



Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

Prototypische Entwicklung eines multimedialen Museumsführers für die Staatliche Kunsthalle Karlsruhe mit Augmented Reality für Tablets

Bachelor-Thesis zur Erlangung des Grades Bachelor of Arts (B. A.)
im Studiengang Kommunikation und Medienmanagement
an der Fakultät für Informationsmanagement und Medien

Rebecca Durm
Matrikelnummer: 41773
rebecca.durm@web.de

Bearbeitungszeitraum:
15. März 2017 – 14. Juli 2017

Betreuer:
Prof. Dipl.-Ing. Martin Schober
Prof. Dr. Petra Drewer

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Ausführungen, die anderen veröffentlichten oder nicht veröffentlichten Schriften wörtlich oder sinngemäß entnommen wurden, habe ich kenntlich gemacht.

Die Arbeit hat in gleicher oder ähnlicher Fassung noch keiner anderen Prüfungsbehörde vorgelegen.

Ort, Datum

Unterschrift

Inhaltsverzeichnis

EIDESSTATTLICHE ERKLÄRUNG	III
ABKÜRZUNGSVERZEICHNIS	IX
1 EINLEITUNG	1
1.1 Ausgangssituation	2
1.2 Aufgabenstellung.....	4
1.3 Struktur der Arbeit.....	5
2 AUGMENTED REALITY.....	7
2.1 Augmented Reality im Allgemeinen	8
2.1.1 Definition Augmented Reality, Mixed Reality und Abgrenzung zu Virtual Reality.....	8
2.1.1.1 Definition von „real“ und „virtuell“	8
2.1.1.2 Definitionsansätze von Augmented Reality und Virtual Reality.....	8
2.1.2 Stand der Technik.....	10
2.1.2.1 Historie von Augmented Reality	10
2.1.2.2 Zukunftsaussichten.....	11
2.1.3 Charakteristika einer Augmented-Reality-Anwendung	12
2.1.4 Anwendungsszenarien und Anwendungsbeispiele.....	14
2.1.5 Nutzen von Augmented-Reality-Anwendungen	15
2.2 Technische Grundlagen von AR	16
2.2.1 Ablauf bei AR-Anwendungen	16
2.2.2 Tracking.....	16
2.2.2.1 Tracking-Verfahren.....	17
2.2.2.2 Tracking-Prinzipien	22
2.2.3 Registrierung.....	23
2.2.3.1 Geometrische Registrierung	23
2.2.3.2 Photometrische Registrierung.....	25
2.2.4 Ausgabe	26
2.2.4.1 Datenbrillen.....	26
2.2.4.2 Projektionsbasierte Ausgabe	29

2.2.4.3	Handheld-Geräte	29
2.2.4.4	Kontaktlinsen	30
2.2.4.5	Nichtvisuelle Ausgabemöglichkeiten	31
2.3	AR, Multimedia und E-Learning.....	32
2.3.1	Einordnung in diese Arbeit	32
2.3.2	Definition E-Learning.....	32
2.3.3	Definition Multimedia	33
2.3.4	Einflussfaktoren für erfolgreichen Umgang mit Multimedia	33
2.3.4.1	Zielgruppe.....	34
2.3.4.2	Motivation.....	34
2.3.4.3	Stress	35
2.3.4.4	Soziale Effekte.....	36
2.3.5	Auswirkungen von Multicodierung und Multimodalität ...	36
2.3.6	Augmented Reality und Multimedia.....	38
2.3.7	Interaktivität.....	39
2.3.7.1	Definition Interaktivität	39
2.3.7.2	Stufen der Interaktivität	39
2.3.7.3	Interaktivität bei AR.....	40
3	LERNEN UND GEDÄCHTNIS.....	43
3.1	Definition Lernen	44
3.2	Gedächtnis.....	45
3.2.1	Gedächtnismodell.....	45
3.2.2	Gedächtnissysteme des Langzeitgedächtnisses	46
3.3	Lernen mit AR	48
4	KONZEPT.....	49
4.1	Vorgehensweise bei der Konzepterarbeitung	50
4.2	Inhaltliches und gestalterisches Konzept	51
4.2.1	Zielgruppe.....	51
4.2.1.1	Allgemeine Zielgruppenbeschreibung für Kunstmuseen.....	51
4.2.1.2	Zielgruppe der SKK.....	51
4.2.2	Lehrziele und Aufgaben der App.....	52

4.2.3	Lernziel der Nutzer.....	53
4.2.4	Grundstruktur der Informationen	53
4.2.5	Inhalte	54
4.2.5.1	Umgang mit Text und Ton.....	55
4.2.5.2	Formulierung der Texte	56
4.2.5.3	Audioinformationen	56
4.2.5.4	Internetlinks.....	56
4.2.6	Interaktivität, Stress und Motivation.....	56
4.2.7	Gestaltung	57
4.2.7.1	Schrift	57
4.2.7.2	Farben.....	57
4.2.7.3	Kontrast.....	58
4.2.7.4	Icons und Symbole.....	58
4.3	Technisches Konzept	59
4.3.1	Ausgabegerät.....	59
4.3.2	Trackingverfahren	59
4.3.3	Aufbau der XML-Datei	60
5	UMSETZUNG UND PROGRAMMIERUNG	65
5.1	Methodisches Vorgehen	66
5.2	Software-Architektur	67
5.2.1	Überblick über verwendete Softwares und Plug-Ins.....	67
5.2.1.1	Unity von Unity Technologies	67
5.2.1.2	Vuforia von PTC, Inc.....	67
5.2.1.3	TTS Android Plug-In von Takohi Games.....	67
5.2.1.4	Android Studio von Google Inc.	67
5.2.1.5	Java SE Development Kit (JDK) von Oracle Corp.....	68
5.2.1.6	Unity Remote 5 von Unity Technologies.....	68
5.2.2	Zusammenhänge zwischen Software-Tools und Plug-Ins..	68
5.2.3	Szenen.....	70
5.2.4	Skripte und ihre Verbindung untereinander.....	73
5.2.5	Objekthierarchie in Unity.....	77
5.3	Erweiterung der Anwendung um Targets.....	78
5.3.1	Vorbereitung.....	78
5.3.1.1	Dateien an- und ablegen	78
5.3.1.2	Bilder in Sprite verwandeln	79

5.3.2	Target in Datenbank laden.....	79
5.3.3	Target-Datenbank in Unity hinzufügen	81
5.3.4	ImageTarget-Objekt hinzufügen	82
5.3.5	Hotspot-Collection anlegen.....	83
5.3.6	Skript „DefaultTrackableEventHandler“ anpassen	84
5.3.7	Skripte und Objekte den Skripten zuweisen	85
5.3.8	Skripte nachbearbeiten.....	86
5.3.8.1	„targetData“ nachbearbeiten.....	87
5.3.8.2	„loadXMLFile“ nachbearbeiten.....	88
5.3.8.3	„touchHotspot“ nachbearbeiten	89
5.3.9	Build-Settings einrichten.....	90
5.3.10	App bilden und installieren	92
6	EVALUATION.....	93
6.1	Experten-Evaluation.....	94
6.2	Test in der Staatlichen Kunsthalle	96
7	FAZIT UND AUSBLICK.....	97
7.1	Technisches Ergebnis.....	98
7.2	Optimierungspotenzial der bisherigen Anwendung	99
7.3	Ansätze für weitere Arbeiten auf Basis dieser Bachelorthesis.....	101
	LITERATURVERZEICHNIS	CIV
	ANHÄNGE	CX
	Nutzungshinweise für Unity Remote 5	CX
	Eingebundene Gemälde	CXI
	Quellen für die Informationen in der App	CXIV
	XML-Datei mit den Inhaltstexten für „Die 7 Todsünden“	CXV
	Quellcode.....	CXIX
	defaultTrackableEventHandler.....	CXIX
	languageButtons.cs.....	CXXI

lightIntensity.cs	CXXIV
loadXMLFile.cs.....	CXXV
menu.cs	CXXX
openSettingWindow.cs	CXXXII
targetData.cs	CXXXIII
TextToSpeech.cs	CXXXVI
touchHotspot.cs	CXXXIX
Fragebogen an die Staatliche Kunsthalle	CXLIII
Email von Fr. Tabea Mernberger	CXLVI

Abkürzungsverzeichnis

Abkürzung	Langform
AR	Augmented Reality
DOF	Degrees of Freedom, Freiheitsgrade
GPS	Global Positioning System
HMD	Head-Mounted Display, Datenbrille
JDK	Java SE Development Kit
MR	Mixed Reality
OST	Optical See Through
RE	Real Environment
RV Continuum	Reality-Virtuality Continuum
SDK	Software Development Kit
SfM	Structure from Motion
SKK	Staatliche Kunsthalle Karlsruhe
SLAM	Simultaneous Localization and Mapping
TTS	Text To Speech
TUI	Tangible User Interface
VE	Virtual Environment
VR	Virtual Reality
VST	Video See Through

1 Einleitung

1.1 Ausgangssituation

Computer, Tablet und Smartphone bestimmen unseren Alltag zunehmend und beeinflussen das menschliche Zusammenleben. Die „neuen Medien“ führen zu einem sozialen Wandel und spielen in der Gesellschaft eine große Rolle (vgl. Hemmerling 2011: 7). Ein Trend, der „kein vorübergehender Hype“ ist (vgl. Mehler-Bicher / Steiger 2014: 4), ist erweiterte Realität, auch augmentierte Realität oder Augmented Reality (AR) genannt. Spätestens seit dem großen Erfolg des Smartphone-Spiels Pokémon Go von Niantic, das bis Februar 2017 weltweit mehr als 650 Millionen Mal auf Android- und IOS-Geräte heruntergeladen wurde (vgl. Sarkar 2017), ist das Phänomen Augmented Reality der Allgemeinheit bekannt.

Augmented Reality ermöglicht es, die physikalische, „echte“ Welt mit digitalen Informationen anzureichern und beide Welten zu verbinden. Die Stärke, aber auch Herausforderung von AR liegt darin, die relevanten und benötigten Informationen zur richtigen Zeit zu liefern. (vgl. Schmalstieg / Höllerer 2016: ix; Hemmerling 2011: 8)

AR findet nicht nur im Entertainment-Bereich Anwendung, sondern in sämtlichen Lebensbereichen, in denen Menschen mit Technik und Maschinen interagieren müssen. Beispielsweise gibt es eine App, die bei Operationen zur Lebertumor-Entfernung anzeigt, wie Blutgefäße verlaufen und so bei der chirurgischen Schnittführung helfen (vgl. Fraunhofer MEVIS 2013). Die Firma Trumpf z. B. setzt Augmented Reality ein, um die Anlagenplanung vor Ort zu visualisieren; bei der Firma Kolbus helfen Augmented-Reality-Anwendungen in der manuellen Produktion, um Fehler beim Zusammenbau von Einzelteilen zu vermeiden. (vgl. Schart o. J.). Bisher werden AR-Anwendungen in der Industrie noch zurückhaltend eingesetzt, weil viele Unternehmen abwarten wollen, wie sich diese Technik weiterentwickelt, oder weil sie noch nicht erkannt haben, welchen Mehrwert die Nutzung von erweiterter Realität für ihr Geschäft haben kann. (vgl. Mehler-Bicher / Steiger 2014: 6)

Auch in Museen werden schon mobile Endgeräte genutzt, um einen interaktiven Museumsbesuch zu gestalten, wie beispielsweise im Bayerischen Nationalmuseum in München, im British Museum in London oder im Sukiennice Museum in Krakau. (vgl. Culture to go 2017)

Die Staatliche Kunsthalle Karlsruhe (SKK) hat in der Vergangenheit eine App entwickelt, die mit dem Google Arts Projects verbunden ist und daraus gespeist wird (vgl. Mernberger (10.04.2017, 10:43 MESZ)). Hierbei werden „Rundgänge“ angeboten, die mit einer Sammlung verschiedener Gemälde aus der Ausstellung und einigen beschreibenden Informationen dazu angereichert sind (Name, Erstelljahr, Künstler, Abmessungen, Herkunft, Material).

Die App wurde bisher 100 bis 500 Mal downgeloadet, innerhalb des letzten Jahres allerdings nicht erweitert (vgl. Mernberger (10.04.2017, 10:43 MESZ)).

Ausstellungsabhängig gibt es zwar einen Multimedialoguide, der allerdings ein deutlich kleineres Display als Tablets hat und für eine AR-Anwendung nicht geeignet ist.

1.2 Aufgabenstellung

Im Rahmen dieser Bachelorthesis wurde ein Konzept für einen interaktiven Museumsführer entwickelt und prototypisch umgesetzt. Dafür wurden Gemälde genutzt, die im Besitz der SKK sind.

Die erstellte Anwendung wurde für Tablets entwickelt, gestaltet und optimiert. Sie ermöglicht es dem Besucher der Kunsthalle, die Kamera des Tablets auf ein Gemälde zu richten und zu diesem Kunstwerk Informationen auf seinem Tablet-Display angezeigt zu bekommen. Hierbei handelt es sich beispielsweise um Informationen wie Name des Kunstwerks, Erstelljahr und Künstler.

Des Weiteren soll die App dazu führen, dass der Betrachter das Bild besser verstehen, einordnen und interpretieren kann. Die App soll Informationen liefern, die für die meisten nicht auf den ersten Blick ersichtlich sind und oft bei Museumsführungen vermittelt werden, z. B. Charakteristika des Malstils, der Epoche, der verwendeten Symbole und der Darstellungen. Die Zusatzinformationen, die in unterschiedlicher medialer Form vorliegen (z. B. Texte, Bildmaterialien, 3D-Grafiken, Videos, Audiospuren etc.), können sich der Nutzer über den Klick auf Markierungen einblenden lassen. Die Wahl der Medien und Inhalte wird für jedes Werk individuell getroffen.

Beim Erstellen des Konzeptes und des Programms wurde darauf geachtet, dass die Anwendung erweitert werden kann, ohne unnötig großen Programmieraufwand nach sich zu ziehen.

Für die Umsetzung wurden die Software-Tools „Unity“ und „Vuforia“ genutzt.

Schließlich wurde sich mit dem Ergebnis kritisch auseinandergesetzt und überlegt, an welcher Stelle noch Optimierungsbedarf besteht.

1.3 Struktur der Arbeit

Zunächst werden in Kapitel 2 die wissenschaftlichen und theoretischen Grundlagen von Augmented Reality erklärt und in Beziehung zu Virtual Reality gebracht. Dabei werden technische Grundlagen erläutert und ein Zusammenhang zum Begriff Multimedia geschaffen.

Anschließend wird in Kapitel 3 kurz erläutert, was unter „Lernen“ zu verstehen ist, wie das Gedächtnis Informationen verarbeitet und welchen Einfluss das Lernen mit Augmented Reality darauf hat.

In Kapitel 4 wird das Konzept des Museumsführers skizziert. Dabei werden die Anforderungen der Zielgruppe mit den Inhalten und Zielen der Anwendung verbunden. Des Weiteren werden der gestalterische und technische Entscheidungen erläutert.

Im nachfolgenden Kapitel wird betrachtet, wie das Konzept mit den vorhandenen Tools umgesetzt wurde. Dabei wird sowohl auf die Unity-Objekte, deren Struktur und Verbindung untereinander als auch auf den Programmiercode Bezug genommen. Das Kapitel schließt mit einer Anleitung ab, wie die Anwendung mit weiteren Gemälden erweitert werden kann. Dabei wird die Struktur der Anwendung deutlich und Teile der Programmiercodes erläutert.

Im Kapitel „Evaluation“ wird erläutert, wie sich die App in einem kleinen Usability-Test und in der SKK bewährt hat. Dabei werden die Kritikpunkte der Kunstvermittlerinnen ebenso beachtet wie die einer Kunststudentin.

Zum Schluss wird ein Fazit gezogen. Hierbei werden Ideen angeführt, wie die App im Anschluss an diese Arbeit optimiert und in anderen Projekt- und Abschlussarbeiten erweitert werden könnte.

In der vorliegenden Arbeit wird das generische Maskulin verwendet. Angesprochen werden stets beide Geschlechter. Aus Gründen der Lesbarkeit wird auf die Nennung beider Formen verzichtet.

2 Augmented Reality

2.1 Augmented Reality im Allgemeinen

2.1.1 Definition Augmented Reality, Mixed Reality und Abgrenzung zu Virtual Reality

2.1.1.1 Definition von „real“ und „virtuell“

Zunächst einmal stellt sich die Frage, was genauer unter „Realität“ zu verstehen ist. In der Lerntheorie des Konstruktivismus gilt die Annahme, dass jeder Mensch seine eigene Realität schafft, die durch seine Erfahrungen und sein Hintergrundwissen beeinflusst wird. (vgl. Boeckmann 2008: 19; Kron 2004: 183, 189). Auch Hemmerling (2011: 14) spricht von der Realität als ein „emergentes Phänomen, das in unserem Gehirn entsteht“.

Der Duden nennt als Synonym zu Realität „Wirklichkeit“ und definiert es als „Zustand, wie man ihn tatsächlich antrifft, erlebt; Bereich dessen, was als Gegebenheit, Erscheinung wahrnehmbar ist“. (Bibliographisches Institut & F. A. Brockhaus AG Mannheim 2006)

In beiden Definitionen wird betont, dass Realität von jedem einzelnen Menschen wahrgenommen wird und nicht für alle Menschen gleich ist.

In dieser Arbeit wird unter Realität die vom Menschen wahrgenommene Umwelt verstanden, was der Definition des Dudens weitestgehend entspricht. Der Umstand, dass sich jeder seine eigene Realität konstruiert, ist zur Unterscheidung zwischen physikalisch existierenden Objekten und computergenerierten Modellen in dieser Arbeit weniger relevant.

Bei näherer Betrachtung der Duden-Definition von „virtuell“ scheint sie im Widerspruch zu „real“ zu stehen: „[...] nicht echt, nicht in Wirklichkeit vorhanden, aber echt erscheinend“. (Bibliographisches Institut GmbH 2017) „echt erscheinend“ spielt vor allem in der Definition von virtueller Realität eine Rolle.

2.1.1.2 Definitionsansätze von Augmented Reality und Virtual Reality

Um einen Kontext für die Begriffe real und virtuell zu schaffen, entwickelte Milgram die Idee des Reality-Virtuality Continuum (RV Continuum): (vgl. Milgram / Colquhoun 1999: 2-9)

„Virtual Environment“ (VE) und „Real Environment“ (RE) stehen jeweils an einem Ende des RV-Continuums. VE muss zwingend vollständig am Computer modelliert worden sein. Auf der anderen Seite liegt das RE, das die „echte Welt“ repräsentiert. Sie ist völlig unmodelliert und frei von computergenerierten Darstellungen.

Alles zwischen diesen beiden Extremfällen ist also teilweise modelliert. Je näher man der Mitte dieser beiden Extremfälle kommt, desto undurchsichtiger ist, ob die Virtualität nun mit Realität erweitert wird (Augmented Virtuality, AV vom engl. „to augment“, zunehmen, erweitern) oder umgekehrt (Augmented Reality, AR). Die Übergänge sind sehr fließend und können auch innerhalb von Anwendungen wechseln. Deshalb führten Milgram / Colquhoun 1999 den Begriff „Mixed Reality“ (MR) ein, der alle Übergangsphasen zwischen RE und VE beinhaltet, nicht jedoch die beiden Extreme (siehe Abbildung 2.1).

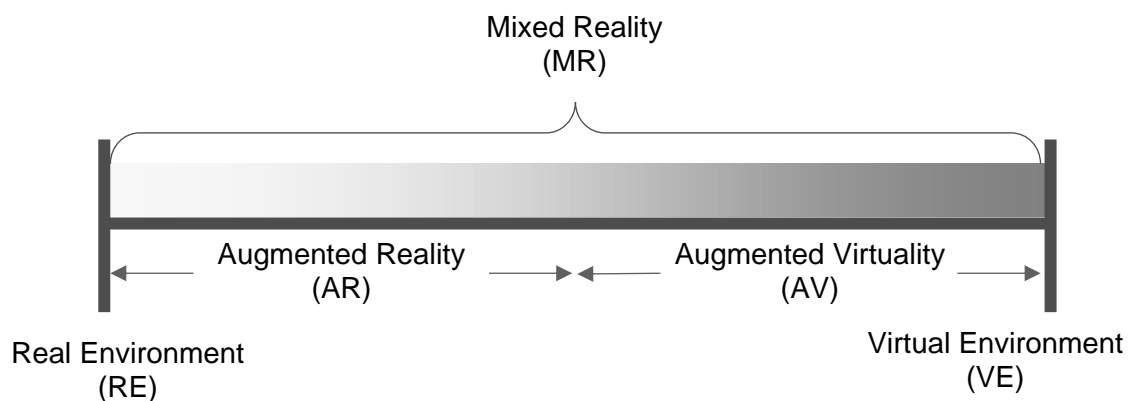


Abbildung 2.1: Reality-Virtuality Continuum (Abbildung in Anlehnung an Milgram / Colquhoun 1999)

Nach Milgram ist Augmented Reality also die Überlagerung der „echten“ Welt mit computergenerierten – also virtuellen – Objekten oder Bildern.

Klein (2009: 1) ergänzt in seiner Definition von VR die von Milgram um die Aspekte „Echtzeit“ und „Interaktivität“. Auch für die Augmentierte Realität, die auf „eine Anreicherung der bestehenden realen Welt um Zusatzinformationen“ abzielt¹ (Klein 2009: 1) ist es zwingend erforderlich, Informationen in Echtzeit zu liefern. Das deckt sich mit Milgrams Grundgedanke.

Eine weitere Definition stellte Azuma 1997 auf. Er erweiterte seine Definition um drei Charakteristika, die in den Definitionen von Milgram und Klein weniger Beachtung finden:

¹ Original: „Augmented reality overlays extra information on real scenes: Typically computer-generated graphics are overlaid into the users field-of-view to provide extra information about their surroundings, or to provide visual guidance for the completion of a task.“ (Übersetzung R. D.)

- Kombination von virtueller Realität und realer Umwelt
- Interaktivität in Echtzeit
- Dreidimensionaler Bezug virtueller und realer Objekte²

(vgl. Azuma 1997: 356)

Vor allem der dreidimensionale Bezug der virtuellen und realen Objekte zueinander, den Azuma erwähnt, führt dazu, dass eine Anwendung oft nicht als Augmented Reality definiert werden kann. Dies ist z. B. der Fall, wenn zweidimensionale Textinformationen eingeblendet werden. Wenn dieser dreidimensionale Bezug zwischen der echten Welt und der Anzeige des virtuellen Objektes nicht gegeben ist, spricht Mehler-Bicher / Steiger (2014: 11f) von „AR im weiteren Sinne“.

In dieser Arbeit wird auf eine Unterscheidung zwischen AR im engeren bzw. weiteren Sinne verzichtet. Die Definition von Azuma ist sehr restriktiv, sodass in dieser Arbeit die Definition von Klein herangezogen wird.

2.1.2 Stand der Technik

2.1.2.1 Historie von Augmented Reality

Vor allem die Spieleindustrie war und ist eine treibende Kraft für die Entwicklung realitätsnaher Darstellungen für ein außergewöhnliches Spielerlebnis und das vollständige Eintauchen in die virtuelle Welt. Gerade im Spielbereich ist das Agieren in Echtzeit und Interaktivität von großer Bedeutung. (Hemmerling 2011a: 16)

Schon 1962 versuchte Morton Heilig mit seinem Sensorama, einem Projektor, einen möglichst hohen Grad an Immersion, also dem völligen Eintauchen in die virtuelle Welt, zu erzeugen. Dafür nutzte er Vibrationen, Geruchsgeneratoren, 3D-Displays, Windmaschinen und Stereo-Sound. (vgl. Hemmerling 2011: 17; Axworthy 2016)

Ivan Sutherland arbeitete 1968 als einer der ersten an der Entwicklung eines Head-Mounted Displays (siehe Abschnitt 2.2.4.1), das wegen seines hohen Gewichtes von der Raumdecke gehalten werden musste. Hier war der Nutzer in der Lage, die reale Umgebung vollständig auszublenden und völlig in die virtuelle Welt einzutauchen. (vgl. Schmalstieg / Höllerer 2016: 5; Hemmerling 2011: 16f).

In einer Arbeit von Tom Caudell und David Mizell Anfang der 1990er Jahre wurde das Wort „Augmented Reality“ erstmals verwendet. (vgl. Schmalstieg / Höllerer 2016: 5).

² Original: „1. Combines real and virtual, 2. Is interactive in real time, 3. Is registered in three dimensions“ (Übersetzung R. D.)

1993 entwickelten Steven Feiner, Blair MacIntyre und Doree Seligmann das System KARMA, das erste AR-System, das automatisch richtige Anleitungssequenzen für Reparatur- und Wartungsprozeduren auf einem Head-Mounted Display anzeigen konnte. (vgl. Schmalstieg / Höllerer 2016: 6; Mehler-Bicher / Steiger 2014: 13)

1993 entstand außerdem das System „Chameleon“, ein Vorreiter der Handheld AR. Es zeigte kontextsensitive Informationen an, wenn der Nutzer das tragbare LCD-Display im Raum bewegte und z. B. auf eine Karte richtete. (vgl. Schmalstieg / Höllerer 2016: 6)

1999 kam ARToolKit als erste Open-Source-Software-Plattform für Augmented Reality auf den Markt. Die zweidimensionalen schwarz-weißen Marker sind Teil einer Tracking Library und können mit einem handelsüblichen Laserdrucker ausgedruckt werden. (vgl. Schmalstieg / Höllerer 2016: 12)

2.1.2.2 Zukunftsaussichten

Aktuell gibt es noch einige Herausforderungen in den Technologien bei den verwendeten Technologien, die noch bewältigt werden müssen. Beispielsweise sind GPS-Signale, die für ein Tracking genutzt werden könnten, noch zu ungenau, um in Räumen zu funktionieren (vgl. Abschnitt 2.2.2.1.1.4). Auch im Bereich Motion Tracking gibt es noch Verbesserungsbedarf (vgl. Stannel / Zyda / Hale 2015: 10). Vor allem Sensoren und Hardware werden sich sehr wahrscheinlich in den kommenden Jahren verbessern. Der Forschungsbedarf und das Potenzial, weitere technische Neuerung zu entwickeln, ist sehr hoch. Das Project Tango beispielsweise, eine Initiative von Google, nutzt und stellt Smartphones und Tablets her, deren Sensoren pro Sekunde 250.000 3D-Messungen durchführen und so ein 3D-Modell der Umwelt in Echtzeit aufnehmen. Dieses bildet die Basis für eine Erweiterung der Realität in Echtzeit. (vgl. Google ATAP 20.02.2014)

Neben den bereits genannten Problemfeldern stellt die Bildregistrierung häufig eine Herausforderung dar. (vgl. Barfield 2016: 10, Azuma 1997: 367-375) Barfield (2016: 10) geht davon aus, dass diese Herausforderung durch die steigende Anzahl an Sensoren in unserem Alltag gelöst werden kann. Ebenso ordnet er es als wahrscheinlich ein, dass Augmented Reality weiteren Einzug in unser Leben erhalten wird, wenn die Technologien am eigenen Körper getragen werden können. Erste Versuche mit Kontaktlinsen gibt es bereits (siehe Abschnitt 2.2.4.2), sodass diese möglicherweise mobile Datenbrillen ablösen könnten.

Durch die starke Verbreitung und die allgegenwärtige Nutzung von Smartphones, die ununterbrochen Daten über unsere Interessen, unseren Standort, unsere Bewegungsabläufe und Ähnliches sammeln, werden Themen wie Privatsphäre und Datensicherheit immer wichtiger. Die gesammelten Daten können beeinflussen, wie die „reale Welt“ ergänzt wird und somit weitreichende Folgen haben. Während die Technologie

sich weiterentwickelt, müssen gleichzeitig Gesetze und Regularien beschlossen werden, die den Missbrauch der gesammelten Daten vermeiden und ein sicheres Zusammenleben und die Privatsphäre schützen. (vgl. Barfield 2016: 7f)

2.1.3 Charakteristika einer Augmented-Reality-Anwendung

Augmented-Reality-Anwendungen haben den Zweck, die „reale“ Welt um virtuelle Objekte zu erweitern. Dabei ist bei der Konzipierung einer Anwendung abzuwägen, ob nur die visuelle oder auch die auditive und haptische Wahrnehmung angesprochen werden sollen, was zu einer höheren Immersion führen kann (siehe Abschnitt 2.3.7.3). (vgl. Azuma 1997: 361)

Eine Eigenschaft, die AR von VR unterscheidet und berücksichtigt werden muss ist die Tragbarkeit. Bei Virtual Reality ist der Nutzer an einen Ort in der realen Welt gebunden; die Anwendung funktioniert nur, wenn der Nutzer sich an einem bestimmten Ort aufhält, z. B. wenn die Sensoren für das Tracking fix an einem Ort stehen. Bei Augmented Reality ist es gewollt, dass sich der Nutzer in seiner Umgebung bewegt, umherläuft und eine deutlich höhere Bewegungsfreiheit hat. Dennoch muss sich der Nutzer an der Stelle befinden, an der die Aufgabe des AR-Systems erfüllt werden kann. (vgl. Azuma 1997: 366)

Unabhängig von den angesprochenen Sinnen und der eingesetzten Technologie sollten vor allem mobile AR-Anwendungen folgende Charakteristika aufweisen, damit sie sich auf dem Markt überhaupt durchsetzen können: (vgl. Wagner, 2007: 10)

- **Geringe Kosten**

Die Geräte müssen kostengünstig sein und jederzeit an jedem Ort zur Verfügung stehen.

- **Robuster und fehlertoleranter Umgang**

Vor allem ungeübte Nutzer sollten das Gerät und die Anwendung intuitiv bedienen können.

- **Autarkes und netzwerkbezogenes Arbeiten**

Das System sollte sowohl alleine als auch in Verbindung mit anderen Geräten arbeiten können.

- **Tracking in Echtzeit**

Die Trackingsysteme müssen verlässlich und einfach sein. Dafür sollten sie von den eingebauten Funktionalitäten des Endgerätes profitieren.

- **Schnelles Erstellen von Prototypen**

Um den Markt schnell mit AR-Anwendungen zu füllen, müssen Konzepte für Apps schnell entstehen und umgesetzt werden können.

- **Standardisiertes Erstellen von Inhalten**

Um einen praktischen Nutzen aus der Anwendung zu ziehen, ist es nötig, Inhalt dafür nicht „irgendwie“, sondern auf standardisiertem Wege in die Anwendung einzupflegen.

Beim Aufbau der Anwendung sollten folgende Anforderungen erfüllt sein, um eine hohe Nutzerfreundlichkeit – unabhängig vom inhaltlichen Aufbau – zu erreichen: (vgl. Mehler-Bicher / Steiger 2014: 78)

- **Funktionalität**

Die Anwendung „muss dem Sachverhalt angemessen sein und die Funktionen bieten, die in der gewünschten Situation erforderlich sind.“ (Mehler-Bicher / Steiger 2014: 78)

- **Intuitive Handhabung**

Die Anwendung muss intuitiv bedienbar sein bzw. dem Nutzer dieses Gefühl vermitteln.

- **Direkte Manipulierbarkeit**

Die Anwendung muss auf die Aktionen des Nutzers reagieren: Bewegungen führen zu Änderungen des Bildschirminhaltes.

- **Echtzeitanwendung**

Die Anwendung muss alle Berechnungen in Echtzeit durchführen.

- **Spezifität**

Die Anwendung bestimmt, welche Marker, Tracking-Verfahren etc. genutzt werden.

- **Anwendbarkeit**

Die Anwendung muss auf Endgeräten zur Verfügung stehen, die der Nutzer idealerweise schon besitzt.

Bei der Wahl der Ausgabegeräte müssen zusätzlich einige technischen Parameter berücksichtigt werden. Allgemein gilt, dass die Anforderungen an Head-Mounted Displays – wie Farben und Auflösungen – bei AR in der Regel weniger hoch als bei VR sind, da der Nutzer immer noch die echte Welt in seine Wahrnehmung miteinbezieht. (vgl. Azuma 1997: 367)

2.1.4 Anwendungsszenarien und Anwendungsbeispiele

AR ist vor allem im Alltag für unterschiedliche Einsatzbereiche interessant, da die reale Umwelt mit einbezogen wird. Im Folgenden werden einige Einsatzgebiete beispielhaft vorgestellt, um einen groben Überblick zu geben. (vgl. Broll 2013: 288ff) Die Liste erhebt keinen Anspruch auf Vollständigkeit, da beinahe täglich neue vielfältige Anwendungen entwickelt werden und der Fantasie kaum Grenzen gesetzt sind.

- **Training und Wartung**

z. B. durch Einblenden von Hinweisen, Anweisungen, Wegen, Werkzeugen, Ersatzteile etc. oder wenn ein Einzelner nicht für alle eintretenden Fälle geschult sein kann

- **Fernsehübertragungen**

z. B. durch Einblenden von Hilfsinformationen beim Football (Line of Scrimmage und Goallinie), Fußball (Abseitslinie), Bobfahren (ideale Kurvenlage) etc.

- **Militärische Applikationen**

z. B. in Helmen zur Darstellung von Liniengrafiken. In naher Zukunft wahrscheinlich auch mit See-Through-Displays (siehe Abschnitt 2.2.2.2) für Geländeinformationen, Satelliten, Flugzeugdaten etc.

- **Lehre, Ausbildung und Museen**

z. B. durch das interaktive Erleben von Inhalten, durch das Erklären eines Effekts in Verbindung mit dem Exponat etc.

- **Architektur und Städteplanung**

z. B. für städtebauliche Veranschaulichung der zukünftigen Innenstadt mit Einbeziehung der Umwelt

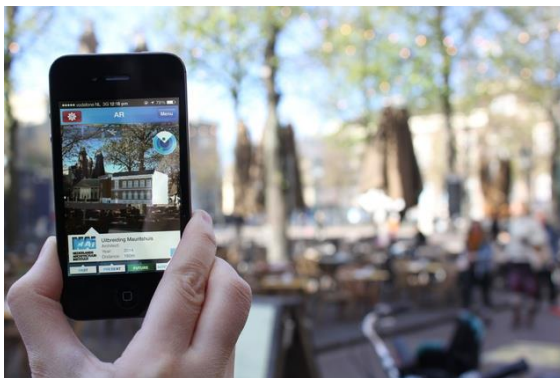


Abbildung 2.2: UAR-App für Den Haag, die in Den Haag alte und zukünftige Gebäude zeigt (Abbound 2014)

- **Medizin**
z. B. für Chirurgie (minimalinvasive Eingriffe, Leberoperationen etc.) durch Kombination mit weiteren Bilddaten
- **Information, Navigation und Tourismus**
z. B. für Sehenswürdigkeiten oder Parkassistenten
- **Archäologie und Geschichte**
z. B. zur Ergänzung unvollständiger Fundstücke und Gebäude, zur Darstellung früherer Zeiten oder zur Ergänzung eines Szenarios
- **Spiele und Unterhaltung**
z. B. für Smartphones und Spielekonsolen, aber auch ergänzend zu Brettspielen (z. B. Scotland Yard Master von Ravensburger)

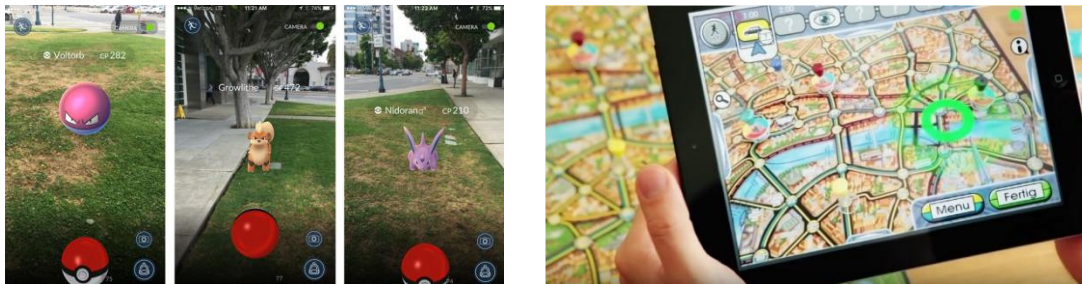


Abbildung 2.3: Links oben: Pokémon Go: Pokémons und Pokébälle werden als virtuelle Objekte über die reale Welt gelegt (Matthew Szymczyk 2016). Rechts oben: das Brettspiel Scotland Yard – Master (Ravensburger AG 2013: Sek. 29).

2.1.5 Nutzen von Augmented-Reality-Anwendungen

Wie aus den Anwendungsbeispielen hervorgeht, kann AR für den Menschen folgenden Nutzen haben: (vgl. Mehler-Bicher / Steiger 2014: 21)

- Erweiterung der menschlichen Wahrnehmung durch computergenerierte Objekte
- Visualisierung von komplexen, abstrakten oder umfangreichen Informationen
- Unterstützung bei der Bewältigung komplexer Aufgaben (z. B. im Bereich Medizin, Konstruktion etc.)
- Verringerung von „Time-to-Content“ („Zeitdauer, die benötigt wird, bis ein Anwender über die von ihm gewünschte Informationen verfügt“ (Mehler-Bicher / Steiger 2014: 21))
- Kontextabhängige Anpassung der Inhalte durch den Nutzer
- Verbinden von haptischem und digitalem Erlebnis

2.2 Technische Grundlagen von AR

2.2.1 Ablauf bei AR-Anwendungen

Augmented Reality lässt sich in fünf kleinere Prozeduren unterteilen: (vgl. Broll 2013: 242-245)

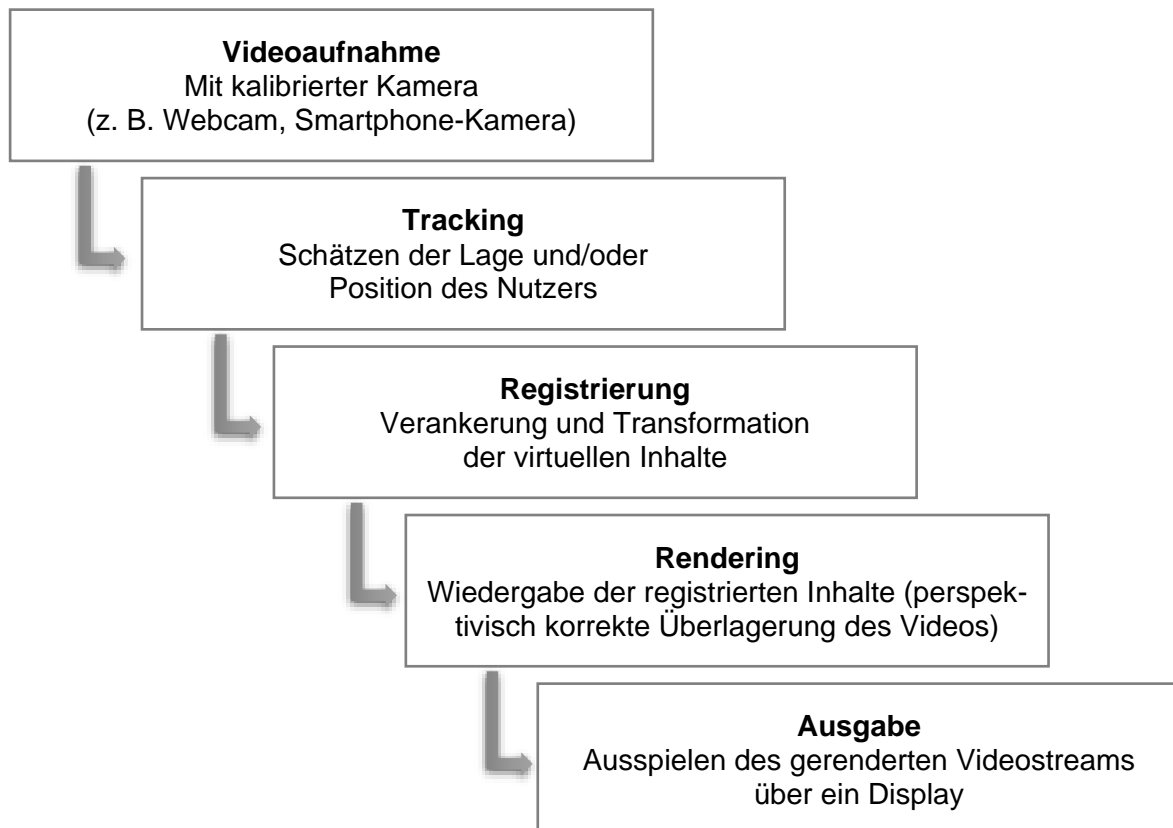


Abbildung 2.4: Ablauf einer AR-Anwendung

Die Kamera zur Videoaufnahme muss kalibriert sein. Bei einer Kalibrierung werden Sensoren eingestellt und an eine Referenz, also bekannten Werten oder einem Koordinatensystem, angepasst, um zu gewährleisten, dass die gemessenen Werte in einer bekannten Größenskala vorliegen und verarbeitet werden können. Oft findet die Kalibrierung im Hintergrund und automatisch statt. (vgl. Schmalstieg / Höllerer 2016: 86f)

Tracking, Registrierung und Ausgabe werden in den folgenden Kapiteln genauer beschrieben.

2.2.2 Tracking

Tracking (zu Deutsch „Verfolgung“) ist die Schätzung der Position und/oder der Lage/Orientierung der verwendeten Kamera. Damit werden dann die Koordinaten aus

dem Koordinatensystems des aufgenommenen Videostreams in das Koordinatensystem der virtuellen Welt transformiert Grund dafür ist, dass der Bildpunkt des Betrachters für eine durchgehende Darstellung zu jeder Zeit bekannt sein muss. Dies geschieht meist durch verschiedene Sensoren oder künstliche Marker. (vgl. Broll 2013: 242f; Schmalstieg / Höllerer 2016: 86).

Da sich der Nutzer in seiner realen Umgebung bewegt, ändert sich die Position ständig. Die Bewegung eines Objektes in einem Raum kann unterteilt werden in Translation (Verschiebung) und Rotation (Drehung) entlang der drei Achsen des Koordinatensystems. Das hat zur Folge, dass auch die Bewegung eines Körpers durch sechs Werte definiert werden kann: drei Koordinaten zur Beschreibung der Position im Raum und drei Winkel zur Beschreibung der Orientierung. „Diese voneinander unabhängigen Bewegungsmöglichkeiten werden als Freiheitsgrade (Degrees of Freedom, DOF) bezeichnet“. (Grimm et al. 2013: 99). Beim Tracking werden diese Freiheitsgrade kontinuierlich geschätzt. (vgl. Grimm et al. 2013: 98f)

Tracking-Systeme können nach mehreren Kriterien unterschieden werden. Die gängigste Unterteilung ist die nach optischen und nichtoptischen Verfahren. Eine weitere Unterteilung ist nach dem Einsatz und der Funktion der Kameras bzw. Sensoren der AR-Systeme (hier in Abschnitt 2.2.2.2 nach Mehler-Bicher / Steiger 2014 „Prinzipien“ genannt).

2.2.2.1 Tracking-Verfahren

2.2.2.1.1 Nichtoptische Trackingverfahren

2.2.2.1.1.1 Mechanisches Tracking

Beim mechanischen Tracking werden die Bewegungen des Nutzers über Mechaniken wie Seilzüge oder Gestänge aufgenommen. An den Gelenken, Zahnrädern o. Ä. werden die Veränderungen der Winkel und Abstände gemessen und an das System weitergeleitet. Die Eingabegeräte können zudem als Ausgabegeräte dienen. Obwohl die Messungen sehr genau sind, werden mechanische Trackingsysteme im AR-Bereich kaum eingesetzt, da sie die Mobilität der Nutzer deutlich einschränken. Dieses Trackingverfahren ist ein stationäres Trackingverfahren. (vgl. Grimm et al. 2013: 110f, vgl. Schmalstieg / Höllerer 2016: 96f)

2.2.2.1.1.2 Akustisches Tracking

Akustische Trackingsysteme, oft auch Ultraschall-Tracking genannt, funktionieren mit Schallwellen im Ultraschall-Bereich (ab 20.000 Hz), der für den Menschen nicht hörbar ist. Gemessen wird dabei die Zeit, die Schallwellen brauchen, um vom Sender zum Empfänger zu kommen. Entweder der Sender oder der Empfänger ist am sich bewege-

genden Objekt angebracht. Um alle sechs Freiheitsgrade zu messen, werden drei Sender und drei Empfänger benötigt, für die drei Freiheitsgrade der Translation ein Sender und drei Empfänger. Die Sender reagieren sehr empfindlich auf Luftdruck- und Temperaturänderungen, sodass die Systeme häufig kalibriert werden müssen. Außerdem dürfen für eine aussagekräftige Messung die Wellen nicht behindert werden. Dieses Trackingverfahren ist ein stationäres Trackingverfahren. (vgl. Grimm et al. 2013: 111f; Schmalstieg / Höllerer 2016: 98)

2.2.2.1.1.3 Elektromagnetisches Tracking

Beim elektromagnetischen Tracking werden mit Spulen Magnetfelder aufgebaut. Die Magnetfelder ändern sich über die Zeit. Als Sensoren dienen ebenfalls Spulen, in die bei der Bewegung durch das Magnetfeld fortwährend Ströme induziert werden. Die Ströme, die in der Empfängerspule gemessen werden, dienen als Messung für Orientierung und Position, also für alle sechs Freiheitsgrade. Dieses Tracking-Verfahren findet allerdings nur Anwendung, wenn es keine anderen störenden Magnetfelder oder ferromagnetische Materialien im Raum gibt. Elektromagnetisches Tracking wird heutzutage nur noch selten verwendet. Dieses Trackingverfahren ist ein stationäres Trackingverfahren. (vgl. Grimm et al. 2013: 112; Schmalstieg / Höllerer 2016: 97)

2.2.2.1.1.4 GPS-Tracking

GPS-Sensoren (Global Positioning System) messen die Laufzeit (Time of Flight) von Radiosignalen aus der Erdumlaufbahn. Da die Lage der Satelliten, die die Strahlen aussenden, bekannt ist, kann ferner die Position (nicht allerdings die Orientierung) des GPS-Empfängers bestimmt werden. Die Genauigkeit variiert zwischen 1 und 100 Metern, da die Messung von der Anzahl der Satelliten, der Qualität des Empfängers und den Umständen des Signalempfangs abhängt. GPS-Tracking funktioniert zuverlässig nur im Freien, da die Radiosignale von Wänden zurückgeworfen werden. (Genauigkeit bei Smartphone-GPS: 5-10 m in Stadtgebieten („Urban Canyons“), 0.5-5 m im „freien Feld“). Dieses Trackingverfahren ist ein mobiles Trackingverfahren. (vgl. Schmalstieg / Höllerer 2016: 99ff)

Genauere Messungen können durch verschiedenen Differenzmethoden (z. B. Differential GPS oder Real-time Kinematics GPS) erzielt werden. (vgl. Schmalstieg / Höllerer 2016: 100; Broll 2013: 254) Auf diese soll an dieser Stelle aber nicht näher eingegangen werden.

2.2.2.1.1.5 Kabellose Netzwerke

In einem kabellosen Netzwerk, das z. B. durch WLAN oder Bluetooth aufgebaut wird, erhält jedes mobile Gerät eine ID. Nachdem die ID in der Datenbank identifiziert wurde, kann die Position des Gerätes durch Basisstationen im Netzwerk auf einige Meter genau bestimmt werden. Durch Wände o. Ä. werden Gebiete abgeschirmt. Daher

eignet sich für Indoor-Tracking Bluetooth-Tracking besonders gut. Hierbei sparen kleine Sender, sogenannte Beacons, Energie und sind kostengünstig. Diese Trackingverfahren sind mobile Trackingverfahren. (vgl. Schmalstieg / Höllerer 2016: 101f)

2.2.2.1.1.6 Magnetometer

Weiterhin können Objekte mit einem elektronischen Kompass (Magnetometer) getrackt werden. Dabei wird die Lage des Geräts an drei Achsen (also drei Freiheitsgrade) relativ zum Magnetfeld, genauer gesagt zum Nordpol, berechnet. Die Berechnungen sind allerdings oft unzuverlässig, da das Magnetfeld z. B. von anderen elektronischen Geräten beeinflusst werden kann. Dieses Trackingverfahren ist ein mobiles Trackingverfahren. (vgl. Schmalstieg / Höllerer 2016: 102)

2.2.2.1.1.7 Inertial-Tracking

„Inertial-Tracking basiert auf Inertial-Sensoren (auch Trägheits- oder Beschleunigungssensoren genannt), das heißt auf Sensoren, welche die Beschleunigung messen.“ (Grimm et al. 2013: 112) Es gibt jeweils Sensoren, die entweder die Beschleunigungen entlang einer Achse (Lineare Accelerometer) oder die Beschleunigungen bei Rotationen um eine Achse (Gyroskop) messen. Die Inertial-Sensoren dienen primär dazu, die drei DOF der Orientierung zu messen. In mobilen Endgeräten werden in der Regel jeweils drei senkrecht zueinanderstehende Magnetometer, Gyroskope und lineare Beschleunigungsmesser eingebaut. Dieses Trackingverfahren ist ein mobiles Trackingverfahren. (vgl. Broll 2013: 255)

2.2.2.1.1.8 Hodometer

Hodometer wurden bisher bei Robotern eingesetzt, um zurückgelegte Wegstrecken zu messen. Das Prinzip ist wie bei einer mechanischen Computermaus mit Gummikugel: Ein Rad misst die Anzahl der Richtungsänderungen der Kugel. Dieses Trackingverfahren ist ein mobiles Trackingverfahren. (vgl. Schmalstieg / Höllerer 2016: 104)

2.2.2.1.2 Optisches Tracking

Beim optischen Tracking werden Bilder der Umgebung aufgenommen und aus diesen Bildern anschließend wichtige Informationen ausgelesen, um damit dann die Position und Orientierung im Raum zu bestimmen. (vgl. Mehler-Bicher / Steiger 2014: 27)

Optische Trackingsysteme können auf verschiedene Arten weiter unterteilt werden. In dieser Arbeit wird die Einteilung von Maiero (2009: 35f) (basierend auf Teichrieb et al. 2007: 2) zur Hand genommen:

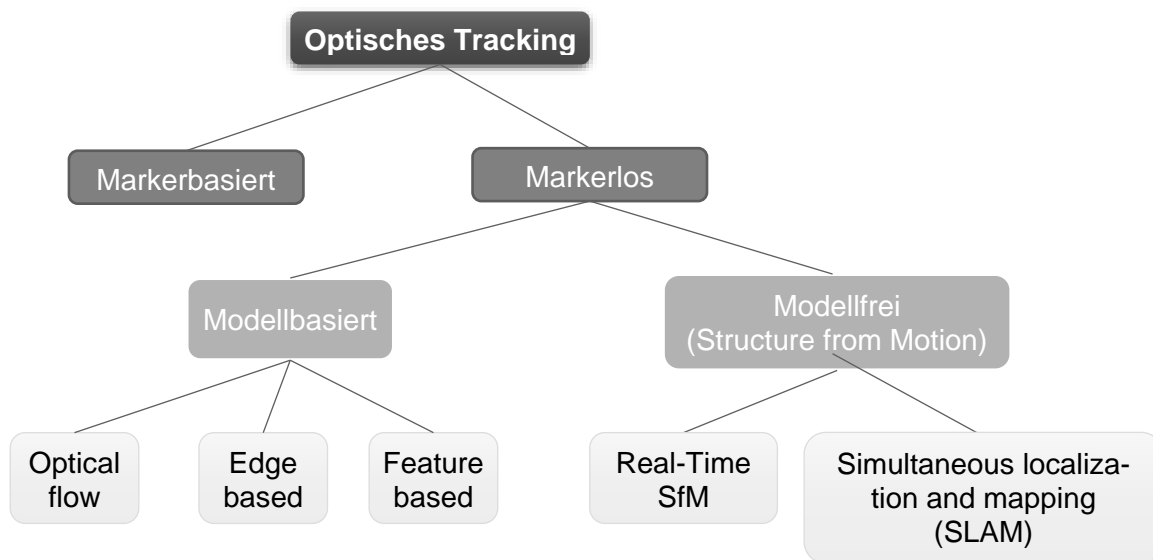


Abbildung 2.5: Einteilung der optischen Tracking-Verfahren (in Anlehnung an (Teichrieb et al. 2007: 2))

Nach dieser Einteilung werden die Verfahren grundlegend nach markerbasiert oder markerlos unterteilt. Hierbei ist ein Marker ein „zwei- oder dreidimensionales Objekt, das durch seine Art und Form leicht durch eine Kamera identifiziert (getrackt) werden kann.“ (Mehler-Bicher / Steiger 2014: 28) Die markerlosen Verfahren werden wiederum unterschieden, ob sie die aufgenommenen Bilder mit einem Referenzmodell abgleichen oder nicht.

2.2.2.1.2.1 Markerbasiert

Beim markerbasierten Verfahren werden meist schwarz-weiße künstliche Marker eingesetzt. Sie sind so gestaltet, dass der Trackingvorgang möglichst schnell starten kann und die Kamerapose möglichst schnell berechnet wird. Marken müssen oft komplett im Sichtfeld der Kamera liegen, damit sie erkannt werden, dürfen aber nicht zu klein sein, damit sie sich noch deutlich von anderen Markern unterscheiden. (vgl. Mehler-Bicher / Steiger 2014: 28; Broll 2013: 256)

Künstliche Marker werden eingesetzt, wenn die Kamerabilder keine natürlichen Merkmale enthalten (s. u.), wenn die Objekte glänzen und so die Aufnahme je nach Blickwinkel beeinträchtigen, oder wenn sich Elemente auf dem zu trackenden Objekt wiederholen und damit nicht klar wird, auf welche Stelle des Objektes die Kamera gerade genau gerichtet wird. (vgl. Schmalstieg / Höllerer 2016: 109)

Marker können durch Software-Tools wie z. B. ARToolkit effizient erstellt werden. Der erstellte Marker wird ausgedruckt und auf dem Objekt oder auf Wänden angebracht.

Jedoch muss beachtet werden, dass Marker von der Anwendung nicht mehr erkannt werden können, wenn sie teilweise verdeckt werden. (vgl. Broll 2013: 257)

Beim Tracken wird zunächst im aufgenommenen Videostream nach zusammenhängenden Liniensegmenten gesucht, die mit den vordefinierten Markern abgeglichen werden. Sollte ein definierter Marker gefunden werden, können aufgrund der Eckpunkte die Position und Orientierung der Kamera zum Marker berechnet bzw. geschätzt werden. (vgl. Broll 2013: 258)

2.2.2.1.2.2 Markerloses modellbasiertes Tracking

Beim modellbasierten Tracking liegen CAD-Daten vor. Die Daten aus dem aufgenommenen 2D-Videostream werden mit vorher hinterlegten 3D-Modellen abgeglichen. Daraufhin wird das Ausgabebild mit den richtigen Positionen und Orientierungen gerendert. (vgl. Teichrieb et al. 2007: 2)

Beispiele für die Anwendung von modellbasiertem Tracking finden sich in der Computerspielindustrie für die Verfolgung menschlicher Bewegungen. Hierfür werden mittlerweile auch in Smartphones RGB-D-Kameras (D steht für engl. Depth, also Tiefe) eingebaut, damit die Bildtiefe ebenso in die Berechnungen einfließen kann. (vgl. Schmalstieg / Höllerer 2016: 108f)

Modellbasiertes Tracking kann über drei unterschiedliche Algorithmentypen stattfinden, die die Merkmale aus dem Videostream herausziehen: kantenbasiert (edge based), optischer Fluss (optical flow) und texturbasiert (feature based). (vgl. Maiero, 2009: 36f) In dieser Arbeit soll auf diese und weitere Algorithmen und Techniken nicht eingegangen werden.

2.2.2.1.2.3 Markerloses modellfreies Tracking

Wenn das Modell während des Trackings erstellt wird, also „on the fly“, spricht man von modellfreiem Tracken, weil keine 3D-Modelldatei hinterlegt sind. Eine andere gebräuchliche Bezeichnung ist Structure from Motion, SfM (vgl. Teichrieb et al. 2007: 5). Das modellfreie Tracking ist wesentlich flexibler als das modellbasierte Tracking. Mit diesem Ansatz wird das Objekt relativ zum Startpunkt in die reale Welt eingebaut; es gibt keine Möglichkeit, das Objekt vorher zu registrieren. (vgl. Schmalstieg / Höllerer 2016: 106) Simultaneous Localization and Mapping (SLAM) ist eine Weiterentwicklung des Structure-from-Motion-Ansatzes mit einem komplexen Algorithmus. An dieser Stelle soll hierauf nicht weiter eingegangen werden.

2.2.2.1.2.4 Merkmalbasiertes Tracking

Ein anderer Ansatz, wie Trackingsysteme gruppiert werden können, ist der Ansatz der Merkmalsuche. Bei klassischen Schwarz-Weiß-Markern (engl. auch „fiducial“ genannt) werden künstliche Merkmale (engl. „feature“) aus dem Kamerabild herausgesucht. Ziel

der Entwicklung ist es aber, auf künstliche Marker verzichten zu können und stattdessen die Umwelt zu nutzen. Dafür werden natürliche Merkmale genutzt. Damit natürliche Merkmale erkannt werden, ist es essentiell, eine gute Bildqualität und somit eine gute Kamera zu benutzen. (vgl. Schmalstieg / Höllerer 2016: 109-112)

Beim Arbeiten mit natürlichen Merkmalen werden Interest Points per Algorithmus aus dem Bild extrahiert und bestimmt. Interest Points sind herausragende, auffallende Stellen auf dem Zielobjekt, die allerdings sehr dicht auf einer unregelmäßigen Oberflächenstruktur vorkommen sollten. Das aufgenommene Kamerabild wird mit Referenzbildern (engl. „keyframes“) verglichen, die vorher von einem bestimmten Aussichtspunkt aus aufgenommen wurden. (vgl. Schmalstieg / Höllerer 2016: 112f)

Ein Marker basierend auf der merkmalsbasierten Technologie „besteht somit aus einer Stelle oder Fläche im Bild und einem Maß, welches das Merkmal beschreibt [...]“. (Kathe, 2015: 7) Das merkmalsbasierte Trackingsystem besteht aus drei Komponenten. Zunächst einmal müssen die Merkmale aus dem Kamerabild extrahiert werden; das geschieht mit Hilfe des Extraktors (z. B. der SIFT- oder SURF-Algorithmus). Der Deskriptor gibt diese Merkmale an den Bezeichner, der später andere Merkmale aus einem anderen Kamerabild den extrahierten Merkmalen zuordnet. Der Zuordner ordnet anschließend die Merkmale zwischen dem Kamerabild und dem Referenzobjekt zu. Aus der Zuordnung wird dann die Position und Orientierung im Raum berechnet. (vgl. Kathe, 2015: 7f)

2.2.2.2 Tracking-Prinzipien

2.2.2.2.1 Outside-In

Beim Outside-In-Prinzip werden Kameras (oder andere Sensoren) von außen auf ein bewegtes Objekt gerichtet (z. B. den Nutzer mit einem Head-Mounted-Display), dessen Position geschätzt werden soll. Meistens werden mehrere Kameras/Sensoren verwendet, um den Raum zu vergrößern, in dem getrackt werden kann. Der Nutzer muss zwar keine schwere Technik tragen, allerdings häufig Marker an sich haben. (vgl. Grimm et al. 2013: 109) Das Outside-In-Verfahren ist wegen der zum Teil sehr großen Anzahl an Kameras und der aufwändigen Technik sehr teuer. (vgl. Mehler-Bicher / Steiger 2014: 26)

2.2.2.2.2 Inside-Out

Beim Inside-Out-Prinzip sind die Kameras/Sensoren am sich bewegendem und zu trackenden Objekt angebracht. Die Positionsbestimmung erfolgt dann durch Referenzpunkte in der Umgebung. Mit diesem Verfahren hat der Nutzer eine große Bewegungsfreiheit. (vgl. Grimm et al. 2013: 109)

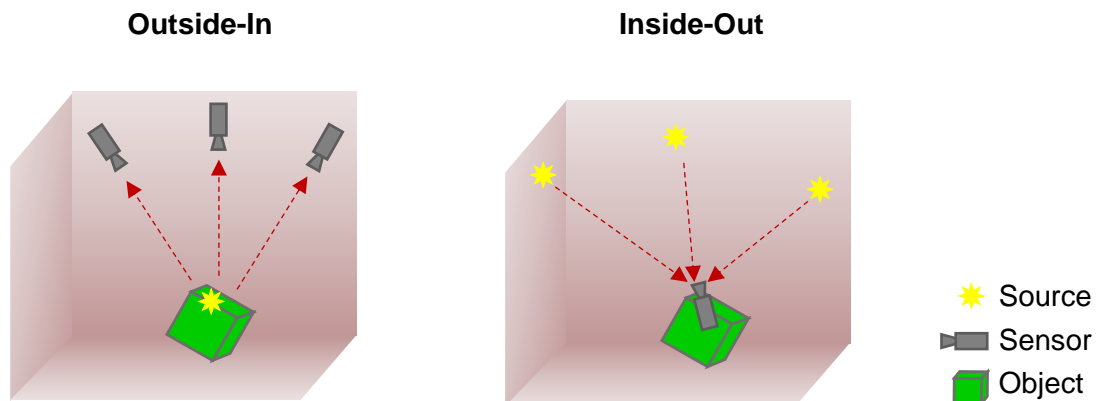


Abbildung 2.6. Unterschied zwischen Outside-In- und Inside-Out-Tracking (Schmalstieg / Höllerer 2016: 93)

2.2.3 Registrierung

Unter Registrierung im Zusammenhang mit Augmented Reality versteht man „die Verankerung oder das korrekte Einpassen der künstlichen virtuellen Inhalte in die Realität.“ (Broll 2013: 243). Die Koordinatensysteme der computergenerierten Inhalte und das Koordinatensystem, das aus dem Tracking hervorgegangen ist und die reale Welt abbildet, werden so übereinandergelegt, dass die virtuellen Objekte fest verankert („registriert“) erscheinen. Damit hat jedes virtuelle Objekt einen festen Platz in der Realität (solange sich das virtuelle Objekt nicht bewegt). (vgl. Broll 2013: 243)

Damit die künstlichen Inhalte möglichst realistisch aussehen, muss zum einen die Perspektive, zum anderen die Beleuchtung angepasst werden.

2.2.3.1 Geometrische Registrierung

Die perspektivische Anpassung wird als geometrische Registrierung bezeichnet. Bei der geometrischen Registrierung werden die computergenerierten Inhalte perspektivisch korrekt in die reale Welt eingebaut. Die Position und Orientierung, die im Tracking geschätzt worden sind, sind dafür die Basis. Wenn sich der Nutzer bewegt und damit seinen Blickpunkt ändert, muss sich die Ausgabe auf dem Display anpassen. Die Transformationen werden neu berechnet. Es soll immer noch so aussehen, als ob sich das virtuelle Objekt an der gleichen Stelle in der Realität befindet. Eine Ausnahme kann vorkommen, wenn es sich um eine Animation handelt und sich das virtuelle Objekt bewegen soll.



Abbildung 2.7: Links: Das Objekt ist korrekt geometrisch registriert. Rechts oben: Das Objekt erscheint an der exakt gleichen Stelle. Ohne Registrierung wird die Perspektivenänderung beim Ändern des Blickwinkels nicht miteinbezogen. Rechts unten: Mit geometrischer Registrierung kann das virtuelle Objekt perspektivisch korrekt gemäß dem Blickpunkt und der Blickrichtung eingepasst werden. Grundlage hierfür sind die Tracking-Daten. (Broll 2013: 265)

Damit die Qualität der Registrierung gut ist, muss darauf geachtet werden, dass die Tracking-Rate, also die Anzahl der Trackingergebnisse pro Sekunde, genauso hoch ist, wie die Frequenz der ausgespielten Bilder. In der Regel handelt es sich um jeweils 60 Bilder pro Sekunde. Sollte die Tracking-Rate zu niedrig sein, kann es vorkommen, dass das virtuelle Objekt noch an der Stelle stehen bleibt, an der es sich befand, bevor der Nutzer die Kamera geschwenkt hat. Selbst, wenn bei Video-See-Through-Geräten die Bildrate der geringen Tracking-Rate anpasst wird, fühlen sich die Nutzer oftmals nicht wohl, da eine Diskrepanz zwischen der eigenen Körperbewegung und der visuellen Wahrnehmung entsteht.

Daneben kann eine hohe Latenz zu Problemen führen. Unter Latenz versteht man „die Zeitspanne, die ein System für die Reaktion auf eine Eingabe benötigt [...] (engl. Latency)“ (Buhr et al. 2013: 196) Im Falle der AR kann eine „Eingabe“ auch die Bewegung der Kamera oder des verfolgten Objekts sein. Die „Reaktion“ ist die Objekttransformation, die nach dem Tracking berechnet werden muss. Anders ausgedrückt ist die Latenz in diesem Fall die Zeitspanne zwischen dem Schätzen der Position und/oder Orientierung und der Objekttransformation. Der Effekt ist ähnlich wie bei einer geringen Tracking-Rate: Die Anzeige erfolgt mit Verzögerungen. Nur wenn die Latenz möglichst gering ist und unter der Wahrnehmungsschwelle des Menschen liegt, empfindet der Nutzer das System als echtzeitfähig und beginnt, völlig in die Anwendung einzutauchen. Gründe für eine hohe Latenz können zum einen die Tracking-Verfahren und die dazugehörige Rate sein, zum anderen lange Signallaufzeiten. (vgl. Broll 2013: 266ff)

2.2.3.2 Photometrische Registrierung

Unter photometrischer Registrierung bezeichnet die Anpassung der virtuellen Objekte an die aktuelle Beleuchtungssituation aufgrund von Schätzungen. Diese Art der Registrierung findet heute kaum noch Anwendung. (vgl. Broll 2013: 268)

Für die photometrische Registrierung muss es möglich sein, dass virtuelle Objekte und die Beleuchtungen von Objekten die Beleuchtung der Realität beeinflussen können. Zum Beispiel sollte es möglich sein, dass ein virtuelles Objekt einen Schatten auf ein reales Objekt wirft. Auf reale Ebenen wie eine Tischplatte lässt sich das noch gut umsetzen, auf andere Objekte mit komplizierten Geometrien ist das deutlich schwieriger. (vgl. Broll 2013: 269)

Für die Schätzung der Daten wird eine meist schwarze Kugel (sog. Light Probe) in die Szene eingebettet. Je nach Reflektion auf der Kugeloberfläche werden virtuelle Lichtquellen geschätzt und in die virtuelle Szene eingebaut. Sie sind die Basis für die Berechnung von beispielsweise Schatten. Ein anderer Weg, die Beleuchtungsdaten zu erhalten, ist die Auswertung von Tiefenkamerabildern wie die der Microsoft Kinect (1. Generation). Damit ist es ferner möglich, die Änderungen in der Realität zu erfassen und einzubeziehen. (vgl. Broll 2013: 268ff)



Abbildung 2.8: Links: keine photometrische Registrierung. Rechts: mit photometrische Registrierung. Das rote Blatt reflektiert Licht auf den computergenerierten Drachen. Außerdem wirft der Drache einen Schatten (Broll 2013: 270)

2.2.4 Ausgabe

Nach dem Rendern, also der Wiedergabe von virtuellen Inhalten auf Basis der Transformationen aus dem Rendering und der Kameraperspektive, wird der erweiterte, gerenderte Videostream auf einem Ausgabegerät ausgespielt. Die gängigsten Ausgabegeräte im Augmented-Reality-Bereich werden hier vorgestellt.

2.2.4.1 Datenbrillen

2.2.4.1.1 Video-See-Through-Displays

Beim Video-See-Through-Display (VST-Display) werden mit 1-2 eingebauten Kameras Aufnahmen der Umgebung gemacht. Dieses Video wird an den Grafikprozessor weitergeleitet. Beim Rendern werden diese Aufnahmen als Hintergrundbilder verwendet. Dabei werden sie perspektivisch korrekt eingebaut, sodass der Nutzer den Eindruck hat, als würde er die Realität direkt durch die Datenbrille (engl. Head-Mounted Display, HMD) wahrnehmen. Tatsächlich sieht er aber nur ein Video.

Es ist besonders wichtig, dass Blickwinkel, Blickpunkt und Blickrichtung von Videokamera und Ausgabegerät korrekt berechnet werden. Wenn die Kamera nicht den gleichen Strahlengang wie das Auge hat (was häufig vorkommt, weil die Kamera leicht versetzt an der Datenbrille montiert ist), muss diese Verschiebung herausgerechnet werden, da der Nutzer sonst Schwierigkeiten mit Größeneinschätzungen und Entfernungen haben könnte. Die Nachberechnung kann vermieden werden, indem ein Prisma eingesetzt wird, um die Strahlen direkt in die Kamera zu lenken. (vgl. Broll 2013: 248-252, 272f; Schmalstieg / Höllerer 2016: 40ff)

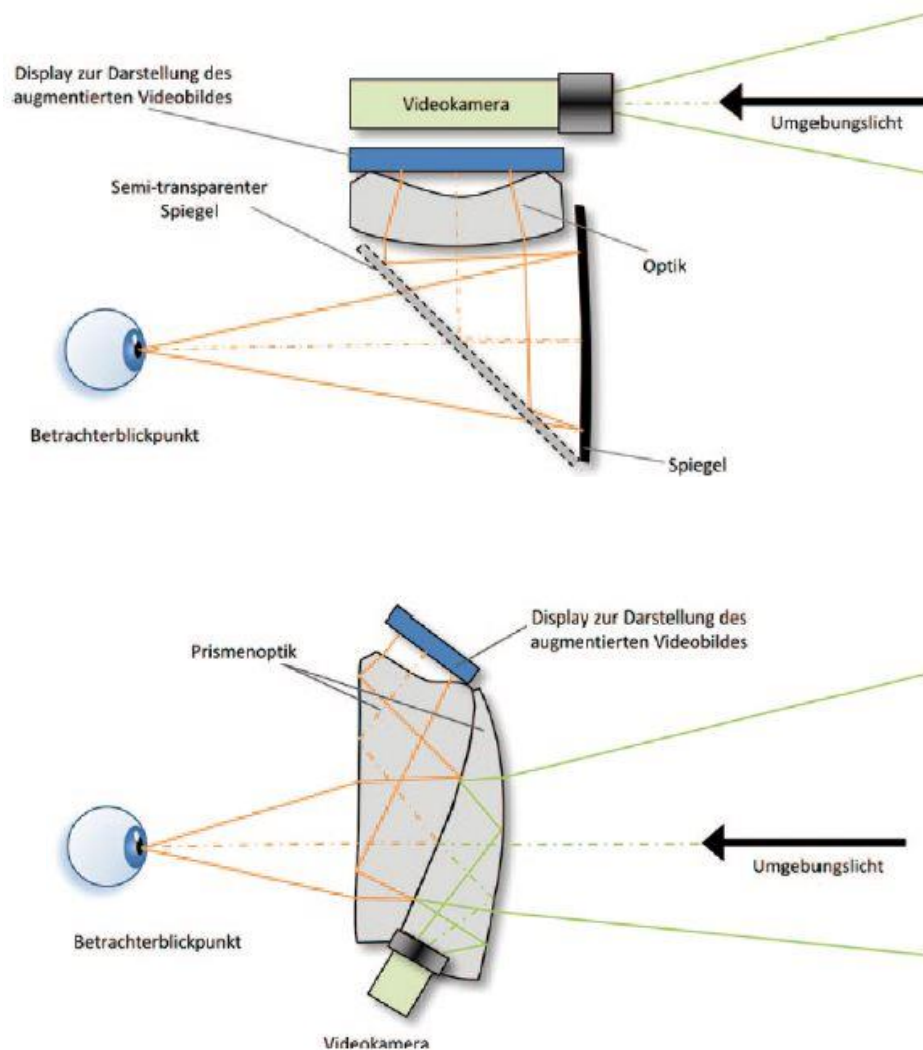


Abbildung 2.9: Schematischer Aufbau der VST-Brillen. Oben: mit semitransparentem Spiegel. Unten: mit Prisma

2.2.4.1.2 Optical-See-Through-Displays

Im Gegensatz zu VST-Displays zeigen Optical-See-Through-Displays (OST-Displays) die Realität direkt an, sodass keine Schwierigkeiten mit Auflösungen und Qualität der realen Objekte auftreten. Zur Ausgabe ist ein semitransparentes Display nötig, das sowohl Licht durchlässt und somit die reale Umwelt wahrnehmen lässt, als auch die Darstellungen der virtuellen Objekte reflektiert. Um dies zu erreichen, muss die einfallende Lichtmenge um bis zu 75 % verringert werden, sodass die Umwelt dunkler als ohne Brille erscheint. Um die Perspektiven zu überlagern, ist es nötig, die Relation zwischen Display und Blickpunkt mittels eines Sensors zu bestimmen. (vgl. Broll 2013: 248-252; Broll 2013: 274 Schmalstieg / Höllerer 2016: 40ff)

Broll (2013: 274-278) beschreibt unterschiedliche Bauweisen für Optical-See-Through-Displays: OST mit semitransparenten Spiegeln, prismenbasierte OST, retinale Datenbrille und See-Through-Displays mit integrierten optischen Elementen. Auf diese soll in dieser Arbeit nicht weiter eingegangen werden.

2.2.4.1.3 Mono-, bi- und binokulare Bauweisen

Bei den Bauarten der Datenbrillen wird unterschieden in monokulare, bi-okulare und binokulare. Wenn ein Auge überdeckt wird und mit dem anderen Auge die Umwelt uneingeschränkt wahrgenommen werden kann, spricht man von *monokular*. Wenn beide Augen bedeckt sind, aber das gleiche Bild (also nur ein Videostream) projiziert bekommen, wird eine *bi-okulare* Datenbrille genutzt. *Binokular* sind Datenbrillen dann, wenn beide Augen bedeckt sind, aber beide ein eigenes Bild eingespielt bekommen. So entsteht ein stereoskopischer Effekt. Diese Variante ist die kostenintensivste. (vgl. Schmalstieg / Höllerer 2016: 42f)

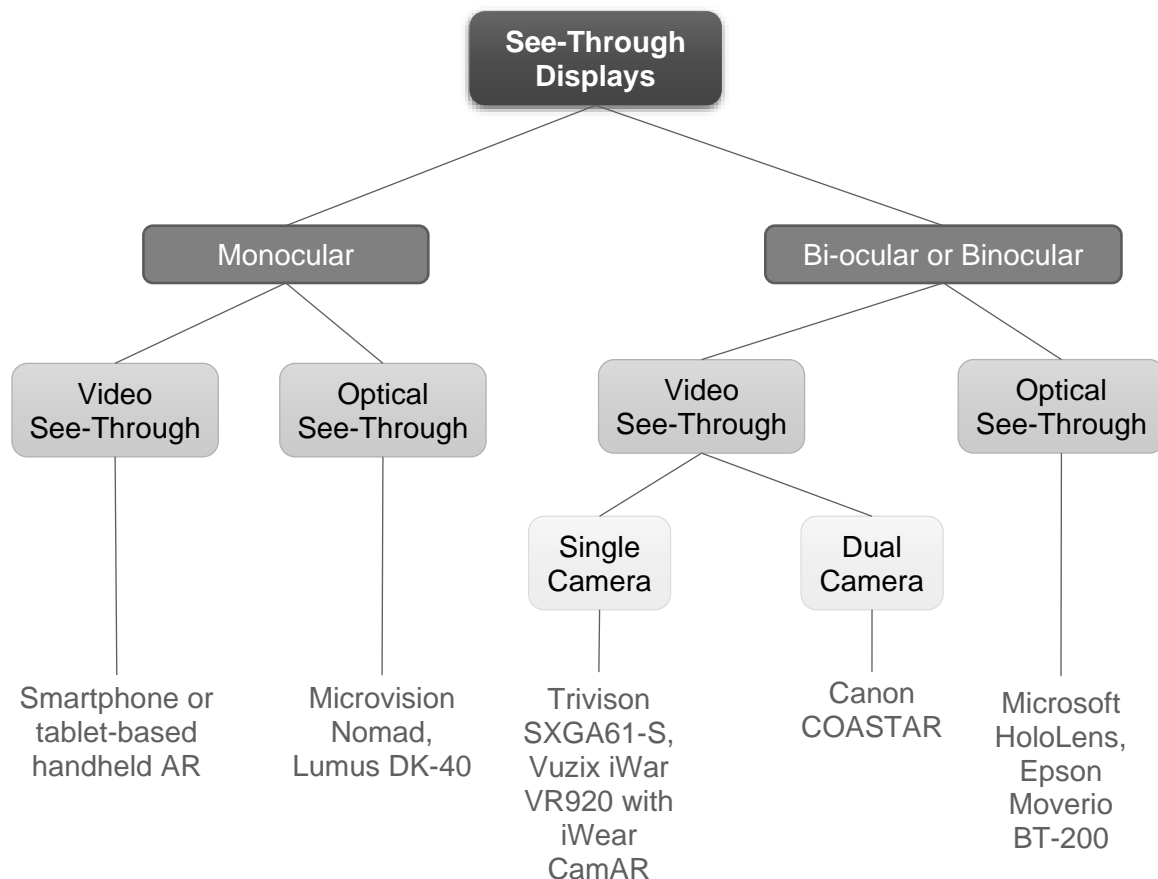


Abbildung 2.10: Kategorisierung von See-Through Displays (in Anlehnung an Schmalstieg / Höllerer 2016: 44)

2.2.4.2 Projektionsbasierte Ausgabe

Neben der Ausgabe über Datenbrillen ist eine projektionsbasierte Ausgabe möglich. Hierbei werden computergenierte Bilder direkt auf andere Gegenstände in der realen Welt projiziert; es werden keine neuen räumlichen Strukturen geschaffen, sondern die Wahrnehmung der vorhandenen Gegenständen beeinflusst. Die virtuellen Inhalte werden nur auf die bestehenden Oberflächen projiziert, sodass die Positionierung der Objekte nicht völlig frei stattfindet, sondern vom Standort der Projektionsoberfläche abhängt. Die Position der Projektoren und die Position der Oberflächen (inkl. Modell des Oberflächenobjektes) müssen bekannt sein. Vor der Ausgabe wird das virtuelle Bild auf das Modell der Oberflächengegenstände gerendert und mit einem kalibrierten Projektor ausgegeben. (vgl. Broll 2013: 248-252; Broll 2013: 281; Schmalstieg / Höllerer 2016: 40ff)

Da die Projektionsflächen meist nicht weiß, matt und eben, sondern strukturiert, glänzend und farbig sind und die Umgebung das Projektionsergebnis mit Helligkeiten, Schatten usw. beeinflusst, ist eine photometrische Kalibrierung für eine gute Ausgabe notwendig. (vgl. Broll 2013: 282)

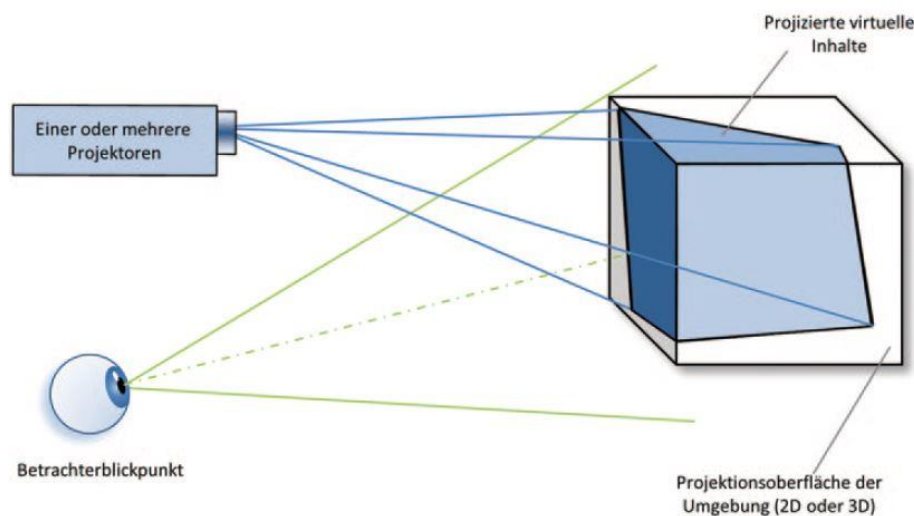


Abbildung 2.11: Projektionsbasierte Ausgabe (Broll 2013: 282)

2.2.4.3 Handheld-Geräte

Handheld-Geräte, auch mobile Endgeräte genannt, umfassen primär die Gruppe der Smartphones und Tablet-Computer. Diese Gerätetypen haben auf der Rückseite eine Kamera, die einen Videostream aufnimmt. Es wird der VST-Ansatz verwendet, der die perspektivische Anpassung für das Display des Endgerätes berechnet – nicht für den Blickwinkel des Nutzers. Das Blickfeld des Nutzers steht im Verhältnis zum Display, die Ausgabe auf dem Display wird aber auf Basis des Kamerablickwinkels berechnet. (vgl. Broll 2013: 271)

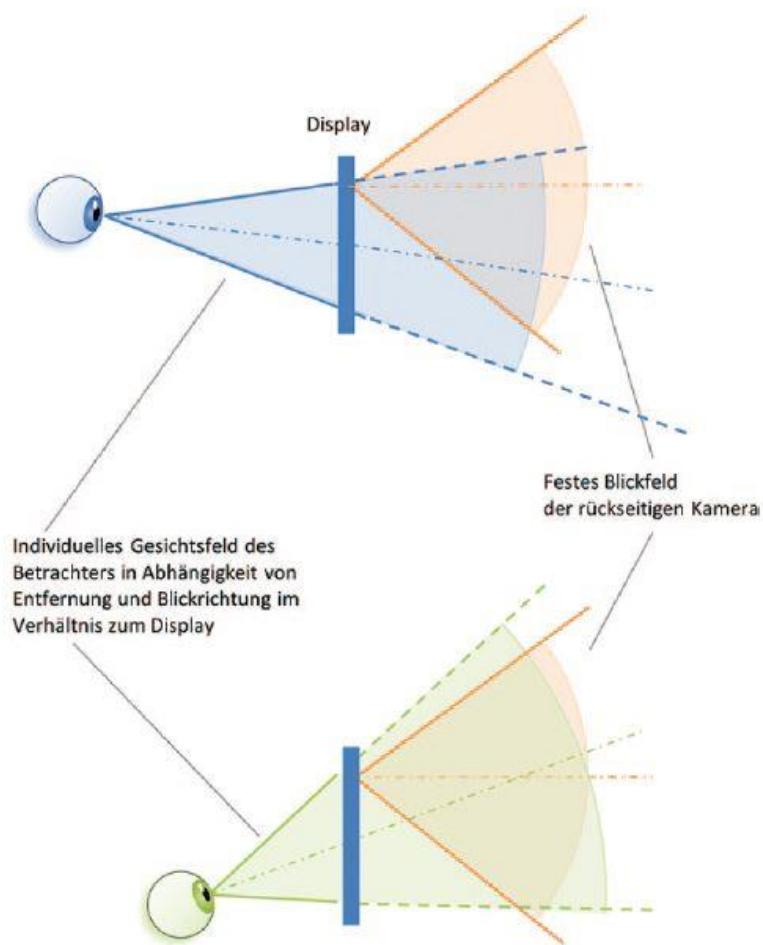


Abbildung 2.12: : Zwei unterschiedliche Blickfelder des Betrachters auf das Display des mobilen Endgerätes (Broll 2013: 272)

2.2.4.4 Kontaktlinsen

Um eine Ausgabe ohne Brille oder „unhandliches“ Endgerät zu ermöglichen, werden Kontaktlinsen entwickelt. Die bisherige Technik fokussiert sich auf LED-Displays in den Linsen, doch Entwicklungen mit LCD-Technik sind schon in Arbeit. Die bisherigen Ansätze der Forschung hatten das Ergebnis, dass entweder ein Bild projiziert wird, das einen halben Meter entfernt erscheint, oder dass ein Bild direkt auf die Retina projiziert wird. (vgl. Virtual Reality Magazin 2012; Mehler-Bicher / Steiger 2014: 48)

Die Technik ist noch nicht alltagstauglich, da z. B. immer noch Kontaktlinsen und Datenbrillen gleichzeitig verwendet werden müssen, die Stromversorgung noch problematisch ist und Objekte nur fokussiert werden können, wenn sie sich weiter als 10 cm vom Auge entfernt befinden. (vgl. Mehler-Bicher / Steiger 2014: 48)

2.2.4.5 Nichtvisuelle Ausgabemöglichkeiten

Die bisherige Technik legt den Fokus auf die visuelle Ausgabe, da die visuelle Wahrnehmung die dominanteste ist. Es gibt allerdings auch weitere Ausgabemöglichkeiten für die anderen menschlichen Sinne.

- **Auditive Ausgabe**

Audioinformationen können je nach Position in einem Gebäude wie einem Museum (z. B. durch Infrarot-Sensoren) oder im Freien durch GPS angesprochen werden. Die Ausgabe kann z. B. über Kopfhörer oder in Datenbrillen eingebaute Lautsprecher erfolgen. (vgl. Schmalstieg / Höllerer 2016: 34f)

- **Haptische Ausgabe**

Was in der Welt der Virtuellen Realität schon existiert, im Bereich AR allerdings noch etwas eingeschränkt ist, sind haptischen Ausgabearten. Dies kann z. B. durch mechanische Eingabegeräte oder elektrische Signale realisiert werden (vgl. Abschnitt 2.2.2.2). Häufig sind haptische Systeme auf einen kleinen Raum beschränkt. (vgl. Schmalstieg / Höllerer 2016: 35f)

- **Gustatorische und olfaktorische Ausgabe**

Mit dem Sensorama versuchte Morton Heilig Gerüche zu vermitteln (siehe Abschnitt 2.1.2.1). Heutzutage ist es möglich, dem Nutzer vor einem zweidimensionalen Bildschirm durch kleine Ventilatoren Gerüche zuzuspielen. Es gibt auch erste Versuche mit tragbaren Displays. Vor allem für die Gastronomie könnte die Verwendung von olfaktorischen und gustatorischen Anwendungen attraktiv sein. (vgl. Schmalstieg / Höllerer 2016: 37f)

2.3 AR, Multimedia und E-Learning

2.3.1 Einordnung in diese Arbeit

In dieser Arbeit soll prototypisch eine Anwendung programmiert werden, die Wissen über Kunstwerke, ihre Bedeutung, Entstehung und Deutung vermittelt. Dahinter steckt letztendlich die Absicht, dass der Nutzer nach der Anwendung der App etwas gelernt hat. Was unter „Lernen“ zu verstehen ist und wie Gelerntes im Gehirn verarbeitet wird, wird in Kapitel 3 definiert.

Zunächst wird kurz erläutert, was unter „E-Learning“ zu verstehen ist. Mit mobilen Endgeräten bietet sich die Darstellung der Lerninhalte mit mehreren Medien, also multimedial, an. In wie weit dies sinnvoll ist und welche wissenschaftliche Erkenntnisse diesbezüglich vorliegen, soll in den folgenden Kapiteln erörtert werden. Besondere Erwähnung findet hier das Thema Interaktivität, da sie sowohl in den Definitionen von AR als auch von Multimedia eine Rolle spielt.

2.3.2 Definition E-Learning

Ob eine AR-Anwendung unter E-Learning gezählt werden kann, lässt sich nur schwer eindeutig beantworten, da es für E-Learning keine eindeutig feststehende Definition gibt. In dieser Arbeit wird die Definition von Kerres herangezogen, der E-Learnings als Oberbegriff versteht für „alle Varianten der Nutzung digitaler Medien zu Lehr- und Lernzwecken [...], etwa um Wissen zu vermitteln, für den zwischenmenschlichen Austausch oder das gemeinsame Arbeiten an digitalen Artefakten“ (Kerres 2013: 6)

Ogleich „Lernen“ häufig eine Assoziation mit Schule, Universität, Vorträgen oder Ähnliches weckt, wird auch im Museum versucht, den Nutzern der App neue Informationen anzubieten, in der Hoffnung, dass sie dadurch ihr Wissen erweitern. Das Lernen erfolgt hier selbstgesteuert: Nur der Nutzer entscheidet, ob und welche Informationen er sehen möchte und ob er sich diese merkt. Was der Nutzer lernt, wird nicht überprüft.

E-Learning-Anwendungen sind neue Lernmittel. Ob ihre positiver Aspekt mit der Zeit verloren geht, kann noch nicht abgeschätzt werden. (vgl. Jäncke 2005: 92-110)

2.3.3 Definition Multimedia

„Multimedia“ stammt ursprünglich aus dem Lateinischen und bedeutet „viele Medien“, z. B. Texte, Bilder, Videos, Audiodateien etc. Es ist eine genauere Definition nötig, um mehr als eine reine Aufzählung zu erhalten und den Begriff genau zu spezifizieren. (vgl. Weidenmann 2002: 45)

Medien sind nach Weidemann „Objekte, technische Geräte oder Konfigurationen, mit denen sich Botschaften speichern und kommunizieren lassen“. (Weidenmann 2002: 46) Als Botschaften sind „absichtsvoll codierte und strukturierte Inhalte“ zu verstehen. (Weidenmann 2002: 46f) Das Medium ist der Träger für kommunikative Inhalte, z. B. ein Buch oder ein Computer.

Weidemann führte darauf basierend die Begriffe Multimodalität und Multicodalität ergänzend zu Multimedia ein: (vgl. Weidenmann 2002: 45ff)

Begriff	Beschreibung	Beispiel
Multimedialität	Unterschiedliche Speicher- und Präsentationstechnologien werden „integriert“ präsentiert. Das heißt, sie arbeiten miteinander und liegen nicht separat wie bei einem Medienverbund vor.	PC + Smartphone
Multicodalität	Unterschiedliche Symbolsysteme und Codierungen werden verwendet.	Text + Bilder, Grafik + Beschriftung
Multimodalität	Unterschiedliche Sinnesmodalitäten werden angesprochen.	Sehen + Hören (audiovisuell)

Tabelle 2.1: Definitionen Multimedialität, Multicodalität, Multimodalität

2.3.4 Einflussfaktoren für erfolgreichen Umgang mit Multimedia

Damit Multimedia wirklich einen positiven Einfluss auf den Lernerfolg der Nutzer hat, ist es wichtig, bei der Anwendungsgestaltung, darauf zu achten, dass sie

- zielgruppengerecht
- motivierend
- mit angemessenem Stresslevel versehen
- an soziale Umstände angepasst

ist. Im Folgenden werden die einzelnen Punkte genauer erklärt.

2.3.4.1 Zielgruppe

Damit eine multimediale E-Learning-Anwendung ihren Zweck erfüllt, müssen vor der Erstellung die Zielgruppe, die Lehrinhalte und die Lehrziele definiert werden. (vgl. Holzinger 2001: 234)

In der folgenden Tabelle sind einige Zielgruppenmerkmale aufgelistet.

	Merkmal	Kurzbeschreibung
Allgemeine Zielgruppenmerkmale	Größe der Zielgruppe	Anzahl der Benutzer/Teilnehmer
	Geografische Verteilung der Zielgruppe	Regional begrenzt? Landesweit? International?
	Alter und Geschlecht	Heterogene oder homogene Gruppe? Altersgruppe?
	Höchster Schulabschluss	Durchschnittsangabe und Spanne des höchsten und niedrigsten Abschlusses
	Weitere Merkmale	Alleinerziehende? Hürden? Milieu? Ethische/Migrationshintergründe?
Weitere Merkmale	Vorwissen	Ist der Nutzer Anfänger, Fortgeschrittener oder Experte? Hat er viel oder wenig Vorwissen? Was könnte der Nutzer schon wissen?
	Motivation	Warum verwendet der Nutzer die Anwendung? Will oder soll er lernen? (siehe Abschnitt 2.3.4.2)
	Einstellungen und Erfahrungen	Sind die Nutzer gewohnt, E-Learnings zu benutzen? Haben sie Erfahrungen mit den Technologien und fremd- bzw. selbstgesteuertem Lernen? Wie stehen sie zum Thema, zu E-Learnings, zu den Lernformen?
	Lernorte	Wird das E-Learning zu Hause oder am Arbeitsplatz genutzt? Können alle Medienkomponenten genutzt werden (z. B. Audio und Videos)?

Tabelle 2.2: Zielgruppenmerkmale (vgl. Kerres 2013: 281-286)

2.3.4.2 Motivation

2.3.4.2.1 Definition „Motivation“

Der Begriff Motivation wird in unterschiedlichen Forschungsbereichen unterschiedlich definiert. In der Kognitionswissenschaft beinhaltet der Begriff Motivation „die Gesamtheit der Bedingungen und Prozesse, die einer *Handlungsbereitschaft* zugrunde liegen [sic!]) und [...] neben der Verfügbarkeit relevanter Fähigkeiten und Fertigkeiten eine *notwendige Voraussetzung* für zielgerichtetes Handeln und Lernen.“ (Holzinger 2001: 248; Hervorhebungen im Original)

In anderen Worten bedeutet das, dass kognitive Prozesse und Bedingungen dazu führen, dass der Lernende bereit ist, zu handeln. Außerdem ist die Motivation (zusammen mit Fähigkeiten und Fertigkeiten) eine Grundvoraussetzung, damit überhaupt zielgerichtet gelernt werden kann. Diese Definition ist im Rahmen dieser Arbeit am passendsten und wird daher nachfolgend verwendet.

2.3.4.2.2 Intrinsische und extrinsische Motivation

Motivation kann in intrinsische und extrinsische Motivation unterschieden werden. Bei der *intrinsischen Motivation* kommt der Antrieb, warum man ein Ziel erreichen will, von innen. Dies kann z. B. Neugierde, Interesse, Wissensdrang, Wettkampfgeist etc. sein. Im Gegensatz dazu regen bei *extrinsischer Motivation* externe Faktoren zur Handlung an. Diese Faktoren können unter anderem Belohnungen (auch Lob) oder aber Gruppendruck sein. Intrinsische Motivation („ich will“-Haltung) ist von längerer Dauer und wirkungsvoller als extrinsische Motivation („ich muss“-Haltung). (vgl. Holzinger 2001: 253f)

Bei intrinsischer Motivation ist es sinnvoll, verstärkt darauf zu achten, dass die Anwendungen möglichst immersiv sind, also den Nutzer völlig „eintauchen“ lassen. Dem Nutzer sollte zudem ermöglicht werden, selbstständig und selbstkontrolliert Inhalte aufzurufen oder Tests durchzuführen, sowie die Neugierde aufrechtzuerhalten und mit Präsentationen zu fördern. (vgl. Kerres 2013: 284)

Bei extrinsischer Motivation ist es hilfreicher, konsistent, strukturiert und in kleinen Schritten vorzugehen. Die Aufmerksamkeit der Nutzer muss gebündelt und regelmäßig Rückmeldung gegeben werden. (vgl. Kerres 2013: 285)

2.3.4.3 Stress

Stress wird in unserer Gesellschaft häufig sehr negativ wahrgenommen, obwohl er für Menschen wichtig ist. (vgl. Willmann 2016)

Stress ist zunächst einmal ein psychophysiologischer Prozess, der zu Stressreaktionen (Stresssyndromen) führt. Diese Stressreaktionen führen dazu, dass sich das Gehirn ebenso wie Pflanzen und andere Organismen mit Dingen auseinandersetzt, mit denen er vorher keine Berührung hatte, und sich damit weiterentwickelt. Erinnerungen und neues Wissen, die mit dem Stressauslöser zu tun haben, lassen sich besser behalten, wenn das Gehirn beim Lernen unter Stress stand (vgl. Holzinger 2001: 266; Willmann 2016) Nimmt der Stress allerdings überhand, kann die Informationsaufnahme gehemmt werden. (vgl. Holzinger 2001: 44)

Auch wenn der Körper Stress „braucht“, um leistungsfähiger zu sein, ist es wichtig, den Lernenden nicht das Gefühl zu geben, dass sie immer 100% Leistung bringen müssen.

Der aufgebaute innerliche Druck würde zu einem Leistungsabfall führen Solange der Lernende das Gefühl hat, die Situation unter Kontrolle zu haben, empfindet er Stress als nichts Schlimmes. (vgl. Holzinger 2001: 266f; Willmann 2016)

Daneben kann der Umgang mit Neuem und Ungewohntem, z. B. Computern oder AR, Stress auslösen. Hierbei ist besonders darauf zu achten, dass der Nutzer das Gefühl hat, Kontrolle über die Situation zu haben.

2.3.4.4 Soziale Effekte

Der Mensch hat ein grundsätzliches Bedürfnis, sich einer sozialen Gruppe zuzuordnen und darin aufzugehen, sich zu entfalten. Er möchte mit anderen interagieren, sich austauschen, eine Rolle einnehmen, eine Wirkung auf andere haben. Diese Eigenschaften haben einen großen Einfluss auf die Motivation des Lernenden. (vgl. Holzinger 2001: 269f) Bei AR könnte dies durch interaktive Avatare verwirklicht werden, mit denen der Nutzer kommuniziert und dessen Verhalten er beeinflusst. Außerdem kann innerhalb der Anwendungen über das Internet Zugänge zu Foren und Chatgruppen ermöglicht werden, in denen die Nutzer sich dann austauschen können.

Nicht nur in der Anwendung sind soziale Kontakte und Gruppenbildung wichtig. Die Anwendung selbst und die Einstellung ihr gegenüber führt zu einer Positionierung in der Gesellschaft. Die Haltung, Erwartung und Wertung, die ein Mensch gegenüber einer unbekanntem Sache hat, entscheidet häufig darüber, ob er sich mit dem Thema beschäftigt und bereit ist, seine Meinung zu ändern. Bisher zeigen Untersuchungen, dass die Akzeptanz neuer Technologien steigt, wenn der Nutzer damit Erfolgserlebnisse hat und positive Erfahrungen macht. (vgl. Holzinger 2001: 267f) Vor allem in einem neuen Bereich wie Augmented Reality ist es wichtig, dass der Nutzer eine positive Haltung gegenüber der Technologie hat bzw. entwickelt, damit er bei einer weiteren Verbreitung dem Thema nicht kritisch gegenübersteht, sondern die positiven Effekte einschätzen kann. Dementsprechend positioniert er sich in der Gesellschaft.

2.3.5 Auswirkungen von Multicodierung und Multimodalität

Grundsätzlich ist festzuhalten, dass Multimedia nicht automatisch das Lernen verbessert. Durch den Einsatz von Multimedia werden die Motivation und die Aufmerksamkeit erhöht. Dadurch beschäftigt sich der Lernende intensiver mit dem Lernstoff. (vgl. Holzinger 2001: 107)

Entgegen der weitläufigen Meinung ist nie wissenschaftlich bewiesen worden, dass die Behaltensleistung verbessert wird, wenn mehrere Sinneskanäle angesprochen werden. (vgl. Weidenmann 2002: 48)

Richter (o. J.: 3f) führt folgende Punkte auf, die für bzw. gegen die Verwendung von Multimedia sprechen. Sie basieren auf empirischen Befunden.

- Begleitende Illustrationen verbessern das Behalten von Text, da verbale und visuelle Elemente im Gedächtnis verknüpft werden und eine mentale Karte bilden.
- Listen mit Abbildungen bekannter Objekte werden besser behalten als Listen mit Wörtern (Bildüberlegenheitseffekt).
- Die Kategorisierung in auditive und visuelle Lerner ist nicht eindeutig empirisch belegbar. Es ist weder bekannt, ob es sich nur um Präferenzen hinsichtlich der Modalitäten oder des Informationsangebotes handelt.
- Die Sinne sind anfällig für Überlastung. Die Überlastung sollte reduziert werden, indem die Informationen auf unterschiedliche Sinne verteilt und unterschiedliche Modalitäten benutzt werden.
- Bilder werden länger betrachtet, wenn akustisch Zusatzinformationen angeboten werden.

Vor allem letzteres ist für das Konzept dieser Arbeit von großer Bedeutung. Weidenmann empfiehlt, vor allem auch auditive Modalitäten einzuarbeiten, da gesprochene Sprache einprägsam ist, Aufmerksamkeit weck und persönlicher ist. Vor allem beim Erklären von Bildern kann es entlastend sein, wenn die Erklärung auditiv erfolgt. (vgl. Weidenmann 2002: 53)

Um zu überprüfen, ob multimediale Anwendungen und somit Anwendungen mit AR den Nutzer besser motivieren, müssen die zwei folgenden Thesen näher betrachtet werden.

1) Unterschätzungsthese (nach Salomon)

Wenn der Lernende das Gefühl hat, sich mental stark angestrengt zu haben, ist der Lernerfolg größer. Wenn der Lerner das Medium als „leicht“ empfindet, bringt er weniger mentale Anstrengung auf, was zu einem geringeren Lernerfolg führt. Multimediale Anwendungen werden in der Regel als „leichter“ empfunden als beispielsweise Printmedien.

2) Hemmungsthese (nach Sturm)

Rasche Bildsequenzen und das Angebot von Sprache *und* Bildern zur gleichen Zeit, Special Effects o. Ä. erschweren die Verarbeitung der Informationen. Zusätzlich existieren Untersuchungen, die die Annahme stützen, dass weniger eigene bildliche Vorstellungen entwickelt werden, wenn Informationen über Bildmaterial codiert werden.

(vgl. Weidenmann 2002: 56f)

Es lässt sich also sagen, dass die Motivation durch Multicodalität und Multimodalität verbessern, aber auch verschlechtert werden kann, je nachdem wie die Anwendungen aufgebaut sind und der Lernende damit umgeht. Sind zu viele Wechsel in der Darstellungsform kann dies den Lernerfolg hemmen, da sich der Lernende überfordert fühlt; ist die Anwendung und ihre Darstellung zu wenig fordernd, kann dies ebenfalls den Lernerfolg verringern. (vgl. Holzinger 2001: 237)

Des Weiteren kann festgehalten werden, dass multicodal und multimodal präsentierte Informationen den Lernenden stimulieren und so das Wissen besser abrufbar ist. Durch die multimodale bzw. -codale Präsentation können komplexe Zusammenhänge abstrahiert und aus verschiedenen Perspektiven betrachtet werden, was zu mentalen Modellen und Wissensaufbau führt. Wenn die Informationspräsentation den Nutzer interaktiv miteinbezieht, erweitert er seine Lernstrategien und Lernerfahrungen. (vgl. Weidenmann 2002: 61)

2.3.6 Augmented Reality und Multimedia

Gemäß den Definitionen von Weidemann kann AR alle Bereiche von Multimedia abdecken. In AR-Anwendungen können mehrere Sinnesmodalitäten angesprochen werden; primär werden visuelle Ausgaben verwendet, es sind aber auch Audiodateien üblich und haptische Ausgaben möglich. Also sind die Anwendungen prinzipiell als multimodale Anwendung konzipierbar.

Die Ausgabe multicodal vorliegender Informationen ist in AR-Systemen möglich, da die Einbindung von Text- und Bildinformationen möglich ist.

Weiterhin können AR-Anwendungen auf mobilen Endgeräten wie z. B. Smartphones oder Tablet-PCs angewendet werden. Bei den meisten AR-Anwendungen wird allerdings nur ein Endgerät angewendet. Die Applikation kann zwar für mehrere Medien vorliegen (z. B. Smartphones, Datenbrillen, Tablets), diese Angebote werden aber nicht integriert präsentiert und würden in diesem Fall nicht multimedial sein. Beispielsweise könnten allerdings Flyer und Bücher verwendet werden, die die Brücke zwischen dem elektronischen Medium und dem Printmedium schlagen, sodass die Anwendung multimedial wäre.

Es lässt sich also sagen, dass AR-Anwendungen und Systeme das Potenzial für Multicodalität, Multimodalität und Multimedialität besitzen.

2.3.7 Interaktivität

2.3.7.1 Definition Interaktivität

Eine Eigenschaft und Stärke, die für Multimedien genauso wichtig ist wie für Augmented Reality ist die Interaktivität. Diese findet sowohl in der AR- Definition von Azuma (1997) Beachtung (vgl. Abschnitt 2.1.1), als auch bei Kerres (2002: 23), der Interaktivität als „wesentliches Charakteristikum von Multimedien“ bezeichnet, da es großes Potenzial in didaktischer Hinsicht bietet.

Interaktion kommt aus dem Lateinischen „inter“ für „zwischen“ und „agere“, was „tun, treiben, handeln“ bedeutet. Darunter versteht man das In-Kontakt-Treten und Sich-Austauschen mit anderen. Dies beschränkt sich nicht nur auf das soziale Leben, sondern kann auch auf multimediale Anwendungen zwischen Mensch und Maschine bezogen werden. (vgl. Haack 2002: 128) Im Glossar von Issing / Klimsa wird Interaktivität definiert als „umfassender Begriff für solche Eigenschaften eines Computersystems, die dem Benutzer Eingriffs- und Steuermöglichkeiten eröffnen, im Idealfall auch die wechselnde Dialoginitiative von Mensch und Computer sowie über ein Computernetz mit anderen Menschen.“ (Issing / Klimsa 2002) Zusammengefasst heißt das, dass der Mensch die Darstellung der multimedialen Anwendung beeinflussen können muss, damit sie als interaktiv gilt.

2.3.7.2 Stufen der Interaktivität

Bei Interaktivität gibt es verschiedene Stufen, die von verschiedenen Institutionen und Wissenschaftlern unterschiedlich definiert wurden. Schulmeister (2005: 3-16) teilt den Begriff der Interaktivität in folgende Stufen ein:

Stufe	Beschreibung
1 Objekte betrachten und rezipieren	Betrachter kann Komponenten (z. B. Filme, Animationen, Bilder) anwählen und abspielen, hat aber keinerlei Einfluss auf die Darstellung. Komponenten sollen nur die Informationen illustrieren.
2 Multiple Darstellungen betrachten und rezipieren	Komponenten können innerhalb eines bestimmten Rahmens ausgetauscht werden (z. B. bei einer Diashow, beim Abspielen mehrerer Audiodateien etc.).
3 Repräsentationsform variieren	Direkter Einfluss auf das Verhalten der Komponente, z. B. Grafiken skalieren, 3D-Modelle drehen etc.

	Nur Repräsentationsform wird geändert, Inhalt wird vom Nutzer nicht beeinflusst.
4 Inhalt der Komponente beeinflussen: Variation durch Parameter- oder Datenvariation	Inhalte der Komponenten werden durch Nutzer generiert, z. B. Parameter ändern oder Daten eingeben → Programm errechnet die Darstellung. Vor allem bei Animationen, Ton und Diagrammen geläufig
5 Objekt bzw. Inhalt der Repräsentation konstruieren und Prozesse generieren	Nutzer entwickelt selbst Grafiken, die seine Gedanken visualisieren, oder kreiert eigene Objekte. Programm dient als Werkzeug.
6 Konstruktive und manipulierende Handlungen mit situationsabhängigen Rückmeldungen	System gibt je nach Verhalten und Handlung des Nutzers und nach Situation eine sinnvolle Rückmeldung, z. B. korrigiertes Lernobjekt. Programm kann eine beliebige Nutzereingabe auswerten.

Tabelle 2.3: Stufen der Interaktion nach Schulmeister

2.3.7.3 Interaktivität bei AR

Interaktivität in Echtzeit spielt sowohl bei VR als auch bei AR eine wichtige Rolle. Interaktivität ist eng mit dem Begriff Immersion verknüpft. Unter Immersion (engl. „immersion“) versteht man den „Grad des Eintauchens bedingt durch objektive, quantifizierbare Stimuli, das heißt, multimodale Stimulationen der menschlichen Wahrnehmung.“ (Dörner / Steinicke 2013: 46) Je größer diese Immersion ist, desto größer ist die „Suspension of Disbelief“, also das willentliche Ausblenden von Unglauben. Das heißt, der Nutzer ist sich bewusst, dass er sich in einer virtuellen Umgebung befindet und ist deshalb bereit, die auftretenden Widersprüche zwischen virtueller und realer Welt auszublenden³. (vgl. Dunleavy / Dede o. J.: 4; Dörner et al. 2013: 8) Wenn die Immersion hoch ist, tritt Präsenz ein.

Als Präsenz (engl. „presence“) wird das subjektive Gefühl des Nutzers bezeichnet, wenn er in die virtuelle Welt eintaucht. Der Nutzer weiß, dass er sich in einer virtuellen Umgebung befindet, sie wird jedoch real für ihn; er verhält sich, als würde er sich in der realen Welt befinden. Da der Nutzer von nichts Außenstehendem abgelenkt wird, rich-

³ Ein Beispiel aus dem nicht-virtuellen Bereich: Ein Leser beschwerte sich bei einem Cartoonisten, dass sich ein Eisbär mit einem Pinguin auf einer Eisscholle unterhält, wo doch Pinguine am Südpol und Eisbären am Nordpol beheimatet sind. Dass sie sprechen konnten, störte ihn nicht. (vgl. Dörner et al. 2013: 8)

tet er seine volle Aufmerksamkeit auf die neue Umgebung. Dies unterstützt die intrinsische Motivation und führt zu einem positiven Gefühl. (vgl. Dörner / Steinicke 2013: 46; Alsdorf / Bannwart 2002: 470)

In gleicher Weise soll bei AR-Anwendungen ein hoher Grad an Immersion erreicht werden, da die Grenze zwischen realer und virtueller Welt verschwindet. Bei der Steuerung kommen mit AR-/VR-Anwendungen auch die natürlichen Bewegungen des Körpers im Raum ins Spiel, die bei klassischen multimedialen Anwendungen nicht berücksichtigt werden können. So kann der Nutzer beispielsweise durch die erweiterte Welt navigieren, in dem er sich durch den realen Raum bewegt. Dies erhöht die Interaktion und somit Immersion, da der Nutzer nun anfängt, selbst noch aktiver zu werden und die Umwelt bzw. die Rückmeldung des Systems aktiv durch sein Tun zu lenken. (vgl. Alsdorf / Bannwart 2002: 470; Broll 2013: 285f)

Beispiele für Interaktionen in einer Anwendung sind das Drücken eines Knopfes, das Bewegen der Hand und das Fixieren des Blicks. Letzteres könnte per Eye-Tracking umgesetzt werden, bei dem die Blickrichtung des Nutzers mit einer zusätzlichen aufwändig kalibrierten Kamera erfasst wird, um damit herauszufinden, welches Objekt er fokussiert und auswählen möchte. Dies erlaubt es, dass der Nutzer nicht mit einem separaten Eingabegerät hantieren muss. Da der Aufwand hierfür recht hoch ist, wird oft die Funktionalität eingebaut, dass ein Objekt „automatisch“ ausgewählt wird, wenn die Kamera einen Moment auf dem Objekt verweilt. (vgl. Broll 2013: 287)

Eine spezielle Interaktionstechnik bei AR ist das Tangible User Interface (TUI). Dabei wird die Benutzeroberfläche „anfassbar“, das heißt, der Nutzer hat Einfluss auf das virtuelle Objekt oder seine Eigenschaften, wenn er einen realen Gegenstand verändert, der mit dem virtuellen Objekt in Zusammenhang steht. Dies kann in direkter Form geschehen, wenn der Nutzer z. B. einen Marker bewegt, dreht etc. und sich dann die Darstellung ändert, da sich das virtuelle Objekt mitbewegt. Die direkte ist die am meisten angewandte. Bei der indirekten Form werden physische Eigenschaften eines realen Gegenstandes auf die Eigenschaften eines virtuellen Gegenstandes übertragen. Beispielsweise ändert sich je nach Position eines realen Würfels die Farbe oder Größe einer virtuellen Darstellung. (vgl. Broll 2013: 287f)

Beim Erstellen einer Anwendung muss festgelegt werden, welche Arten der Bewegung (z. B. Gesten oder die Verfolgung eines Fingers, der auf etwas zeigt) und von welchem Körperteil (z. B. Kopf, Arm, Hand) diese erkannt werden sollen. (vgl. Grimm et al. 2013: 98)

3 Lernen und Gedächtnis

3.1 Definition Lernen

Jäncke definiert Lernen als „erfahrungsbedingte und reversible Verhaltens- und Wissensänderung“. (Jäncke 2005: 84) Holzinger (2001: 106f) ergänzt diese Definition mit den Ursachen für Lernen (Reize, Signale oder Situationen) und dem Aspekt, dass es irrelevant ist, ob der Mensch beabsichtigt oder unbeabsichtigt lernt. Wie und mit welchen Mechanismen der Mensch lernt, ist unterschiedlich und von verschiedenen Faktoren wie z. B. Motivation und körperlichem Befinden abhängig. (vgl. Jäncke 2005: 84)

In der Lernforschung werden allgemein folgende Teilschritte des Lernprozesses angenommen (vgl. Holzinger 2001: 106):

- Informationsaufnahme
- Informationsverarbeitung
- Informationssicherung
- Informationsspeicherung (Gedächtnis; siehe Abschnitt 3.2)
- Informationsanwendung in konkreter Problemstellung
- Informationsverlust (vergessen)

Neurophysiologisch betrachtet werden neue Informationen zunächst elektrisch encodiert, dann werden sie in verschiedenen anatomischen Strukturen verarbeitet und kurzfristig gespeichert. Hierbei ist die Störanfälligkeit rechthoch. Damit es zu einer „dauerhaften“ Abspeicherung kommt, müssen zwei Bedingungen erfüllt sein (vgl. Jäncke 2005: 85-89):

- 1) Die elektrischen Erregungen bleiben über einen längeren Zeitraum (ab 20 Minuten) erhalten.
- 2) Die Neuronen oder die Neural Assemblies (Nervenzellengruppen), die verbunden werden sollen, müssen gleichzeitig aktiviert sein.

Dadurch werden die Kontakte zwischen den Neuronen verbessert, sodass sich die anatomischen Strukturen ändern. Dadurch werden Informationen schneller ausgetauscht und die Speicherkapazität größer. Wenn weit entfernte Neural Assemblies verknüpft werden, werden auch unterschiedliche Funktionen miteinander verknüpft, sodass z. B. visuelle und auditive oder emotionale und nichtemotionale Informationen miteinander verknüpft sind. Je mehr Verknüpfungen eine Information mit anderen Informationen hat, desto besser ist sie verankert und abrufbar. Stabile Verbindungen entstehen nur durch Wiederholung. (vgl. Jäncke 2005: 85-89)

Schließlich werden die Informationen aus dem Gedächtnis wieder abgerufen. (vgl. Jäncke 2005: 85-89)

3.2 Gedächtnis

3.2.1 Gedächtnismodell

Das Gedächtnis und der Lernprozess stehen in komplexen Zusammenhängen, die noch nicht vollständig erforscht sind.

Über die Jahre entwickelten sich mehrere Modelle und Vorstellungen davon, wie der Mensch lernt. Das populärste Gedächtnismodell wurde von Atkinson und Shiffrin 1968 eingeführt. Sie unterscheiden zwischen Ultrakurzzeit-, Kurzzeit- und Langzeitgedächtnis. (vgl. Holzinger 2001: 39)

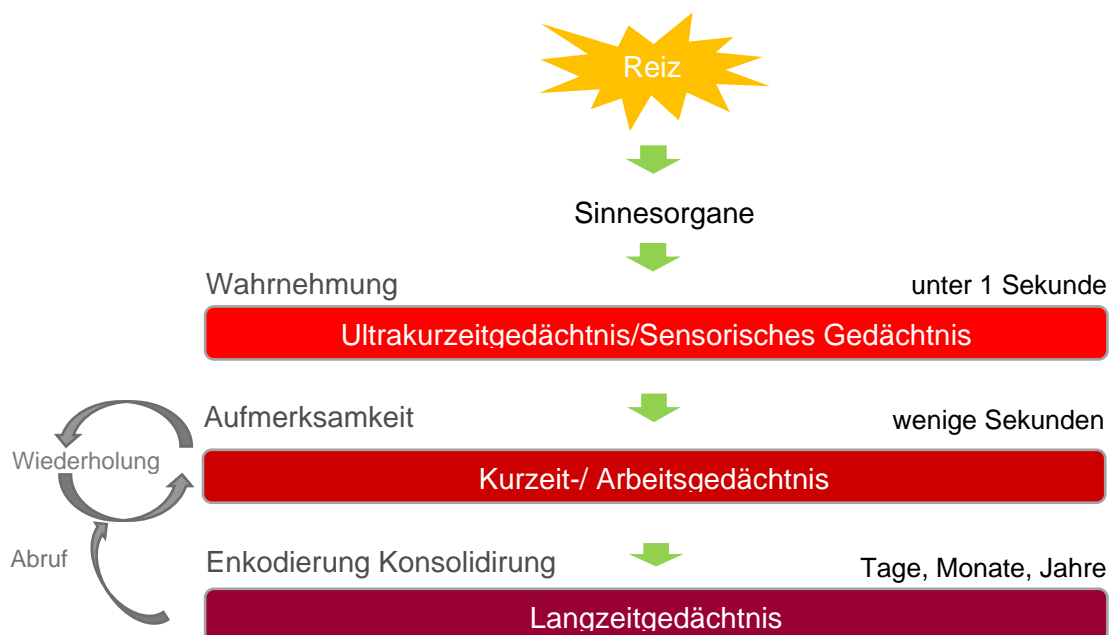


Abbildung 3.1: Die drei Ebenen des Gedächtnisses (in Anlehnung an Editions Atlas SA o. J.: "Das Gedächtnis")

Das *Ultrakurzzeitgedächtnis* nimmt Sinneswahrnehmungen auf und speichert es nur für Sekundenbruchteile. Es wird auch sensorisches Gedächtnis genannt. Nur, wenn Wahrnehmungen Aufmerksamkeit geschenkt wird, gelangen sie ins Kurzzeitgedächtnis. (vgl. Editions Atlas SA o. J.: "Das Gedächtnis"; Holzinger 2001: 39)

Erst das *Kurzzeitgedächtnis* (auch Arbeitsgedächtnis genannt) speichert 7+/- 2 sog. Chunks für wenige Sekunden ab. Ein Chunk ist eine Informationseinheit, die subjektiv relevant scheint, unterschiedlich komplex ist und von jedem anders gebildet wird. (vgl. Editions Atlas SA o. J.: "Das Gedächtnis"; Holzinger 2001: 40)

Im *Langzeitgedächtnis* werden Informationen als Erinnerungen „dauerhaft“ gespeichert und per Abruf für das Arbeitsgedächtnis bereitgestellt. (vgl. Holzinger 2001: 40) Informationen können nach Störungen verloren gehen oder durch zu wenige Abrufe

vergessen werden. Häufig ist das Wissen nur passiv vorhanden, das heißt, Dinge werden erkannt, können aber nicht aktiv abgerufen werden. Beim Lernen werden alte kognitive Schemata, die sich jeder Mensch im Laufe seines Lebens aufbaut, mit den neuen Informationen aus dem Arbeitsgedächtnis erweitert und bei regelmäßiger Abrufung automatisiert. (vgl. Editions Atlas SA o. J.: "Das Gedächtnis"; Holzinger 2001: 40f; Kerres 2013: 155)

Das Langzeitgedächtnis bildet sich durch die Veränderung an den Synapsen, wodurch sich die Erregungsübertragung verbessert. Ferner spielen Proteine und Gene bei der Bildung des Langzeitgedächtnisses eine Rolle. (vgl. DocCheck Medical Services GmbH o. J.; Vontobel 2006: 15)

3.2.2 Gedächtnissysteme des Langzeitgedächtnisses

Lernprozesse und Gedächtnisprozesse sind eng miteinander verwoben. Unterschieden werden jedoch fünf Gedächtnissysteme des Langzeitgedächtnisses. Diese Gedächtnissysteme lassen sich zwei Gedächtnisprinzipien zuordnen, dem expliziten und dem impliziten Gedächtnis. (vgl. (Jäncke 2005: 84; Editions Atlas SA o. J.: "Das Gedächtnis")

	Merkmal	Kurzbeschreibung
Explizites Gedächtnis	Semantisches Gedächtnis	Faktenwissen, bewusstes Gedächtnis
	Episodisches Gedächtnis	Zeitliche Bezüge, bewusstes Gedächtnis, persönlich erlebte Ereignisse
Implizites Gedächtnis	Perzeptuelles Gedächtnis	Bewusste und unbewusste Wahrnehmungen
	Prozedurales Gedächtnis	Handlungsabläufe, unbewusst (Kleinhirn)
	Priming-System	Vorbewusstes Gedächtnis für Wahrnehmung und Kognition

Tabelle 3.1: Gedächtnissysteme und Gedächtnisprinzipien des Langzeitgedächtnisses

Das *explizite Gedächtnis* (auch deklaratives oder bewusstes Gedächtnis) beinhaltet das semantische und das episodische Gedächtnis. Verarbeitet werden solche Informationen im Frontalkortex (Großhirnrinde an der Stirnseite des Hirns), der Informationen selektiert, organisiert und modifiziert. Hier werden aus den neuen Informationen die wichtigen von den unwichtigen Informationen getrennt und priorisiert. Anschließend

werden die Informationen organisiert, gruppiert, häufig interpretiert und modifiziert – Diese Prozesse laufen in gleichem Maße beim Abruf von Informationen ab. Das bedeutet, dass Informationen schon kurz nach der Aufnahme interpretiert werden und die Selektion daher abhängig von individuellen Informationen und Interpretationsergebnissen ist. (vgl. Jäncke 2005: 87)

Das *implizite Gedächtnis* speichert unbewusst Informationen wie erlernte Fähigkeiten, Handlungsfolgen und Wahrnehmungsergebnisse ab. Dieses Gedächtnisprinzip findet sich in anderen Bereichen des Gehirns, zum Teil in den Basalganglien. Problematisch ist, dass bei der Informationsaufnahme der Frontalkortex beim Abruf älterer Informationen die Funktion der Basalganglien hemmt. Vor allem beim Lernen von impliziten Informationen (z. B. beim Erlernen einer Handlung) ist es wichtig, dass nicht zu viele explizite Informationen (z. B. verbale Kommandos) verwendet werden. (vgl. Jäncke 2005: 88f)

Der Neuropsychologe Jäncke (2005: 90) führt weitere Erkenntnisse an: Emotionsgeladene Informationen werden besser behalten als „neutrale“. Dies betrifft vor allem unangenehme Informationen. Zudem lernen die meisten Individuen besser, wenn sie die Informationen selbst strukturieren und eigenverantwortlich aufnehmen können. Dies bestätigt Holzinger (2001: 42f), der ergänzt, dass Informationen besser reproduziert werden können, wenn sie nicht nur oberflächlich wahrgenommen wurden, sondern als Basis einer intensiven Auseinandersetzung dienen. Dies unterstützt der Ansatz der Interaktivität (siehe Abschnitt 2.3.7).

3.3 Lernen mit AR

Wie in Abschnitt 2.3.7.3 beschrieben, führt die Interaktivität zu einem gewissen Grad an Immersion. Studien haben gezeigt, dass die Immersion in eine digitale Umwelt den Wissenserwerb verbessert, da er verschiedene Perspektiven, situiertes Lernen und Transfer ermöglicht. (vgl. Dunleavy / Dedebach o. J.: 4)

Es gibt zwei Ansätze, bei denen die Vorteile von virtuellen Lerninhalten auf der Hand liegen: beim situierten Lernen und beim konstruktivistischen Lernen⁴. Beim *situierten Lernen* geht es darum, dass Wissen immer in einem bestimmten Kontext erlernt wird und dieser Kontext die Lernerfolge beeinflusst. Das erlernte Wissen muss dann transferiert, also in andere Situationen und Kontexte übertragen werden. Durch die Anwendung von virtuell erweiterten Kontexten können reale Probleme dargestellt werden, sodass der Transfer in die echte Welt näher ist. Dies ist beispielsweise bei Flugsimulationen und medizinischen Eingriffen der Fall. (vgl. Dunleavy / Dedebach o. J.: 5f)

Der Ansatz des *Konstruktivismus* geht davon aus, dass Lernende neues Wissen aufbauen, in dem sie die neuen Informationen in Beziehung zu ihrem Vorwissen, ihren Erfahrungen, ihrem soziokulturellen Hintergrund und dem Kontext setzen. Die Aufgaben, die gelernt werden sollen, müssen in sinnvolle, realistische Situationen integriert sein und sollten selbstgeleitet, aktiv und unterschiedlich dargestellt sein. AR bringt den Lernenden in einen sozialen und physikalischen Kontext und führt und unterstützt seinen Lernprozess mit verschiedenen Arten der Darstellung. (vgl. Dunleavy / Dedebach o. J.: 7f)

Somit kann AR eingesetzt werden, um auf ein Problem verschiedene, sich ergänzende Perspektiven zu schaffen. Aus dem Ergänzen und Overlaying von Informationen in einen echten physikalischen Raum wird dieser zu einer Quelle für vielerlei ungenutzte Lernmöglichkeiten. Durch die Verwendung mobiler Endgeräten können weitere Ressourcen wie das Internet oder weitere Software genutzt werden, um das Lernen zu vereinfachen. (vgl. Dunleavy / Dedebach o. J.: 12f)

Dunleavy / Dedebach (o. J.: 13) haben aber ebenfalls herausgefunden, dass Lernende und Schüler bei AR bisher mit Entscheidungen, Komplexität und Navigation kognitiv überfordert werden. Hierbei kann helfen, mit einer einfachen Struktur anzufangen und die Komplexität schrittweise zu steigern, ausdrücklich Schritt für Schritt zu unterstützen, die Objektanzahl zu minimieren und Text durch Audio zu ersetzen.

⁴ Trotz Überlappungen in der Lerntheorie werden diese zwei Ansätze in einigen Literaturen unabhängig voneinander betrachtet (vgl. Kerres 2013: 142-146)

4 Konzept

4.1 Vorgehensweise bei der Konzepterarbeitung

Ziel dieser Arbeit ist die Entwicklung einer prototypischen App, die als interaktiver Museumsführer dient (siehe hierzu Abschnitt 4.2.2). Um die Entwicklung und Konzeption einem realistischen Projekt nachzuempfinden, unterstützte die Kunstvermittlung der SKK, vor allem Dr. Sybille Brosi, Petra Erler-Striebel und Sandra Trevisan. Sie lieferten inhaltliche Informationen, halfen bei der Auswahl eines geeigneten Beispielsbildes, erklärten, welche Informationen erwähnenswert sind u. Ä. (vgl. Brosi / Erler-Striebel / Trevisan (06.04.2017, 16:00 MESZ) Die Anwendung und das Konzept könnte allerdings auch auf jede andere Kunsthalle angewandt und genutzt werden.

Zunächst half ein einfacher Fragebogen (siehe Anhang), die Zielgruppe gemeinsam mit den Kunstvermittlerinnen zu definieren und Informationen über Designvorgaben einzuholen. Zusammen mit dem Fragebogen und eigenen Überlegungen entstand ein zielgruppenorientiertes Konzept, das sowohl die Gestaltung, als auch die Umsetzungsmöglichkeiten berücksichtigt.

4.2 Inhaltliches und gestalterisches Konzept

4.2.1 Zielgruppe

4.2.1.1 Allgemeine Zielgruppenbeschreibung für Kunstmuseen

Die Zahl der Museumsbesucher von Kunstmuseen in Deutschland hat sich 2015 im Vergleich zum Vorjahr um 4,6% gesteigert. (vgl. Institut für Museumsforschung 2016: 7) Das zeigt deutlich, dass weiterhin Nachfrage und Interesse an Kunst und ihrem Verstehen herrscht.

Um das Lernziel der Anwendung zu erreichen, ist es notwendig, bei der Erarbeitung eines Konzeptes die Zielgruppe vor Augen zu haben (siehe Abschnitt 2.3.4.1). Eine Möglichkeit, seine Zielgruppe einzuschätzen, ist die Einteilung in soziale Milieus, bei denen Werte und Werteprioritäten, Gesellschaftsbilder, Lebensstile und berufliche Lage bedacht werden. (vgl. Kleinhüchelkotten 2012: 6)

In einer Statistik von 2009 werden Zielgruppen für Kunstmuseen mithilfe von Sinus-Milieus⁵ beschrieben. Dabei stellte sich heraus, dass Kunstmuseumsbesucher hauptsächlich den Milieus der Konservativen, Etablierten und Postmateriellen zugeordnet werden können und somit die Hauptzielgruppe bilden. Alle drei Milieus gehören der Oberschicht bzw. oberen Mittelschicht an. (vgl. Kleinhüchelkotten 2012: 8f)

Laut Kleinhüchelkotten (2012: 15-18) sollten Etablierte, die als anspruchsvoll, intelligent und ästhetisch gelten, nicht mit grellen und aufdringlichen Darstellungen angesprochen werden. Postmaterielle, die oft originell, intelligent und ungewöhnlich seien, mögen es zwar grell, jedoch nur, wenn es mit Humor verknüpft ist. Konservative gälten als sachlich, informativ und seriös. Deshalb sollten modisch aktuelle Ästhetik verhalten eingesetzt werden, da hier große Skepsis zu erwarten sei.

Deshalb wurde beim Design der Oberfläche darauf geachtet, schlicht zu bleiben, die Farben einzuschränken und einen klaren, strukturellen Aufbau beizubehalten.

4.2.1.2 Zielgruppe der SKK

In der SKK sind vormittags vor allem Schulklassen und Gruppen bis ca. 12 Jahre anzutreffen. Darüber hinaus besuchen viele Erwachsene die Ausstellungen, das Durchschnittsalter und die Herkunft der Besucher ist allerdings sehr von den jeweiligen Ausstellungen abhängig. Meist sind die Besucher (tendenziell mehr Frauen) aus der Region

⁵ „Die Sinus-Milieus sind das Ergebnis von über 30 Jahren sozialwissenschaftlicher Forschung. [...] Die Sinus-Milieus gruppieren Menschen, die sich in ihrer Lebensauffassung und Lebensweise ähneln“ (SINUS Markt- und Sozialforschung GmbH 2015: 3)

Karlsruhe oder aus dem weiteren Baden-Württemberg. Die meisten haben ihre Schul- ausbildung mit der Hochschulreife abgeschlossen. (vgl. Brosi / Erler-Striebel / Trevisan (06.04.2017, 16:00 MESZ))

Die bisherigen Angaben lassen darauf schließen, dass die Zielgruppe aus der Mittel- bis Oberschicht kommt und über eine gute Bildung verfügt. Nach Aussagen der Kunst- vermittlerinnen ist das Vorwissen unterschiedlich und von der Ausstellung abhängig. (vgl. Brosi / Erler-Striebel / Trevisan (06.04.2017, 16:00 MESZ))

Sie schätzen die Besucher, egal welchen Alters, als sehr aufgeschlossen gegenüber tech- nischen Neuerungen (Smartphones, AR, Tablets etc.) ein, sodass den meisten der Um- gang damit geläufig sein dürfte bzw. sie bereit sind, durch „Learning by Doing“ das Gerät und das Programm zu erkunden. Für die meisten dürfte Augmented Reality neu sein, was nach Meinung der Kunstvermittlerinnen trotzdem zu Begeisterung und posi- tiven Überraschung führen sollte.

4.2.2 Lehrziele und Aufgaben der App

Die App, die in dieser Bachelorarbeit prototypisch entwickelt wird, ist kein klassisches E-Learning. Es verfolgt zwar den Zweck, den Nutzern Neues beizubringen, dennoch gibt es in der App keine Quizfragen, keine Rückmeldung über mögliche Lernerfolge, keine Inhalte, die angeklickt werden *müssen*, um ein bestimmtes Ziel zu erreichen.

Die App soll die (wahrscheinlich) vom Nutzer gewünschten Informationen im genau richtigen Moment und unmittelbar anzeigen. Darüber hinaus sollen Informationen an- gezeigt werden, die dem Nutzer helfen, die (angenommenen) Gedanken des Künstlers besser zu verstehen. Das heißt, durch die App soll der Nutzer auf Dinge aufmerksam gemacht werden, die Rückschlüsse auf Interpretationen des Bildes zulassen und damit die Wahrnehmung und das Verständnis des Bildes verändern. Bei den Informationen handelt es sich primär um Faktenwissen und episodisches Wissen, welches dann im deklarativen bzw. episodischen Gedächtnis abgespeichert wird (vgl. Abschnitt 3.2.2).

Die Hauptaufgabe der App ist es, den Nutzer durch das Museum zu führen und ihm die Dinge zu erklären, die ein Museumspädagoge, Kunsthistoriker etc. erklären würde. Ein Vorteil der App liegt im Vergleich zur persönlichen Führung darin, dass der Nutzer völlig individuell entscheiden kann, welches Bild und welche Bildbereiche ihn interes- sieren, wie lange er verweilen will, welche Inhalte genau seine Aufmerksamkeit erre- gen etc.

Dadurch, dass durch die Gemälde an sich Emotionen geweckt werden, die durch die Zusatzinformationen konkretisiert und geleitet werden können, ist es wahrscheinlich,

dass die gebotenen Informationen auch behalten werden. Dies ist allerdings sehr vom Werk, den Informationen und dem Betrachter abhängig.

4.2.3 Lernziel der Nutzer

Die Lernziele legt jeder Nutzer selbst fest. Entweder der Nutzer erhält „nur“ die Informationen, wie das Gemälde heißt, wann es entstand und wer es erschaffen hat, oder er lässt sich auf die Zusatzinformationen ein, die er außerdem angeboten bekommt. In welcher Intensität, mit welcher Informationstiefe und mit welcher Absicht jeder Einzelne die App nutzt, bleibt ihm überlassen.

Inwieweit sich der Wissensstand ändert, ist sehr vom Nutzer selbst abhängig. Es wird davon ausgegangen, dass sie die App freiwillig nutzen, weil sie intrinsisch motiviert sind und sich neues Wissen aneignen bzw. neue Informationen über ein Gemälde erhalten wollen. Durch die intrinsische Motivation ist es wahrscheinlich, dass Inhalte ins Langzeitgedächtnis aufgenommen werden und der Nutzer lernt.

Selbst wenn der Nutzer die App nur verwendet, weil er von der Technik begeistert ist und sie ausprobieren möchte, ist es wahrscheinlich, dass positive Emotionen und der Spaß dazu führen, dass Informationen abgespeichert werden.

4.2.4 Grundstruktur der Informationen

Der „Lernstoff“ – im vorliegenden Fall die unterschiedlichen Informationen, die dem Nutzer angeboten werden – werden nicht linear vermittelt. Der Nutzer wählt selbst aufgrund seines Interesses und seines Vorwissens Informationen aus. Dies kann durch ein konstruktivistisches Lernsetting ermöglicht werden. (vgl. Vontobel 2006: 9 und Abschnitt 3.3)

Jedes Gemälde bzw. Werk, das in der App hinterlegt wird, bildet ein eigenes „Topic“, eine Einheit, der Informationen zugeordnet werden.

In der App gibt es zwei unterschiedliche Informationsarten: Informationen, die der Nutzer nach dem erfolgreichen Tracking unmittelbar angezeigt bekommt, und Informationen, die der Nutzer nach Interesse auswählt. Die Informationen, die der Nutzer nach Interesse auswählt sind, sind in Abbildung 4.1 als gelbe Sterne dargestellt. Die Zusatzinformationen variieren abhängig von jedem einzelnen Werk. An diesen Stellen sind Informationen hinterlegt, die dem Nutzer helfen sollen, ein Gemälde besser zu verstehen, die Entstehungsgeschichte miteinbeziehen zu können, Charakteristika im Malstil herauszuarbeiten etc. Die Informationen sind voneinander unabhängig und dürfen nicht aufeinander aufbauen, da der Nutzer die Möglichkeit haben muss, das

Bild zu verstehen, ohne dass er die anderen Informationen zwingend gelesen hat. Es soll möglich sein, dass er nur auswählt, was seine Aufmerksamkeit weckt. Sie können in unterschiedlicher medialer Form vorliegen. Texte sind das wahrscheinlichste. Denkbar wäre aber auch ein Video oder Bilder.

Die „Basisinformationen“ (in Abbildung 4.1 rechts vom Gemälde dargestellt, am roten Stern hängend) werden unmittelbar angezeigt, sobald das Gemälde von der Software erkannt wird. Die Informationen „Lebensspanne“, „Nationalität“ und „Portrait“ hängen an der Information „Künstler“. Diese Abhängigkeit ist lediglich für die Strukturierung der Informationen in der verwendeten XML-Datei relevant, nicht aber für die Ausgabe (vgl. Abschnitt 4.3.3).

Um die Aufmerksamkeit auf die interessanten Stellen (im Folgenden als „Hotspot“ bezeichnet) des Gemäldes zu lenken, werden an dieser Stelle zunächst leuchtende Markierungen erscheinen. Erst bei Klick bzw. Touch auf den Hotspot werden die Informationen angezeigt. So wird ein „information overload“ verhindert.

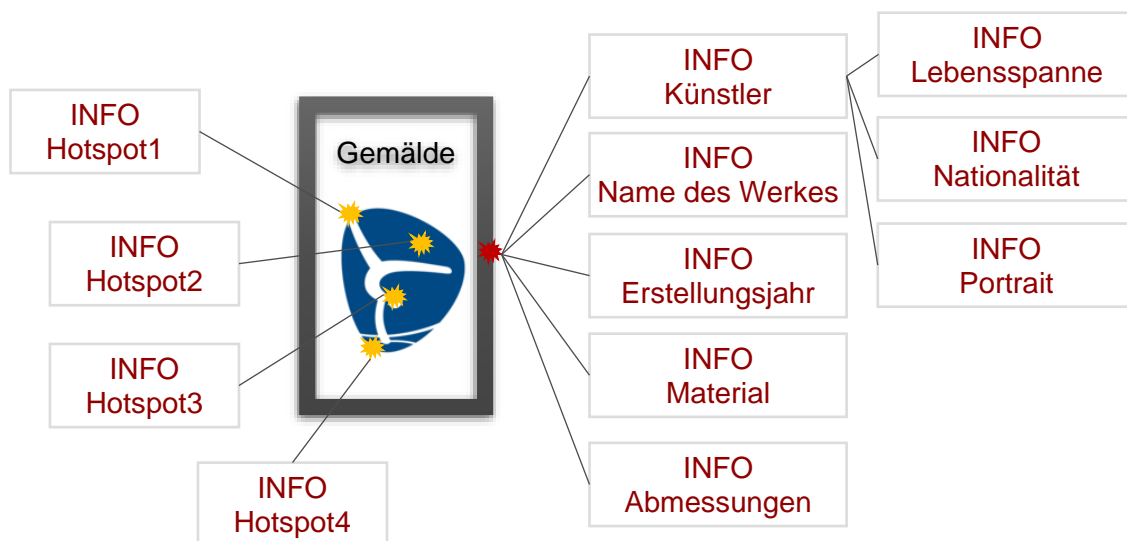


Abbildung 4.1: Informationsarten und Aufbau eines "Topics"

4.2.5 Inhalte

Die Textinformationen für die einzelnen Hotspots werden in XML-Dateien geschrieben. Der Grund dafür ist die angestrebte Mehrsprachigkeit. Da der Text in einer separaten Datei steht, könnte von einem Übersetzer eine entsprechende XML-Datei in der gewünschten Sprache erstellt werden. Zusätzlicher Vorteil einer XML-Datei ist, dass Änderungen schnell vorgenommen werden können. Die Anwendung müsste von der Software nur erneut ausgespielt und auf den Geräten aktualisiert werden.

Welche inhaltlichen Elemente verarbeitet werden, ist abhängig vom Gemälde und den zur Verfügung stehenden Materialien. Vorstellbar sind hier 3D-Modelle, Videos etc.

Die App wird erst zu „AR im engeren Sinne“ (vgl. Abschnitt 2.1.1.2), wenn der Bezug eines virtuellen 3D-Objekts zur echten Welt hergestellt wird. Bei den meisten Gemälden wird dies wahrscheinlich nicht der Fall sein. Ein denkbares Szenario, um „AR im engeren Sinne“ zu erreichen, ist es, zusätzlich zu einem Gemälde eine Statue als 3D-Objekt in der realen Welt zu „verankern“. Hierbei zeigt sich ein Vorteil von AR: Der reale Raum lässt Platz, damit der Mensch sich frei bewegen kann und die Gemälde wirken zu lassen, durch die Einblendung virtueller Objekte es aber möglich ist, den breiten Schaffungsrahmen eines Künstlers vollständig darzustellen und seine Ausstellung zu erweitern ohne Ausgaben für weitere Werke.

4.2.5.1 Umgang mit Text und Ton

Beim Lernen legen Menschen im Gedächtnis Modelle und Schemata an. Anfänger, die noch nicht so tief im Thema sind, haben es oft schwerer, die neuen Informationen zu behalten, da ihnen geeignete Schemata fehlen, in die sie die neuen Informationen einordnen können. Deshalb müssen die präsentierten Informationen reduziert werden, um das Arbeitsgedächtnis nicht zu überlasten. (vgl. Kerres 2013: 167; Rey, o. J.: 26) Das bedeutet, dass die Textblöcke recht kurz sein müssen. Das ist auch wichtig im Hinblick auf die erweiterte Anwendung mit mehreren Topics. Wenn jedes Topic sehr viele lange Textinformationen anbietet, wird der Nutzer schnell müde, die Anwendung könnte als anstrengend empfunden werden und die Behaltensleistung sinken.

Der Nutzer kann in der Anwendung auswählen, ob er lieber einen Text lesen, oder ob er den Text zusätzlich vorgelesen haben möchte. Obwohl das Hören von geschriebenen Texten bei gleichzeitigem Lesen für das Arbeitsgedächtnis eine unnötige Anstrengung darstellt und die Wahrscheinlichkeit mindert, dass die Informationen behalten werden (vgl. Kerres 2013: 168f), werden bei der Aktivierung der Vorlesefunktion die Texte nicht ausgeblendet. Da der Text viele Beschreibungen enthält, ist es sehr wahrscheinlich, dass der Nutzer das Bild genauer betrachtet und nicht den Text mitliest.

Wenn Bildinformationen und verbale Informationen nahe beieinander präsentiert werden, lassen sie sich besser erhalten. (vgl. Kerres 2013: 167). Dies ist eines der Grundprinzipien, die bei der Anwendung berücksichtigt werden. Da der Nutzer das Bild vor sich sieht (sowohl in Realität als auch auf dem Display seines Tablets) erhält er sofort interessante Informationen in verbaler Form (entweder geschriebene Texte oder Audiobeschreibungen). Da das Bild zusätzlich auf dem Display erscheint, muss der Nutzer nicht seinen Fokus verstellen, sondern kann durchgehend auf das Tablet schauen.

4.2.5.2 Formulierung der Texte

Die Anwendung nutzt eine Text-To-Speech-Software, die den geschriebenen Text vorliest. Dies hat den Vorteil, dass keine neuen Audiodateien aufgenommen werden müssen, wenn sich die Texte noch einmal ändern sollte oder noch weitere Hotspots hinzukommen. Außerdem ist diese Variante deutlich günstiger als Aufnahmen im Tonstudio von professionellen Sprechern. Die SKK greift zwar bei der Erstellung der Audioguides auf professionelle Sprecher zurück, in diesem Projekt ist dies aber wegen mangelnder Ressourcen nicht möglich. Da die Texte, die gelesen werden, mit den gehörten übereinstimmen ist es besonders wichtig, dass kurze, prägnante Sätze enthalten, leicht verständlich sind, wenige Fremdworte enthalten und nicht monoton wirken. Damit die Texte als Audiotexte funktionieren und nicht zu anstrengend für den Hörer sind, orientieren sich Satzkonstruktionen und Wortwahl an der Alltagssprache. (vgl. Ballstaedt 1997: 94f)

Die dargestellten Objekte auf den Gemälden sollten so spezifisch wie möglich benannt werden, da dadurch visuelle Details besser behalten werden. Beispielsweise sollte ein dargestelltes Brotmesser besser als ein solches bezeichnet werden statt einfach nur als ein Messer. (vgl. Ballstaedt 1997: 254) Da in der Kunst Detailtreue für das Verständnis wichtig ist, müssen die Texte besonders präzise formuliert werden.

4.2.5.3 Audioinformationen

Wie in Abschnitt 2.3.5 erwähnt, wecken Audioinformationen die Aufmerksamkeit, führen zu einer längeren Betrachtung des Bildes und somit zu einer intensiveren Auseinandersetzung mit dem Abgebildeten und wirken persönlicher. Überdies führen sie dazu, dass das Bild länger betrachtet wird, was für das Lehrziel der App förderlich ist. (vgl. Rey, o. J.: 24)

4.2.5.4 Internetlinks

Da in der SKK das WLAN-Netzwerk der Stadt Karlsruhe KA-WLAN genutzt werden kann, ist es möglich, den Nutzer auf externe Seiten für weitere Informationen zu verlinken. Eine Möglichkeit wäre es z. B., weitere Gemälde des Künstlers vorzustellen und auf das Museum zu verlinken, in dem das Werk ausgestellt ist.

4.2.6 Interaktivität, Stress und Motivation

Wenn die Interaktivitätsstufen nach Schulmeister (siehe Abschnitt 2.3.7.2) betrachtet werden, wird mit dieser Anwendung die Stufe 2 erreicht (Multiple Darstellungen betrachten und rezipieren), da der Nutzer individuell die Hotspots auswählt. Je nach verwendetem Material kann auch Stufe 3 (Repräsentationsform variieren) oder 4 (Inhalt

der Komponente beeinflussen: Variation durch Parameter- oder Datenvariation) erreicht werden. Stufe 4 trüfe beispielsweise zu, falls Animationen oder variable 3D-Modelle zum Einsatz kommen. Dies dürfte aber eher die Ausnahme sein.

Durch die leuchtenden Hotspots wird im Gehirn Stress ausgelöst. Dieser Stress soll förderlich sein und die Aufmerksamkeit steuern. Dadurch, dass die Aufmerksamkeit auf die Hotspots gelenkt wird, soll der Nutzer schon beginnen, das Gemälde selbstständig zu interpretieren, in dem er sich über die markierte Stelle (z. B. einen düsteren Hintergrund) Gedanken macht, bevor er die zugehörigen Informationen lesen kann. Sie werden erst auf Touch-Befehl eingeblendet.

Ein wichtiger Motivationsfaktor ist die Neugierde, die der Betrachter schon mit in die Kunsthalle bringt, da ihn das Thema schon vorher interessant erscheint. Durch die Hotspots steigt die Neugierde, welche noch unbekannt Informationen hinter dem Gemälde stecken könnte bzw. ob die eigenen Vermutungen sich mit den Deutungsvorschlägen decken.

4.2.7 Gestaltung

4.2.7.1 Schrift

Es wird davon ausgegangen, dass Lesen am Bildschirm anstrengender, langsamer und ermüdender ist als Lesen auf Papier. Deshalb sollte die Schriftart leserlich sein. (vgl. Ballstaedt 1997: 87; Ganapathy 2013: 177) Durch die relativ kleinen Textboxen ist der Text linksbündig ausgerichtet, damit große Wortzwischenräume und häufige Worttrennungen wie im Blocksatz vermieden werden. (vgl. Ballstaedt 1997: 89)

Es wird empfohlen, eine serifenlose Schrift wie Helvetica in der Schriftgröße 12 Pt oder größer zu verwenden. (vgl. Ballstaedt (1997: 89)

Im Corporate Design der SKK werden die Schriftarten Frutiger Next Pro für Überschriften, Absatzüberschriften sowie Bildunterschriften und Sabon Oldstyle Figures für Fließtexte vorgeschrieben. (vgl. Mernberger (10.04.2017, 10:43 MESZ)) Frutiger Next Pro ist eine serifenlose, Sabon Oldstyle Figures eine Serifenschrift. Auf die Verwendung der Serifenschrift Sabon wurde wegen der schlechteren Leserlichkeit auf Bildschirmen verzichtet und somit nur Frutiger verwendet.

4.2.7.2 Farben

Die Hauptfarbe der SKK ist der Rotton 15-100-100-0 im CMYK-Farbschema (in RGB 217-0-0). Er stellt die Basisfarbe dar. Davon wurden weitere Rottöne für die App abgeleitet: 217-130-130 und 190-0-0. Ansonsten werden Weißtöne und Schwarz verwendet.

4.2.7.3 Kontrast

Der Text muss auf dem Hintergrund gut leserlich sein, was durch einen hohen Kontrast zwischen Vorder- und Hintergrund zustande kommt. (vgl. Ballstaedt 1997: 92) Da der Text nicht direkt über das Gemälde eingeblendet werden kann, steht der Text in eine Box mit Hintergrundfarbe. Um einen hohen Kontrast zu erhalten, der nicht zu stark ist, wurde für die Schriftfarbe ein mittlerer Grauton (50-50-50) gewählt; der Hintergrund ist weiß.

4.2.7.4 Icons und Symbole

Die ausgewählten Icons müssen auf den ersten Blick identifizierbar sein. (vgl. Ganapathy 2013: 179) Sie dienen wie Piktogramme dazu „ohne Vermittlung durch die Sprache einen Begriff [zu] aktivieren oder eine Handlung aus[zulösen“ (Ballstaedt 1997: 271) Hierzu wurden bekannte Icons gewählt, die auch in anderen Anwendungen und Programmen auf Computer und Smartphones bekannt sind und als eindeutig verstanden werden⁶:





Icon	Bedeutung
	Kehrt auf die „Welcome“/Home-Seite zurück
	Öffnet das Einstellungsfenster, um die Sprache zu ändern und die Vorlesefunktion einzuschalten.
	Ändert die Spracheinstellung zwischen Französisch, Deutsch und Englisch
	Schaltet die Vorlesefunktion ein bzw. aus
	Startet und stoppt das Vorlesen des Textes

Tabelle 4.1: Verwendete Icons und ihre Bedeutungen

Da die Aufmerksamkeit der Nutzer überwiegend auf der linken Bildschirmseite liegt, werden die Icons (bis auf den Start-/Stopp-Button) links oben platziert. Sie sind inhaltlich gruppiert. (vgl. Ballstaedt 1997: 285)

⁶ Bei der Auswahl der Icons wurde nicht auf Kulturneutralität geachtet. Dies war im Rahmen dieser Arbeit leider nicht möglich

4.3 Technisches Konzept

4.3.1 Ausgabegerät

Die App wird für Tablets optimiert. Diese haben den Vorteil, dass ihr Display im Vergleich zu dem von Smartphones größer ist, sodass alle Informationen dargestellt werden können, ohne das reale Bild völlig zu verdecken.

Eine denkbare Alternative wäre die Nutzung von Datenbrillen. Tablets sind allerdings im Moment noch kostengünstiger und der Umgang mit ihnen ist weitläufig bekannt. Des Weiteren ist es vorstellbar, dass der Nutzer durch die Dauerpräsenz der Informationen bei einer Datenbrille überfordert wird. Er kann das Bild nicht ohne Informationen betrachten, ohne die Brille absetzen zu müssen. Dies würde vermutlich als nicht sehr nutzerfreundlich wahrgenommen werden.

Für die SKK ist es im Moment aus Budgetgründen nicht möglich, Tablets zur Verfügung zu stellen, der Wunsch ist jedoch da. Konkrete Umsetzungsmöglichkeiten liegen jedoch in der Verantwortung der SKK und werden nicht in dieser Thesis betrachtet.

Als Ausgabegerät dient im Rahmen dieser Arbeit das SAMSUNG GT-P5210 Galaxy Tab 3. Es hat eine Bildschirmdiagonale von 25,7 cm (10.1 Zoll), eine Bildschirmauflösung von 1200 x 800 Pixel und läuft mit dem Betriebssystem Android 4.4.2.

4.3.2 Trackingverfahren

Als Trackingverfahren wird das optische Tracking mit dem Inside-Out-Prinzip verwendet: Die Kameras sind am Tablet befestigt und mit Hilfe des Markers wird die Position des Tablets im Raum berechnet. Hierbei dienen die Gemälde als optische Marker. Falls sich das Bild nicht als Trackingobjekt eignet, müssten künstliche Marker erzeugt werden und neben dem Gemälde angebracht werden, gleichwohl dies aus optischen Gründen unerwünscht ist.

Andere Trackingverfahren sind weniger geeignet, da GPS beispielsweise innerhalb von Gebäuden nicht ausreichend funktioniert. Eine andere Möglichkeit könnte das Tracken in kabellosen Netzwerken wie Bluetooth oder WLAN sein. Da zunächst die Genauigkeit dieser Systeme getestet werden müsste, ist die Umsetzung im Rahmen dieser Thesis leider nicht möglich.

Falls die App weiterentwickelt werden sollte und andere Kunstmuseen an diesem Projekt teilnehmen, könnten GPS-Trackings verwendet werden, um abhängig vom Standort automatisiert die Inhalte für die App zu laden. Viele Nutzer möchten aber aus Da-

tenschutzgründen ihren Standort nicht zur Verfügung stellen, sodass diese Problematik über eine simple Abfrage, per manuelle Eingabe oder über das Scannen eines Markers am Museumseingang gelöst werden könnte.

4.3.3 Aufbau der XML-Datei

Damit ein Austausch von Text abhängig von der ausgewählten Sprache schnell möglich ist, sollen XML-Dateien eingesetzt werden.

Für jedes Kunstwerk wird eine XML-Datei in der jeweiligen Sprache erstellt. Für die SKK sind momentan Englisch, Deutsch und Französisch am sinnvollsten. Je nach gewählter Sprache wird die entsprechende Datei geladen. Die Dateien werden wie folgt benannt:

- **[nachname]-[vorname]_[name-des-werkes]_[Kürzel der Sprache]**
Bsp.: dix-otto_die-sieben-todsuenden_de.xml

Die XML-Datei ist wie folgt aufgebaut (siehe Abbildung 4.2 und Tabelle 4.2):

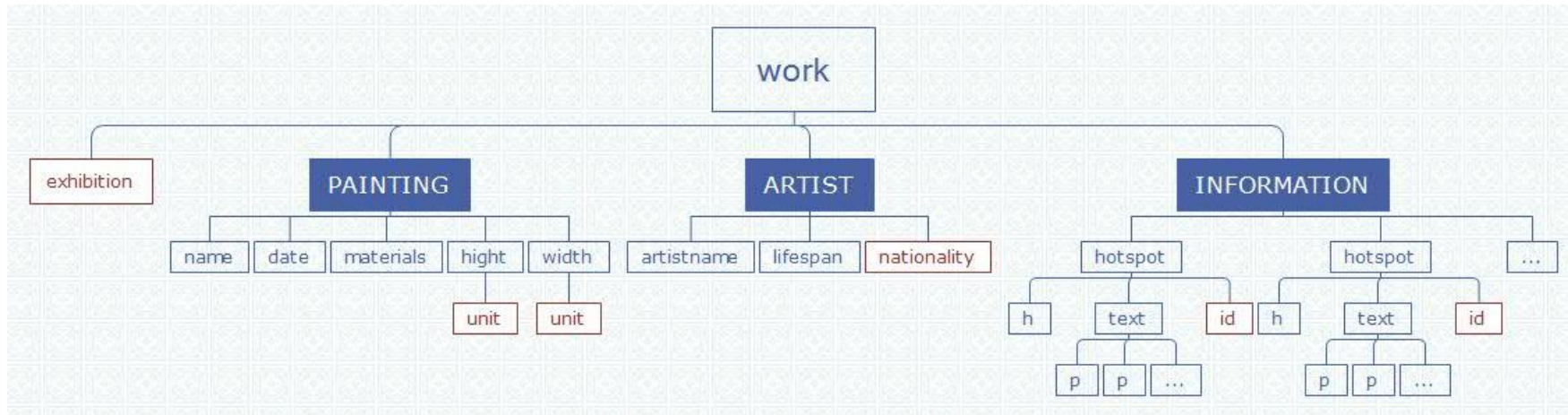


Abbildung 4.2: Grobstruktur der XML-Dateien

XML-Knoten	Beschreibung
work	Wurzelknoten für das Kunstwerk. Attribut gibt an, in welcher Ausstellung es zu sehen ist
<i>exhibition</i>	<i>Hier kann hinterlegt werden, in welcher Ausstellung das Bild zu sehen ist. (Information für die Programmierung der App nicht notwendig)</i>
painting	Sammelknoten für Informationen über das Gemälde
name	Name des Gemäldes
date	Erstelljahr
materials	Verwendete Materialien des Künstlers
height, width	Höhe und Breite des Gemäldes.
<i>unit</i>	<i>Attribut für die Maßeinheit der Höhe und Breite des Bildes</i>
artist	Sammelknoten für Informationen über den Künstler
lifespan	Lebensspanne des Künstlers
artistname	Name des Künstlers
<i>nationality</i>	<i>Attribut für die Nationalität des Künstlers ⁷</i>
information	Sammelknoten für Informationen über die Hotspots
hotspot	Knoten für Hotspots.
h	Überschrift
<i>id</i>	<i>Attribut zur eindeutigen Identifizierung der Hotspots</i>
text	Beinhaltet den Beschreibungstext des Hotspots, der in der App angezeigt wird.
p	Absätze

Tabelle 4.2: Elemente der XML-Struktur

Die Lebensdaten des Künstlers ändern sich nicht. Deshalb liegt die Überlegung nahe, die Informationen über den Künstler (Name, Lebensspanne, Nationalität; siehe Abbildung 4.1) in eine separate XML-Datei zu schreiben und diese XML-Datei dann mit der XML-Datei zu verknüpfen, die die Informationen über das Kunstwerk enthält. Dies hätte den Vorteil, dass die Künstlerinformationen nur an einer Stelle liegen und nicht bei jeder neuerstellten Datei die Informationen übernommen oder abgetippt werden müssen, da sich dabei gerne Fehler einschleichen. Ein Grund, der gegen dieses Vorgehen spricht, ist die Tatsache, dass manche Künstler Namenszusätze haben wie beispielsweise „der Ältere“. Da diese Information genauso übersetzt werden muss wie die anderen Texte (z. B. „the Elder“), ist es zur Vermeidung von zu vielen Dateien ratsam, die Informationen über den Künstler trotzdem mit den Informationen über das Gemälde in einer XML-Datei zu bündeln. So liegen pro Gemälde statt drei XML-Dateien

⁷ In einer DTD sollte dafür eine Liste angelegt werden. Bisher wurden die Abkürzungen gemäß ISO 3166 ALPHA-2 in Kleinschreibung benutzt (de, fr).

für das Gemälde und drei XML-Dateien für den Künstler (je eine in Deutsch, Französisch und Englisch) nur drei XML-Dateien mit allen Informationen vor

Die XML-Dateistruktur kann noch erweitert werden, reicht jedoch für die prototypische Umsetzung aus. Die Elemente haben englische Bezeichnungen, da sie bei der Übersetzung wegen der großen Verbreitung des Englischen überall verstanden werden dürften. Damit die Dateien austauschbar sind, muss gewährleistet werden, dass die Struktur immer gleich ist. Deshalb sollte noch eine DTD (Document Type Definition) entsprechend erstellt und verknüpft werden. Dies ist im Rahmen dieser Arbeit leider nicht möglich.

5 Umsetzung und Programmierung

5.1 Methodisches Vorgehen

Das Konzept wird beispielhaft am Werk „Die 7 Todsünden“ (1933) von Otto Dix umgesetzt. Dieses Gemälde wurde gemeinsam mit den Kunstvermittlerinnen der SKK ausgewählt (siehe Fragebogen im Anhang). Die textlichen Inhalte in der XML-Datei (siehe Anhang) wurden von den Kunstvermittlerinnen lektoriert.

Der Fokus der Arbeit liegt nicht auf der inhaltlichen Erarbeitung, sondern auf der technischen Umsetzung und Darstellungsweise mit Augmented Reality. Dennoch soll fundiertes Wissen über das Beispielgemälde mit der prototypischen Umsetzung vermittelt werden und die Texte realistisch, verständlich und korrekt formuliert sein. Die Anwendung erkennt noch zwei andere Gemälde aus der SKK, dessen vollständige inhaltliche Aufbereitung im Rahmen dieser Arbeit leider nicht möglich war. Sie dienten als Hilfestellung bei der technischen Umsetzung des Konzeptes.

Bei der Erstellung der Anwendung wurde nach einer groben Einarbeitung in Unity zunächst die Target-Datenbank für Vuforia eingerichtet. Als Target bezeichnet Vuforia den Marker. Bei dieser Anwendung ist das eine Bilddatei des Gemäldes. Die Target-Datenbank wurde während der Erarbeitung mit zwei weiteren Bildern erweitert, so dass die Datenbank drei Marker umfasst und damit drei Gemälde erkannt werden.

Anschließend wurde die Benutzeroberfläche der Szene „museumsfuehrer“ aufgebaut (siehe hierzu Abbildung 5.4 in Abschnitt 0).

Dann wurde die Datenbank aus Vuforia heruntergeladen und in das Unity-Projekt importiert. Nach dem Anlegen des Targets für das Beispielbild wurden verschiedene Darstellungsvarianten für die Hotspots betrachtet und sich für Lichtpunkte entschieden. Dabei entstand ein Skript, das die Leuchten blinken lässt. Bei der Umsetzung mit dem Beispielbild von Dix wurde darauf aber verzichtet, da die Ablenkung des Nutzers vermutlich zu hoch ist.

Um die Panels bzw. darin liegenden Textfelder gemäldeabhängig mit den Informationen aus den XML-Dateien zu füllen, entstanden verschieden Skripte, die die Sprachauswahl, das Target und die Hotspots bei der Anzeige der Texte berücksichtigen.

Im letzten Schritt in der Szene „museumsfuehrer“ entstand ein Skript zur Steuerung der Vorlese-Funktion. Um die Sprachausgabe über eine TTS zu ermöglichen, wurde im Asset Store das Text-to-Speech-Plug-In für Android heruntergeladen.

Um zu vermeiden, dass der Nutzer ahnungslos die Anwendung öffnet und sich überfordert und verloren fühlt, wurde zusätzlich zur „museumsfuehrer“-Szene eine „Startseite“ erstellt. Sie heißt den Nutzer willkommen und liefert Informationen über die Entstehung der Anwendung sowie eine kleine Anleitung (siehe Abbildung 5.3).

Target
=
Marker

5.2 Software-Architektur

5.2.1 Überblick über verwendete Softwares und Plug-Ins

5.2.1.1 Unity von Unity Technologies

Unity ist ursprünglich eine Multiplattform-Spiel-Engine für Windows- und Mac-Systeme, wird jedoch häufig auch für andere Zwecke als Spiele verwendet. (vgl. Seifert 2015: 1) Unity ermöglicht nach dem Erstellen eines Projektes die Veröffentlichung für mehrere Plattformen wie iOS, Android, Windows etc.

Unity wurde in der 32-Bit-Version installiert (Software-Version 5.6).

5.2.1.2 Vuforia von PTC, Inc.

Vuforia ist ein Software Development Kit (SDK) für die Entwicklung von AR-Anwendungen. Es ermöglicht das optische Tracken von Objekten und Bildern.

Bei Vuforia handelt es sich um ein Plug-In für Unity. Das *.unitypackage wurde importiert; dadurch wurde z. B. das Skript „DefaultTrackableEventHandler“ oder das Prefab „ImageTarget“ hinzugefügt, welches zur Erkennung der Marker nötig ist. (siehe Abschnitt 0). Target ist eine Bezeichnung von Vuforia und wird in dieser Arbeit synonym zu Marker verwendet.

Vuforia SDK wurde in der Version 6.2.10 installiert. Da sich die erstellte App in der Entwicklung befindet und nicht veröffentlicht wird, ist die Software kostenlos.

5.2.1.3 TTS Android Plug-In von Takohi Games

Bei TTS Android handelt es sich ebenfalls um ein Plug-In, das im Unity Asset Store erworben wurde. Es ermöglicht die Text-to-Speech-Ausgabe auf Android-Geräten in den Sprachen Chinesisch, Englisch, Französisch, Deutsch, Italienisch, Japanisch und Koreanisch. Das Plug-In wurde über den Asset Store heruntergeladen und darüber importiert.

5.2.1.4 Android Studio von Google Inc.

Android Studio ist ein SDK, das für das Ausspielen der App auf dem Computer installiert sein muss. Beim Ausspielen der App in Unity (über *File > Build*) muss der Programmpfad des Android Studio SDK angegeben werden. Das Tool selbst muss für das Erstellen einer App mit Unity nicht geöffnet werden.

Android Studio wurde in der Version 2.3.1 installiert.

5 Umsetzung und Programmierung

5.2.1.5 Java SE Development Kit (JDK) von Oracle Corp.

Das Java SE Development Kit muss ebenfalls installiert sein. Auch hier wird der Programmpfad für das Ausspielen der App festgelegt. Das Tool selbst muss für das Erstellen einer App mit Unity nicht geöffnet werden.

Java SDK wurde in der Version 8 Update 131 installiert.

5.2.1.6 Unity Remote 5 von Unity Technologies

Unity Remote 5 steht in keiner direkten Beziehung zum Erstellprozess der Anwendung. Unity Remote 5 ist eine App, die auf das **Android-Gerät** installiert werden kann und dazu dient, dass Apps auf dem Tablet getestet werden können, ohne sie jedes Mal ausspielen und installieren zu müssen. Dies ist vor allen Dingen hilfreich, wenn Anwendungen entwickelt werden, die Touch-Befehle abfragen.

Die Anwendung kann über den Google Play Store heruntergeladen wie gewöhnlich auf dem Testgerät installiert werden. Im Anhang befindet sich eine kurze Anleitung zur problemfreien Nutzung von Unity Remote.

5.2.2 Zusammenhänge zwischen Software-Tools und Plug-Ins

Um die Tools nutzen zu können, muss zunächst je ein **Account für Unity und Vuforia** angelegt werden. Dann können die Programme **Unity, Android Studio SKD und JDK** heruntergeladen und installiert werden.

In Unity müssen anschließend die **Verbindungen** zwischen Unity und dem Android Studio sowie zwischen Unity und dem JDK geschaffen werden. Dies kann beim ersten Erstellen der App festgelegt oder später über *Edit > Preferences > External Tools* geändert bzw. angegeben werden.

Anschließend muss das **TTS-Plug-In** über den Asset Store von Unity heruntergeladen und importiert werden.

Beim Ausspielen der App greift Unity auf das Android Studio sowie das JDK zu. Vuforia und das TTS-Plug-In erweitern nur die Funktionen in einem Projekt in Unity. Die Plug-Ins müssen für jedes Projekt neu importiert werden.

Das Ergebnis des „Build“-Prozesses ist eine ***.apk-Datei**, die per USB-Verbindung auf das Endgerät gespeichert werden kann. Für die Installation muss der Nutzer dann auf die *.apk-Datei über sein Tablet zugreifen, sie anklicken und installieren (siehe Abbildung 5.1).

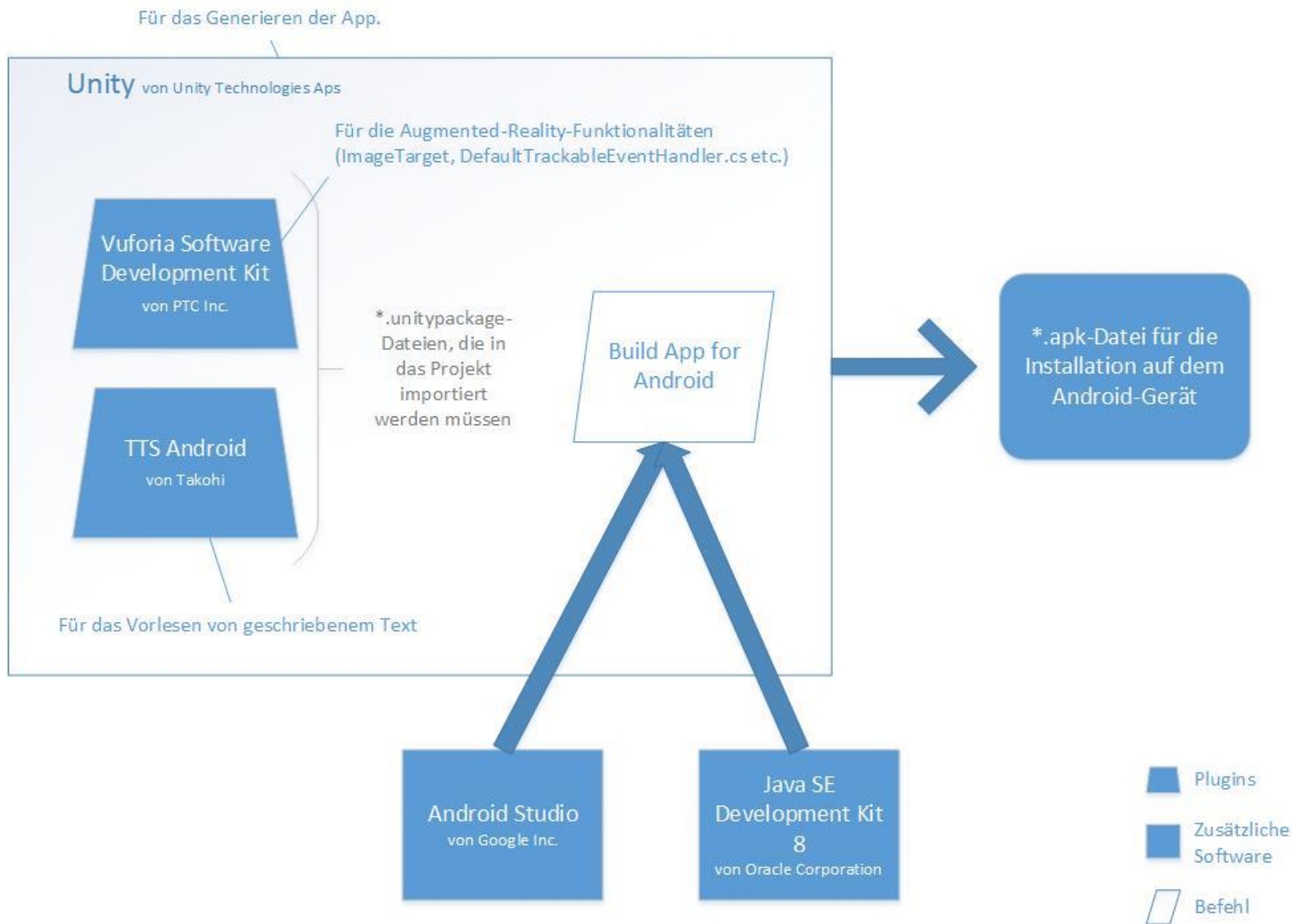


Abbildung 5.1: Verwendete Software-Tools und Plug-Ins, ihre Funktionen und Zusammenhänge

5 Umsetzung und Programmierung

5.2.3 Szenen

Die dreidimensionale Welt, die bei AR mit der realen Welt verschmilzt, wird **Szene** genannt. Sie ist ein **3D-Modell**, das neben den eigentlichen Objekten nicht nur den Blickpunkt (also die Kamera) sondern auch Belichtungs- und Audioquellen beinhaltet. Wenn der Nutzer die Szene durch sein Eingreifen beeinflussen kann, spricht man von einer **interaktiven Szene**. (vgl. Jung / Vitzthum 2013: 69).

In Unity wurde zunächst ein Projekt „MuseumsguideSKK“ angelegt. Im Projekt wurden zwei Szenen erstellt (siehe Abbildung 5.2).

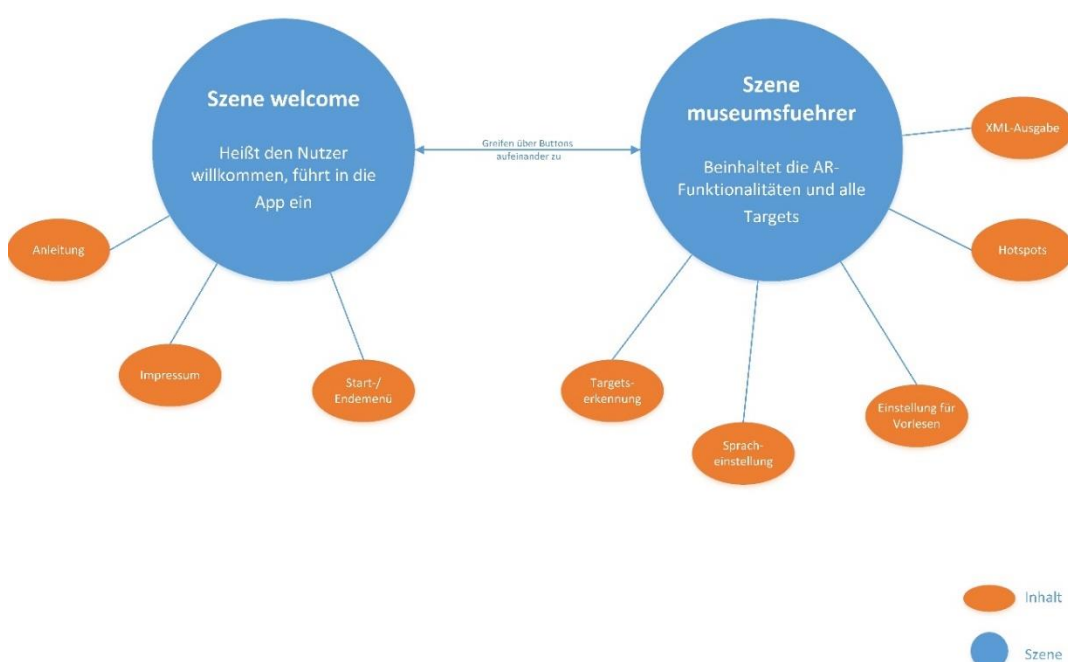


Abbildung 5.2: Szenen und ihre Inhalte

Die **Szene „welcome“** steht am Anfang und begrüßt den Nutzer, wenn er die Anwendung startet (siehe Abbildung 5.3). Über den **Button „Beginnen“** wird die **zweite Szene** geladen, über **„Beenden“** wird die Anwendung beendet, wenn der Nutzer dies nach einer erneuten Abfrage bestätigt. Über **„Anleitung“** erhält der Nutzer eine kurze Einleitung in die Anwendung.



Für Smartphone- und Tablet-Apps ist wie bei Webseiten ein Impressum vorgeschrieben, da sie unter das **Telemediengesetz (TMG)** fallen. Obwohl die App kein „geschäftsmäßige[s], in der Regel gegen Entgelt angebotene[s] Telemedi[um]“ (TMG (i.d.F. v. 21.07.2016) §5(1)) ist, wurde dennoch ein Impressum angefügt, um Abmahnungen von vornherein zu verhindern. Damit das Impressum gemäß der Faustformel „nach 2 Klicks“ erreichbar ist (vgl. Knop 2016) kann es direkt über den Button **„Impressum“** aufgerufen werden (siehe Abbildung 5.3).



Abbildung 5.3: Startseite der App

Die Szene „museumsfuehrer“ vereint die eigentlichen Funktionen der App. Die Kamera erkennt die Targets, die in der Szene hinterlegt sind, der Nutzer kann Hotspots auswählen und die Ausgabesprache ändern.

Im linken oberen Bereich findet der Nutzer den „Zurück-zur-Startseite“-Button sowie den für die Einstellungen (siehe Abbildung 5.4). Hier kann er die Sprachen wechseln und Einstellen, ob er die Vorlese-Funktion einschalten möchte oder nicht. Daneben erscheint nach dem Tracken des Bildes ein Panel, in dem der Name des Gemäldes und das Schaffensjahr steht. Im Panel am rechten Bildschirmrand werden ein Portrait des Künstlers, sein Name und seine Lebensdaten sowie das Material und die Abmessungen des Gemäldes eingeblendet. Die Hotspotinformationen werden, nachdem der Nutzer einen Hotspot berührt hat, in den Abschnitt darunter geschrieben (die Überschriften in Rot über einen scrollbaren Textcontainer für den Haupttext). Wenn kein Bild getrackt ist, werden die beiden Panels nicht eingeblendet.

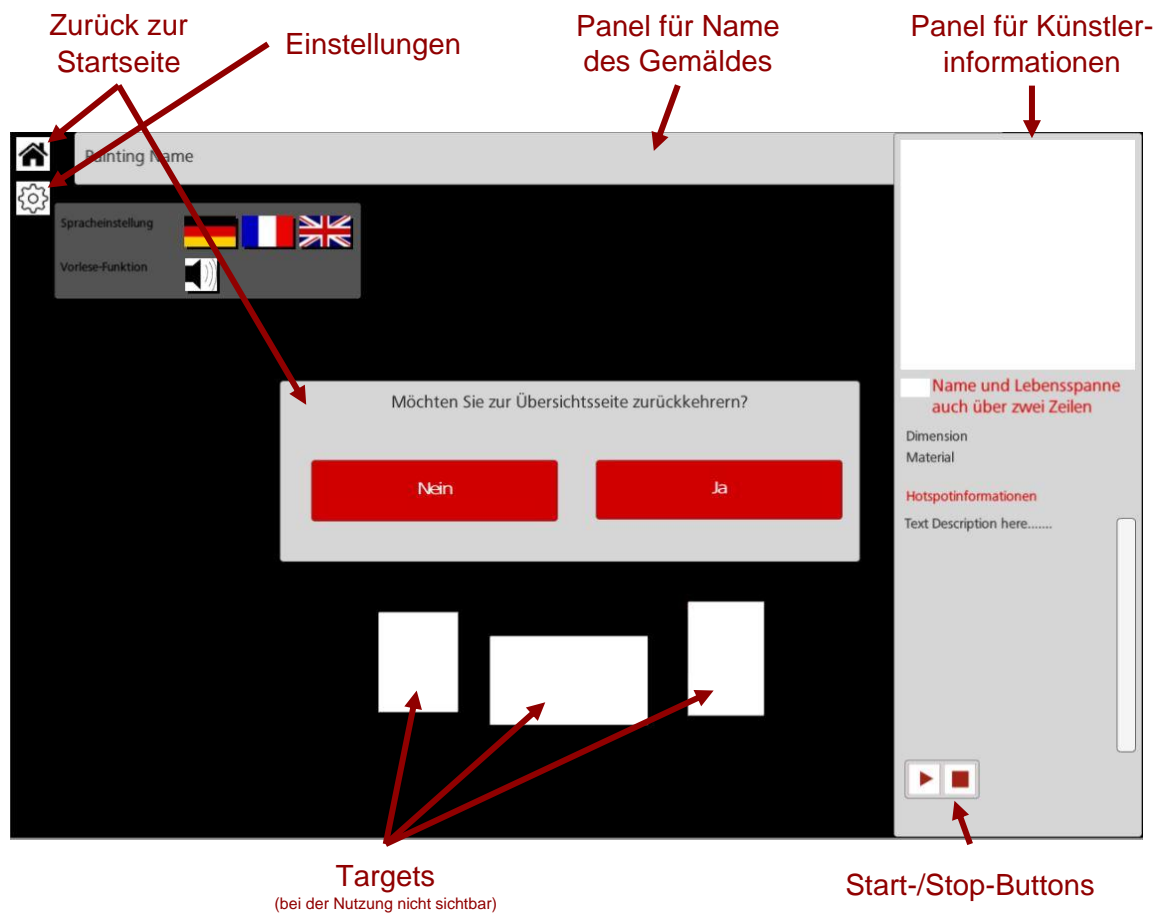


Abbildung 5.4: Schematischer Aufbau der Benutzeroberfläche

5.2.4 Skripte und ihre Verbindung untereinander

Die Skripte, die die Funktionalitäten der einzelnen Objekte regeln, wurden mit **C#** geschrieben. **C#** ist eine objektorientierte Programmiersprache. Sie wurde von Microsoft entwickelt, ursprünglich für das .NET Framework. Unity nutzt eine Open-Source-Variante des .NET Frameworks, sodass auch Nicht-Microsoft-Systeme zum Programmieren genutzt werden können. (vgl. Seifert 2015: 35)

In der Szene „welcome“ kommt nur das Skript „menu“ zum Einsatz. In der Szene „museumsfuehrer“ werden mehrere Skripte verwendet. Die Skripte müssen zum Teil aufeinander zugreifen, um Variablen abzufragen oder zu ändern⁸. In Abbildung 5.5 ist zu sehen, welche Skripte in der Szene „museumsfuehrer“ verwendet werden, an welches GameObject⁹ die Skripte angehängt sind, welche Grundfunktion das Skript hat und welches Skript auf welche Variable eines anderen Skripts zugreift. Die Skripte „DefaultTrackableEventHandler“ und „lightIntensity“¹⁰ haben keine Verbindung mit anderen Skripten.

⁸ Die Skripte „flashingLights“, „hotspot“, „hotspotContainer“, „hotspotLoader“ und „SettingXMLLoad“ im Projektordner von Unity sind nur zu Testzwecken. Die Skripte sind in der Endfassung keinem Objekt zugeordnet und haben keine Funktion mehr. Deshalb befinden sie sich nicht im Anhang.

⁹ „Basis“-Objekte in Unity

¹⁰ Das Skript „lightIntensity“ wurde nur beispielhaft beim Target „ImageTarget-Duerer“ verwendet, um die Funktion zu demonstrieren. Das dauerhafte Leuchten könnte den Nutzer zu stark irritieren.

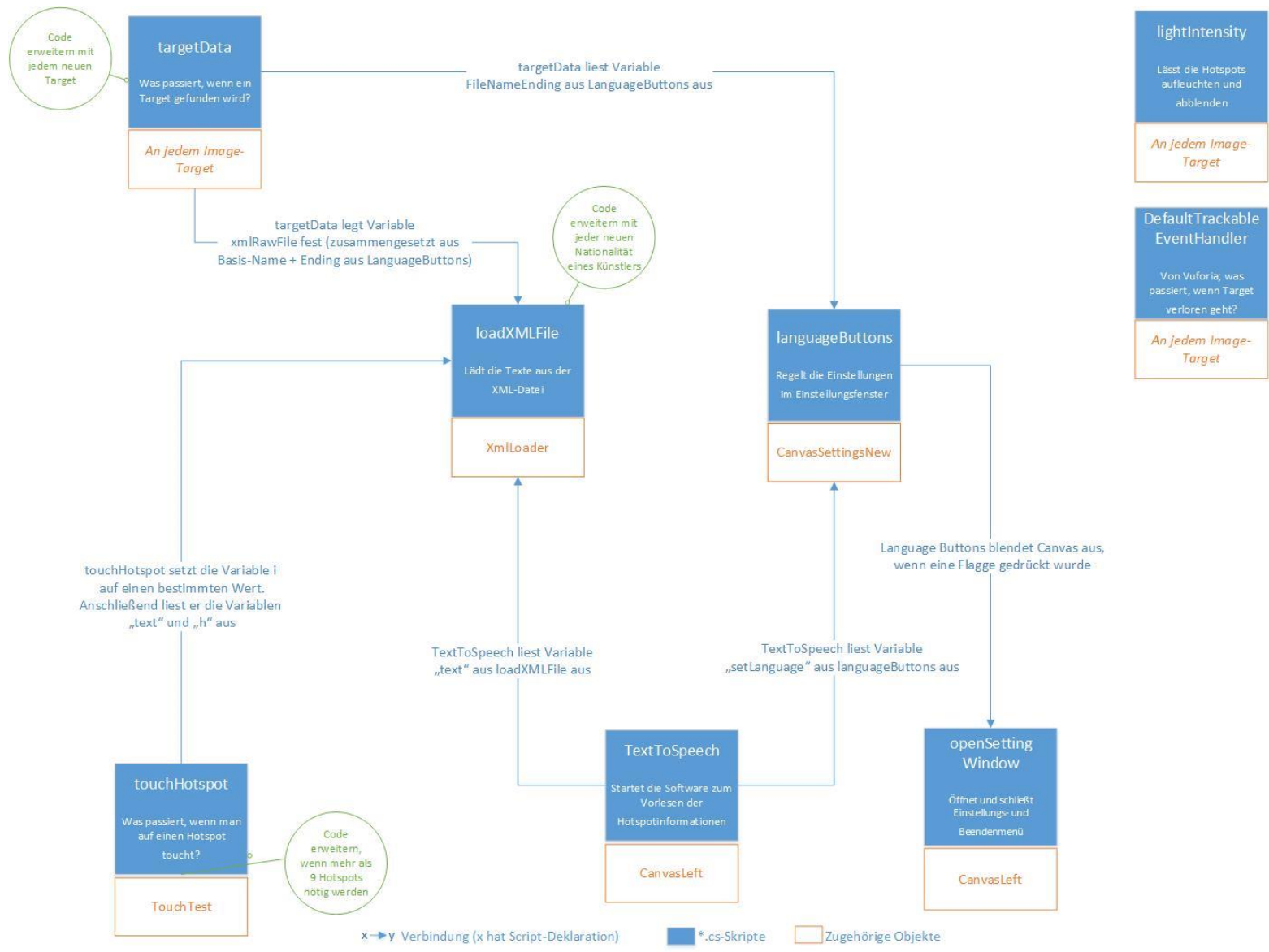


Abbildung 5.5: Skripte in der Szene „museumsfuehrer“ und ihre Beziehung zueinander

Der Zugriff auf andere Skripte und ihre Variablen kann nur erfolgen, wenn die Variablen als „public“ deklariert sind. Im gesamten Projekt wurde nicht zwischen „private“- und „public“-Variablen unterschieden. Die Verbindung zwischen den Skripten wird folgend am Skript „touchHotspot“ erklärt:

Nach dem Festlegen der Klasse (`public class touchHotspot : MonoBehaviour {`; diese Zeile wird automatisch eingefügt, wenn ein neues Skript erstellt wird) müssen zwei Variablen definiert werden. Die erste beschreibt das GameObject, das das Skript enthält, auf das zugegriffen werden soll. Sie ist vom Typ GameObject. Die zweite ist vom Typ mit dem Namen des Skriptes, auf das zugegriffen werden soll.

Im Beispiel wird auf das Skript „loadXMLFile“ zugegriffen. Dieses Skript ist an das GameObject „XMLLoader“ angehängt. Also wird zunächst das GameObject definiert, anschließend das Skript. Im Skript „touchHotspot“ wird für das Skript loadXMLFile der Variablenname „getHotspotInformation“ verwendet:

```
public GameObject XMLLoader;
public loadXMLFile getHotspotInformation;
```

Damit Unity die Verbindung zwischen der Variablen „XMLLoader“ und dem Skript „loadXMLFile“ herstellt, muss über die Unity-Oberfläche dem Skript „touchHotspot“ (welches am GameObject „TouchTest“ hängt) das GameObject zugewiesen werden, an dem das Skript „loadXMLFile“ hängt – in diesem Fall das GameObject XmlLoader (siehe Abbildung 5.6).

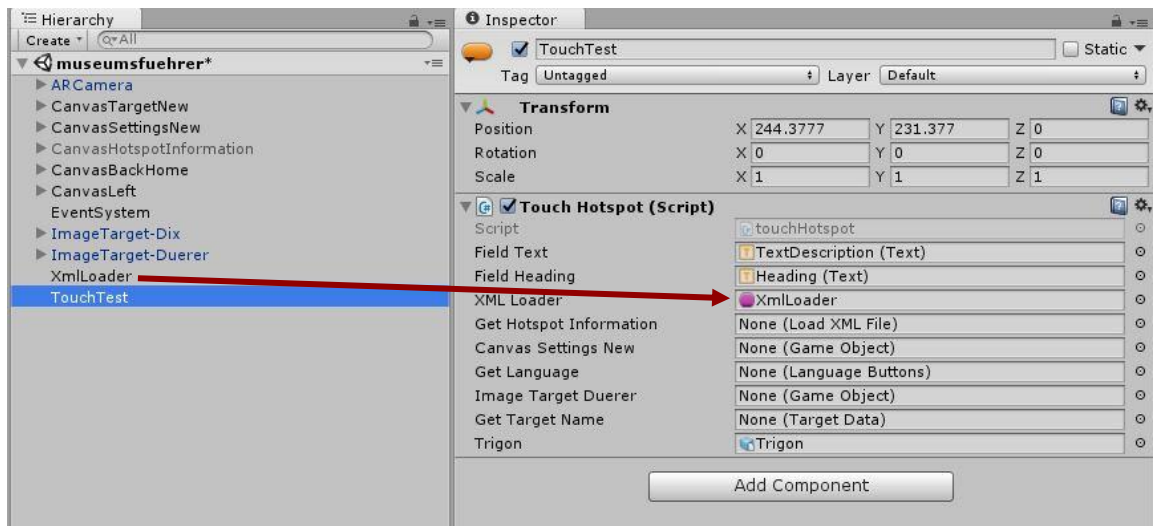


Abbildung 5.6: Das Skript „touchHotspot“ ist dem GameObject "TouchTest" zugewiesen. „touchHotspot“ greift auf das Skript "loadXMLFile" zu, das am GameObject "XMLLoader" hängt.

In der Start-Funktion des Skripts „TouchHotspot“ muss nun noch festgelegt werden, dass die Komponente „loadXMLFile“ des GameObject „XMLLoader“ mit dem Skript gemeint ist. Über die Funktion `gameObject.GetComponent<>()` wird diese Verbindung festgelegt.

```
getHotspotInformation = XMLLoader.GetComponent<loadXMLFile>();  
(Variable vom Typ Skriptname = GameObject mit Zugriffskript.GetComponent<Skriptname>());
```

Der Zugriff auf die Variablen des Skripts „loadXMLFile“ wird im „touchHotspot“-Skript folgend über die Variable „getHotspotInformation“ erfolgen. Beispielsweise wird die Variable `i` des Skripts „loadXMLFile“ über den Code `getHotspotInformation.i = 2;` auf 2 gesetzt.

5.2.5 Objekthierarchie in Unity

In der Szene „museumsfuehrer“ sind in der Objekthierarchie folgende Objekte angelegt:

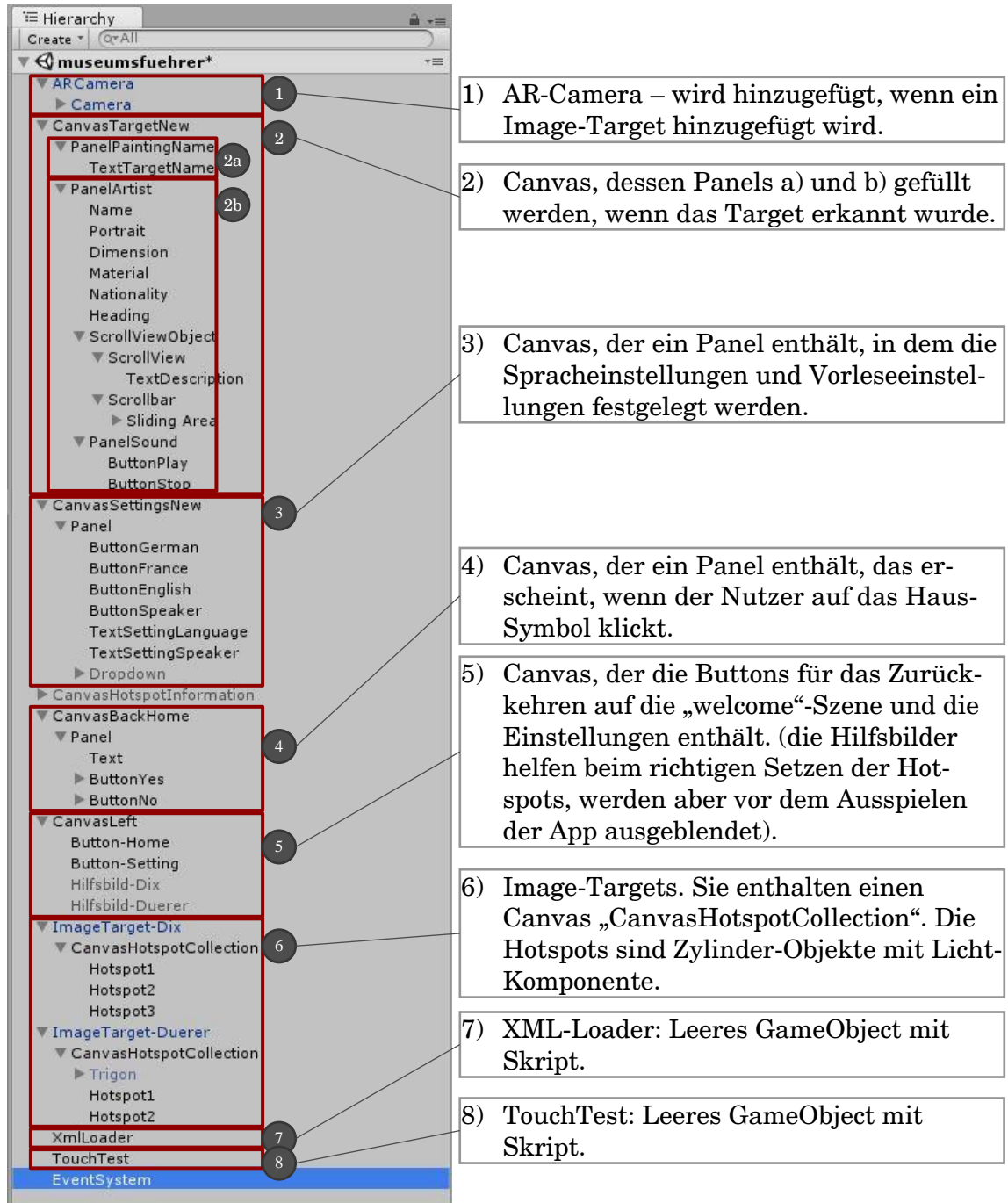


Abbildung 5.7: Objekthierarchie in der Szene „museumsfuehrer“

Über die Skripte werden die verschiedenen Objekte ein- bzw. ausgeblendet, Textfelder gefüllt und XML-Dateien ausgelesen.

5.3 Erweiterung der Anwendung um Targets

Im Moment erkennt die Anwendung drei Gemälde, die Anwendung ist aber so aufgebaut, dass sie theoretisch um beliebig viele Targets erweitert werden könnte. Anhand des folgenden Kapitels soll ein Überblick gegeben werden, welche Schritte durchzuführen sind, um den jetzigen Prototyp so zu erweitern, dass weitere Gemälde erkannt werden. Anhand der schrittweisen Anleitung werden darüber hinaus weitere Zusammenhänge und Funktionen der einzelnen GameObjects und Skripts deutlich.

5.3.1 Vorbereitung

5.3.1.1 Dateien an- und ablegen

Um die Anwendung möglichst schnell mit weiteren Targets zu erweitern, sollten folgende Dateien vorbereitet werden:

- **Bild-Datei (jpeg oder png) mit max. 2 MB**
Wird als Target festgelegt.
Benennung: nachname_vorname-titel-des-bildes.jpg
Beispiel: lorrain-claude_landschaft-mit-der-anbetung-des-goldenen-kalbes.jpg
- **(Selbst-)Portrait des Künstlers (sofern vorhanden)**
Wird nach dem Erkennen des Targets in das Panel 2b (siehe Abbildung 5.7) eingefügt.
Benennung: nachname-vorname.jpg
Beispiel: lorrain-claude.jpg
- **XML-Dateien mit Hotspot-Texten und Informationen in Französisch, Englisch und Deutsch**
Dient der Textbereitstellung.
Benennung: nachname_vorname-titel-des-bildes_sprachraumkuerzel.xml
(siehe Abschnitt 4.3.3)
Beispiel: lorrain-claude_landschaft-mit-der-anbetung-des-goldenen-kalbes_de.xml
- **Flagge der Nationalität des Künstlers**
Wird neben dem Namen des Künstlers angezeigt.
Benennung: nation (in Englischer Sprache).jpg
Beispiel: france.jpg

Diese Dateien müssen in den Resources-Ordner in Unity gelegt werden. Die Dateien können einfach per Drag&Drop hineingezogen werden. Hierbei dürfen keine Unterordner angelegt werden, da die Dateien teilweise automatisch geladen werden und der Pfad auf den Ordner „Resources“ beschränkt ist.

5.3.1.2 Bilder in Sprite verwandeln

Die Flagge, das (Selbst-)Portrait und das Gemälde müssen in Unity in ein Sprite umgewandelt werden, da die verwendete Funktion nur mit Sprites funktioniert. Dazu müssen folgende Schritte ausgeführt werden:

- 1) Das Bild im Projekt-Panel anklicken.
- 2) Im Inspector die Einstellung „Texture Type“ von Default auf „Sprite (2D and UI)“ ändern.
- 3) Im Inspector mit „Apply“ bestätigen.

5.3.2 Target in Datenbank laden

Das Gemälde an sich dient als Marker. Zunächst muss es als Target in den Vuforia Target Manager geladen werden. Dazu muss man sich zuerst auf <https://developer.vuforia.com> mit seinen Accountdaten anmelden.

Danach kann in der Registerkarte „Develop“ der License Manager oder der Target Manger ausgewählt werden. Beim Target Manager muss eine Datenbank angelegt werden. Diese wird anschließend mit den verschiedenen Targets gefüllt.

Für den Target-Typ wird „Image“ ausgewählt, da es sich um ein optisches 2D-Target handelt. Das Bild muss hinzugefügt werden. Wichtig ist, dass der Name des Targets genauso geschrieben ist wie der Name der Datei, was üblicherweise standardmäßig voreingestellt ist. Der Wert „Width“ ist obligatorisch. Hier ist bei den bisherigen Targets die Breite des echten Targets eintragen, also die Breiten-Maße des Gemäldes. Welche Änderungen ein anderer Wert hat, konnte im Rahmen der Thesis nicht überprüft werden.

Add Target

Type:

Single Image Cuboid Cylinder 3D Object

File:

lorrain-claude_landschaft-mit-der-anbetung-des-goldenen-kall Browse...

.jpg or .png (max file 2mb)

Width:

248

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

lorrain-claude_landschaft-mit-der-anbetung-des-goldenen-kalbes

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Cancel Add


Abbildung 5.8: Target-Manager von Vuforia

Nach dem Hochladen des Bildes über den Befehl „Add“ wird es in einer Datenbank für das Projekt abgespeichert. Dort wird bewertet, wie gut es sich als Target eignet. Wenn die Bewertung über 4 Sternen liegt, kann darauf verzichtet werden, einen künstlichen Marker zu erstellen. Bei „Die 7 Todsünden“ war dies der Fall (siehe Abbildung 5.9).

Anfang der 2000er definierten Owen / Xiao / Middlin (o. J.: 1-5) Kriterien für einen guten künstlichen Marker. Dabei betrachteten sie die Form (optimalerweise quadratisch), Farbe (monochrom, um von der Farbauflösung der Kamera und Umgebungsbeleuchtung unabhängig zu sein), Positionierung (hoher Kontrast) und Identifizierung (Einzigartigkeit). Wegen des geringen Umfangs der Datenbank und dem Rahmen der Arbeit kann nicht ermittelt werden, in wie weit die Kriterien von Owen / Xiao / Middlin noch aktuell sind.


5.3 Erweiterung der Anwendung um Targets

Edit Name Remove



Type: Single Image
Status: Active
Target ID: 2cd4def217f848118221aa6ed1f3b6d
Augmentable: ★★★★★
Added: Apr 14, 2017 11:41
Modified: Apr 14, 2017 11:41

Edit Name Remove



Type: Single Image
Status: Active
Target ID: f891a48f107f417fae679c34deac25be
Augmentable: ★★★★★
Added: May 17, 2017 16:00
Modified: May 17, 2017 16:00

Abbildung 5.9: Die Merkmale und Bewertungen der Targets im Vuforia Target Manager

5.3.3 Target-Datenbank in Unity hinzufügen

Nachdem die Datenbank gefüllt wurde, muss sie heruntergeladen werden. Dabei ist als „Development Platform“ „Unity Editor“ auszuwählen. Heruntergeladen wird eine *.unitypackage-Datei mit dem Namen der Datenbank. Diese wird anschließend in Unity importiert.

5.3.4 ImageTarget-Objekt hinzufügen

Als nächstes muss das Target hinzugefügt werden. Alle Targets, also alle Gemälde, die die Anwendung erkennen soll, werden in der Szene „museumsfuehrer“ gespeichert. Jedes Gemälde hat ein eigenes Target, deren Positionierung theoretisch frei wählbar ist. Zur besseren Übersicht werden die Targets nebeneinander angeordnet.

Zunächst wird das Prefab „ImageTarget“ aus dem Project-Fenster (*Vuforia > Prefabs*) in das Hierarchy-Fenster gezogen (siehe Abbildung 5.10).

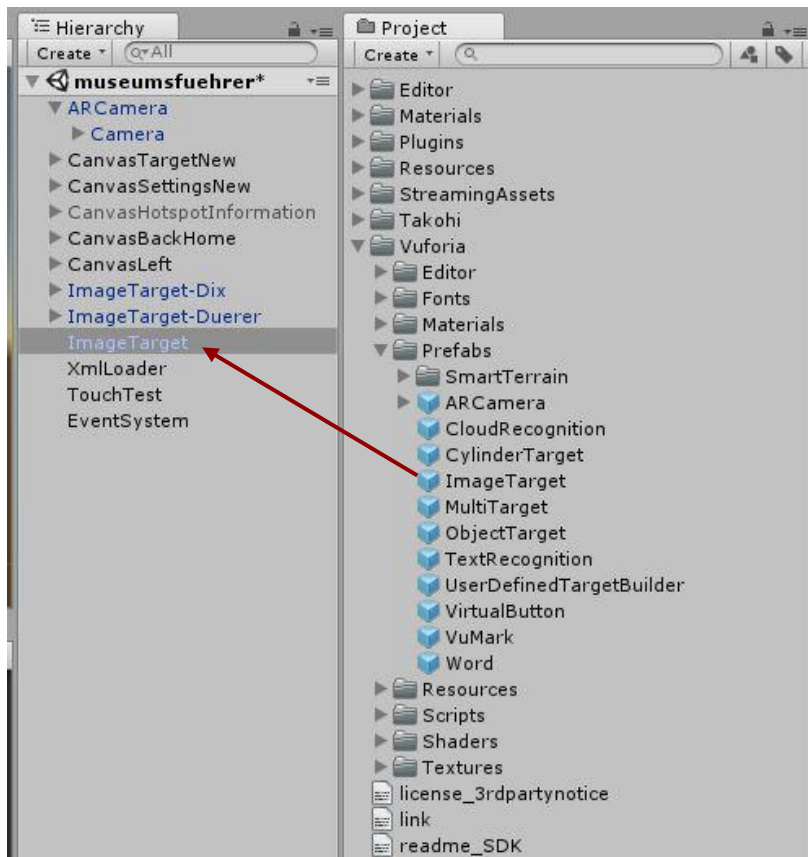


Abbildung 5.10: Einfügen des ImageTarget-Prefabs in die Objekt-Hierarchie

Anschließend ist der Target-Name um den Künstlernachnamen zu ergänzen (z. B. „ImageTarget-Lorrain“).

Damit dahinter auch Gemälde liegt, muss die Komponente „ImageTargetBehaviour“ angepasst werden (siehe Abbildung 5.11). Hierfür wird die importierte Datenbank hinterlegt und das passende Gemälde als ImageTarget ausgewählt.

Nun ist das ImageTarget so auszurichten, dass es senkrecht neben den vorhandenen ImageTargets steht. Dazu muss das ImageTarget ausgewählt werden, um im Inspector die Rotationswerte X, Y und Z der Transform-Komponente auf -90 | 0 | 0 zu

ändern (siehe Abbildung 5.11). Anschließend kann das Objekt neben die anderen Targets geschoben werden.

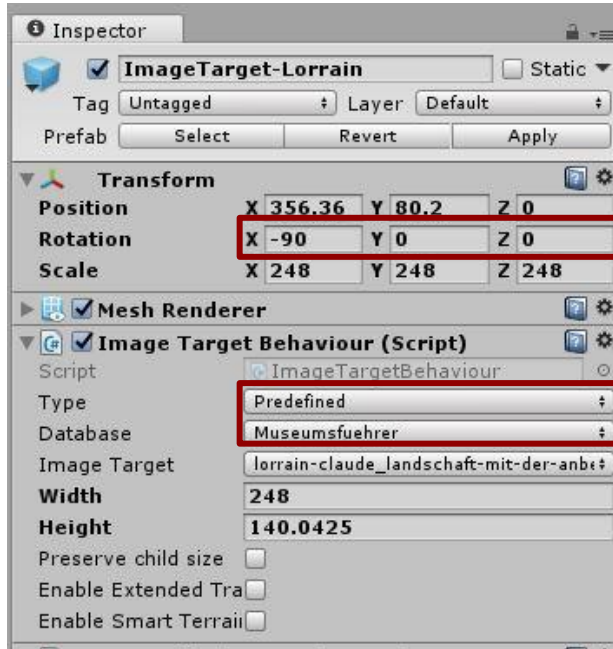


Abbildung 5.11: Änderungen in den Komponenten "ImageTargetBehaviour" und "Transform"

5.3.5 Hotspot-Collection anlegen

Die Hotspots sind alle in einem Canvas¹¹ gesammelt. Um nicht jeden Hotspot komplett neu anlegen zu müssen, bietet es sich an, eine HotspotCollection eines schon vorhanden ImageTargets zu kopieren und als Child-Objekt des neuen ImageTargets einzufügen. Wenn mehr Hotspots in der HotspotCollection benötigt werden, können diese kopiert und eingefügt werden.

Wichtig ist, dass die Hotspots konsequent durchnummeriert werden und gemäß den Benennungsregeln benannt sind. Die Hotspots müssen benannt werden mit „Hotspot1“, „Hotspot2“, „Hotspot3“ usw.¹²

Damit die Hotspots an eine sinnvolle Stelle gesetzt werden können und nicht geschätzt werden müssen, ist es ratsam, ein Hilfsbild einzubauen, welches vor dem Ausspielen der App ausgeblendet werden muss.

Dafür wird am besten im CanvasLeft per Rechtsklick über *UI > Image* ein Image-Objekt angelegt, es mit dem Target-Bild befüllt und über das Target geschoben. Sollte sich das Hilfsbild später verschieben, ist das kein Problem, da es später für die Ausgabe wieder ausgeblendet wird.

¹¹ Element in Unity; primär für User-Interface-Elemente wie Buttons

¹² siehe Skript „touchHotspot“ für die Funktion

Bei den Hotspots handelt es sich um 3D-Objekte (durchsichtige Zylinder-Formen) mit einer Light-Komponente. Die Farbe des Lichtes kann über den Inspektor eingestellt werden¹³. Um den Beleuchtungskegel anzupassen, wird mit aktiviertem Transformieren-Werkzeug der Hotspot ausgewählt und an den gelben Punkten gezogen. Um die Größe des Hotspots zu bearbeiten, muss die Position der blauen Punkte verändert werden (siehe jeweils Abbildung 5.12). Damit die Proportionen erhalten bleiben und es sich weiterhin um einen kreisförmigen Hotspot handelt, ist die Umschalt-Taste zu drücken.



Abbildung 5.12: Ändern des Hotspots in Größe und Beleuchtungskegel

5.3.6 Skript „DefaultTrackableEventHandler“ anpassen

Damit das Target erkannt werden kann, müssen den Variablen aus dem Skript „DefaultTrackableEventHandler“, das als Komponente am Prefab ImageTarget hängt, Objekten zugeordnet werden. Dies zeigt Abbildung 5.13.

In diesem Skript wird festgelegt, was geschieht, wenn Targets generell gefunden werden. Dabei werden die Panels rechts und oben ausgeblendet und die Textfelder, in die die Hotspot-Informationen oder die Gemäldeinformationen geschrieben werden, gelöscht. Die Text-Informationen werden erst über das Skript „targetData“ in die Felder geschrieben.

¹³ Bei „ImageTarget-Dix“ ist ein Hotspot demonstrativ rot gefärbt. Unterschiedliche Farben können dazu dienen, unterschiedliche Arten von Informationen zu unterscheiden. Der rote Hotspot enthält Informationen zur Entstehungsgeschichte des Bildes, die grünen über die Inhalte des Bildes. Ob und in welchem Rahmen dies sinnvoll sein könnte, muss in einem anderen Rahmen entschieden werden.

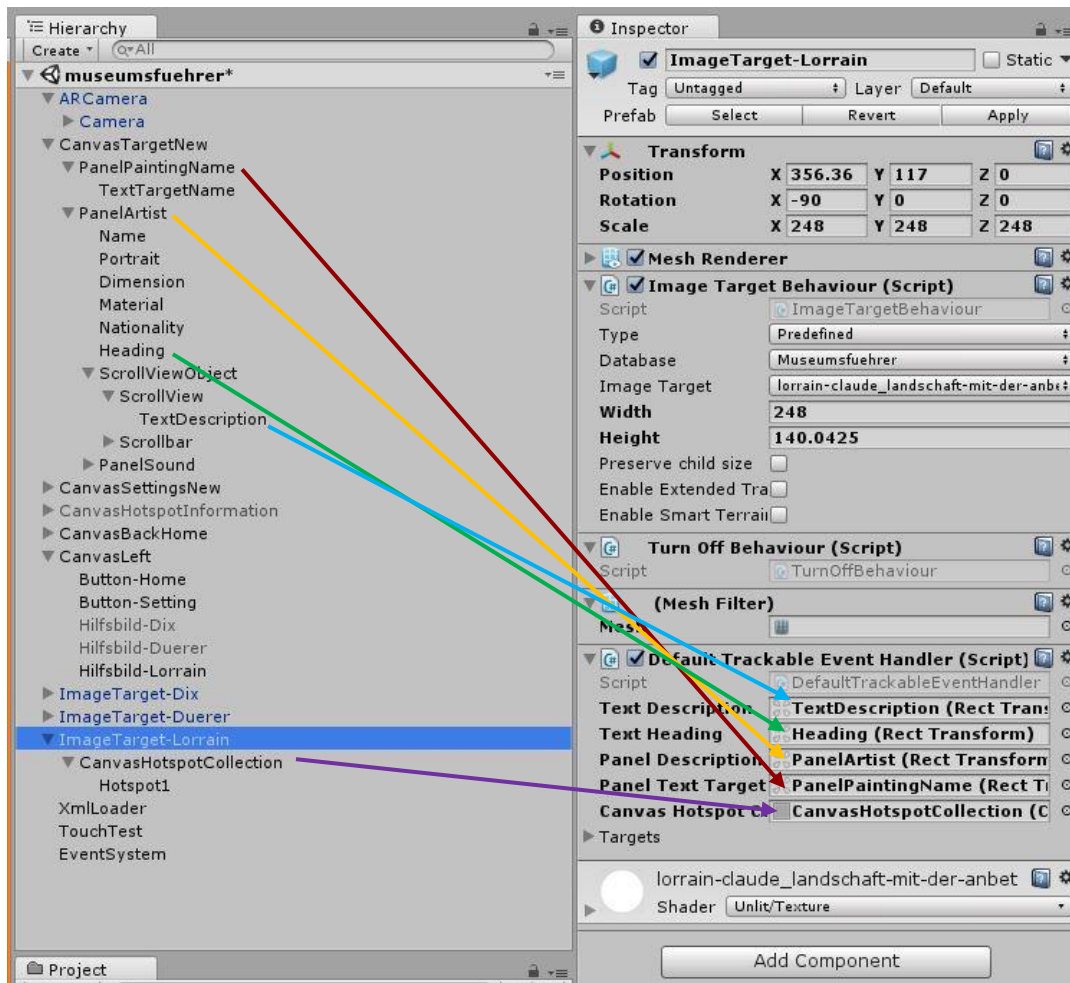


Abbildung 5.13: Zuweisung der Objekte zu DefaultTrackableEventHandler

5.3.7 Skripte und Objekte den Skripten zuweisen

Das ImageTarget braucht noch das Skript targetData. Darin wird z. B. unterschieden, was passieren soll, wenn spezifische Targets (z. B. das Gemälde von Dix oder das Gemälde von Dürer) gefunden werden. Je nach Target wird auf Basis dieses Skripts eine andere XML-Datei geladen.

Dies wird ihm hinzugefügt, in dem das Skript aus dem Project-Fenster auf das ImageTarget gezogen wird. Das Skript befindet sich im Ordner *Resources > Scripts*. Anschließend werden den Variablen des Skripts Objekte zugewiesen (siehe Abbildung 5.14); leere Felder in dieser Zuweisung werden über verschiedene Skripte beim Start der Anwendung gefüllt.

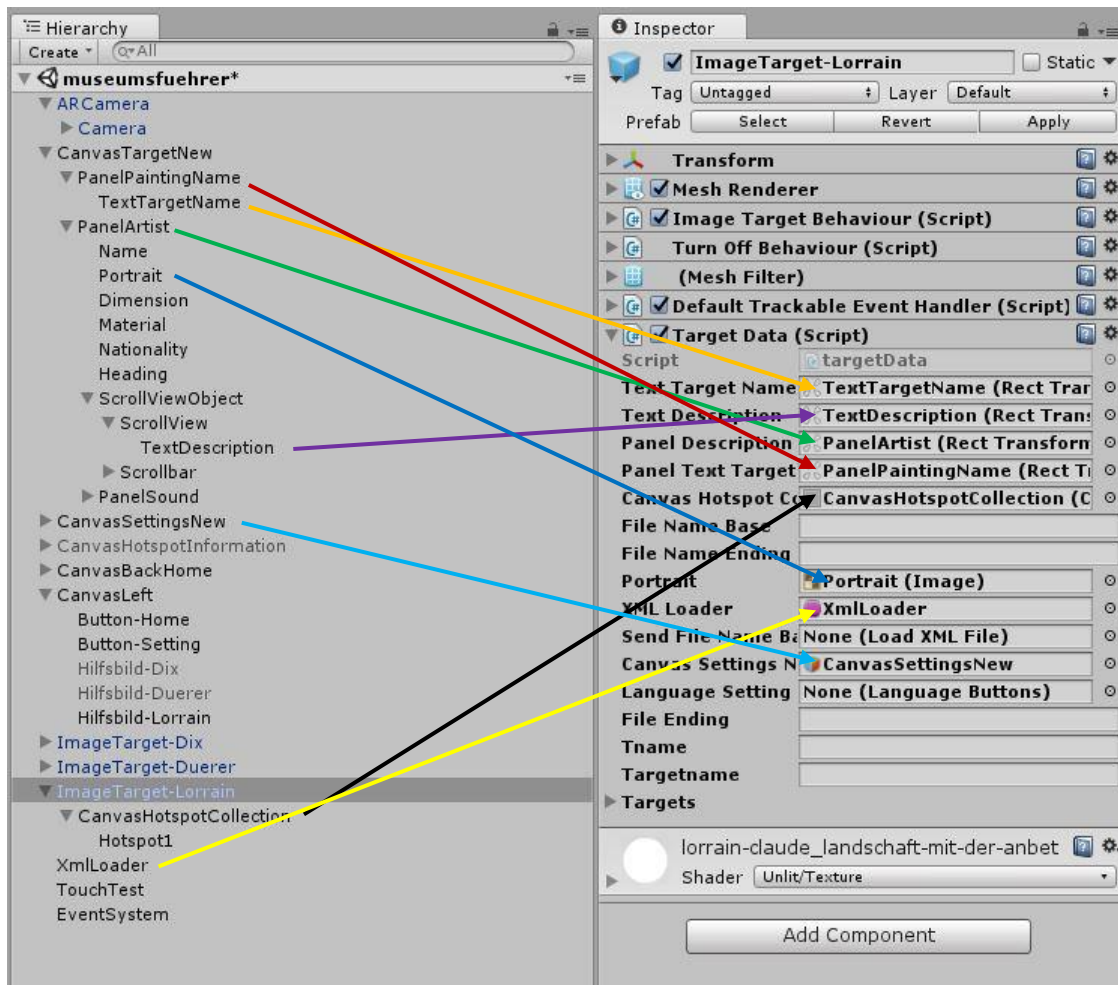


Abbildung 5.14: Zuweisung der Objekte zum Skript „targetData“

5.3.8 Skripte nachbearbeiten

Manche Skripte müssen angepasst werden, wenn ein neues Target in die Datenbank eingespeist werden:

- **targetData**
Muss mit jedem neuen Target erweitert werden.
- **loadXMLFile**
Muss erweitert werden, wenn ein Künstler eine Nationalität hat, die bisher noch nicht verwendet wurde¹⁴
- **touchHotspot**
Muss erweitert werden, wenn mehr als 9 Hotspots benötigt werden.

¹⁴ Hier ist es ratsam, eine Liste anzulegen, um den Überblick zu behalten. Bisher sind die Nationalitäten deutsch und französisch vorgekommen.

5.3.8.1 „targetData“ nachbearbeiten

Zunächst wird abgefragt, ob ein Target entdeckt wird. Es gibt für jedes Target eine eigene if-Abfrage, die nach den Namen der ImageTargets unterscheidet. Wenn ein Target erkannt wird, wird das Child-Objekt (der HotspotContainer) aktiviert. Er wird defaultmäßig von „DefaultTrackableEventHandler“ ausgeschaltet. Um den HotspotContainer anzusprechen, muss der Name des ImageTarget-Prefabs (z. B. „ImageTarget-Dix“) angegeben werden. Durch das Aktivsetzen dieser speziellen HotspotCollection ist es möglich, nur die Hotspots anzuzeigen, die zum getrackten Gemälde passen.

Der Code (ab Z. 94) sieht wie folgt aus:

```
//Wenn der Name des Targets "dix-otto_die-sieben-todsuenden" ist:
    if (tname == "dix-otto_die-sieben-todsuenden")
    {
        //Mache den Hotspotcontainer aktiv (erstes Kind vom Objekt mit
        dem Namen "ImageTarget-Dix")
        GameObject.Find("ImageTarget-Dix").transform.GetChild(0).trans-
        form.gameObject.SetActive(true);
```

In diesem Skript wird abhängig vom Target der passende Name der XML-Datei zusammengestellt. Er besteht aus einer Basis (z. B. dix-otto_die-sieben-todsuenden) und dem Sprachkürzel, das aus dem Skript „languageButtons“ ausgelesen wird. Die Verbindung zum Skript „languageButtons“ wurde über die Variable „languageSettings“ hergestellt. Der Basis-Dateiname muss deshalb mit dem Namen des Targets aus der Vuforia-Datenbank (steht hinter der Variablen „tname“) übereinstimmen. Nachdem der Dateiname zusammengesetzt wurde, kann die Datei geladen werden. Sie wird an das Skript „loadXMLFile“ übergeben (mit der Variable „sendFileNameBase“), in dem die XML-Datei dann ausgelesen wird und die Inhalte der XML-Knoten in weitere Variablen geschrieben wird.

```
fileEnding = languageSetting.fileNameEnding;
string fileName = tname + fileEnding;

//importiert Text-Datei (xml)
TextAsset Asset = (TextAsset)Resources.Load(fileName);
sendFileNameBase.xmlRawFile = Asset;
```

Außerdem wird in diesem Abschnitt das Portrait-Bild festgelegt. Es wird eine Datei aus dem Ordner Resources mit dem Namen [nachnameKünstler_vornameKünstler] geladen und der Variablen „portrait“ zugewiesen. „portrait“ ist vom Typ Image. Image-Objekte können nur mit Sprites gefüllt werden, weshalb es wichtig ist, die hochgeladenen Bilder in Sprites umzuwandeln (siehe Abschnitt 5.3.1.2).

```
//Portrait-Bild mit Namen "dix-otto" wird geladen.
portrait.sprite = Resources.Load<Sprite>("dix-otto");
```

Der Skriptteil, der bei einem neuen Target ergänzt werden muss, könnte z. B. so aussehen:

```
if (tname == " lorrain-claude_landschaft-mit-der-anbetung-des-gol-
denen-kalbes")
{
    GameObject.Find("ImageTarget-Lorraine").trans-
form.GetChild(0).transform.gameObject.SetActive(true);
    fileEnding = languageSetting.fileNameEnding;
    string fileName = tname + fileEnding;

    TextAsset Asset = (TextAsset)Resources.Load(fileName);
    sendFileNameBase.xmlRawFile = Asset;

    Debug.Log("File Name aus Target Data: " + fileName);

    portrait.sprite = Resources.Load<Sprite>("lorrain-claude");
}
```

5.3.8.2 „loadXMLFile“ nachbearbeiten

Damit das Skript einfach nachbearbeitet werden kann, muss das Bild der Nationalflagge im Ordner Resources liegen und entsprechend benannt sein (siehe Abschnitt 5.3.1.1).

Die Nationalität wird in der XML-Datei über das Attribut „nationality“ im Knoten „artist“ angegeben. Die XML-Datei wird im Skript „loadXMLFile“ ausgelesen. Wenn „de“ angegeben ist, soll die deutsche Flagge neben dem Namen des Künstlers stehen, wenn „fr“ angegeben ist, die französische usw.

Der Code, um die Nationalität abzufragen, sieht wie folgt aus (ab Z. 93):

```
//Pfad, den ich ansprechen will.
string xmlPathArtistRoot = "/work";

//wähle alle Knoten aus dem Pfad xmlPathArtistRoot aus und
mach eine Liste draus
XmlNodeList artistRootInformation = xmlDoc.SelectNodes(xmlPa-
thArtistRoot);

//For-Each-Schleife für jedes Kind von "work"
foreach (XmlNode arnode in artistRootInformation)
{
    //lies die Nationalität aus
    nationality = arnode["artist"].GetAttribute("nationa-
lity");
}
```

Nun muss noch abgefragt werden, welchen Wert das Attribut „nationality“ hat. Dies sieht beispielsweise so aus:

```
//Wenn der Wert des Attributs in der XML "de" ist,
if (nationality == "de")
{
//lade in das Image-Object die Flagge mit dem Namen "germany"
    imageNationality.sprite = Resources.Load<Sprite>("germany");
}

```

Mit jeder neuen Datei muss der obenstehende Code kopiert und angepasst werden (z. B. bei einem österreichischen Künstler):

```
if (nationality == "at")
{
    imageNationality.sprite = Resources.Load<Sprite>("austria")
};

```

5.3.8.3 „touchHotspot“ nachbearbeiten

Dieses Skript ermittelt den Hotspot, der geklickt bzw. berührt wurde. Je nachdem, welchen Namen der Hotspot hat („Hotspot1“, „Hotspot2“ etc.) wird die Variable `i` im Skript „loadXMLFile“ verändert. Diese Variable `i` dient dazu, den Hotspot aus der XML-Datei abzurufen, dessen Attribut „id“ mit dem aktuellen Wert von `i` übereinstimmt. In „loadXMLFile“ werden die entsprechenden Texte mit dem Attributwert `id=i` aus der XML in die Variablen „text“ und „h“ geschrieben. Diese werden dann in die Textfelder geschrieben, die dafür vorgesehen sind.

Der Skript-Code (Z. 74) sieht wie folgt aus:

```
//Wenn ein Objekt getroffen wurde mit Namen "Hotspot1"...
    if (hit.collider.gameObject.name == "Hotspot1")
    {
        //setze die Variable i in LoadXMLFile auf 1
        getHotspotInformation.i = 1;
        //Debug.Log("Name: " + hit.transform.name + "\n Tag: " +
hit.transform.tag);

        //Informationen aus LoadXMLFile werden in die Textfelder
geschrieben.
        fieldText.text = getHotspotInformation.text;
        fieldHeading.text = getHotspotInformation.h;
    }

```

An manchen Stellen kann es sinnvoll sein, beim Touch auf einen Hotspot zusätzlich zum Text ein anderes Objekt (in Fall von Dürers Melencolia ein 3D-Objekt) einzublenden. Dies kann z. B. durch folgende Codeerweiterung innerhalb der `if`-Abfrage geschehen:

```
//Wenn das "Großeltern"-Element des Hotspot1 "ImageTar-
get-Duerer" heißt....
if (hit.collider.transform.parent.gameObject.trans-
form.parent.gameObject.name == "ImageTarget-Duerer")
{
    //mache das Objekt aktiv
    Trigon.gameObject.SetActive(true);
}
```

Wenn mehr als 9 Hotspots benötigt werden, muss der Code erweitert werden. Dazu ist die if-Abfrage zu kopieren, der Name des Hotspots zu ändern und i auf die passende id-Nummer zu setzen. Der Code könnte dann z. B. so aussehen:

```
if (hit.collider.gameObject.name == "Hotspot10")
{
    getHotspotInformation.i = 10;
    fieldText.text = getHotspotInformation.text;
    fieldHeading.text = getHotspotInformation.h;
}
```

5.3.9 Build-Settings einrichten

Vor dem Auspielen der App müssen einige Einstellungen vorgenommen werden. Über *File > Build Settings...* muss zunächst festgelegt werden, dass eine Android-App ausgegeben werden soll (dies muss bei einem Wechsel über den Button „Switch Platform“ bestätigt werden). Außerdem ist festzulegen, welche Szenen ausgegeben werden sollen.

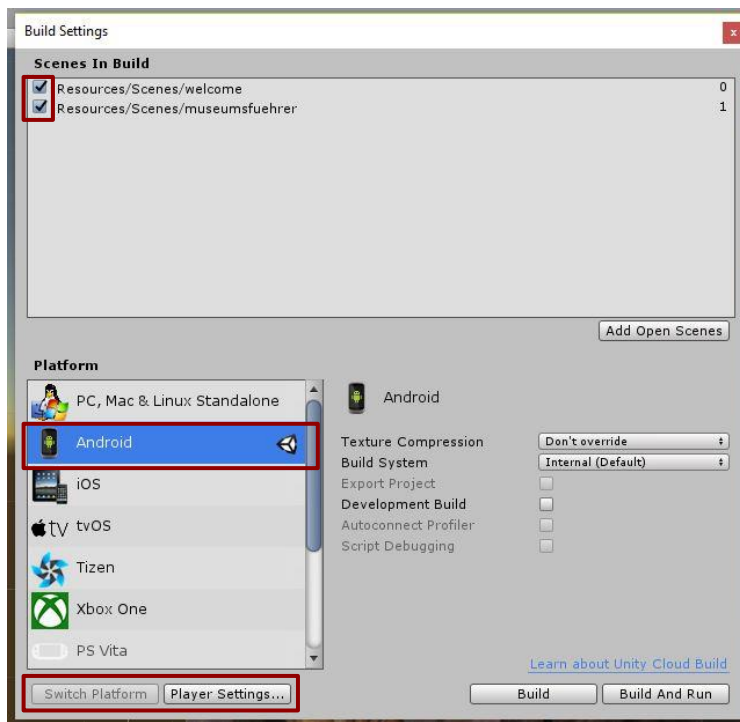


Abbildung 5.15: Build-Settings

Über den Button „Player Settings“ können bzw. müssen weitere Einstellungen geändert werden (siehe Abbildung 5.16):

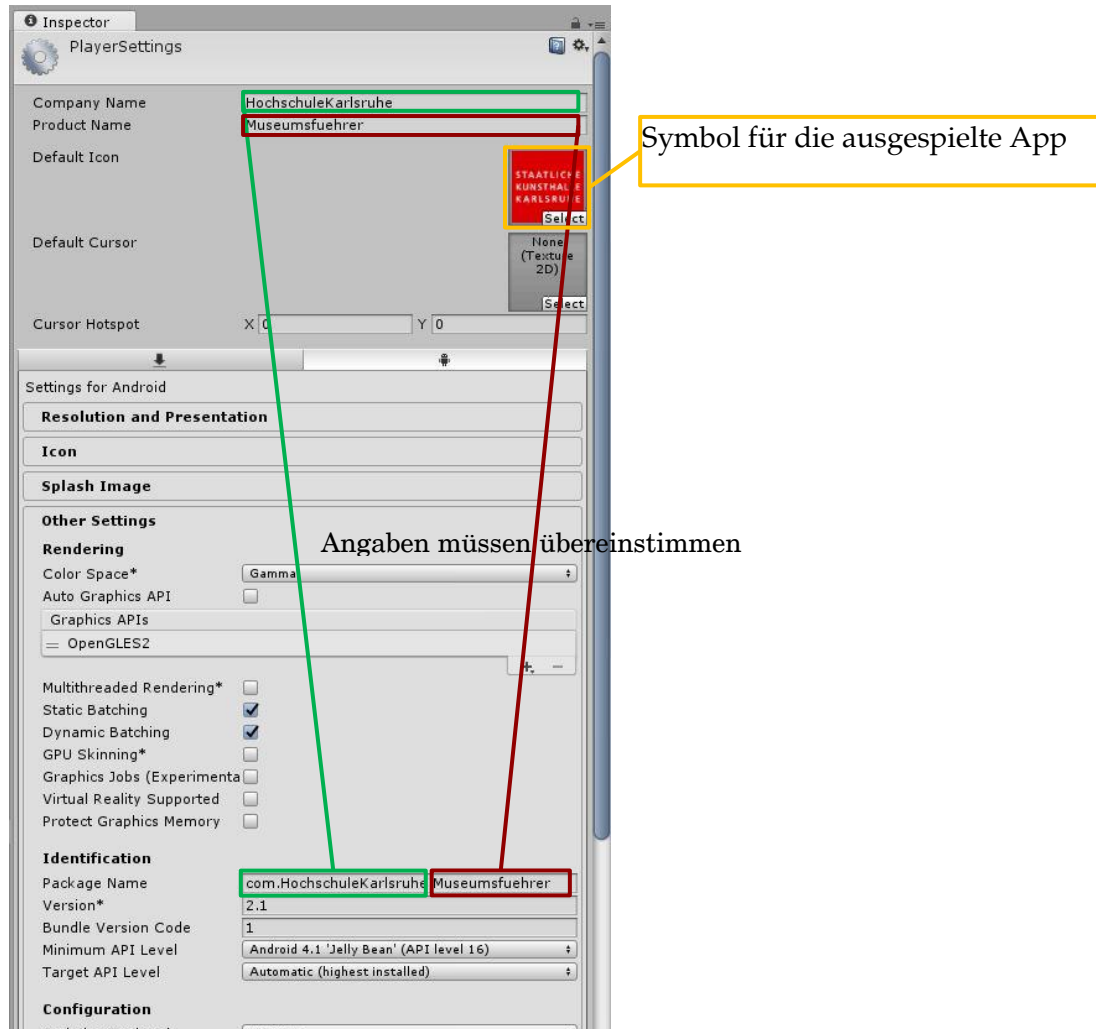


Abbildung 5.16: Änderungen bzw. Anpassungen in "Player Settings"

Vor der Veröffentlichung muss der Lizenzschlüssel aus Vuforia angegeben werden. Dieser wird im Vuforia Development-Portal unter „Licence Manager“ erstellt. Um den Lizenzschlüssel anzugeben, ist in der Objekthierarchie das Objekt „AR Camera“ auszuwählen und in der Komponente „Vuforia Behaviour“ auf den Button „Open Vuforia configuration“ zu drücken. Es ist wichtig, zu prüfen, ob folgende Angaben gemacht sind:

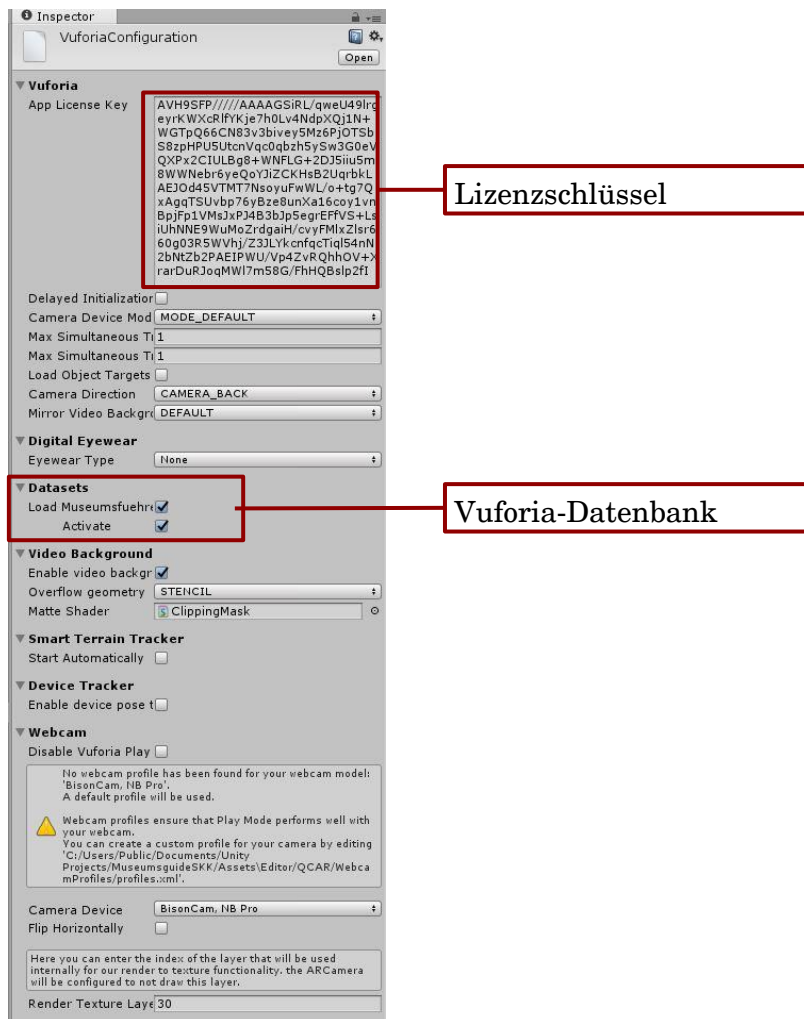


Abbildung 5.17: Vuforia Configuration

5.3.10 App bilden und installieren

Über den Button „Build“ wird eine *.apk-Datei erstellt. Nach Ändern der Einstellungen kann ein Speicherort für die Datei angegeben werden.

Um die App auf dem Gerät zu installieren, muss die *.apk-Datei mittels USB auf das Tablet kopiert werden. Durch Klick auf die Datei im Tablet-Explorer kann die Datei dann installiert werden.

6 Evaluation

6.1 Experten-Evaluation

Die Anwendung wurde von einer Studentin des Bachelorstudiengangs „Kunst – Medien – Ästhetische Bildung“ an der Universität Bremen getestet. Der Test wurde mit verkleinerten Darstellungen (A4-Ausdrucke für „Melencolia I“ und „Landschaft mit der Anbetung des goldenen Kalbes“ bzw. Poster 40 cm x 60 cm der „7 Todsünden“) durchgeführt.

Beim Starten der App erscheint zunächst die „welcome“-Szene. Bei Klick auf „Beginnen“ ist die Ladezeit etwas hoch. Es ist kann leider nicht zu erkennen, dass die App Inhalte lädt. Der Button färbt sich zwar leicht hellrot, dies ist aber kaum zu erkennen.

Beim Test ist aufgefallen, dass die Gemälde nur mit guten Lichtverhältnissen getrackt werden können. Tageslicht ist optimal, aber auch helle künstliche Beleuchtung reicht aus. Ab welcher Lichtstärke Problemen auftauchen, ist nicht bekannt.

Das Konzept gefiel ihr sehr gut, genauso wie die Umsetzung und das Handling mit dem Tablet. Sie könnte sich vorstellen, dass Tablet-Anwendungen dieser Art mittelfristig die bisher üblichen Audioguides ersetzen könnten, da solche Anwendungen den großen Vorteil bieten, Informationen sowohl auditiv als auch visuell darstellen zu können. Sie konnte sich vorstellen, die App hinsichtlich der Audiobeiträge so zu erweitern, dass zunächst ein Beitrag startet, der das Gemälde grob in einen Kontext einbettet und anschließend weitere Informationen über die Hotspots liefert.

Bei allen drei Beispielgemälden handelt es sich um ältere Werke, die ein recht hohes Maß an Symboliken aufweisen. Die Expertin schätzt daher das Potenzial der App vor allem für diese Werke als sehr hoch ein. Bei zeitgenössischer Kunst dürften sich die Inhalte etwas unterscheiden, da diese Werke häufig abstrakter sind und weniger Symboliken aufweisen. Hier bieten sich Informationen über den Künstler oder die Kunstepoche an.

Sie war überrascht, wie schnell die App das Bild erkannte, bemängelte jedoch, dass die Kameradarstellung auf dem Tablet nicht immer flüssig war. Die Länge der Texte und die Inhalte empfand sie als umfassend, weder zu lang noch zu kurz. Die Vorlesefunktion und der Sprachenwechsel funktionierten trotz der ein oder andere Aussprachefehler beim maschinellen Vorlesen gut (z. B. „tausendneunhundertdreiunddreißig“ statt „neunzehnhundertdreiunddreißig“ oder „Nsdap“ statt „N-S-D-A-P“). Die attraktivste Lösung hierfür wäre das Einsprechen in einem Tonstudio. Alternativ könnte geprüft werden, ob die Ausgabe mit einer neueren Sprachausgabe-Software oder einer neueren Betriebssystemversion besser funktioniert bzw. in welcher Hinsicht der Text geändert werden könnte, um Aussprachefehler zu vermeiden.

Eine Funktion, die sie sich gewünscht hätte, wäre, dass die Anwendung das Bild erkennt, der Hotspot ausgewählt wird und das Tablet anschließend zum Lesen der Hotspottexte heruntergenommen werden könnte; also, dass sich die Anwendung das Bild und die Hotspots „merkt“, wenn das Tablet gesenkt wird. Das ist im Moment nicht möglich, da die Panels nur solange angezeigt werden, wie die Tablet-Kamera das Gemälde trackt. Sobald das Tracking unterbrochen wird, werden die Panels ausgeblendet und die Hotspottexte gelöscht.

Leider hat sich die Anwendung während des Testes zwei Mal „aufgehängt“. Es konnte nicht herausgefunden werden, ob die Ursache beim Gerät lag oder an der Anwendung.

6.2 Test in der Staatlichen Kunsthalle

Die Kunstvermittlerinnen waren ähnlicher Meinung wie die Studierende aus Bremen. Wünschenswert wäre es, das Tablet senken zu können, um die Informationen über die Hotspots zu lesen. Dafür müsste der Programmcode nur dahingehend geändert werden, dass das Panel beim Verlieren des Targets nicht ausgeblendet und der Text nicht durch Leerzeichen ersetzt wird (Skript „DefaultTrackableEventHandler“). Sobald ein neuer Hotspot ausgewählt werden soll, müsste der Nutzer das Tablet wieder auf das Gemälde richten. Damit der Nutzer die Hotspots auch in waagrechtlicher Haltung auswählen kann, wäre es denkbar, dass der Nutzer über einen Button ein Abbild des Gemäldes auf sein Display einblenden kann. An dieser Stelle könnten die gleichen Hotspots liegen, wie wenn das Gemälde getrackt würde. So könnten die Hotspots ebenfalls aufgerufen werden, ohne dass der Nutzer das Tablet senkrecht halten muss.

Die Kunstvermittlerinnen empfanden die Sprachausgabe als eher unangenehm. Für den Fall einer Anwendung im Museum wäre eine bessere Ausgabe oder die Aufnahme im Tonstudio unausweichlich. Bei den bisher vorliegenden Audioguides des Museums wurden die Inhalte professionell gesprochen, da sich die Texte im Regelfall nicht mehr ändern. Für das Gemälde von Otto Dix standen leider keine Audiodateien zur Verfügung.

Beim Test im Museum wurde deutlich, dass die Hotspots noch kleiner hätten gewählt werden können, da das Originalgemälde von Dix sehr groß ist (179 cm x 120 cm). Bei kleineren Hotspots würde es zwar mehr geben, doch der Informationsgehalt wäre konstant. Beispielsweise hätte anstelle eines Hotspots auf dem Kopf des Geizes vielleicht einen Hotspot auf die Geldscheine in der rechten Hand und einen Hotspot auf die Kleidung positioniert und die Informationen aufgeteilt werden können.

Am Tag des Tests war es sonnig, sodass das Gemälde in der Orangerie der SKK bei den Lichtverhältnissen gut und zügig erkannt wurde. Ob es Einschränkungen bei Wolken oder nach Einbruch der Dunkelheit geben könnte, ist nicht bekannt.

7 Fazit und Ausblick

7.1 Technisches Ergebnis

Während dieser Bachelorthesis entstand die Datei „museumsfuehrer_4.0.apk“, die auf ein Android-Tablet geladen und installiert werden kann.

Die App erkennt drei Werke:

- „Die 7 Todsünden“ von Otto Dix (1933)
- „Melencolia I“ von Albrecht Dürer (1514)
- „Landschaft mit der Anbetung des goldenen Kalbes“ von Claude Lorrain (1653)

Dabei wurden alle Hotspots für das Werk „Die 7 Todsünden“ von Otto Dix (1933) angelegt und die deutsche XML-Datei mit gemäldebezogenen Informationen gefüllt. Die XML-Dateien für Französisch und Englisch wurden maschinell übersetzt und müssten vor einer Veröffentlichung zwingend überarbeitet werden.

Die anderen Gemälde enthalten beispielhaft einen Hotspot und Blindtext in den XML-Dateien. Bei Dürers „Melencolia“ ist ein 3D-Modell des Rhombusstumpf eingearbeitet, der auf dem Gemälde zu sehen ist, um darzustellen, wie AR im engeren Sinn aussehen kann.

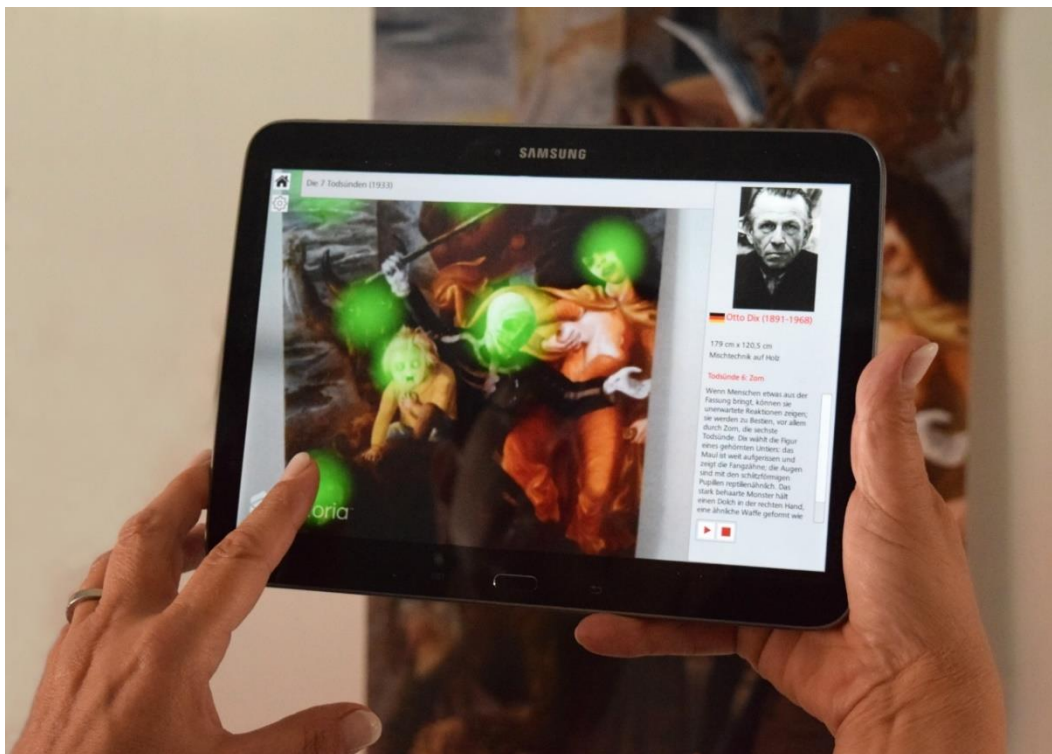


Abbildung 7.1: Die Anwendung in Benutzung

7.2 Optimierungspotenzial der bisherigen Anwendung

In der nachfolgenden Liste werden Aspekte aufgeführt, die die Anwendung sowohl hinsichtlich der Usability als auch hinsichtlich der Programmstruktur verbessern würden. Im Rahmen dieser Arbeit können sie leider nicht umgesetzt werden.

- Bisher können die Hotspottexte nur gelesen werden, wenn die Kamera auf das Gemälde gerichtet ist und dabei das Gemälde trackt. Sobald das Gemälde nicht mehr getrackt wird, werden die Panels ausgeblendet und die Hotspottexte gelöscht. Ideal wäre es, wenn die Anwendung das Bild erkennt und nach dem Touch auf den Hotspot die Textinformationen immer noch anzeigt, wenn das Target verloren geht, beispielsweise weil der Nutzer das Tablet senkt. Realisiert werden könnte dies, in dem das Skript „defaultTrackableEventHandler“ so angepasst wird, dass die Panels nicht ausgeblendet werden, wenn das Target verloren geht.
- Um zu vermeiden, dass der Nutzer das Tablet über einen längeren Zeitraum senkrecht halten muss, um Hotspots auszuwählen, könnte eine „Merken“-Funktion ergänzt werden: Bei Klick auf einen kleinen Button würde ein Bild vom Gemälde auf dem Display erscheinen. Hier wären die Hotspots genauso platziert und anwählbar, wie wenn das Gemälde mit der Kamera getrackt würde. So könnte der Nutzer das Tablet senken, um weitere Hotspots auszuwählen und die Hotspottexte zu lesen.
- Im Moment sind in der C#-Programmierung das Geheimnisprinzip und die Datenkapselung nicht eingehalten, da alle Variablen mit „public“ von überall aus zugänglich sind (vgl. Becker 2014). Die Skripte sollten hinsichtlich ihrer Objektorientierung vor allem mit den „public“- und „private“-Variablen überarbeitet werden.
- Die Audioausgabe über TTS ist zwar eine ökonomische und schnelle Lösung, um zusätzlich zum Text eine Audiospur anzubieten, allerdings werden manche Wörter falsch ausgegeben. Außerdem stößt sie bei vielen Menschen auf Ablehnung. Nach einer vollständigen Überarbeitung der Texte könnten statt der TTS-Technologie Aufnahmen von Audiodateien angefertigt werden. Alternativ könnte eine verbesserte Software für das Android-Gerät erworben werden.
- Falls eine Aufnahme von Audiodateien nicht möglich ist, müsste je nach angeschafften Tablets bzw. Distribution der Anwendung die TTS-Ausgabe für IOS- und Windowsgeräte ergänzt und dementsprechend andere Formate ausgespielt werden.
- Da die Ladezeit nach dem Klick auf „Beginnen“ auf der Welcome-Szene recht lange ist, könnte eine Lade-Anzeige hinzugefügt werden. Aktuell ist nur bei

genauem Hinsehen erkennbar, dass sich die Farbe des Beginnen-Buttons ein wenig heller färbt, während das Programm die Inhalte lädt.

- Die Ladezeit für Bilder innerhalb der App wird sich mit zunehmender Größe der Datenbank deutlich verlängern, da alle Targets in einer Szene gespeichert sind und gleichzeitig geladen werden. Um den unerwarteten Abbruch der App zu verhindern, muss herausgefunden werden, ab wann die Ladezeit einen vertretbaren Rahmen übersteigt und anschließend ein Weg gefunden werden, die Ladezeiten zu verringern.
- Die Anwendung ist responsiv gestaltet, passt sich also der Bildschirmgröße des Ausgabegerätes an. Im Moment ist die Gestaltung der Tablet-Benutzeroberfläche für das Querformat im Format 1280 x 800 Pixel optimiert. Die Darstellung funktioniert zwar im Hochformat, nimmt jedoch recht viel Platz auf dem Display ein und ist eher unübersichtlich. Das Responsive Design kann in Bezug auf die Ausrichtung und die Größe der Inhalte noch optimiert werden.
- Die Hotspot-Textanzeige ist scrollbar. Sobald das Target erkannt wurde und ein Hotspot angeklickt wurde, wird der Textanfang angezeigt. Wenn der Nutzer den Text scrollt und anschließend direkt (ohne erneutes Tracken des Gemäldes) einen weiteren Hotspot anklickt, fängt der Text nicht oben an. Es wird ungefähr die Menge an Text „übersprungen“, die am vorherigen Hotspot weitergescrollt wurde. Ein vorstellbarer Lösungsansatz dieses Problems ist es, den Programmcode so zu erweitern, dass bei jedem Touch auf einen Hotspot der Y-Wert des Pivot-Punktes des „Scroll-View“-Objektes auf 1 gesetzt wird.
- Bisher werden als Medien nur Texte, Bilder, ein 3D-Modell und durch das TTS erstellte Audiodateien dargestellt. Wünschenswert wäre die Erweiterung mit Videos und Links. Denkbar wären Internetseiten, die z. B. das Leben des Künstlers genauer beschreiben. Außerdem könnten im rechten Panel weitere Werke des Künstlers abgebildet werden, sodass der Nutzer sich ein Gesamtbild über seine Schaffensphase machen könnte. Gegebenenfalls könnte in diesem Zusammenhang angegeben werden, in welcher Galerie die erwähnten Gemälde ausgestellt sind und auf die Homepage verlinken. Der Internet-Zugang wäre über das WLAN-Netzwerk der Stadt Karlsruhe (KA-WLAN) möglich.
- Das 3D-Modell, das bisher zu demonstrativen Zwecken eingebaut wurde, bietet bisher keine Interaktion an. Hier wäre es zur Motivation der Nutzer wünschenswert, das Modell drehen und vergrößern bzw. verkleinern zu können. Bisher kann sich der Nutzer das Modell genauer ansehen, indem er um das Gemälde herumgeht.

7.3 Ansätze für weitere Arbeiten auf Basis dieser Bachelorthesis

Die Thesis bietet großes Erweiterungspotenzial, sowohl in technischer als auch in konzeptioneller Hinsicht. Einige Ansätze, in welchem Maße andere Projektarbeiten diese Arbeit weiterentwickeln könnten, werden im Folgenden genannt.

- Die App könnte auf einem Head-Mounted-Display installiert und getestet werden. Je nachdem, ob sich das HMD bezüglich der Usability für eine Anwendung in der Kunsthalle eignet, könnte die App angepasst werden.
- Vorstellbar ist auch, eine Ausstellung mit virtuellen Skulpturen zu erweitern. Der Nutzer könnte 3D-Modelle auf dem Tablet betrachten, wenn er die Tablet-Kamera auf den dafür vorgesehenen Marker richtet. Durch Klicks auf das Modell können dazu weitere Informationen angezeigt werden. Damit die Einbettung möglichst real ist, könnte der Fokus vor allem auf die photometrische und geometrische Registrierung von 3D-Modellen gelegt werden (siehe Abschnitt 2.2.3). In dieser Arbeit wurde prototypisch der Rhombusstumpf aus „Melencolia I“ von Dürer eingearbeitet, allerdings wurde hierbei auf Interaktionsmöglichkeiten und photometrische Registrierung verzichtet.
- Bisher befindet sich die App nur auf Interaktionsstufe 2 (vgl. Abschnitt 2.3.7.2). Wünschenswert wäre es, die Interaktion zwischen Mensch und Maschine zu erhöhen, sodass z. B. verstärkt 3D-Modelle eingebaut werden und diese vom Nutzer gedreht werden können.
- Eine solche App könnte nicht nur für das Tracken von Gemälden dienen. Sie könnte auch für Objekttracking erweitert werden. Das könnte gleichermaßen für andere Museen, z. B. Naturkundemuseen, interessant sein. Im Moment sind die Möglichkeiten des Objekt-Trackings durch die geringe Größe der trackbaren Objekte etwas eingeschränkt. Hier könnten Algorithmen genutzt und erprobt werden, die mit Punktwolken arbeiten.
- Da es eine gewisse Zeit benötigen würde, bis alle Gemälde einer Ausstellung in einer Anwendung hinterlegt sind, könnte auf der Benutzeroberfläche eine kleine Karte eingeblendet werden, die anzeigt, wo sich der Nutzer in der Halle befindet und welche Objekte mit dem Museumsführer erkannt werden. Denkbar wäre auch, dass durch einen Klick auf die Karte deutlich wird, welche Gemälde sich in welchem Stockwerk bzw. Raum befinden. So könnten Flyer gespart werden, die oft in Museen ausliegen, und alles in einer Informationsquelle vereint werden. Zur technischen Umsetzung könnte das Tracken über Bluetooth eine Möglichkeit sein oder der Ansatz des Google-Projektes Tango.

- Da die SKK oft von Schulklassen und Jugendlichen besucht wird, wäre eine Erweiterung des Konzeptes mit spielerischen Elementen denkbar. Beispielsweise könnte ein Quiz eingebaut werden. Das würde auch den Lernerfolg steigern.
- Der Mensch möchte in quasi jeder Lebenslage Teil einer sozialen Gruppe sein (vgl. 2.3.4.4). Der zwischenmenschliche Kontakt fehlt gänzlich im vorliegenden Konzept, da der Schwerpunkt der Arbeit auf der Umsetzung liegt. In einer weiteren Arbeit könnte versucht werden, Kommunikationswege zu eröffnen. Entweder könnten die Nutzer über die App miteinander kommunizieren oder Zugriff auf Foren, Webseiten etc. bekommen. Dies könnte darüber hinaus sehr interessant sein, wenn die Anwendung mehr spielerische Elemente erhalten sollte.
- Der Lernerfolg und die Motivationssteigerung der Nutzer könnte in einem umfanglicher angelegten Usability-Test überprüft werden. Dadurch könnte herausgefunden werden, wo die App noch weiter optimiert werden könnte.
- Zur Verwaltung der XML-Dateien, Fotografien, Gemälden etc. könnte ein Single-Source-Konzept entwickelt werden, damit die Kunsthalle Informationen nicht an mehreren Stellen ablegen muss und somit schnell den Überblick verlieren könnte.
- Es könnte sinnvoll sein, die Anwendungstexte und Bedienung barrierefrei zu gestalten. Hierzu wäre die Optimierung der Hotspot-Beschreibungen hinsichtlich des Detaillierungsgrades nötig. Zusätzlich wäre die Anwendung von Leichter Sprache eine Option, um den Museumsführer für möglichst viele Menschen attraktiv zu gestalten.
- Bisher sind nur die gemäldebezogenen Inhalte multilingual. Wünschenswert wäre die Umstellung der „welcome“-Seite sowie deren Unterseiten.

Literaturverzeichnis

- Abbound, Rana (2014): "Architecture in an Age of Augmented Reality"
<<http://architectureau.com/articles/architecture-in-an-age-of-augmented-reality/>>
[Stand: 07.04.2014. Zugriff: 03.06.2017, 12:50 MESZ]
- Alsdorf, Claudia / Bannwart, Edouard (2002): "Virtuelle Realität - Erfahrbare Informationen im Cyberspace". In: Issing, Ludwig J.; Klimsa, Paul (Hrsg.) (2002), 467–480
- Axworthy, Jon (2016): "The Origins of Virtual Reality"
<<https://www.wearable.com/wearable-tech/origins-of-virtual-reality-2535>>
[Stand: 01.04.2016. Zugriff: 03.06.2017, 12:50 MESZ]
- Azuma, Ronald T. (1997): "A Survey of Augmented Reality". In: Presence: Teleoperators and Virtual Environments (1997), Nr. 4, 355–385"
<<http://www.cs.unc.edu/~azuma/ARpresence.pdf>>
[Stand: k. A. Zugriff: 03.06.2017, 11:25 MESZ]
- Ballstaedt, Steffen-Peter (1997): "Wissensvermittlung - Die Gestaltung von Lernmaterial". Weinheim : Beltz Psychologie-Verl.-Union
- Barfield, Woodrow (Hrsg.) (2016): "Fundamentals of Wearable Computers and Augmented Reality". Second edition. Boca Raton, Florida : CRC Press Taylor & Francis Group
- , - (2016): "Wearable Computers and Augmented Reality - Musing and Future Directions". In: Barfield, Woodrow (Hrsg.) (2016), 3–11
- Becker, Klaus-Peter (2014): "Fachkonzept - Datenkapselung"
<http://www.inf-schule.de/modellierung/oopython/roboter/datenkapselung/konzept_datenkapselung>
[Stand: 24.07.2014. Zugriff: 03.06.2017, 12:50 MESZ]
- Bibliographisches Institut & F. A. Brockhaus AG Mannheim (2006): Duden - das Bedeutungswörterbuch. Bd. 10. 3., neu bearb. und erw. Aufl. Mannheim : Dudenverl.
- Bibliographisches Institut GmbH (2017): "Duden - Rechtschreibung, Bedeutung, Synonyme, Herkunft"
<<http://www.duden.de/rechtschreibung/virtuell>>
[Stand: k. A. Zugriff: 03.03.2017 13:25 MEZ]
- Boeckmann, Klaus-Börge (2008): "eLernen und Fremdsprachendidaktik - reflektierte Praxis, etwas Theorie und ein Beispiel". In: Boeckmann, Klaus-Börge; Rieder-Bünemann, Angelika; Vetter, Eva (Hrsg.) (2008), 13–28
- Boeckmann, Klaus-Börge; Rieder-Bünemann, Angelika; Vetter, Eva (Hrsg.) (2008): "eLernen/eLearning/Apprentissage en ligne in der sprachbezogenen Lehre - Prinzipien, Praxiserfahrungen und Unterrichtskonzepte". Frankfurt, M : Lang

- Broll, Wolfgang (2013): "Augmentierte Realität". In: Dörner, Ralf; Broll, Wolfgang; Grimm, Paul; Jung, Bernhard (Hrsg.) (2013), 241–294
- Brosi, Sybille / Erler-Striebel, Petra / Trevisan, Sandra (Kunstvermittlung, Staatliche Kunsthalle Karlsruhe) (06.04.2017, 16:00 MESZ): Gespräch mit Rebecca Durm
- Buhr, Mathias et al. (2013): "Echtzeitaspekte von VR-Systemen". In: Dörner, Ralf; Broll, Wolfgang; Grimm, Paul; Jung, Bernhard (Hrsg.) (2013), 195–240
- Culture to go (2017): "Augmented Reality im Museum - AR-Projekte, Demos, Experimente"
 <<http://culture-to-go.com/mediathek/augmented-reality-im-museum/>>
 [Stand: k. A. Zugriff: 06.03.2017, 12:50 MESZ]
- DocCheck Medical Services GmbH (o. J.): "Gedächtnis"
 <http://flexikon.doccheck.com/de/Ged%C3%A4chtnis?utm_source=www.doccheck.flexikon&utm_medium=web&utm_campaign=DC%2BSearch>
 [Stand: k. A. Zugriff: 06.03.2017, 13:00 MESZ]
- Dörner, Ralf et al. (2013): "Einleitung". In: Dörner, Ralf; Broll, Wolfgang; Grimm, Paul; Jung, Bernhard (Hrsg.) (2013), 1–32
- Dörner, Ralf; Broll, Wolfgang; Grimm, Paul; Jung, Bernhard (Hrsg.) (2013): "Virtual und Augmented Reality (VR/AR) - Grundlagen und Methoden der Virtuellen und Augmentierten Realität". In: eXamen.press. Berlin : Springer Vieweg.
- Dörner, Ralf / Steinicke, Frank (2013): "Wahrnehmungsaspekte von VR". In: Dörner, Ralf; Broll, Wolfgang; Grimm, Paul; Jung, Bernhard (Hrsg.) (2013), 33–64
- Dunleavy, Matt / Dede, Chris (o. J.): "Augmented Reality Teaching and Learning"
 <<http://isites.harvard.edu/fs/docs/icb.topic1116077.files/DunleavyDedeARfinal.pdf>>
 [Stand: k. A. Zugriff: 03.06.2017, 12:45 MESZ]
- Editions Atlas SA (o. J.): "Das Gedächtnis". In: Editions Atlas SA (o. J.)
- Editions Atlas SA (o. J.): "Der menschliche Körper". Cheseaux-sur-Lausanne : Editions Atlas.
- Fraunhofer MEVIS (2013): "Tablet-PC unterstützt Leberchirurgen - Neue App von Fraunhofer MEVIS erstmals in deutschem OP getestet"
 <<https://www.mevis.fraunhofer.de/de/press-and-scicom/press-release/tablet-pc-supports-liver-surgeons---new-app-from-fraunhofer-mevi.html>>
 [Stand: 20.08.2013. Zugriff: 01.03.2017 11:15 MEZ]
- Ganapathy, Subhashini (2013): "Design Guidelines for Mobile Augmented Reality: User Experience". In: Huang, Weidong; Alem, Leila; Livingston, Mark A. (Hrsg.) (2013), 165–180

- Google ATAP (20.02.2014): "Say hello to Project Tango!"
 <<https://www.youtube.com/watch?v=Qe10ExwzCqk>>. Zugriff: 06.03.2017 12:50
 MEZ
- Grimm, Paul et al. (2013): "VR-Eingabegeräte". In: Dörner, Ralf; Broll, Wolfgang; Grimm, Paul; Jung, Bernhard (Hrsg.) (2013), 97–126
- Haack, Johannes (2002): "Interaktivität als Kennzeichen von Multimedia und Hypermedia". In: Issing, Ludwig J.; Klimsa, Paul (Hrsg.) (2002), 127–136
- Hale, Kelly S.; Stanney, Kay M. (Hrsg.) (2015): "Handbook of Virtual Environments - Design, Implementation, and Applications". In: Human Factors and Ergonomics Series. Second edition. Boca Raton, Florida : CRC Press.
- Hemmerling, Marco (Hrsg.) (2011a): "Augmented Reality - Mensch, Raum und Virtualität". In: PerceptionLab. Bd. 1. Paderborn : Fink.
- , - (2011): "Die Erweiterung der Realität". In: Hemmerling, Marco (Hrsg.) (2011), 13–24
- , - (2011): "Einleitung". In: Hemmerling, Marco (Hrsg.) (2011), 7–11
- Holzinger, Andreas (2001): "Lernen - Kognitive Grundlagen multimedialer Informationssysteme". In: Basiswissen Multimedia. Band 2. 1. Aufl. Würzburg : Vogel.
- Huang, Weidong; Alem, Leila; Livingston, Mark A. (Hrsg.) (2013): "Human Factors in Augmented Reality Environments". New York : Springer Vieweg.
- Institut für Museumsforschung (2016): "Statistische Gesamterhebung an den Museen der Bundesrepublik Deutschland für das Jahr 2015". In: Materialien aus dem Institut für Museumskunde. Heft 70. Berlin : Staatliche Museen PK Institut für Museumskunde
 <http://www.smb.museum/fileadmin/website/Institute/Institut_fuer_Museumsforschung/Publikationen/Materialien/mat70.pdf>
 [Stand: k. A. Zugriff: 03.06.2017, 11:55 MESZ]
- Issing, Ludwig J.; Klimsa, Paul (Hrsg.) (2002): "Information und Lernen mit Multimedia und Internet - Lehrbuch für Studium und Praxis". 3., vollst. überarb. Aufl. Weinheim : Beltz PVU
- Jäncke, Lutz (2005): "E-Learning aus der Sicht der Neuropsychologie". In: Miller, Damian (Hrsg.) (2005), 83–110
- Jung, Bernhard / Vitzthum, Arnd (2013): "Virtuelle Welten". In: Dörner, Ralf; Broll, Wolfgang; Grimm, Paul; Jung, Bernhard (Hrsg.) (2013), 65–96
- Kathe, Florian (2015): "360-Grad-Tracking für 3D-Objekte". Universität Koblenz - Landau, Fachbereich 4: Informatik, Masterthesis, 2015
 <<https://kola.opus.hbz-nrw.de/frontdoor/index/index/docId/909>>
 [Stand: 30.01.2015. Zugriff: 22.02.2017, 11:25 MEZ]

- Kerres, Michael (2002): "Technische Aspekte multi- und telemedialer Lernangebote". In: Issing, Ludwig J.; Klimsa, Paul (Hrsg.) (2002), 19–27
- Kerres, Michael (2013): "Mediendidaktik - Konzeption und Entwicklung mediengestützter Lernangebote". 4., überarb. und aktualisierte Aufl. München : Oldenbourg
- Klein, Georg (2009): "Visual Tracking for Augmented Reality - Edge-based Tracking Techniques for AR Applications". Saarbrücken : VDM Verl. Müller
- Kleinhüeckelkotten, Silke (2012): "Neue Zielgruppen für das Museumsmarketing - Grundlagen und Anregungen für eine milieuorientierte Kommunikation" <http://www.annette-jagla.de/files/kleinhueeckelkotten_zielgruppen_museumsmarketing_mai2012.pdf>
[Stand: 11.05.2012. Zugriff: 03.06.2017, 13:15 MESZ]
- Knop, Sebastian (2016): "Impressumspflicht auch für Apps" <<http://www.impressum-generator.de/2016/02/impressumspflicht-gilt-auch-fuer-apps/>>
[Stand: 09.02.2016. Zugriff: 03.06.2017, 13:05 MESZ]
- Kron, Friedrich W. (2004): Grundwissen Didaktik. 4. Auflage. München : Ernst Reinhardt.
- Maiero, Jens (2009): "Image-based Tracking". Hochschule Bonn-Rhein-Sieg, Fachbereich 02 (Informatik), Masterthesis, 2009 <http://cg.inf.h-bonn-rhein-sieg.de/wp-content/uploads/2009/10/imagebased_tracking_jm09.pdf>
[Stand: 2009. Zugriff: 03.06.2017, 11:35 MESZ]
- Mehler-Bicher, Anett / Steiger, Lothar (2014): "Augmented Reality - Theorie und Praxis". 2., überarbeitete Auflage. München : de Gruyter Oldenbourg
- Mernberger, Tabea (Presse und Digitale Kommunikation, Staatliche Kunsthalle Karlsruhe) (Stand: 10.04.2017, 10:43 MESZ): "AW: Corporate Design für Digitale Materialien der SKK". Email an Rebecca Durm
- Milgram, Paul / Colquhoun, Herman (1999): "A Taxonomy of Real and Virtual World Display Integration" <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.6230&rep=rep1&type=pdf>>
[Stand: k. A. Zugriff: 03.06.2017, 11:25 MESZ]
- Miller, Damian (Hrsg.) (2005): "E-Learning - Eine multiperspektivische Standortbestimmung". Bern : Haupt
- Owen, Charles B. / Xiao, Fan / Middlin, Paul (o. J.): "What Is the Best Fiducial?" <<http://www.vs.inf.ethz.ch/edu/SS2005/DS/papers/ar/owen-bestfiducial.pdf>>
[Stand: k. A. Zugriff: 03.06.2017, 10:40 MESZ]

- Ravensburger AG (2013): "Ravensburger Scotland Yard - Master"
 <<https://www.youtube.com/watch?v=cwRLmfWnWbY>>.
 [Stand: 02.09.2013. Zugriff: 07.06.2017, 12:25 MESZ]
- Rey, Günter D. (o. J.): "Lernen mit Multimedia - Die Gestaltung interaktiver Animationen". Universität Trier, Fachbereich I - Psychologie, Dissertation, o. J.
 <http://ubt.opus.hbz-nrw.de/volltexte/2008/450/pdf/Lernen_mit_Multimedia.pdf>
 [Stand: k. A. Zugriff: 03.06.2017, 11:35 MESZ]
- Richter, Andreas (o. J.): "Informationsrepräsentation und Multimediales Lernen"
 <<http://www.scribd.com/doc/2979242/Informationsrepräsentation-und-multimediales-Lernen>>
 [Stand: k. A. Zugriff: 03.06.2017, 13:20 MESZ]
- Sarkar, Samit (2017): "Pokémon Go Hits 650 Million Downloads"
 <<http://www.polygon.com/2017/2/27/14753570/pokemon-go-downloads-650-million>>
 [Stand: 27.02.2017 4:00 pm EST. Zugriff: 01.03.2017, 16:15 MEZ]
- Schart, Dirk (o. J.): "Fünf Anwendungsfelder für Augmented Reality in Automotive"
 <<https://www.wearear.com/automotive-augmented-reality/>>
 [Stand: k. A. Zugriff: 16.06.2017, 15:50 MESZ]
- Schmalstieg, Dieter / Höllerer, Tobias (2016): "Augmented Reality - Principles and Practice". Boston : Addison-Wesley
- Schulmeister, Rolf (2005): "Interaktivität in Multimedia-Anwendungen"
 <<https://www.e-teaching.org/didaktik/gestaltung/interaktiv/InteraktivitaetSchulmeister.pdf>>
 [Stand: 08.11.2005. Zugriff: 15.03.2017, 17:10 MEZ]
- Seifert, Carsten (2015): "Spiele entwickeln mit Unity 5 - 2D- und 3D-Games mit Unity und C# für Desktop, Web & Mobile". 2., aktualisierte und erweiterte Auflage. München : Hanser
- Stannel, Kay M. / Zyda, Michael / Hale, Kelly S. (2015): "Virtual Environments in the Twenty-First Century". In: Hale, Kelly S.; Stanney, Kay M. (Hrsg.) (2015), 3–22
- Teichrieb, Veronica et al. (2007): "A Survey of Online Monocular Markerless Augmented Reality". In: International Journal of Modeling and Simulation for the Petroleum Industry (2007), 1–7
 <<http://www.ijmspi.org/ojs/index.php/ijmspi/article/view/88/19>>.
 [Stand: k. A. Zugriff: 03.06.2017, 11:35 MESZ]
- TMG (Stand: 26.02.2007 (i.d.F. v. 21.07.2016))
- Virtual Reality Magazin (2012): "Augmented Reality in Kontaktlinsen"
 <<http://www.virtual-reality-magazin.de/augmented-reality-kontaktlinsen>>
 [Stand: 07.12.2012. Zugriff: 03.06.2017, 13:05 MESZ]

- Vontobel, Peter (2006): "Didaktisches Design aus lernpsychologischer Sicht"
<http://www.sciencetonic.de/media/015_digimedia/050_konzepte/LIT_0210_Didaktisches_Design_Vontobel_2006.pdf>
[Stand: 08.05.2006. Zugriff: 03.06.2017, 13:20 MESZ]
- Wagner, Daniel (2007): "Handheld Augmented Reality". Graz University of Technology, Institute for Computer Graphics and Vision, Dissertation, 2007
<http://www.icg.tugraz.at/Members/daniel/HandheldAR_Thesis>
[Stand: 01.10.2007. Zugriff: 06.03.2017, 16:40 MEZ]
- Weidenmann, Bernd (2002): "Multicodierung und Multimodalität im Lernprozess". In: Issing, Ludwig J.; Klimsa, Paul (Hrsg.) (2002), 45–62
- Willmann, Urs (2016): "Schöner Stress"
<<http://www.zeit.de/2016/18/alltag-stress-gesundheit-positive-effekte>>
[Stand: 21.04.2016. Zugriff: 19.03.2017, 15:10 MEZ]

Anhänge

Nutzungshinweise für Unity Remote 5

Bei der Benutzung von Unity Remote ist für einen reibungslosen Ablauf die Schrittfolgenfolge zu beachten:

- 1) Unity Remote auf dem Testgerät starten.
- 2) Das Endgerät über USB an den Computer anschließen. Hierbei muss das Debugging des Endgerätes aktiviert sein und es als Mediengerät verbunden werden. Dies muss in den meisten Fällen in den Einstellungen des Endgerätes bei der ersten Nutzung angepasst werden. Wenn eine Popup-Nachricht erscheint, die die Zugriffserlaubnis abfragt, muss diese bestätigt werden. Die Erlaubnis muss vor Schritt 3 bestätigt werden.
- 3) Unity auf dem Computer starten.

Um die in Unity erstellte Anwendung nun zu testen, wird der Play-Button in der Unity-Benutzeroberfläche gedrückt. Die aktuelle Szene in Unity erscheint auf dem Bildschirm des Endgerätes. Wenn mit Augmented Reality gearbeitet wird, wird zusätzlich die Webcam des Computers angeschaltet, da sie die Aufgabe der Endgerätkamera für den Test übernimmt. Damit getestet werden kann, wie die Anwendung reagiert, wenn die Endgerätkamera das Target trackt, muss das Target vor die Webcam gehalten werden. Die Folgen sind auf dem Bildschirm des Endgerätes zu sehen. Die Darstellung auf dem Bildschirm ist dann etwas verpixelt und unscharf, da die Aufnahme der Web-Cam die virtuell erstellten Objekte „überdeckt“. Um die Funktionalitäten der App zu testen, ist das allerdings keine große Einschränkung.

Eingebundene Gemälde



Abbildung 7.2: "Die 7 Todsünden" von Otto Dix (1933)



Abbildung 7.3: "Melencolia I" von Albrecht Dürer (1514)



Abbildung 7.4: "Landschaft mit der Anbetung des goldenen Kalbes" von Claude Lorraine (1653)

Quellen für die Informationen in der App

- **Textinformationen über „Die 7 Todsünden“**

Schwarz, Birgit / Staatliche Kunsthalle Karlsruhe (Hg.) (1986): Werke von Otto Dix. In der Vortragsreihe: Bildhefte der Staatlichen Kunsthalle Karlsruhe. Nr. 11. Karlsruhe : Staatl. Kunsthalle

- **Künstlerportraits**

Otto Dix

Lebendiges Museum Online (o. J.): „Otto Dix“.

<<https://www.dhm.de/lemo/bestand/objekt/30277>>

[Stand: k. A. Zugriff: 02.06.2017, 19:45 MESZ]

Claude Lorrain

Wikipedia (2015): „Claude Lorrain“.

<<https://de.wikipedia.org/w/index.php?oldid=159858525>>

[Stand: 19.11.2016. Zugriff: 02.06.2017, 19:40 MESZ].

Albrecht Dürer

Schnabel, Norbert (2013): „Seht her, ich bin ein Künstler! – Albrecht Dürers Selbstbildnis in Madrid“.

<<http://syndrome-de-stendhal.blogspot.de/2013/01/seht-her-ich-bin-ein-kunstler.html>>

[Stand: 12.05.2017. Zugriff: 02.06.2017]

- **Textinformationen über Dürers Rhombusstumpf**

Ziegler, Günter M. (24.12.2014): „Dürer-Polyeder – Ein mysteriöser Körper wird 500“.

<<http://www.spiegel.de/wissenschaft/mensch/albrecht-duerers-polyeder-ein-mysterioeses-raetsel-wird-500-a-1009959.html>>

[Stand: k. A. Zugriff: 02.06.2017, 19:30 MESZ]

- **Textinformationen über das goldene Kalb:**

Wikipedia (2015): „Goldenes Kalb“

<<https://de.wikipedia.org/w/index.php?oldid=165142250>>

[Stand: 03.05.2017. Zugriff: 02.06.2017, 18:20 MESZ].

- **Bild Staatliche Kunsthalle**

<http://www.kunsthalle-karlsruhe.de/fileadmin/_processed_/csm_01_KHK-CF011438_a53c8069c3.jpg>

[Stand: k. A. Zugriff: 02.06.2017, 19:30 MESZ]

XML-Datei mit den Inhaltstexten für „Die 7 Todsünden“

```
<?xml version="1.0" encoding="utf-8"?>
<work exhibition="permanent">
  <painting>
    <name>Die 7 Todsünden</name>
    <date>1933</date>
    <materials>Mischtechnik auf Holz</materials>
    <hight unit="cm">179</hight>
    <width unit="cm">120,5</width>
  </painting>
  <artist nationality="de">
    <artistname>Otto Dix</artistname>
    <lifespan>1891-1968</lifespan>
  </artist>
  <information>
    <hotspot id="1">
      <!-- Alte - GEIZ -->
      <h>Todsünde 2: Geiz</h>
      <text>
        <p>Die Alte mit der Hakennase erinnert mit ihrer gebückten Haltung an die Hexen aus Märchen und Gruselgeschichten. Ihre pupillenlosen grünen Augen sind aufmerksam nach vorne gerichtet, ihr Gesichtsausdruck vermittelt Entschlossenheit und Begierde. Sie zieht als Spitze des Zuges vorweg und wird von den anderen dahinter beinahe bedrängt. Sie steht für die zweite Todsünde: den Geiz. Dies symbolisieren sowohl die zerrissenen Kleider, als auch die Geldscheine in ihrer rechten Hand. In der linken Hand hält sie einen abgebrochenen Spazierstock. Das kleine Stückchen Holz verhindert, dass die Alte vollständig auf allen Vieren kriecht. </p>
        <p>Dix orientierte sich an deutschen Malern des Spätmittelalters, die Geiz auch als alte, magere, hässliche Frauen darstellten. </p>
      </text>
    </hotspot>
    <hotspot id="2">
      <!-- Wicht - NEID -->
      <h>Todsünde 4: Neid</h>
      <text>
        <p>Der Wicht auf dem Rücken der Alten scheint der eigentliche Anführer des Zuges zu sein, er fällt sofort ins Auge. Seine Haare sind blond, sein Wams gelb - die Farbe des Neides. Neid ist die vierte Todsünde. Das wahre Gesicht des Wichtes ist nicht erkennbar, er versteckt es hinter einer Maske. Die Augen schielen nach rechts und links gleichzeitig; fast so, als ob sie nach etwas suchen, was zwar Anderen gehört, man aber selbst gerne hätte. Die Mundpartie mit den heruntergezogenen Mundwinkeln strahlt Verachtung aus. </p>
        <p>Das Auffälligste an dem Wicht ist der Oberlippenbart der Maske, der sofort an Adolf Hitler erinnert. Otto Dix hat ihn erst nach dem zweiten Weltkrieg hinzugefügt. Er hatte das Gefühl, dass sich einige Mitmenschen hinter einer politischen Maske verborgen hielten und ihn beneideten, denn er war beliebt und anerkannt. Einer seiner Kollegen an der Dresdener Akademie, Richard Müller, war Mitglied der NSDAP. Er stellte einige Werke von Dix in einer Ausstellung für "entartete Kunst" aus. 1933 sorgte er auch dafür, dass Dix aus dem Professorenamt entlassen wurde. Wenig später wurde er Direktor und ließ Dix' Namen von der Ehrentafel entfernen. </p>
      </text>
    </hotspot>
    <hotspot id="3">
      <!-- Untier - ZORN -->
      <h>Todsünde 6: Zorn</h>
      <text>
```

`<p>Wenn Menschen etwas aus der Fassung bringt, können sie unerwartete Reaktionen zeigen; sie werden zu Bestien, vor allem durch Zorn, die sechste Todsünde. Dix wählt die Figur eines gehörnten Untiers: das Maul ist weit aufgerissen und zeigt die Fangzähne; die Augen sind mit den schlitzförmigen Pupillen reptilienähnlich. Das stark behaarte Monster hält einen Dolch in der rechten Hand, eine ähnliche Waffe geformt wie die Sense. Er droht und verbreitet Angst und Schrecken - wozu ist er wohl noch fähig?</p>`

```
</text>
</hotspot>
<hotspot id="4">
  <!-- Tod - TRÄGHEIT-->
  <h>Todsünde 7: Trägheit</h>
  <text>
```

`<p>Der Mann mit der Sense ist seit Jahrhunderten eine Darstellung des Todes; der Sensenmann führt hier den Totentanz auf. Er wirkt sehr lebhaft und gelöst, jedoch symbolisiert er die siebte und letzte Todsünde: die Trägheit. An Stelle des Herzens sitzt eine Kröte, ein kaltblütiges, langsames und abstoßendes Tier, das bei Todesdarstellungen häufig auftritt. Wenn man genau hinsieht, erkennt man, dass der Tod mit der Frau tanzt, sie bilden ein Paar. Er trägt ein schwarzes Gewand mit einem aufgedruckten Skelett. Hinter dem weißen Tuch über dem Kopf zeigen sich die Augenhöhlen über der Stelle, wo sich früher einmal die Nase befand; sie ist schon verwest.</p>`

`<p>Der Totentanz ist eine klassische Darstellung. Sie sollen an die Nichtigkeit aller irdischen Güter erinnern. Der Mensch soll Buße tun und so leben, wie es Gott gefällt.</p>`

```
</text>
</hotspot>
<hotspot id="5">
  <!-- Weib - WOLLUST -->
  <h>Todsünde 3: Wollust</h>
  <text>
```

`<p>Ohne Zweifel stellt die Frau Todsünde Nummer 3 dar: Wollust. Das sehr enge orange Kleid bedeckt sie nur halb. An den Beinen weht es auf und zeigt das rote Innenfutter. Das rote Innenfutter und die Form des orangenen Stoffes erinnern an eine Vagina. Die Frau ist das Abbild einer Prostituierten. Dies erkennt man an der Geste, wie sie ihren Busen anbietet und am Schorf an der Oberlippe. Der Schorf kommt von einer Syphilis-Erkrankung, an der die Menschen ohne Penicillin starben. Die Wollust ist nicht wie in vielen älteren Darstellungen als junges, hübsches Mädchen dargestellt: Sie ist kein unschuldiges Opfer, sondern ein Todesengel mit Syphilis, der mit dem Tod tanzt.</p>`

`<p>Als Vorbild diente Otto Dix eine alte Zeichnung von seiner langjährigen Geliebten und Modell Käthe König. Als die Zeichnung entstand, dachte Dix noch nicht daran, sie als Vorlage für die Todsünde Wollust zu benutzen.</p>`

```
</text>
</hotspot>
<hotspot id="6">
  <!-- Schwellkopf - HOCHMUT, STOLZ -->
  <h>Todsünde 1: Stolz</h>
  <text>
```

`<p>Die Nase in den Himmel gestreckt symbolisiert diese Figur die erste Todsünde: Hochmut. Der Schwellkopf wird von einem Menschen getragen; man erkennt die linke Hand im Ohr der Maske. Schwellköpfe kommen aus dem Karneval. Der Hochmut bildet mit den anderen einen Art Fastnachtsumzug. Die Augen der Maske sind geschlossen. Die erhobene Nase macht den Blick auf den afterartigen Mund frei. Die Wangen und die Nase sind mit Schorf bedeckt.</p>`

`<p>Die Figur ähnelt Dix' Kollegen Ferdinand Dorsch. Er wurde nach`

Richard Müller, der für Dix' Verweis aus dem Professorenamt mitverantwortlich war, Rektor der Dresdner Akademie. Ob er bewusst für die Figur des Stolzes gewählt wurde, ist unbekannt. Wahrscheinlich war das Verhältnis zwischen Dix und Dorsch eher distanziert. </p>

</text>

</hotspot>

<hotspot id="7">

<!-- Suppentopfjunge - VÖLLEREI -->

<h>Todsünde 5: Völlerei</h>

<text>

<p>Das dicke Kind mit dem fülligen Gesicht hat einen Suppentopf über dem Kopf. Es hält in der einen Hand eine Brezel, in der anderen Hand eine Holzstange mit Wurstpaaen. Es steht für die Völlerei - Todsünde Nummer 5. Da der Kessel mit Augen und Nase bemalt ist, sieht es so aus, als ob das Kind von Suppentopf verschlungen wurde. Der Körper des Jungen und seine Arme sind von Würsten umwickelt. Nicht nur der Junge ist Symbol für die Gier nach Essen. </p>

<p>Bei dieser Figur werden Elemente des Karnevals aufgegriffen: Die Brezel gilt als ein Fastengebäck. Fleischwaren sind in der Fastenzeit bis Ostern tabu. An Karneval wird aber noch einmal gesündigt.</p>

</text>

</hotspot>

<hotspot id="8">

<!-- Zitat Nietzsche, Ambiente -->

<h>Hintergrund</h>

<text>

<p>Die kahle, öde Landschaft erinnert an eine Wüste. Otto Dix hat während seiner Dienstzeit im ersten Weltkrieg diese Landschaften häufig gesehen und immer wieder in seinen Zeichnungen aufgegriffen. Sie dient als Symbol für die zerstörerische Kraft des Bösen. "Die Wüste wächst. Wehe dem, der Wüsten birgt", ist auf der linken, bröckelnden Hauswand zu lesen. Dieses Zitat stammt von Friedrich Nietzsche. </p>

</text>

</hotspot>

<hotspot id="9">

<!-- Geschichtlicher Hintergrund -->

<h>Über das Gemälde und seine Entstehung</h>

<text>

<p>"Sie haben Bilder gemalt, die geeignet sind, den Wehrwillen zu beeinträchtigen. Danach bieten Sie nicht die Gewähr dafür, dass Sie jederzeit rückhaltlos für den nationalen Staat eintreten". So lauteten die Worte, mit denen Otto Dix aus seinem Professorenamt enthoben wurde. Kurz nach der Wahl der Nationalsozialisten 1933 war der angesehene Professor der Dresdener Akademie geächtet und blickte in eine unsichere Zukunft. Dix wollte zunächst in Dresden bleiben und malte im Sommer 1933 "Die 7 Todsünden". Ende des Jahres beugte er sich dem Druck der Nationalsozialisten und zog an den Hohentwiel. </p>

<p>Die 7 Todsünden werden in der Bibel als Hauptlaster beschrieben: Sie sind die Ursache für alle anderen Sünden wie Mord (z. B. aus Gier) oder Diebstahl (z. B. aus Neid). Sie führen dazu, dass der Mensch die göttliche Gnade verliert und zum ewigen Tod verdammt ist. Außerdem führen sie zur Herrschaft des Teufels und des Todes. </p>

<p>"Die 7 Todsünden" ist das erste und einzige Werk von Otto Dix mit christlicher Morallehre als Thema. Immer wieder wollte sich Dix mit den alten deutschen Malern messen wie Albrecht Dürer, Meister Francke oder Pieter Bruegel dem Älteren messen. Deshalb wählte er eine Thematik mit kunsthistorischer Tradition und orientierte sich an den Werken seiner Vorbilder. Bei den 7 Todsünden handelt sich um eine Allegorie. Das ist die Darstellung eines abstrakten Begriffs durch Personifizierungen. </p>

</text>

</hotspot>

```
</information>  
</work>
```

Quellcode¹⁵

defaultTrackableEventHandler

```
/*=====
====
Copyright (c) 2010-2014 Qualcomm Connected Experiences, Inc.
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
=====
===*/

using UnityEngine;
using UnityEngine.UI;

namespace Vuforia
{
    /// <summary>
    /// A custom handler that implements the ITrackableEventHandler inter-
    face.
    /// </summary>
    public class DefaultTrackableEventHandler : MonoBehaviour,
        ITrackableEventHandler
    {
        #region Nachgefügte Deklarationen für die Ein- und Ausblendungen
        //public Transform TextTargetName;
        //public Transform PanelHotspotInformation;

        public Transform TextDescription;
        public Transform TextHeading;
        public Transform PanelDescription;
        public Transform PanelTextTargetName;
        public Canvas CanvasHotspotCollection;

        public GameObject[] targets;

        #endregion

        #region PRIVATE_MEMBER_VARIABLES
        private TrackableBehaviour mTrackableBehaviour;
        #endregion // PRIVATE_MEMBER_VARIABLES

        #region UNITY_MONOBEHAVIOUR_METHODS
        void Start()
        {
            mTrackableBehaviour = GetComponent<TrackableBehaviour>();
            if (mTrackableBehaviour)
            {
                mTrackableBehaviour.RegisterTrackableEventHandler(this);
            }
        }
    }
}
```

¹⁵ Längere auskommentierte Abschnitte wurden entfernt

```

    }

    #endregion // UNTIY_MONOBEHAVIOUR_METHODS

    #region PUBLIC_METHODS

    /// <summary>
    /// Implementation of the ITrackableEventHandler function called
when the
    /// tracking state changes.
    /// </summary>
    public void OnTrackableStateChanged(
        TrackableBehaviour.Status previ-
ousStatus,
        TrackableBehaviour.Status newSta-
tus)
    {
        if (newStatus == TrackableBehaviour.Status.DETECTED ||
            newStatus == TrackableBehaviour.Status.TRACKED ||
            newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
        {
            OnTrackingFound();
        }
        else
        {
            OnTrackingLost();
        }
    }

    #endregion // PUBLIC_METHODS

    #region PRIVATE_METHODS

    private void OnTrackingFound()
    {
        Renderer[] rendererComponents = GetComponentsInChildren<Ren-
derer>(true);
        Collider[] colliderComponents = GetComponentsInChildren<Col-
lider>(true);

        // Enable rendering:
        foreach (Renderer component in rendererComponents)
        {
            component.enabled = true;
        }

        // Enable colliders:
        foreach (Collider component in colliderComponents)
        {
            component.enabled = true;
        }

        Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + "
found");
    }
}

```

```

        private void OnTrackingLost()
        {
            Renderer[] rendererComponents = GetComponentsInChildren<Render-
derer>(true);
            Collider[] colliderComponents = GetComponentsInChildren<Col-
lider>(true);

            // Disable rendering:
            foreach (Renderer component in rendererComponents)
            {
                component.enabled = false;
            }

            // Disable colliders:
            foreach (Collider component in colliderComponents)
            {
                component.enabled = false;
            }

            Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + "
lost");

            #region Zugefügt: Was passiert beim Verlieren des Bildes

            //Beim Verlieren des Targets: Panele werden inaktiv, Hotspot-
Collection wird unsichtbar, Texte für Hotspotbeschreibung und Überschrift
werden gelöscht
            PanelDescription.gameObject.SetActive(false);
            PanelTextTargetName.gameObject.SetActive(false);
            CanvasHotspotCollection.gameObject.SetActive(false);
            TextDescription.gameObject.GetComponent<Text>().text = " ";
            TextHeading.gameObject.GetComponent<Text>().text = " ";

            //TextTargetName.GetComponent<Text>().text = "Kein Bild
getrackt";
            //TextDescription.gameObject.SetActive(false);
            //PanelHotspotInformation.gameObject.SetActive(false);

            #endregion

        }

        #endregion // PRIVATE_METHODS
    }
}

```

languageButtons.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO;
using System.Xml;
using UnityEngine.UI;

public class languageButtons : MonoBehaviour

```

```

{
    #region Deklarationen der Variablen
    //Panele und Textfelder, in denen später Text geschrieben wird.
    public Transform panelSound;
    public Transform TextDescription;
    public Transform TextHeading;

    //Buttons, die gedrückt werden können
    public Button germanButton;
    public Button frenchButton;
    public Button englishButton;
    public Button speakingButton;

    //Schatten-Komponenten, die den Image-Buttons zugefügt wurde
    public Shadow shadowGermanButton;
    public Shadow shadowFrenchButton;
    public Shadow shadowEnglishButton;
    public Shadow shadowSpeakingButton;

    //Zeichenvariable für die Dateierdung abhängig von der gewählten Sprache / Flagge
    public string fileNameEnding;
    //Zeichenvariable, die die Sprache enthält (wird in anderem Skript ausgelesen)
    public string setLanguage;
    #endregion

    #region Deklaration für Zugriff auf Skript
    //Skriptzugriff auf openSettingWindow
    public GameObject CanvasLeft;
    public openSettingWindow closeWindow;
    #endregion

    void Start()
    {
        //Zugriff auf Button-Komponenten
        germanButton = germanButton.GetComponent<Button>();
        frenchButton = frenchButton.GetComponent<Button>();
        englishButton = englishButton.GetComponent<Button>();
        speakingButton = speakingButton.GetComponent<Button>();

        //PanelSound ist zu Beginn ausgeblendet
        panelSound.gameObject.SetActive(false);

        //Zugriff auf Skript OpenSettingWindow
        closeWindow = CanvasLeft.GetComponent<openSettingWindow>();

        //Zugriff auf die Shadow-Komponente (setzt die Variable shadowGermanButton in Verbindung mit der Komponente "Shadow")
        shadowGermanButton = germanButton.GetComponent<Shadow>();
        shadowFrenchButton = frenchButton.GetComponent<Shadow>();
        shadowEnglishButton = englishButton.GetComponent<Shadow>();
        shadowSpeakingButton = speakingButton.GetComponent<Shadow>();

        //am Anfang ist Deutsch die Standardssprache, also wird der deutsche Schatten angezeigt.
        shadowGermanButton.enabled = true;
        shadowEnglishButton.enabled = false;
    }
}

```

```

shadowFrenchButton.enabled = false;
shadowSpeakingButton.enabled = false;

//Per Default werden die deutschen Dateien geladen.
fileNameEnding = "_de";
setLanguage = "german";

}

//Beim Klick auf eine Flagge
public void onGermanButton()
{
    //Lege die Dateinamen-Endung fest
    fileNameEnding = "_de";
    Debug.Log(fileNameEnding);

    //setze die entsprechenden Schatten. Die Sprache, die ausgewählt
ist, bekommt einen Schatten.
    shadowGermanButton.enabled = true;
    shadowFrenchButton.enabled = false;
    shadowEnglishButton.enabled = false;

    //schreibe die Sprache in die String-Variable
    setLanguage = "german";

    //Beim Ändern der Einstellung soll der Canvas ausgeblendet werden,
das Einstellungsfenster also verschwinden
    closeWindow.canvasSettings.enabled = false;

    //Direkt nach Änderung der Spracheinstellung müssen die Ausgabefel-
der der Hotspots wieder leer sein, damit sich die Sprachen nicht vermi-
schen.
    TextDescription.gameObject.GetComponent<Text>().text = " ";
    TextHeading.gameObject.GetComponent<Text>().text = " ";
}

public void onFrenchButton()
{
    fileNameEnding = "_fr";
    Debug.Log(fileNameEnding);

    shadowGermanButton.enabled = false;
    shadowFrenchButton.enabled = true;
    shadowEnglishButton.enabled = false;

    setLanguage = "french";

    closeWindow.canvasSettings.enabled = false;

    TextDescription.gameObject.GetComponent<Text>().text = " ";
    TextHeading.gameObject.GetComponent<Text>().text = " ";

}

public void onEnglishButton()
{
    fileNameEnding = "_en";
    Debug.Log(fileNameEnding);
}

```

```

shadowGermanButton.enabled = false;
shadowFrenchButton.enabled = false;
shadowEnglishButton.enabled = true;

setLanguage = "english";

closeWindow.canvasSettings.enabled = false;

TextDescription.gameObject.GetComponent<Text>().text = " ";
TextHeading.gameObject.GetComponent<Text>().text = " ";
}

//Wenn ich auf den Lautsprecher drücke
public void onSpeakerButton()
{
    Debug.Log("Speaker gedrückt");

    //Wenn der Schatten inaktiv ist
    if (shadowSpeakingButton.enabled == false)
    {
        //setze ihn aktiv
        shadowSpeakingButton.enabled = true;

        //blende das Sound-Panel ein
        panelSound.gameObject.SetActive(true);

        TTSManager.Initialize(transform.name, "OnTTSInit");
    }
    else
    {
        shadowSpeakingButton.enabled = false;
        panelSound.gameObject.SetActive(false);
    }

    closeWindow.canvasSettings.enabled = false;
}
}

```

lightIntensity.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class lightIntensity : MonoBehaviour
{
    public float duration = 1.0f;
    public Light lt;

    //2 public int counter = 0; //Zähler, Am Anfang 0
    //2 public int counterend = 20; //Zählerende, nach 5 mal An-

```

```

UND-AUS
    //2 public float waitingTime = 0.5F;    //Zeit zwischen den Blinker

void Start()
{
    lt = GetComponent<Light>();
}

void Update()
{
    //1 IEnumerator Test()
    //1{
        //2 for (counter = 0; counter < counterend; counter++)
        //2 {
            float phi = Time.time / duration * 2 * Mathf.PI;
//Time.time = Zeit nach Start des Spiels
            float amplitude = Mathf.Cos(phi) * 0.5F + 0.5F;
            lt.intensity = amplitude;
            //1 yield return new WaitForSeconds(waitingTime);
            //2 }

            // 1}
        }
    }
}

```

loadXMLFile.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO;
using System.Xml;
using UnityEngine.UI;
using System.Xml.Linq;
using System.Linq;

public class loadXMLFile : MonoBehaviour {

    //Variable, die für das richtige Auslesen der Hotspotttexte aus der XML
    zuständig ist.
    public int i;

    #region Variablendeklaration, um darauf über touchHotspot zuzugreifen
    //Variablen für die Inhalte der XML-Knoten
    public string paintingName;
    public string materials;
    public string date;
    public string width;
    public string hight;
    public string unitHight;
    public string unitWidth;

```

```

public string nationality;
public string hotspot;

public string artistname;
public string lifespan;

public string id;
public string h;
public string p;
public string text;

public string pValue;
#endregion

#region Deklarationen der Ausgabefeldfelder und dafür zusammengestellte
Zeichenketten
//Zeichenketten, die später ins Textfeld geschrieben werden
public string outputArtistName;
public string outputMaterial;
public string outputPaintingName;
public string outputDimension;
public string outputDebugText;

public string outputPtag;

//Da kommt später der ausgelesene Text rein
public Text fieldArtist;
public Text fieldMaterial;
public Text fieldPaintingName;
public Text fieldDimension;
public Text fieldDebugText;
public Text fieldPtag;

#endregion

#region weitere Deklarationen für Flaggenbild und XML-Dateiname
//Da kommt später die XML-File rein. Dateiname wird in "targetData" zu-
sammengesetzt.
public TextAsset xmlRawFile;

//Bild, wo später die Flagge reinsoll
public Image imageNationality;
#endregion

private void Update()
{
    string data = xmlRawFile.text;
    /***ALT!!!!
    TextAsset txtAsset = (TextAsset)Resources.Load(data);           //im-
portiert Text-Datei (xml)
    xmlRawFile = txtAsset;
    */
    Debug.Log("Datei geladen");

    //übergibt den Namen der XML-Datei an die Funktion
    parseXmlFile(data);
}

void parseXmlFile(string xmlData)

```

```

{
    Debug.Log("Funktion parseXmlFile erreicht");

    //Lade die XML-Datei, die den Namen hat wie xmlData (entspricht
"data" aus Z. 70, da sie als Parameter übergeben wird. )
    XmlDocument xmlDoc = new XmlDocument();
    xmlDoc.Load(new StringReader(xmlData));

    /*****
#region für die Kinder des Rootknoten "work"
*****/

    //Pfad, den ich ansprechen will.
    string xmlPathArtistRoot = "/work";

    //wähle alle Knoten aus dem Pfad xmlPathArtistRoot aus und mach
eine Liste draus
    XmlNodeList artistRootInformation = xmlDoc.SelectNodes(xmlPathArt-
istRoot);

    //For-Each-Schleife für jedes Kind von "work"
    foreach (XmlNode arnode in artistRootInformation)
    {
        //lies die Nationalität aus
        nationality = arnode["artist"].GetAttribute("nationality");

        //!!!!!! Muss erweitert werden, wenn neue Nationalität dazu-
kommt! Bilder der Flaggen ablegen und DTD um Nationalitätskürzel (de, fr
etc) erweitern.
        //Wenn der Wert des Attributs in der XML "de" ist,
        if (nationality == "de")
        {
            //lade in das Image-Object die Flagge mit dem Namen "ger-
many"
            imageNationality.sprite = Resources.Load<Sprite>("ger-
many");
        }

        if (nationality == "fr")
        {
            imageNationality.sprite = Resources.Load<Sprite>("france");
        }
    }

    #endregion

    /*****
#region für die Kinder des Knoten "work/painting"
*****/

    string xmlPathPainting = "/work/painting";

```

```

ing);

XmlNodeList paintingInformation = xmlDoc.SelectNodes(xmlPathPaint-
ing);

foreach (XmlNode node in paintingInformation)
{
    //Abspeicherung in die Variablen
    paintingName = node["name"].InnerText;
    materials = node["materials"].InnerText;
    date = node["date"].InnerText;
    width = node["width"].InnerText;
    height = node["height"].InnerText;
    //string unit = node.Attributes["unit"].Value;
    //1 XmlNode width = height.NextSibling;
    unitHeight = node["height"].GetAttribute("unit");
    unitWidth = node["width"].GetAttribute("unit");

    //Ausgabe für die Textfelder
    outputMaterial = materials;
    fieldMaterial.text = outputMaterial;

    outputPaintingName = paintingName + " (" + date + ")";
    fieldPaintingName.text = outputPaintingName;

    outputDimension = height + " " + unitHeight + " x " + width + " "
+ unitWidth;
    fieldDimension.text = outputDimension;
}
#endregion

/*****/
#region für die Kinder des Knoten "work/artist"
/*****/

string xmlPathArtist = "/work/artist";
XmlNodeList artistInformation = xmlDoc.SelectNodes(xmlPathArtist);

foreach (XmlNode anode in artistInformation)
{
    artistname = anode["artistname"].InnerText;
    lifespan = anode["lifespan"].InnerText;

    //Ausgabe ins richtige Textfeld
    outputArtistName = artistname + " (" + lifespan + ")";
    fieldArtist.text = outputArtistName;
}
#endregion

/*****/
#region für die Kinder des Knoten "work/information"
/*****/

string xmlPathInformation = "work/information";
XmlNodeList informationInformation = xmlDoc.SelectNodes(xmlPathIn-
formation); //nimmt alle Knoten von work/information, also alle Hotspots
und Kommentare

```

```

foreach (XmlNode inode in informationInformation)
{
    /*
    if (inode.HasChildNodes)
    {
        int i = 0;
        int sum;
        //Zählt die Anzahl der Hotspots
        do
        {
            sum = i + 1;
            Debug.Log(i);
            i++;

        } while (i < inode.ChildNodes.Count);

        Debug.Log("Anzahl der Hotspots: " + sum);
        /*
        for (int i = 0; i < inode.ChildNodes.Count; i++)
        {

        }

        */
    }
}

*/

}

#endregion

/*****/
#region für die Kinder des Knoten "work/information/hotspot"
/*****/
string xmlPathHotspot = "work/information/hotspot";
XmlNodeList hotspotInformation = xmlDoc.SelectNodes(xmlPathHotspot); //nimmt alle Knoten von work/information/hotspot, also alle Kommentare, h, p, text

foreach (XmlNode hnode in hotspotInformation)
{
    //wenn der Wert des Attributs id gleich ist mit dem i (wird aus touchHotspot übergeben)
    if (hnode.Attributes["id"].Value == i.ToString())
    {

        //alle h-Überschriften
        h = hnode["h"].InnerText;

        //Text zur jeweiligen Überschrift
        text = hnode["text"].InnerText;

        //Debug.Log("Inhalt der Hotspotnummer: " + h + "\n Text: " +
text);
    }
}

```

```

        //string hotspotnummer = inode.SelectSingleNode("work/infor-
        mation/hotspot[@id='1']").InnerText;

    }

    #endregion

}
}

```

menu.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class menu : MonoBehaviour {

    //Deklaration der Canvas und der Button
    public Canvas impressum;
    public Canvas quitMenu;
    public Canvas howTo;

    public Button startText;
    public Button exitText;
    public Button impressumText;
    public Button howToText;

    // Use this for initialization
    void Start () {

        //Zugriff auf die Komponenten (vor allem wegen der Buttons wichtig)
        impressum = impressum.GetComponent<Canvas>();
        quitMenu = quitMenu.GetComponent<Canvas>();
        howTo = howTo.GetComponent<Canvas>();

        startText = startText.GetComponent<Button>();
        exitText = exitText.GetComponent<Button>();
        impressumText = impressumText.GetComponent<Button>();
        howToText = howToText.GetComponent<Button>();

        //Am Anfang sind die Menü-Panelen bzw. Canvases ausgeblendet (Quit
        Menu, Impressum, Anleitung)
        quitMenu.enabled = false;
        impressum.enabled = false;
        howTo.enabled = false;
    }

    //Wenn ich Exit drücke...
    public void ExitPress()
    {
        Debug.Log("Endebutton gedrückt");
    }
}

```

```

        //Menü "Quit" erscheint
        quitMenu.enabled = true;

        //andere Buttons werden inaktiv und können nicht mehr gedrückt wer-
den
        startText.enabled = false;
        exitText.enabled = false;
        impressumText.enabled = false;
        howToText.enabled = false;
    }

    //Wenn ich "Beginnen" drücke..
    public void StartLevel()
    {
        //Lade andere Szene mit Namen "museumsfuehrer"
        Debug.Log("Startbutton gedrückt");
        SceneManager.LoadScene("museumsfuehrer");
    }

    //Wenn ich im QuitMenü "ja" drücke...
    public void ExitGame()
    {
        //Beende Applikation
        Application.Quit();
    }

    //Wenn ich "Impressum" drücke...
    public void ImpressumPress()
    {
        //Impressumspanel erscheint
        impressum.enabled = true;

        //andere Buttons werden inaktiv und können nicht mehr gedrückt wer-
den
        startText.enabled = false;
        exitText.enabled = false;
        impressumText.enabled = false;
        howToText.enabled = false;
    }

    //Wenn ich "Anleitung" drücke...
    public void HowToPress()
    {
        //Anleitungspanel erscheint
        howTo.enabled = true;

        //andere Buttons werden inaktiv und können nicht mehr gedrückt wer-
den
        startText.enabled = false;
        exitText.enabled = false;
        impressumText.enabled = false;
        howToText.enabled = false;
    }

    //Wenn ich im Impressum oder in der Anleitung "Zurück" drücke oder im
Quitmenü "Nein"

```

```

public void BackToMenu()
{
    //Alle Menüs werden ausgeblendet
    impressum.enabled = false;
    howTo.enabled = false;
    quitMenu.enabled = false;

    //Button werden wieder aktiv
    startText.enabled = true;
    exitText.enabled = true;
    impressumText.enabled = true;
    howToText.enabled = true;
}
}

```

openSettingWindow.cs

```

// Activate and deactivate a GameObject.
// The GUI function implements two buttons. These buttons
// toggle the attached GameObject.
// Note that this script needs to be attached to a
// GameObject that isn't disabled.

using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class openSettingWindow : MonoBehaviour
{
    //Deklaration Canvas und Buttons
    public Canvas canvasSettings;
    public Canvas canvasHome;
    public Button homeButton;
    public Button settingButton;

    public void Start()
    {
        //Zugriff auf entsprechenden Komponenten
        canvasHome = canvasHome.GetComponent<Canvas>();
        canvasSettings = canvasSettings.GetComponent<Canvas>();
        homeButton = homeButton.GetComponent<Button>();
        settingButton = settingButton.GetComponent<Button>();

        //Sowohl das Einstellungsfenster als auch das Abfragefenster fürs
        //Beenden sind ausgeblendet
        canvasSettings.enabled = false;
        canvasHome.enabled = false;
    }

    //Wenn man das Zahnrad drückt...
    public void OnSetting()
    {
        //falls das Menu ausgeblendet ist...
        if (canvasSettings.enabled == false)
        {

```

```

        //blende das Menu ein
        canvasSettings.enabled = true;
    }
    else
    {
        //ansonsten blende es aus
        canvasSettings.enabled = false;
    }
}

//Wenn man das Haus drückt... //KEINE IF-ABFRAGE; DER NUTZER SOLL BE-
WUSSTER ÜBER DAS MENÜ DAS PROGRAMM BEENDEN UND "nein" DRÜCKEN
public void OnHome()
{
    //blende das Menü ein
    canvasHome.enabled = true;

    //Mache die Buttons inaktiv
    homeButton.enabled = false;
    settingButton.enabled = false;
}

//Wenn man "nein" drückt...
public void NoPress()
{
    //blende das Menu aus
    canvasHome.enabled = false;

    //Mache die Buttons aktiv
    homeButton.enabled = true;
    settingButton.enabled = true;
}

//wenn man "ja" drückt...
public void YesPress()
{
    //rufe das "Willkommen"-Menu auf
    SceneManager.LoadScene("welcome");
}
}

```

targetData.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO;
using System.Xml;
using UnityEngine.UI;
using Vuforia;

public class targetData : MonoBehaviour
{

```

```

// Deklarationen (später im Inspector festgelegt)
public Transform TextTargetName;
public Transform TextDescription;
public Transform PanelDescription;
public Transform PanelTextTargetName;
//public GameObject ObjectScrollViewObject;

//Damit die Lichter auch ausgehen, wenn das Tracking Objekt fehlt, werden
//alle Lichter bzw. Objekte mit Lichtern in einem Canvas gesammelt.
public Canvas CanvasHotspotCollection;
public string fileNameBase;
public string fileNameEnding;

//ausführliche Deklaration mit Namespace. Ansonsten kommt es zu Ambiguitäten
//zwischen den Namespace von Vuforia und Unity
public UnityEngine.UI.Image portrait;

//Skriptzugriff:
public GameObject XMLLoader;
public loadXMLFile sendFileNameBase;

public GameObject CanvasSettingsNew;
public languageButtons languageSetting;

//Variablen, die ich übersende:
//public TextAsset sentXMLFile;

public string fileEnding;
public string tname;

public string targetname;
public GameObject[] targets;

void Start()
{
    //Zugriff auf Skripte languageButtons und loadxmlfile
    sendFileNameBase = XMLLoader.GetComponent<loadXMLFile>();
    languageSetting = CanvasSettingsNew.GetComponent<languageButtons>();

}

// Update is called once per frame
void Update()
{
    StateManager sm = TrackerManager.Instance.GetStateManager();
    IEnumerable<TrackableBehaviour> tbs = sm.GetActiveTrackableBehaviours();

    foreach (TrackableBehaviour tb in tbs)
    {
        tname = tb.TrackableName;
        ImageTarget it = tb.Trackable as ImageTarget;
    }
}

```

```

Vector2 size = it.GetSize();

Debug.Log("Active image target:" + name + " -size: " + size.x
+ ", " + size.y);

//Evertime the target is found it will show "name of target" on
the TextTargetName. Description and Panel will be visible (active)
//TextTargetName.GetComponent<Text>().text = name;
TextDescription.gameObject.SetActive(true);
PanelDescription.gameObject.SetActive(true);
PanelTextTargetName.gameObject.SetActive(true);
//ObjectScrollViewObject.gameObject.SetActive(true);

/***** KOPIEREN UND EINFÜGEN, WENN NEUES TARGET VOR-
LIEGT! *****/

//Wenn der Name des Targets "dix-otto_die-sieben-todsunden"
ist, dann ....
if (tname == "dix-otto_die-sieben-todsunden")
{
    //Mache den Hotspotcontainer aktiv (erstes Kind vom Objekt
mit dem Namen "ImageTarget-Dix")
    GameObject.Find("ImageTarget-Dix").trans-
form.GetChild(0).transform.gameObject.SetActive(true);

    fileEnding = languageSetting.fileNameEnding;
    string fileName = tname + fileEnding;

    //importiert Text-Datei (xml)
    TextAsset Asset = (TextAsset)Resources.Load(fileName);
    sendFileNameBase.xmlRawFile = Asset;
    //sendFileNameBase.fileName = fileName;

    Debug.Log("File Name aus Target Data: " + fileName);

    //Portrait-Bild mit Namen "dix-otto" wird geladen.
    portrait.sprite = Resources.Load<Sprite>("dix-otto");
}

if (tname == "duererer-albrecht_melancholia-I")
{
    GameObject.Find("ImageTarget-Duerer").trans-
form.GetChild(0).transform.gameObject.SetActive(true);

    fileEnding = languageSetting.fileNameEnding;
    string fileName = tname + fileEnding;

    TextAsset Asset = (TextAsset)Resources.Load(fileName);
    sendFileNameBase.xmlRawFile = Asset;

    Debug.Log("File Name aus Target Data: " + fileName);

    portrait.sprite = Resources.Load<Sprite>("duererer-alb-
recht");
}

if (tname == "lorrain-claude_landschaft-mit-der-anbetung-des-

```

```

goldenen-kalbes")
    {
        GameObject.Find("ImageTarget-Lorraine").transform.GetChild(0).transform.gameObject.SetActive(true);

        fileEnding = languageSetting.fileNameEnding;
        string fileName = tname + fileEnding;

        TextAsset Asset = (TextAsset)Resources.Load(fileName);
        sendFileNameBase.xmlRawFile = Asset;

        Debug.Log("File Name aus Target Data: " + fileName);

        portrait.sprite = Resources.Load<Sprite>("lorrain-claude");
    }
}

void deactivateOtherTargets(string targetname)
{
    targets = GameObject.FindGameObjectsWithTag("Target");
    foreach (GameObject target in targets)
    {
        if (target.name != targetname)
        {
            target.SetActive(false);
        }
    }
}
}
}

```

TextToSpeech.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class TextToSpeech : MonoBehaviour
{
    #region Deklarationen
    //Zugriff auf Script LoadXMLFile
    public GameObject XMLLoader;
    public loadXMLFile getHotspotInfo;

    //Zugriff auf LanguageButton
    public GameObject canvasSetting;
    public languageButtons getLanguage;

    //Deklaration der Buttons im PanelSound und der Textfelder, deren Inhalt ausgespielt wird.

```

```

public Button buttonPlay;
public Button buttonStop;

public Text textfield;
public string textfieldContent;

//Variablen für den TTSManager
private bool _initializeError = false;
private string[] _localeStrings;

#endregion

// Use this for initialization
void Start()
{
    //Zugriff auf "loadXMLFile"
    getHotspotInfo = XMLLoader.GetComponent<loadXMLFile>();

    getLanguage = canvasSetting.GetComponent<languageButtons>();

    //Initialisierung des Text-To-Speech-Plugins
    TTSManager.Initialize(transform.name, "OnTTSInit");
}

private void Update()
{
    /*
    textfieldContent = textfield.text;
    Debug.Log(textfieldContent);
    */
}

//Wenn man den Play-Button drückt...
public void OnPlay()
{
    Debug.Log("Play button pressed");
    Debug.Log("Ausgabe aus TTS (aus xmlloader): " + getHotspotInfo.text);

    //Wenn die Variable "setLanguage" im Skript LanguageButtons auf
    "german" gesetzt ist...
    if (GameObject.Find("CanvasSettingsNew").GetComponent<languageButtons>().setLanguage=="german")
    {
        Debug.Log("Spracheinstellung: Deutsch");
        //Stelle die Ausgabesprache auf GERMAN
        TTSManager.SetLanguage(TTSManager.GERMAN);
    }

    if (GameObject.Find("CanvasSettingsNew").GetComponent<languageButtons>().setLanguage == "french")
    {
        Debug.Log("Spracheinstellung: Französisch");
        TTSManager.SetLanguage(TTSManager.FRENCH);
    }
}

```

```

    }

    if (GameObject.Find("CanvasSettingsNew").GetComponent<languageButtons>().setLanguage == "english")
    {
        Debug.Log("Spracheinstellung: English");
        TTSManager.SetLanguage(TTSManager.ENGLISH);
    }

    //Wenn der TTSManager initialisiert wurde...
    if (TTSManager.IsInitialized())
    {
        //Spricht die Software den Text, der im Skript LoadXMLFile in
        die Variable "text" geschrieben wird
        TTSManager.Speak(getHotspotInfo.text, false, volume: 1);
    }

    else if (_initializeError)
    {
        GUI.contentColor = Color.red;
        GUILayout.Label("Error during initialization of TTS.\nIs it installed on your device?");
    }

}

//Wenn der Stop-Button gedrückt wird
public void OnStop()
{
    //hört der TTSManager auf zu sprechen
    TTSManager.Stop();
}

//Callback-Methode, die bei der Initialisierung aufgerufen wird. (Aus Demo von TTS-Plugin)
void OnTTSInit(string message)
{
    int response = int.Parse(message);

    switch (response)
    {
        case TTSManager.SUCCESS:
            List<TTSManager.Locale> l = TTSManager.GetAvailableLanguages();
            _localeStrings = new string[l.Count];
            for (int i = 0; i < _localeStrings.Length; ++i)
                _localeStrings[i] = l[i].Name;

            break;
        case TTSManager.ERROR:
            _initializeError = true;
            break;
    }
}

}

```

touchHotspot.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class touchHotspot : MonoBehaviour {

    //darein kann das geschaffene Ray gespeichert werden, das durch den
    Touch ausgelöst wird.
    Ray ray;
    //Variable speichert Informationen über das, was getroffen wird
    RaycastHit hit;

    //Deklaration der Textfelder für die Ausgabe
    public Text fieldText;
    public Text fieldHeading;

    //Zugriff auf Script LoadXMLFile
    public GameObject XMLLoader;
    public loadXMLFile getHotspotInformation;

    //Zugriff auf LanguageButton
    public GameObject canvasSettingsNew;
    public languageButtons getLanguage;

    //Zugriff auf Script TargetData
    public GameObject ImageTargetDuerer;
    public targetData getTargetName;

    //Für das Erkennen von Albrecht Dürer
    public GameObject Trigon;

    void Start()
    {
        //Zugriff auf loadXMLFile
        getHotspotInformation = XMLLoader.GetComponent<loadXMLFile>();

        //Die Felder für Hotspotinformationen sind zu Beginn leer
        fieldText.text = " ";
        fieldHeading.text = " ";

        //Zugriff auf languageButtons
        getLanguage = canvasSettingsNew.GetComponent<languageButtons>();

        //Zugriff auf targetData
        getTargetName = ImageTargetDuerer.GetComponent<targetData>();
    }

    void Update ()
```

```

{
    //Bei einem Touch (nur kurzer Touch: Input.Get-
    Touch(0).phase==TouchPhase.Began)
    if (Input.touchCount > 0 )
    {
        //Touchpunkt wird in Ray konvertiert
        ray = Camera.main.ScreenPointToRay(Input.GetTouch(0).position);

        //vom Ausgangspunkt in seine Richtung, 20-fache seiner Länge)
        //Debug.DrawRay(ray.origin, ray.direction*20, Color.red);

        //alle Informationen über das Objekt, das getroffen wurde, wer-
        den in "hit" gespeichert
        if (Physics.Raycast(ray, out hit, Mathf.Infinity))
        {
            Debug.Log("hit something");
            //hitObject = hit.transform.gameObject;

            //Wenn ein Objekt getroffen wurde mit Namen "Hotspot1"...
            if (hit.collider.gameObject.name == "Hotspot1")
            {
                //setze die Variable i in LoadXMLFile auf 1
                getHotspotInformation.i = 1;
                //Debug.Log("Name: " + hit.transform.name + "\n Tag: "
                + hit.transform.tag);

                //Informationen aus LoadXMLFile werden in die Textfel-
                der geschrieben.
                fieldText.text = getHotspotInformation.text;
                fieldHeading.text = getHotspotInformation.h;

                //Wenn das "Großeltern"-Element des Hotspot1 "ImageTar-
                get-Duerer" heißt....
                if (hit.collider.transform.parent.gameObject.trans-
                form.parent.gameObject.name == "ImageTarget-Duerer")
                {
                    //mache das Objekt aktiv
                    Trigon.gameObject.SetActive(true);
                }
            }

            if (hit.collider.gameObject.name == "Hotspot2")
            {
                getHotspotInformation.i = 2;
                //Debug.Log("Name: " + hit.transform.name + "\n Tag: "
                + hit.transform.tag);
                fieldText.text = getHotspotInformation.text;
                fieldHeading.text = getHotspotInformation.h;
            }

            if (hit.collider.gameObject.name == "Hotspot3")
            {
                getHotspotInformation.i = 3;
                //Debug.Log("Name: " + hit.transform.name + "\n Tag: "
                + hit.transform.tag);
                fieldText.text = getHotspotInformation.text;
            }
        }
    }
}

```

```

        fieldHeading.text = getHotspotInformation.h;
    }

    if (hit.collider.gameObject.name == "Hotspot4")
    {
        getHotspotInformation.i = 4;
        //Debug.Log("Name: " + hit.transform.name + "\n Tag: "
+ hit.transform.tag);
        fieldText.text = getHotspotInformation.text;
        fieldHeading.text = getHotspotInformation.h;
    }

    if (hit.collider.gameObject.name == "Hotspot5")
    {
        getHotspotInformation.i = 5;
        //Debug.Log("Name: " + hit.transform.name + "\n Tag: "
+ hit.transform.tag);
        fieldText.text = getHotspotInformation.text;
        fieldHeading.text = getHotspotInformation.h;
    }

    if (hit.collider.gameObject.name == "Hotspot6")
    {
        getHotspotInformation.i = 6;
        //Debug.Log("Name: " + hit.transform.name + "\n Tag: "
+ hit.transform.tag);
        fieldText.text = getHotspotInformation.text;
        fieldHeading.text = getHotspotInformation.h;
    }

    if (hit.collider.gameObject.name == "Hotspot7")
    {
        getHotspotInformation.i = 7;
        //Debug.Log("Name: " + hit.transform.name + "\n Tag: "
+ hit.transform.tag);
        fieldText.text = getHotspotInformation.text;
        fieldHeading.text = getHotspotInformation.h;
    }

    if (hit.collider.gameObject.name == "Hotspot8")
    {
        getHotspotInformation.i = 8;
        //Debug.Log("Name: " + hit.transform.name + "\n Tag: "
+ hit.transform.tag);
        fieldText.text = getHotspotInformation.text;
        fieldHeading.text = getHotspotInformation.h;
    }

    if (hit.collider.gameObject.name == "Hotspot9")
    {
        getHotspotInformation.i = 9;
        //Debug.Log("Name: " + hit.transform.name + "\n Tag: "
+ hit.transform.tag);
        fieldText.text = getHotspotInformation.text;
        fieldHeading.text = getHotspotInformation.h;
    }
}

```

```
        /*
        if (Physics.Raycast(ray, Mathf.Infinity)) //unendilcher Strahl,
der aus dem Ray oben generiert wird. wenn der Strahl sich mit einem Objekt
schneidet, ist die Bedingung wahr
        {
            Debug.Log("hit something");
        }
        */
    }
}
}
```

Fragebogen an die Staatliche Kunsthalle

Fragebogen Staatliche Kunsthalle

Zielgruppe

1) Gibt es Besuchererhebungen, die über die Zielgruppe Auskunft erteilen?

Ja Nein

Könnte ich die Daten sehen?

*morgens Klassen / Schulen
bis 12 sehr viele; Ausstellung
auch viel Erwachsene abhängig*

FALLS DIE ANTWORT NEIN IST: (SCHÄTZUNGEN DER MITARBEITER)

2) Aus welchem Gebiet kommt die Zielgruppe überwiegend?

Region Baden-Württemberg aus Deutschland International *von 4. abhängig*

3) Wie alt ist die Zielgruppe im Schnitt?

unter 12 12-19 20-39 40-59 60-79 80+

*16-30
BSE*

4) Welche Geschlechter gehören überwiegend zur Zielgruppe?

überwiegend Frauen Überwiegend Männer ausgeglichen

tendenziell

5) Welchen Schulabschluss haben die Mitglieder der Zielgruppe durchschnittlich?

Hauptschulabschluss Realschulabschluss Abitur Hochschule/Universität
 Promotion/Professur

6) Über wie viel Vorwissen verfügen die Besucher durchschnittlich über Kunst, Kunstepochen, Künstler, zeitliche Rahmen, Entstehungsgeschichte der Werke etc.?

sehr wenig wenig viel sehr viel keine Angabe

unterschiedlich

7) Worüber würden Sie sich mehr Vorwissen wünschen?

8) Wie schätzen Sie die Einstellung Ihrer Zielgruppe gegenüber Smartphones ein?

Ablehnend eher ablehnend eher aufgeschlossen aufgeschlossen

9) Wie schätzen Sie die Einstellung Ihrer Zielgruppe gegenüber Tablets ein?

Ablehnend eher ablehnend eher aufgeschlossen aufgeschlossen

10) Wie schätzen Sie die Einstellung Ihrer Zielgruppe gegenüber Datenbrillen ein?

Ablehnend eher ablehnend eher aufgeschlossen aufgeschlossen

11) Wie schätzen Sie die Einstellung Ihrer Zielgruppe gegenüber Augmented Reality ein?

Ablehnend eher ablehnend eher aufgeschlossen aufgeschlossen

12) Denken Sie, Ihre Zielgruppe hat schon Erfahrungen mit Augmented Reality gesammelt?

ja nein

13) Welche Reaktion erwarten Sie auf die Tablet-Nutzung mit AR?

"WOW!" Interaktivität kommt gut an, überrascht,
Positiv & begeistert.

Inhalte für die neue App

Folgendes Kunstwerk eignet sich am besten für die prototypische Umsetzung:

"7 Todsünden" von Otto Dix ⚠ Rechte von Otto Dix werden von
der VG Bild-Kunst vertreten

14) Welche Inhalte können Sie zur Verfügung stellen? Videos, Bilder etc.?

Hemmenhofen - Otto Dix Haus
Kunstmuseum Stuttgart ← → Heinrich George
Jazz-Guide - was gibt's wo? → weitere Werke Dix

15) Gibt es Broschüren zum ausgewählten Bild/Künstler? → Living Print eventuell umsetzbar (theoretisch!)

Ja Nein Ausstellungskataloge
könnte genutzt werden

Bisherige App

16) Wie viel Prozent Ihrer Zielgruppe nutzt schätzungsweise die bisherige App?

? Mernberger fragen, → bisher 100-500 Downloads *

17) Wie wurde die bisherige App erstellt?

Fr. Mernberger
mit Präsenz bei Google Arts Projects verknüpft & daraus gespeist

18) Wie häufig werden der App neue Inhalte hinzugefügt?

Fr. Mernberger
nicht sehr häufig, nicht innerhalb des
letzten Jahres.

ICH KANN BEI DER ENTWICKLUNG NICHT DARAUF RÜCKSICHT NEHMEN, OB DIE ZWEI „APP-ARTEN“ MITEINANDER HARMONISIEREN UND AUF EINANDER AUFBAUEN KÖNNEN. DIE APP WIRD UNABHÄNGIG VON DER VORHANDENEN APP GESTALTET UND ERSTELLT.

* Email von Fr. Mernberger (10.04.17; 10:43)

Technische Voraussetzungen

19) Könnte das Museum Tablets zur Verfügung stellen oder müsste jeder Nutzer sein eigenes Tablet mitbringen?

- eigenes Tablet Museum stellt Tablets *→ Wunsch in Zukunft*
↳ Budgetgründe

20) Gibt es in der SKK WLAN oder andere kabellose Netzwerke?

- Ja Nein
KA-WLAN

Rechte

DIE RECHTE DER APP LIEGEN BEI DER HOCHSCHULE KARLSRUHE – TECHNIK UND WIRTSCHAFT. BEI DER GESTALTETEN APP HANDELT ES SICH UM EINEN PROTOTYPEN.

21) Wie darf ich die Materialien der SKK nutzen?

22) Muss wegen der Materialien ein Sperrvermerk in die Thesis?

- Ja Nein

Sonstiges

23) Gibt es Unterlagen über das Corporate Design?

- ja nein
Könnte ich sie einsehen? Eigenes Design entwickeln

*ste-Beauftragte
mehnerberger@kunsthall.
-karlsruhe.de bildunterse
Frutiger Next Pro (Überschr.)
Sabon Oldstyle Figures (Fließ-
text)
Hausfarbe: Rotton
(CMYK) 15-100-100-0*

24) Worauf legt die Kunsthalle wert im Hinblick auf Design?

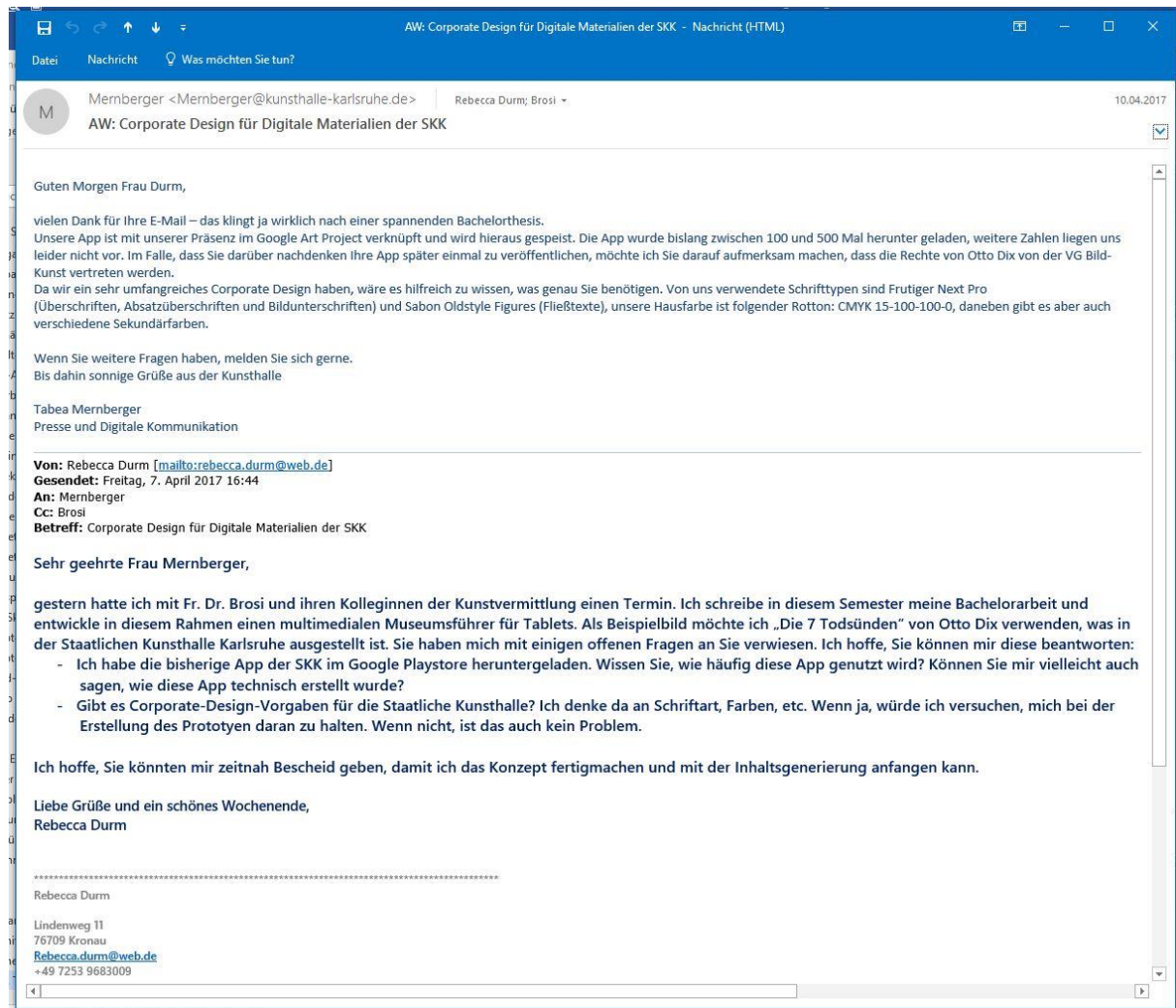
minimalistisch, klar, Raster

25) Was würde sich die Kunsthalle wünschen?

Interessante Bilder für mich

- „Melencolia I“ von Albrecht Dürer *→ nicht ausgestellt*
- „Die 7 Todsünden“ von Otto Dix
- „Die Störung“ von Adolph Menzel
- Die Kreuzigung auf dem Tauberbischofsheimer Altar von Mathias Grünewald
- „Die Anbetung des goldenen Kalbes“ von Claude Lorrain
- „Abtransport der Sphinx“ von Max Beckmann *→ ausgeliehen*

Email von Fr. Tabea Mernberger



AW: Corporate Design für Digitale Materialien der SKK - Nachricht (HTML)

10.04.2017

Memberger <Mernberger@kunsthalle-karlsruhe.de> | Rebecca Durm; Brosi

AW: Corporate Design für Digitale Materialien der SKK

Guten Morgen Frau Durm,

vielen Dank für Ihre E-Mail – das klingt ja wirklich nach einer spannenden Bachelorthesis. Unsere App ist mit unserer Präsenz im Google Art Project verknüpft und wird hieraus gespeist. Die App wurde bislang zwischen 100 und 500 Mal herunter geladen, weitere Zahlen liegen uns leider nicht vor. Im Falle, dass Sie darüber nachdenken Ihre App später einmal zu veröffentlichen, möchte ich Sie darauf aufmerksam machen, dass die Rechte von Otto Dix von der VG Bild-Kunst vertreten werden.

Da wir ein sehr umfangreiches Corporate Design haben, wäre es hilfreich zu wissen, was genau Sie benötigen. Von uns verwendete Schrifttypen sind Frutiger Next Pro (Überschriften, Absatzüberschriften und Bildunterschriften) und Sabon Oldstyle Figures (Fließtexte), unsere Hausfarbe ist folgender Rotton: CMYK 15-100-100-0, daneben gibt es aber auch verschiedene Sekundärfarben.

Wenn Sie weitere Fragen haben, melden Sie sich gerne.
Bis dahin sonnige Grüße aus der Kunsthalle

Tabea Mernberger
Presse und Digitale Kommunikation

Von: Rebecca Durm [<mailto:rebecca.durm@web.de>]
Gesendet: Freitag, 7. April 2017 16:44
An: Mernberger
Cc: Brosi
Betreff: Corporate Design für Digitale Materialien der SKK

Sehr geehrte Frau Mernberger,

gestern hatte ich mit Fr. Dr. Brosi und ihren Kolleginnen der Kunstvermittlung einen Termin. Ich schreibe in diesem Semester meine Bachelorarbeit und entwickle in diesem Rahmen einen multimedialen Museumsführer für Tablets. Als Beispielbild möchte ich „Die 7 Todsünden“ von Otto Dix verwenden, was in der Staatlichen Kunsthalle Karlsruhe ausgestellt ist. Sie haben mich mit einigen offenen Fragen an Sie verwiesen. Ich hoffe, Sie können mir diese beantworten:

- Ich habe die bisherige App der SKK im Google Playstore heruntergeladen. Wissen Sie, wie häufig diese App genutzt wird? Können Sie mir vielleicht auch sagen, wie diese App technisch erstellt wurde?
- Gibt es Corporate-Design-Vorgaben für die Staatliche Kunsthalle? Ich denke da an Schriftart, Farben, etc. Wenn ja, würde ich versuchen, mich bei der Erstellung des Prototypen daran zu halten. Wenn nicht, ist das auch kein Problem.

Ich hoffe, Sie könnten mir zeitnah Bescheid geben, damit ich das Konzept fertigmachen und mit der Inhaltsgenerierung anfangen kann.

Liebe Grüße und ein schönes Wochenende,
Rebecca Durm

Rebecca Durm
Lindenweg 11
76709 Kronau
Rebecca.durm@web.de
+49 7253 9683009