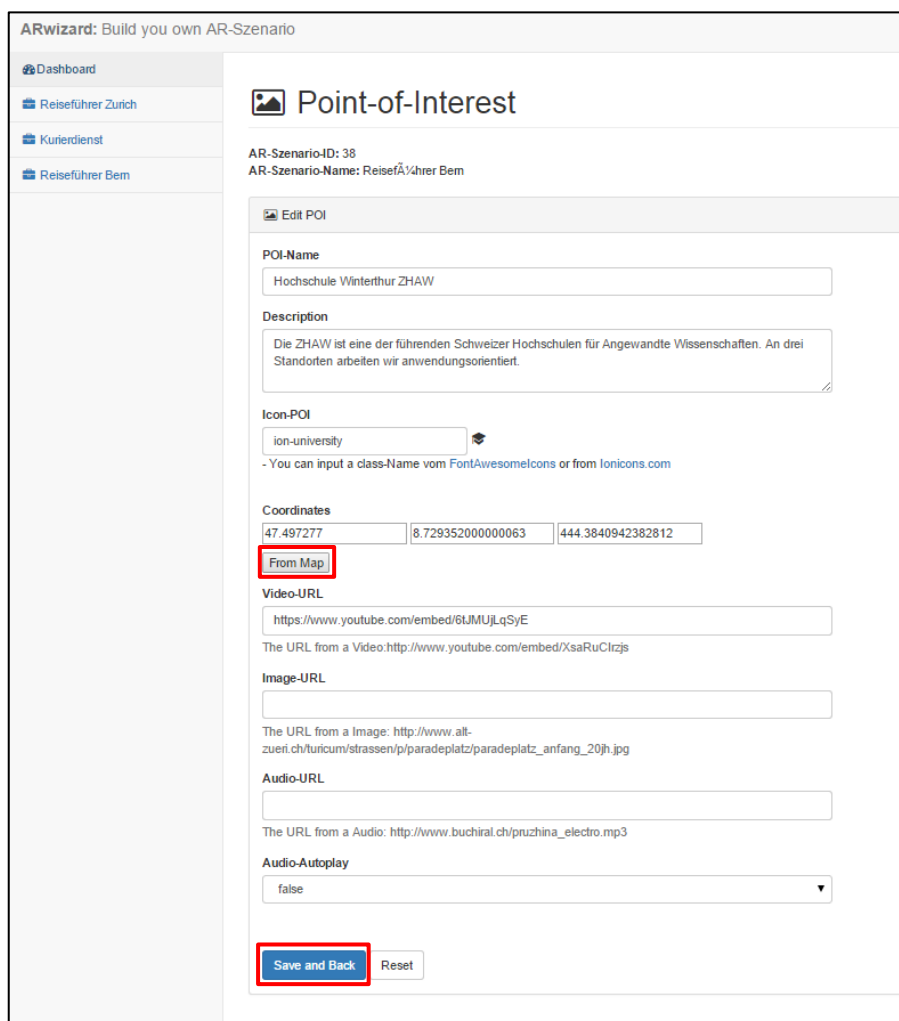
Showing 0 to 0 of 0 entries

Previous Next

Add new POI

Abbildung 30: Neuer POI hinzufügen [Add new POI]

Durch die Auswahl von [Add new POI] wird das Formular für einen neuen POI angezeigt.



ARwizard: Build your own AR-Szenario

Dashboard

Reiseführer Zurich

Kurierdienst

Reiseführer Bern

Point-of-Interest

AR-Szenario-ID: 38
AR-Szenario-Name: Reiseführer Bern

Edit POI

POI-Name

Hochschule Winterthur ZHAW

Description

Die ZHAW ist eine der führenden Schweizer Hochschulen für Angewandte Wissenschaften. An drei Standorten arbeiten wir anwendungsorientiert.

Icon-POI

ion-university

- You can input a class-Name vom FontAwesomelcons or from Ionicons.com

Coordinates

47.497277 8.729352000000063 444.3840942382812

From Map

Video-URL

https://www.youtube.com/embed/6tJMUJLqSyE

The URL from a Video: http://www.youtube.com/embed/XsaRuClrzjs

Image-URL

The URL from a Image: http://www.alt-zueri.ch/turicum/strassen/p/paradeplatz/paradeplatz_anfang_20jh.jpg

Audio-URL

The URL from a Audio: http://www.buchiral.ch/pruzhina_electro.mp3

Audio-Autoplay

false

Save and Back Reset

Abbildung 31: Neuer POI erfassen

Nun können die Informationen zum POI eingegeben werden. Über den Button [From Map] können die POIs zusätzlich mittels Google Maps gesucht und ausgewählt werden. Dabei werden durch die Google Maps API automatisch die Koordinaten abgefragt.

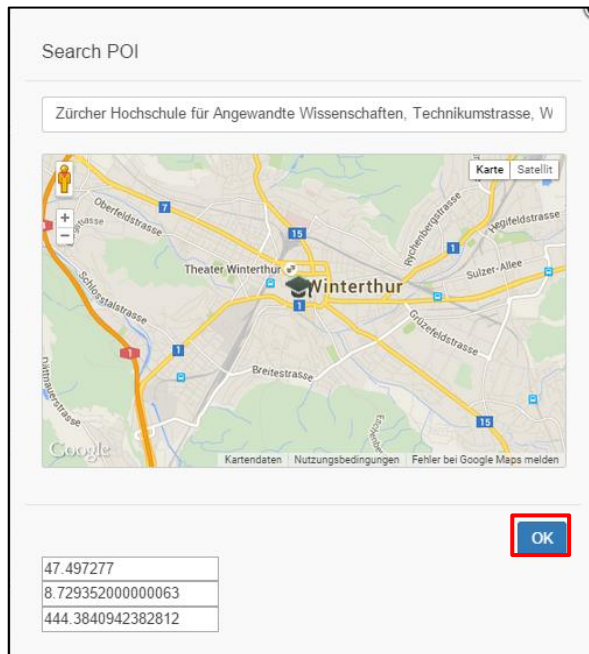


Abbildung 32: POI von Google Maps auswählen

Durch die Bestätigung mit [OK] und [Save and Back] wird der POI gespeichert und erscheint in der Verwaltungsseite des AR-Szenarios.

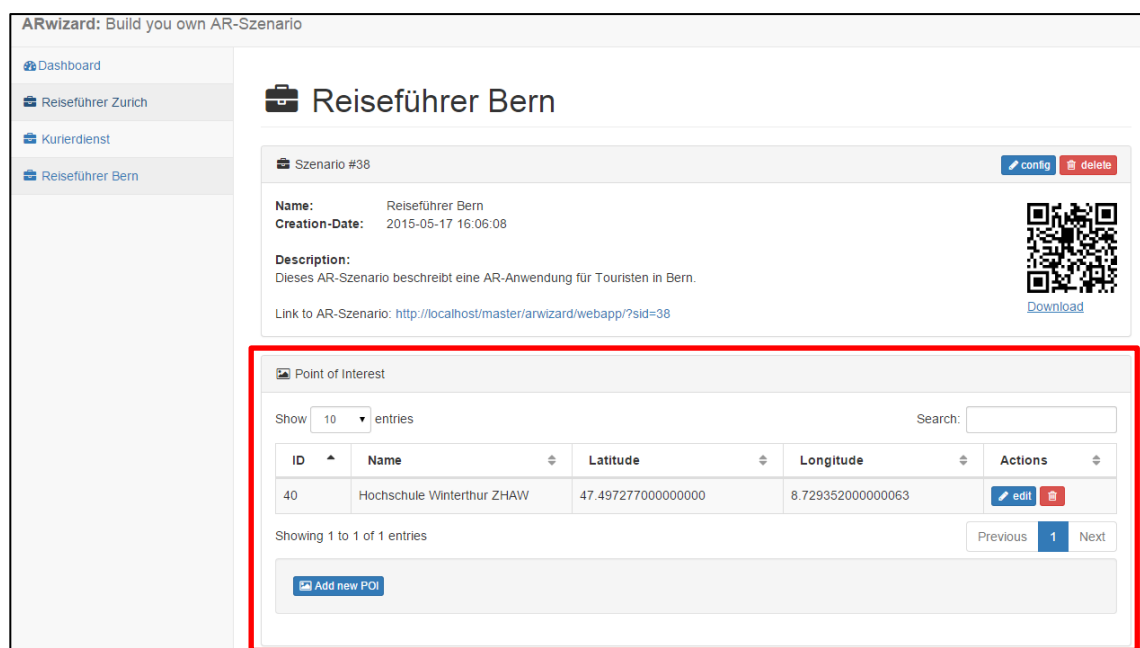


Abbildung 33: Verwaltungsseite POI

4.1.6 POI löschen

In diesem Abschnitt wird mit der nachfolgenden Abbildung 34 die Löschung eines POIs beschrieben. Wie schon bei der Löschung des AR-Szenarios ist die Ausgangslage die Verwaltungsseite des AR-Szenarios. Im Bereich der POIs kann dann bei den zu löschenden POIs der Löschbutton ausgewählt werden.

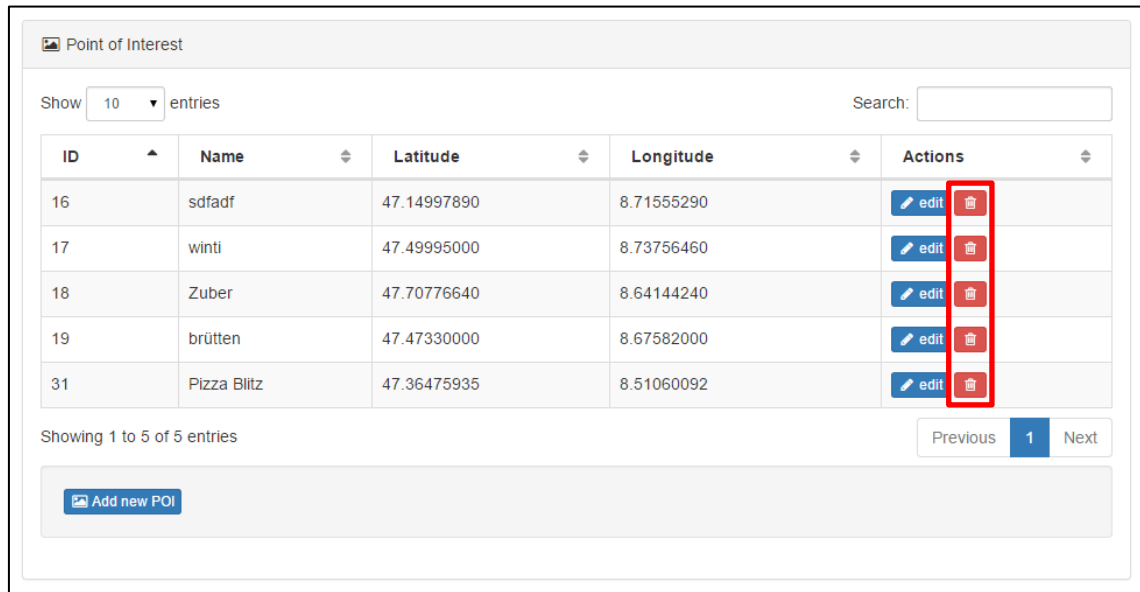


Abbildung 34: POI löschen

4.1.7 POI bearbeiten

Bearbeitet werden kann der POI mit dem Button [edit]. Die Auswahl befindet sich auf der Verwaltungsseite des AR-Szenarios. Durch die Auswahl dieses Buttons wird das Bearbeitungsformular geladen und der POI kann bearbeitet werden. Da das Formular dasselbe ist wie beim Hinzufügen eines POIs, wird hier auf das Kapitel 4.1.5 verwiesen.

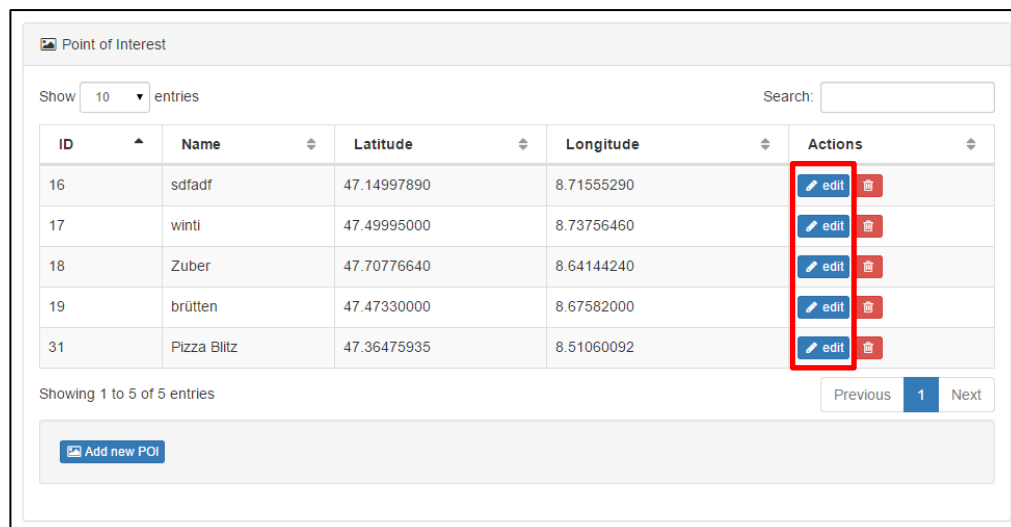


Abbildung 35: POI bearbeiten

4.1.8 Externe POI Quelle hinzufügen

Damit extern verwaltete POIs in das AR-Szenario eingebunden werden können, muss beim AR-Szenario im ARwizard die URL der externen JSON-Quelle angegeben werden. Dies geschieht mit dem Button [Add Source] und ist in der Abbildung 36 ersichtlich.

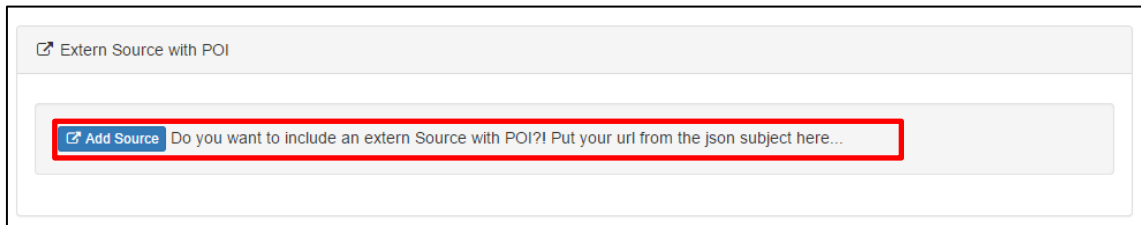


Abbildung 36: Externe Quellen hinzufügen - Schritt 1

Nun erscheint ein Eingabefeld in das die URL eingetragen wird. Mit dem Button [Save] wird die Eingabe gespeichert (Abbildung 37).



Abbildung 37: Externe Quellen hinzufügen - Schritt 2

In der Verwaltungsansicht des AR-Szenarios wird der Eintrag ersichtlich (Abbildung 28).

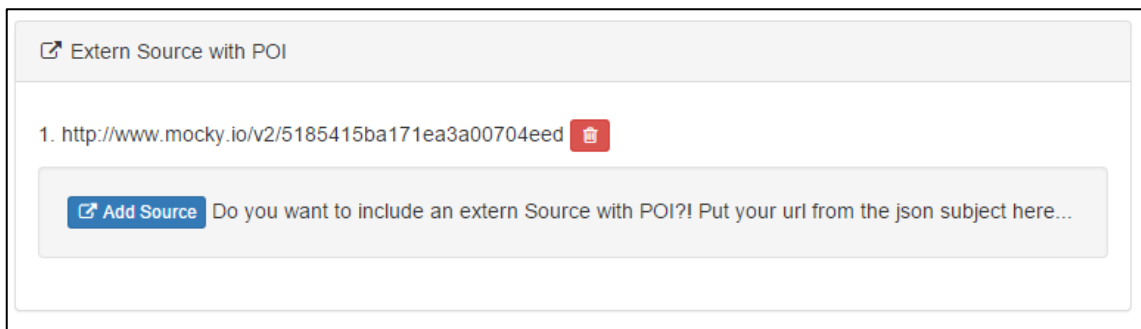


Abbildung 38: Übersicht Externe Quellen

4.1.9 Externe POI-Quelle löschen

Mit dem Löschbutton kann der Eintrag über eine externe POI-Quelle wieder gelöscht werden (Abbildung 39).

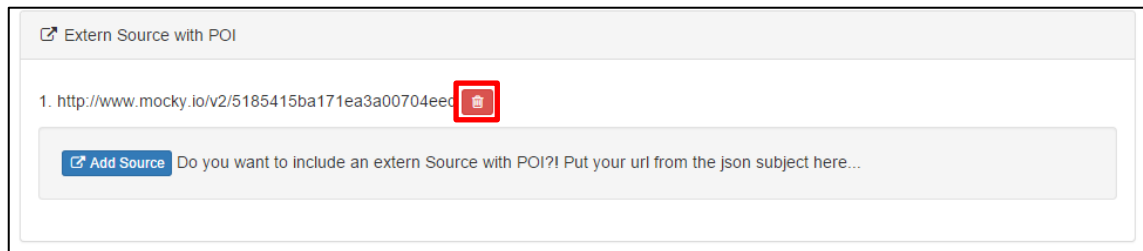


Abbildung 39: Externe Quelle löschen

4.2 AR-Browser

4.2.1 AR-Browser starten

Der AR-Browser eines AR-Szenarios wird mit dem automatisch generierten Link aus dem Administrationsbereich gestartet. Der Link muss dabei im Web-Browser des mobilen Endgeräts geöffnet werden. Nachdem die Seite geladen ist, erscheinen für den Zugriff auf die Kamera und den Zugriff auf die GSP-Sensoren zwei Berechtigungsfragen, die mit [Zulassen] bestätigt werden müssen (Abbildung 30). Im Hintergrund werden zudem die Darstellungsoptionen und Einstellungen geladen, die im Administrationsbereich gespeichert wurden.

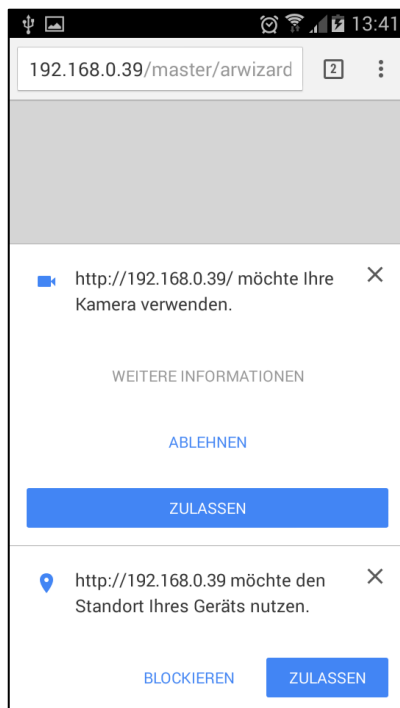


Abbildung 40: AR-Browser Startprozess

4.2.2 AR-Browser Grunddarstellung

Nun kommen wir zur eigentlichen AR-Anwendung. In Abbildung 41 werden die verschiedenen Funktionalitäten des ARwizards dargestellt.

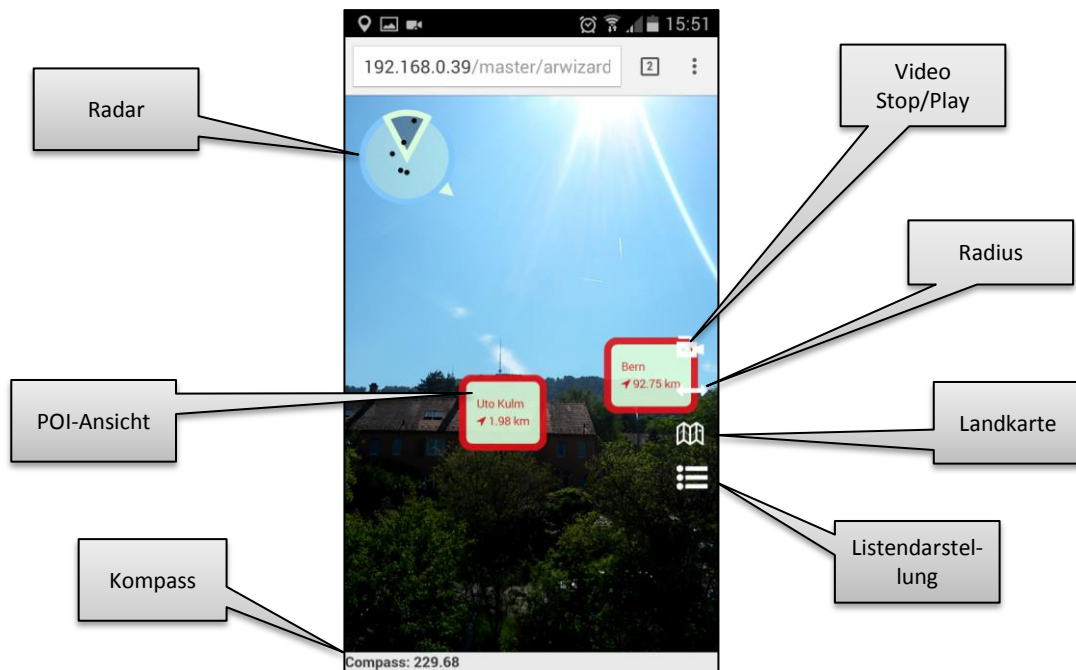
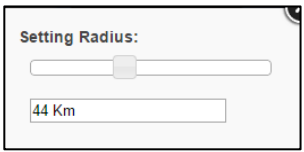


Abbildung 41: AR-Browser Darstellung mit Funktionen

Funktion	Interaktionsmöglichkeit	Beschreibung
Radar	Nein	Die Darstellung wird vom Administrationsbereich vorgegeben und je nach Auswahl des Umfeld-Radius werden mehr POIs im Radar angezeigt oder nicht. Wie bereits im Kapitel 3.4.7 beschrieben, gibt das Radar eine gute Übersicht für den Anwender.
POI-Ansicht	Ja	In der aktuellen Abbildung erkennt man zwei POIs. Der angezeigte Text ist der Name des POIs, dieser kann im Administrationsbereich jederzeit geändert werden. Im kommenden Kapitel 4.2.3 wird noch genauer auf diese Funktionalität eingegangen.
Kompass	Nein	Die Angabe des Kompasses wird in Abhängigkeit der aktuellen Position und der Ausrichtung angezeigt. Gemäss Abbildung ist die Ausrichtung momentan 229.68 Grad von Norden auch Azimut genannt.
Video Stop/Play	Ja	Durch diese Funktion kann die Kamera abgeschaltet werden. Die AR-Anwendung läuft auch ohne Kamera weiter und erscheint ein Standbild. Dies kann genutzt werden, falls der Akkuverbrauch verkleinert werden möchte. Auch gibt es Situationen, wie beispielsweise bei Nacht, in denen die Kamera nicht verwendet werden kann. Durch den Button kann die Kamera auch wieder aktiviert werden.
Radius	Ja	Wie bereits in Kapitel 3.4.6 beschrieben, kann der Radius zu einer besseren Usability beitragen. Durch die Auswahl im AR-Browser öffnet sich eine Box in der mit einem Slider der Radius eingestellt werden kann. 

Landkarte	Ja	Durch die Auswahl dieses Buttons wird der Landkarten-Modus gestartet. Diese Komponente wurde in dieser Arbeit in Kapitel 3.4.9 erklärt. Weiter werden in Kapitel 4.2.4 weitere Abbildungen aufgezeigt, wie diese Komponente im ARwizard aussieht.
Listendarstellung	Ja	Durch diese Auswahl wird eine Liste mit allen POIs angezeigt. Der Anwender hat dadurch einen Überblick über die vorhandenen POIs (vgl. Kapitel 3.4.8). Zusätzlich kann der Anwender die einzelnen POIs öffnen (vgl. Kapitel 4.2.3). <div data-bbox="1141 302 1412 638" data-label="Image"> <p>List with POI:</p> <ul style="list-style-type: none"> Paradeplatz Distance: 2.04 km Uto Kulm Distance: 3.968 km Horgen Distance: 15.006 km Hochschule Winterthur ZHAW Distance: 20.549 km Bern Distance: 94.301 km </div>

Tabelle 8: Funktionenbeschreibung des AR-Browsers

4.2.3 POI-Ansicht

Wenn ein POI im AR-Browser ausgewählt wird, öffnet sich die Informationsbox vom ausgewählten POI. Hierbei werden die hinterlegten Bilder, Videos, Tondateien oder Texte angezeigt (Abbildung 42 bis Abbildung 45). Zudem kann direkt der Navigationsmodus zu dem ausgewählten POI gestartet werden. Je nach Einstellung des Navigationstyps im Administrationsbereich werden nicht alle Transportmittel für die Navigation zu Verfügung stehen.

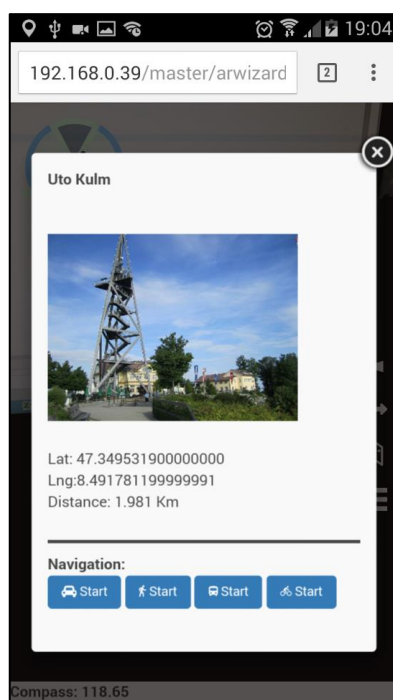


Abbildung 42: POI-Ansicht Uto-Kulm mit Bild

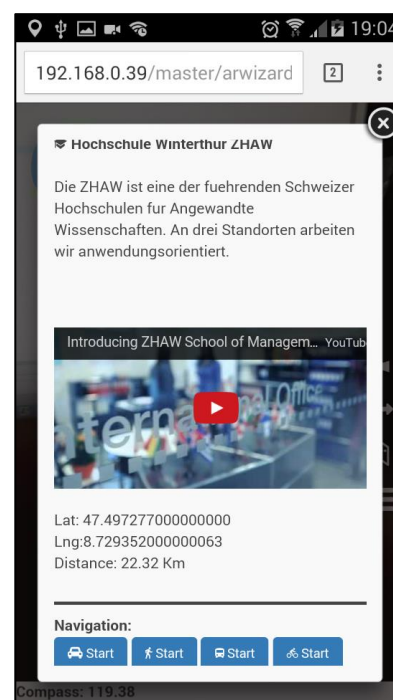


Abbildung 43: POI-Ansicht ZHAW mit Video

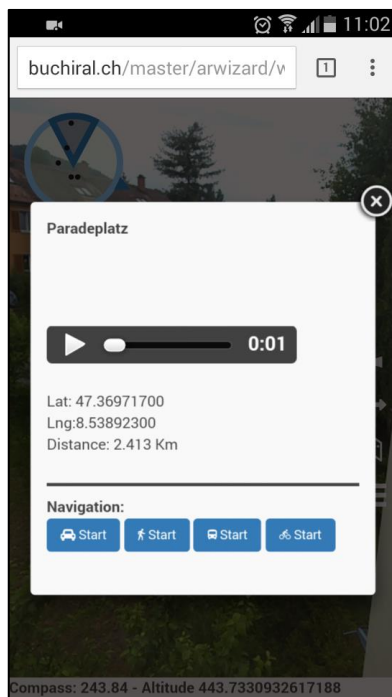


Abbildung 44: POI-Ansicht Paradeplatz mit Audio



Abbildung 45: POI-Ansicht Bern nur Text

4.2.4 Kartenansicht

Mit der Auswahl der Kartenansicht wird der AR-Modus ausgeblendet und eine Karte (Google Maps) wird auf dem mobilen Endgerät angezeigt (Abbildung 47 und Abbildung 48). Auf der Karte werden alle POIs markiert und können wiederum ausgewählt werden, um die POI-Ansicht zu öffnen. Zusätzlich zu den POIs wird auch die aktuelle Position des Anwenders bzw. mobilen Endgerätes mit einem blauen Punkt angezeigt. Dadurch kann sich der Anwender optimal orientieren.



Abbildung 46: Kartenansicht starten



Abbildung 47: Beispiel Landkarte 1

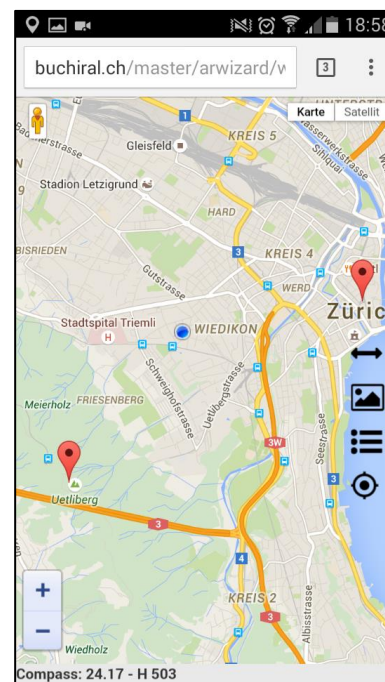


Abbildung 48: Beispiel Landkarte 2

Um zum AR-Modus zurückzukehren zu kehren, muss in der Menüleiste das in der nachfolgenden Abbildung ersichtliche Symbol ausgewählt werden.

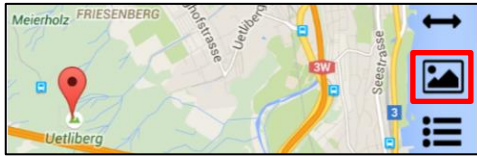


Abbildung 49: Kartenansicht beenden

Eine weitere Funktion ist die Zentrierung der Karte. Mit dieser Funktion (Abbildung 50) wird die Karte so ausgerichtet, dass der aktuelle Standort das Zentrum der Karte ist. Falls der Standort ändert, zentriert die Karte automatisch wieder. Sobald der Benutzer die Karte verschiebt, wird die Zentrierung aufgelöst und muss bei Bedarf wieder mit einem Klick initialisiert werden.

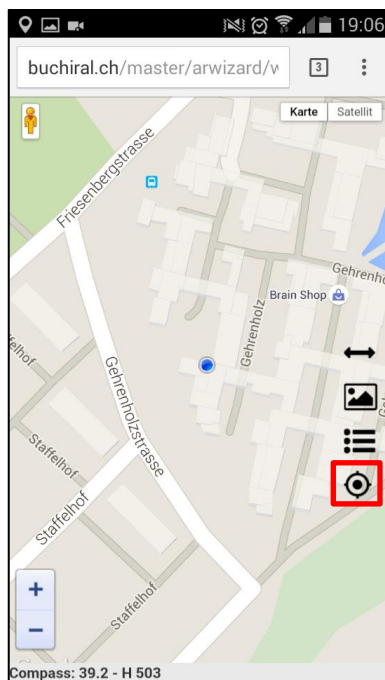


Abbildung 50: Karte zentrieren

4.2.5 Navigationsmodus

Der Navigationsmodus kann aus der POI-Ansicht mittels den folgenden Buttons gestartet werden. Die Auswahlmöglichkeit dieser Buttons ist je nach Einstellung im Administrationsbereich eingeschränkt.

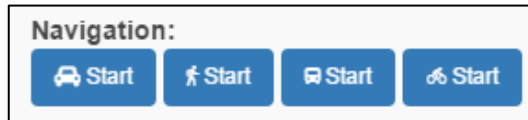


Abbildung 51: Button für Navigationsmodus

Durch die Aktivierung des Navigationsmodus werden im AR-Browser die Richtungsangabe und eine Textinformation über die Route eingeblendet (Abbildung 52). Wie in Kapitel 3.4.5 erwähnt, werden die Routenangaben für die nächsten 30 Meter grafisch dargestellt. Andere POIs werden dabei weiterhin im AR-Browser angezeigt.

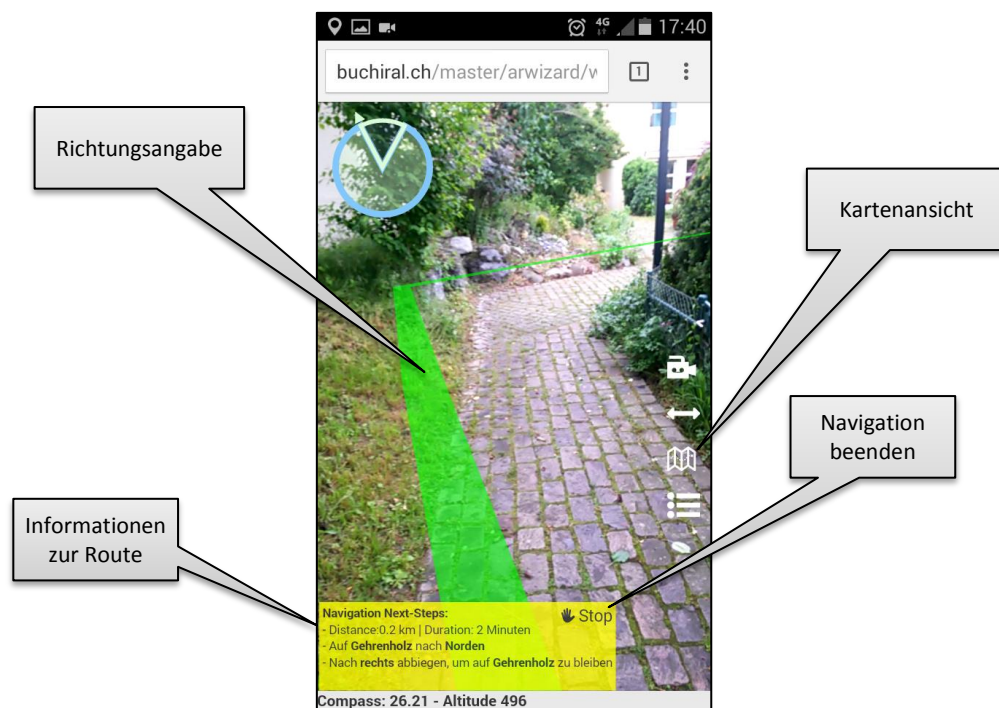


Abbildung 52: Navigationsmodus im ARwizard

Um die Orientierung nicht zu verlieren und einen Überblick über den aktuellen Standort zu erhalten, kann, wie im Kapitel 4.2.4 erläutert, in die Kartenansicht gewechselt werden (Abbildung 53). Die Route wird dabei auf der Karte angezeigt, welche verschoben und vergrößert bzw. verkleinert werden kann. Mit der Option [Karte zentrieren] kann die Karte so eingestellt werden, dass sie dem aktuellen Standort und der Route automatisch folgt.

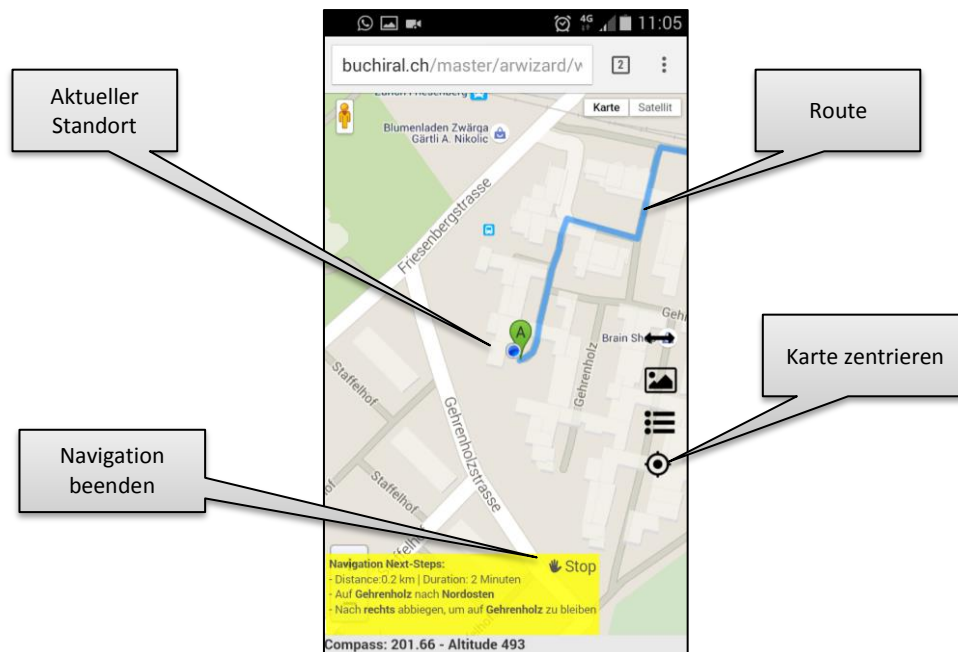


Abbildung 53: Kartenmodus mit Navigation

Sobald das Ziel innerhalb der nächsten 15 Meter liegt, erscheint eine Meldung und der Navigationsmodus wird beendet (Abbildung 54). Der Anwender kann dann mittels Karte oder AR-Browser zum Ziel finden. Der Grund für dieses Vorgehen ist, dass das GPS-Signal bei den Tests teilweise nicht präzise genug und die Navigation auf den letzten Metern fehlerhaft war. Eine ähnliche Meldung erscheint, falls die Navigation durch den Anwender gestartet wird und die Distanz zum Ziel-POI weniger als 15 Meter beträgt (Abbildung 55).

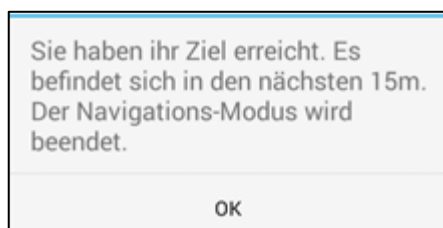


Abbildung 54: Meldung: Ziel erreicht

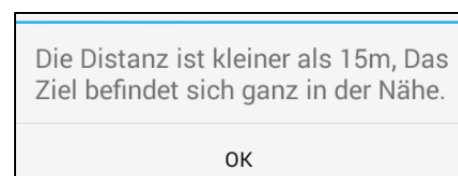


Abbildung 55: Meldung: Distanz zu klein

5 Flexibilität der Anwendung „ARwizard“

5.1 Szenario: Reiseführer „Skulpturen“

Szenario-Beschreibung

Das erste Beispielszenario soll eine Reiseführer-Anwendung für Skulpturen darstellen. In diesem Szenario sollen alle Skulpturen einer Stadt, Gemeinde oder Dorf in einem AR-Browser angezeigt werden können und dem Anwender, in diesem Fall einem Touristen, auf effiziente Weise dargestellt werden. Dabei sollen dem Anwender eine Vielfalt von Informationen über die einzelnen Skulpturen in Form von einem Text, einem Video, einem Bild oder einer Tondatei angeboten werden. Auch soll dem Touristen eine Übersicht dargeboten werden, welche Skulpturen in seiner aktuellen Umgebung existieren. Die Anwendung stellt zudem eine Orientierungshilfe an einem fremden Ort dar.

Lösung mit ARwizard

Mit den verschiedenen Komponenten des ARwizards kann dieses Szenario umgesetzt werden. Über den Administrationsbereich werden die Skulpturen dem AR-Szenario hinzugefügt. Weiter können Informationen gemäss den Anforderungen des Szenarios in verschiedenen Formen (Text, Bilder, Video und Audio) dem POI hinzugefügt werden. Zusätzlich kann dem Anwender mit Hilfe der Listendarstellung, dem Radar und der Radius-Funktion ein Überblick über alle Skulpturen in der Umgebung gegeben werden. Die Anforderung einer Orientierungshilfe durch den ARwizard kann mit der Landkarten-Funktion erfüllt werden. Wie in Kapitel 4.2.4 beschrieben, wird mit dieser Funktion eine interaktive Landkarte (Google Maps) auf dem mobilen Endgerät angezeigt, die es dem Tourist ermöglicht sich zu orientieren. Die webbasierte Umsetzung des ARwizards bringt dem Touristen insofern einen Nutzen, dass er mit dem mobilen Endgerät (unter der Voraussetzung, dass die Anforderungen gemäss Kapitel 3.1.1 erfüllt sind) mittels Web-Browser auf die Anwendung zugreifen kann, ohne vorgängig eine Installation durchführen zu müssen.

5.2 Szenario: Denkmalgeschützte Gebäude (externe POIs)

Szenario-Beschreibung

In diesem Abschnitt wird das Beispielszenario für das Anzeigen von denkmalgeschützten Gebäuden beschrieben. Jede Gemeinde oder Stadt besitzt Katasterpläne mit Informationen über Gebäude, Anlagen, Häuser, etc., die mittels AR-Browser angezeigt werden können. Daraus ergibt sich der Nutzen, dass die Anwender vor Ort Informationen über die Gebäude erhalten und diese nicht mit Aufwand in Verzeichnissen suchen müssen.

Lösung mit ARwizard

Durch die Schnittstelle von ARwizard können externe POIs im AR-Browser angezeigt werden. Bei diesem Szenario können so die Katasterinformationen dem ARwizard über eine Schnittstelle zur Verfügung gestellt werden. Der Vorteil ist, dass die Katasterinformationen nicht im ARwizard verwaltet werden müssen. Auch sind die Daten, welche im ARwizard angezeigt werden, immer aktuell. Es besteht zudem die Möglichkeit zur Individualisierung, indem bestimmte AR-Funktionen aktiviert bzw. deaktiviert werden. In diesem Beispiel wird eine AR-Navigation nicht zwingend benötigt und im Administrationsbereich des ARwizards deaktiviert. Die Radius-Funktion sollte hingegen aktiviert werden, da der Anwender durch Änderung des Radius die POIs im Sichtfeld eingrenzen kann. Dies hat den Vorteil, dass nicht unzählige POIs im Kamerabild erscheinen.

5.3 Szenario: AR-Navigation Kurierdienst

Szenario-Beschreibung

Ein weiteres Beispielszenario ist eine AR-Navigationsanwendung für einen Lebensmittel-Kurierdienst. Die Anforderungen sind, dass der Kurier in der AR-Anwendung einen Überblick über die Lieferorte hat und direkt zum Lieferort navigiert wird, in dem er das nächste Ziel selber auswählen kann. Dabei kann der Kurier entweder mit dem Auto, dem Fahrrad, zu Fuss oder mit den öffentlichen Verkehrsmitteln unterwegs sein.

Lösung mit ARwizard

Das aufgezeigte Szenario eines Kurierdienstes kann mit dem ARwizard durch die Funktionalität der AR-Navigation erreicht werden. Es können je nach Anwendungsfall die verschiedenen Komponenten des ARwizards zur optimalen Lösung für einen Kurierdienst beitragen. Durch die Schnittstelle, mit der externe POIs geladen werden können, kann der Kurierdienst seine aktuellen Zielorte in die AR-Anwendung laden. Der Kurier, der die Lieferung ausführt, hat die Möglichkeit auf seinem mobilen Endgerät die Lieferorte zu betrachten und kann den nächsten Lieferort auswählen. Dies wird durch die Listendarstellung (vgl. Kapitel 3.4.8) oder die Kartenansicht (vgl. Kapitel 4.2.4) des ARwizards ermöglicht. In einem nächsten Schritt kann er einen Zielort auswählen bzw. die weiteren Details dazu ansehen. Es kommt dabei darauf an, wie die Informationen eingegeben wurden. Optimal wäre es, wenn im Description-Feld des Lieferortes (POI) ersichtlich ist, welche Objekte geliefert werden müssen und wie die genauen Kontaktdaten (inkl. Telefonnummer) des Kunden lauten. Weiter hat der Kurier die Möglichkeit direkt die Navigation zu diesem Lieferort zu starten. Für die Navigation bietet der ARwizard dank der integrierten Google Maps API vier verschiedene Navigationsarten an, zu Fuss, mit dem Auto, mit den öffentlichen Verkehrsmitteln oder mit dem Fahrrad. Sobald die AR-Navigation gestartet wurde, werden im Kamerabild die Richtungsangaben (vgl. Kapitel 3.4.5) angezeigt. Zusätzlich besteht im ARwizard die Möglichkeit den Darstellungsmodus der Navigation zu ändern, sodass eine Landkarte dargestellt wird, was zur Unterstützung des Anwenders beitragen kann.

5.4 Szenario: AR-Navigation Eventveranstaltung

Szenario-Beschreibung

Im folgenden Szenario geht es darum, potenziellen Kunden mit marketingtechnischen Mitteln, wie zum Beispiel Flyern oder Zeitungsinseraten über eine Veranstaltung zu informieren und direkt mit seinem Smartphone zum Event zu führen. Das heisst, der Kunde soll mit einem Link oder einem Barcode die Navigation zum Veranstaltungsort direkt starten können. Das Szenario kann auch so erweitert werden, dass bei einer Grossveranstaltung Flyer von Verpflegungsständen oder Ausstellern verteilt werden und der Kunde direkt zu diesen geführt wird.

Lösung mit ARwizard

Das Szenario beschreibt den Geschäftsfall, dass potenzielle Kunden zum Anbieter geführt werden. Mit den verschiedenen Funktionalitäten des ARwizards ist es durchaus möglich solch ein Szenario zu unterstützen. Der ARwizard bietet zu jedem AR-Szenario einen Barcode an, mit dem das AR-Szenario im Webbrowser geöffnet werden kann. Dieser kann in einen Flyer oder ein Zeitungsinserat integriert werden. Der Kunde kann anschliessend durch das Scannen des Barcodes, ohne eine weitere Installation einer Applikation oder Software, ganz unkompliziert die Navigationsfunktion starten und wird direkt zum Ort geführt. Im Administrationsbereich des ARwizards können darüber hinaus zusätzliche Informationen über den Zielort angegeben werden. Beim Beispiel einer Veranstaltung kann die Zeit der Türöffnung angegeben werden, beim Beispiel eines Verpflegungsstandes die Preise und das Angebot.

6 Konklusion

6.1 Forschungsergebnis

Das Ziel dieser Masterthesis war es, die Beantwortung der Forschungsfragen mit Hilfe des IT-Artefakts ARwizard zu beantworten. Dazu wurde ein Prototyp geschaffen, der zum einen aus einem Administrationsbereich und zum andern aus einem AR-Browser besteht. Der Administrationsbereich stellt eine Webplattform für Personen dar, die ein eigenes AR-Szenario bilden möchten. Hierbei wurde darauf Wert gelegt, dass die Grundfunktionalitäten einer standortbezogenen AR-Anwendung, wie man sie auf dem freien Markt kennt, angepasst und verwaltet werden können. Dieser Administrationsbereich wurde aus der Anforderungsanalyse hergeleitet, welche die optimalen Komponenten und Funktionalitäten eines solchen Systems aufzeigt.

Zur Beantwortung der Forschungsfrage „Wie kann die Entwicklung von standortbezogenen AR-Anwendungen für mobile Endgeräte generell unterstützt werden?“ wurden die einzelnen Komponenten der AR-Anwendung erklärt und auf dieser Basis die Webplattform entwickelt, mit der ein Entwickler auf elegante Art und Weise und ohne Programmierung eine AR-Anwendung bilden kann. Somit stellt ARwizard eine Lösung für diese Forschungsfrage dar.

Die weiteren Unterforschungsfragen („Welche Anforderungen müssen erfüllt sein, damit ohne Programmierkenntnissen eine standortbezogene AR-Applikation gebildet werden kann?“, „Ist es möglich ausschliesslich mit existierenden Webtechnologien AR-Applikationen zu bilden?“, „Welche Anforderungen haben standortbezogene AR-Anwendungen?“ sowie „Welche Anforderungen muss eine Plattform haben, mit der man AR-Anwendungen erstellen kann?“) konnten mit der Anforderungsanalyse und dem AR-Browser beantwortet werden. Der AR-Browser wurde dabei ausschliesslich mit Webtechnologien entwickelt und ist in einem Webbrowser auf einem mobilen Endgerät ausführbar. Dadurch wird bewiesen, dass eine AR-Anwendung mit den existierenden Webtechnologien gebildet werden kann.

In der Masterthesis wurden zudem die Forschungsrichtlinien nach Hevner et al. (2004) eingehalten. In nachfolgender Tabelle wird die Masterthesis anhand der sieben Richtlinien nach Hevner et al. (2004) evaluiert.

Richtlinie	Evaluation
Design as an Artefact	Durch die Erarbeitung des IT-Artefakts ARwizard wurde eine neuartige Lösung entwickelt, mit der die Problemstellung gemäss Kapitel 1.2 gelöst werden konnte.
Problem Relevance	Die Problemstellung wurde zu Beginn der Masterthesis aufgezeigt und das Ergebnis orientiert sich klar daran.
Design Evaluation	Durch kontinuierliche Tests bei der Entwicklung wurde die Nützlichkeit und Qualität des Forschungsergebnisses durchgehend überprüft. Auch wurde der Aufbau der Masterthesis zu Beginn definiert und abgearbeitet.
Research Contributions	Mit dem Ergebnis, dass eine AR-Anwendung mit den existierenden Webtechnologien gebildet werden kann und mit der Generalisierung von AR-Anwendungen, konnte das IT-Artefakt ARwizard ein eindeutiger Beitrag zur Wissenschaft leisten.
Research Rigor	Der Prozess von der Herleitung der Problemstellung, über die Darlegung des theoretischen Hintergrunds, zur sukzessiven Entwicklung des IT-Artefaktes stellt ein strukturiertes wissenschaftliches Vorgehen dar.
Design as a Search Process	Durch kontinuierliche Tests bei der Entwicklung des IT-Artefaktes wurde das Forschungsergebnis mehrfach kontrolliert und evaluiert. Im Verlauf der Masterthesis wurden zudem Teilergebnisse mit dem Betreuer ausgetauscht und evaluiert.
Communication of Research	Die Forschungsergebnisse werden in Form der vorliegenden Arbeit zielgruppengerecht kommuniziert. Die Neuartigkeit und das Konzept des IT-Artefakts ARwizard wird dabei explizit erklärt und dargelegt.

Tabelle 9: Evaluation der Richtlinien nach Hevner et al. (2004)

6.2 Offene Punkte und Ausblick

In diesem Abschnitt werden offene Punkte und Verbesserungsvorschläge vorgestellt, die sich bei den Tests herauskristallisiert haben. Zudem werden Informationen über allfällige Erweiterungen angegeben.

Genauigkeit der Positionierung mittels GPS

In den Tests hat sich eine Ungenauigkeit von wenigen Metern bezüglich der GPS-Position gezeigt. Im Normalfall konnte der ARwizard trotz dieser Ungenauigkeit problemlos benutzt werden, doch gerade bei der Fussgängernavigation, welche teilweise schmale Fusswege beinhaltet, waren die Richtungsangaben vereinzelt fehlerhaft. Daher muss die GPS-Positionierung genauer analysiert werden und eine Lösung erarbeitet werden, wie mit ungenauen GPS-Signalen umgegangen werden soll.

POI-Ansicht

In diesem Prototyp kann der Benutzer die geometrische Form des POI nicht anpassen. Der POI wird immer in Form eines Rechtecks mit abgerundeten Ecken angezeigt. Lediglich die Farben können individualisiert werden. Allenfalls könnte dies in einem Folgeprojekt dynamischer gestaltet werden. Dabei soll die geometrische Form eines POI individuell angepasst und der Inhalt der POI-Ansicht modular ausgewählt werden können.

Energieverbrauch

Bei der Implementierung standen primär die Funktionalitäten im Vordergrund. Ein wichtiger Bereich bei mobilen Endgeräten ist jedoch auch der Energieverbrauch. Bei solch einer AR-Anwendung stehen verschiedene Komponenten, wie beispielsweise Kamera und GPS durchgehend miteinander in Kontakt und bewirken einen überdurchschnittlichen Energie- und Ressourcenverbrauch. Daher sollte in einem nächsten Projekt oder einer nächsten Überarbeitung eine Optimierung im Energieverbrauch vorgenommen werden.

POI-Informationen

Ferner können in einem Folgeprojekt weitere spezifische Felder implementiert werden, wie zum Beispiel Telefonnummern, Email-Adressen oder Social-Media Buttons. Auch wäre es denkbar, dass individuelle Buttons hinzugefügt werden können, die einen

Webservice einer externen Applikation ansteuern. Dadurch könnte die Datenverarbeitung einer Applikation mit dem Standort verknüpft werden.

Externe Quellen einbinden

Zudem gibt es die Möglichkeit, in Zukunft weitere Schnittstellen und Services zu Opendata-Anbietern zu implementieren. Für solche Erweiterungen gibt es bereits viele Anbieter Datenbanken, zum Beispiel *Wikimapia* oder *OpenStreetMap*. Diese bieten oft bereits eine Schnittstelle an, die angesteuert werden könnte.

Navigation

Bei der Navigationsfunktion gibt es noch einige Verbesserungsmöglichkeiten. Im Rahmen dieser Masterthesis konnten nur die Grundfunktionalitäten implementiert werden. Es hat sich gezeigt, dass im Navigationsmodus ein Hinweis nötig ist, falls sich der Anwender in die entgegengesetzte Richtung bewegt. Momentan wird dieses Szenario nicht optimal unterstützt, der Anwender wird zwar nach links bzw. rechts geführt, doch er erkennt nicht, dass er sich um 180 Grad drehen sollte.

Darüber hinaus ist bei der Darstellung von weiteren Informationen, wie der Angabe der Geschwindigkeit oder der Ankunftszeit, Verbesserungspotenzial vorhanden. Eine mögliche Erweiterung wäre auch die automatisierte Erkennung des aktuellen Fortbewegungsmittels. Dadurch kann der ARwizard die optimale Route von der Google Maps API anfragen und anzeigen. Dies wird beispielsweise benötigt, falls ein Anwender zu Fuss unterwegs ist und eine Teilstrecke mit dem öffentlichen Verkehrsmittel oder dem Fahrrad zurücklegt.

Horizontale Platzierung der POIs im Sichtfeld

Ein weiterer Punkt, der bei den Tests aufgefallen ist, ist die ideale Platzierung der POIs im Sichtfeld. Es hat sich gezeigt, dass je nach Szenario das mobile Endgerät verschieden ausgerichtet ist. Beispielsweise hält ein Fussgänger das Smartphone normalerweise angewinkelt nach unten auf Bauchhöhe. Dies bewirkt, dass die Kamera des Smartphones nach unten zeigt und dadurch POIs nicht angezeigt werden. Hier stellt sich die Frage, ob trotz dieser Ausrichtung die POIs, welche über dem Blickfeld des Anwenders liegen, angezeigt werden sollen. Bei der Anwendung beim Autofahren mit einer Platzierung des mobilen Endgerätes in einer senkrechten Halterung stellt sich analog dazu die Frage, ob grundsätzlich alle POIs angezeigt werden sollen, egal ob sie über oder unter dem Blickfeld liegen.

6.3 Fazit

Zusammenfassend stellt der ARwizard eine solide Lösung dar, um die Entwicklung standortbezogener AR-Anwendungen auf der Basis von webbasierten Technologien und ohne Programmierkenntnisse zu ermöglichen und zu unterstützen. Beim ARwizard handelt es sich um einen Prototyp, der sich bezüglich der Genauigkeit je nach Anwendungsszenario ausbauen und mit weiteren Zusatzfunktionalitäten erweitern lässt.

Literaturverzeichnis

- blippar. (2014). *Mobile Augmented Reality*. URL: <https://blippar.com/> [24.11.2014].
- de Lange, N. (2013). *Geoinformatik in Theorie und Praxis*. Berlin Heidelberg: Springer-Verlag Berlin Heidelberg. S. 145.
- Elektronik Kompendium. (2014). *LBS - Location Based Services (standortbezogene Dienstleistungen)*. URL: <http://www.elektronik-kompendium.de/sites/kom/0905061.htm> [24.11.2014].
- Geiger, P. (2012). *Entwicklung einer Augmented Reality Engine am Beispiel des iOS*. Ulm, DE-89069: Institut für Datenbanken und Informationssysteme.
- Google, (2015). *Directions Service - Google Maps JavaScript API*. URL: <https://developers.google.com/maps/documentation/javascript/directions?hl=de> [05.05.2015].
- Hevner, A., March, S., Park, J., & Ram, S. (2004). Design Science in Information System. *Research. MIS Quaterly Vol. 28 No. 1*, S. 75-105.
- Kluge, M. (2009). *Fussgängernavigation: Reality View - der Einsatz von Computerspielnavigation in der realen Welt*. Salzburg.
- kowoma.de. (2007). *Navigation mit GPS*. URL: <http://www.kowoma.de/gps/Navigation.htm> [03.05.2015].
- Ludwig, C., & Reinmann, C. (2005). Augmented Reality: Information im Fokus. *C-LAB Report Vol. 4 No. 1*.
- Mehler-Bicher, A. (2011). *Augmented Reality: Theorie und Praxis*. München: Oldenbourg Verlag.
- Metaio. (2014). *Creator Features*. URL: <http://www.metaio.com/products/creator/features/> [24.11.2014].
- Michel, J. (2011). *Layar & Co.: Apps für Augmented Reality im Überblick*. URL: <http://www.teltarif.de/augmented-reality-browser-smartphone-internet/news/42173.html> [05.04.2015].
- Milgram, P. (1994). *Augmented Reality: A class of displays on the reality-virtuality continuum*. ATR Communication Systems Research Laboratories.

- Minock, D. (2014). *Two Guys and Some iPads: 6 Platforms to Create Augmented Reality*. URL: <http://www.twoguysandsomeipads.com/2014/02/6-platforms-to-create-augmented-reality.html> [24.11.2014].
- richt. (2014). *html5 - Calculate compass heading from DeviceOrientation Event API - Stack Overflow*. URL: <http://stackoverflow.com/questions/18112729/calculate-compass-heading-from-deviceorientation-event-api> [15.04.2015].
- selfhtml-wiki. (2014). *JavaScript/API/Geolocation – SELFHTML-Wiki*. URL: <http://wiki.selfhtml.org/wiki/JavaScript/API/Geolocation> [30.04.2015].
- Sinnott, R. (1984). Virtues of the haversine. *Sky and Telescope* vol. 68 no. 2, S. 158.
- Steiniger, S. (2006). *Foundations of location based services*. Lecture Notes on LBS.
- Stürzekarn, D. (2010). *Fußgängernavigation mit Augmented Reality auf Android-basierten Smartphones*. Institut für Informatik AG Medieninformatik Universität Osnabrück.
- Veness, C. (2015). *Calculate distance, bearing and more between Latitude/Longitude points*. URL: <http://www.movable-type.co.uk/scripts/latlong.html> [01.05.2015].
- Wikitude.com. (2015). *Wikitude Navigation (Turn-by-turn) - Wikitude*. URL: <http://www.wikitude.com/showcase/wikitude-navigation/> [02.05.2015].
- W3c.(2014). *Editor's Draft 12 March 2014 - DeviceOrientation Event Specification*. URL: <http://w3c.github.io/deviceorientation/spec-source-orientation.html#worked-example> [03.04.2015].

Anhang A: CD

Mit der Abgabe der Arbeit wird auch eine CD abgegeben. In der Tabelle ist die Struktur ersichtlich.

Ordnerstruktur	Beschreibung
CD/	
__ /1_Diagramme	Erstellte Diagramme
__ /2_Literatur	Elektronisch verfügbare Literatur
__ /3_Quellcode_ARwizard	Dateien und Quellcode des Prototypen ARwizard. Im Ordner Webapp befindet sich der AR-Browser.
_ Büchi_Ralph_Masterthesis_FS2015.pdf	Masterthesis als PDF
_ Büchi_Ralph_Masterthesis_FS2015.docx	Masterthesis als Word (docx.)

Tabelle 10: Ordner-/ Dateistruktur CD

Wahrheitserklärung

„Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig, ohne Mithilfe Dritter und nur unter Benützung der angegebenen Quellen verfasst habe und dass ich ohne schriftliche Zustimmung der Studiengangleitung keine Kopien dieser Arbeit an Dritte aushändigen werde.“

Gleichzeitig werden sämtliche Rechte am Werk an die Zürcher Hochschule für Angewandte Wissenschaften (ZHAW) abgetreten. Das Recht auf Nennung der Urheberschaft bleibt davon unberührt.

Name des Studierenden (Druckbuchstaben)

.....

Name des Studierenden (Unterschrift)

.....