Konvergenz mobiler Endgeräte und öffentlicher interaktiver Installation mithilfe moderner Web-Technologien

Diplomarbeit

Ausgeführt zum Zweck der Erlangung des akademischen Grades
Dipl.-Ing. für technisch-wissenschaftliche Berufe

am Masterstudiengang Digitale Medientechnologien an der Fachhochschule St. Pölten, **Vertiefungsrichtung Mobiles Internet**

von:
Gernot Rottermanner, BSc
dm111539

Betreuer und Erstbegutachter: FH-Prof. DI Dr. Grischa Schmiedl
Zweitbegutachterin: DI Kerstin Blumenstein, BSc

Randegg, 17. August 2014

Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.
- ich dieses Thema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der vom Begutachter bzw. der Begutachterin beurteilten Arbeit überein.

....

Ort, Datum Unterschrift

Kurzfassung

Die vorliegende Diplomarbeit behandelt das Zusammenspiel von mobilen Endgeräten und einer interaktiven Installation im öffentlichen Raum. Dabei beschreibt sie wesentliche Herausforderungen und damit verbundene Lösungsansätze. Denn neben der Wahl eines passenden Standortes müssen vorbeigehende PassantInnen auf die Installation aufmerksam gemacht werden und die Möglichkeiten der Interaktion erfassen. Primäres Ziel soll dann die Nutzung des Systems sein. Strategien dafür werden in dieser Arbeit näher erläutert.

Zahlreiche Beispiele aus der Forschung und von marktreifen Produkten zeigen außerdem, dass das mobile Endgerät nicht nur zur Steuerung, sondern auch zur Informationsanreicherung und -mitnahme einer interaktiven Installation geeignet ist.

Im Rahmen dieser Arbeit wurde ein Prototyp in einem Feldversuch getestet. Mithilfe einer innovativen Projektionstechnik *(3D Projection Mapping)* werden Sehenswürdigkeiten auf dreidimensionale Tetraeder dargestellt. Durch diese neuartige Darstellungsform soll die Neugierde vorbeigehender PassantInnen geweckt werden. Die Sehenswürdigkeiten können dann mithilfe einer mobilen Web-Applikation erraten werden. Beobachtet wurde dabei das Verhalten der PassantInnen vor der Projektion und während des Spiels. Die Erkenntnisse wurden gemeinsam mit einer Online-Umfrage und einer Datenaufzeichnung ausgewertet. Diese zeigen zu den bereits vorhandenen Erkenntnissen in einschlägiger Forschungsliteratur auch neue auf, die im Fazit dieser Arbeit als Empfehlungen zusammengefasst sind. So kam es vor, dass Gruppen bestehend aus zwei Personen oder mehr mit nur einem Endgerät spielten. Außerdem sollen BenutzerInnen beim Spielausstieg ein positives Nutzungserlebnis mitnehmen und so motiviert werden, wieder zu kommen.

Abstract

This master thesis describes the interplay between mobile devices and a public interactive installation (which can be a screen or projection). One aim of this thesis is to illustrate the main challenges concerning this topic. People need to get aware of the installation and recognize that it is interactive. Appropriate incentives have to motivate them to use it. Gamification offers powerful strategies to motivate people to use a system.

Examples in research and development have shown, that the mobile device is an ideal counterpart through its possibility of acting as a remote controller, or of sharing and picking up information from an interactive screen.

Within this thesis, the author and a colleague implemented a prototype of a public interactive screen, which makes use of the technique called “3D projection mapping”. It shows images of sights of St. Pölten. This innovative technique should be used to attract people’s attention to come closer. With their mobile devices they have to guess the right answer. This project was tested within a field test on three different days at two places at St. Pölten. Methods like observation, data logging and online survey were used to collect usage data. The findings are presented at the end of this thesis.

Inhaltsverzeichnis

Ehrenwörtliche Erklärung II

Kurzfassung III

Abstract IV

Inhaltsverzeichnis V

1 Einleitung 8

1.1 Problemstellung 8

1.2 Forschungsfragen 9

1.3 Aufbau der Arbeit 10

1.4 Begriffsdefinition 11

2 Allgemeine Herausforderungen 12

2.1 Phasenmodelle der Interaktion 12

2.2 Aufmerksamkeit und Interaktivität 16

2.2.1 Aufmerksamkeit erzeugen 16

2.2.2 Interaktivität kommunizieren 18

2.3 Motivation 19

2.3.1 Herausforderung & Kontrolle 19

2.3.2 Neugier & Exploration 20

2.3.3 Auswahl 21

2.3.4 Kollaboration 21

2.4 Gamification zur Steigerung der Motivation 22

2.4.1 Punkte 22

2.4.2 Ranglisten 23

2.4.3 Abzeichen 23

2.4.4 Levels 24

2.4.5 Challenges und Quests 24

2.4.6 Onboarding 24

2.4.7 Social Engagement Loops 25

3 Anwendungsbeispiele 26

3.1 Personalisierung 26

3.2 Interaktion 27

3.2.1 Positionierung 27

3.2.2 Selektierung (Auswahl) 30

3.2.3 Weitere Möglichkeiten der Interaktion 32

3.3 Co-Displays und Cyberforaging 32

3.4 Information Take-Away 34

4 Technologiebewertung 36

4.1 Alternativen zur händischen URL-Eingabe 36

4.1.1 Quick-Response (QR) Code 37

4.1.2 Near Field Communication (NFC) 39

4.2 Echtzeitkommunikation 42

4.2.1 WebSocket-Protokoll und API 43

4.2.2 Node.js und Socket.IO 46

4.3 Medienzugriff 49

4.3.1 HTML Media Capture 50

4.3.2 WebRTC 51

5 Prototypische interaktive Installation für den öffentlichen Raum 53

5.1 Aufbau 53

5.2 Spielprinzip 54

5.2.1 Web-Anwendung für PassantInnen 54

5.2.2 Web-Anwendung für AdministratorIn 57

5.2.3 Argumentation für dieses Konzept 58

5.3 Mögliche Einsatzszenarien für diese Installation 59

5.3.1 Szenario 1: Tourismus 60

5.3.2 Szenario 2: Werbung 60

5.4 Technische Umsetzung 61

5.4.1 Mobile Web-Applikation 62

5.4.2 Node.js-Server 66

5.4.3 Projektion 70

5.5 Feldversuch 70

5.5.1 Aufbau 70

5.5.2 Fragestellung und Methodik 71

5.5.3 Testablauf 73

5.5.4 Ergebnisse 73

6 Fazit 78

7 Ausblick 82

8 Literaturverzeichnis 83

9 Abbildungsverzeichnis 87

10 Tabellenverzeichnis 89

11 Listingverzeichnis 90

12 Anhang 92

1 Einleitung

„[…] public displays should form the backbone of a new communications medium with the same potential impact on society as radio, television, and the Internet.“ (Davies et al. 2012, S. 58).

1.1 Problemstellung

Davies et al. (2012) beschreiben in ihrem Paper „Open Display Networks, A Communications Medium for the 21st Century“ eine Vision: ein interaktives Bildschirm-Netzwerk im öffentlichen Raum, das für Anwendungen und Inhalte aus unterschiedlichen Quellen offen ist und die vielen kleinen, geschlossenen Systeme ablöst. Nach Ansicht der Autoren könnte diese Entwicklung dann ähnliche Bedeutung wie das Radio, Fernsehen oder das Internet erlangen (vgl. Davies et al. 2012, S. 58). Denn es ist eine Analogie zu der Entwicklung im Bereich des Mobilfunks erkennbar: In den 1990er- und frühen 2000er-Jahren waren mobile Endgeräte bezüglich Funktionalität und Applikationen geschlossene Systeme, kontrolliert durch die Hersteller und Netzbetreiber. Durch die Bereitstellung von Programmierschnittstellen und Distributionskanälen, ist es Dritt-EntwicklerInnen möglich, eigene Anwendungen zu entwickeln und zu vertreiben (vgl. Davies et al. 2012, S. 58). Die Öffnung dieses Ökosystems führte zu einem Durchbruch der Smartphones.

Durch die Vision eines offenen Bildschirm-Netzwerkes ist es nach Meinung des Autors dieser Diplomarbeit sinnvoll, für die Erstellung der Anwendungen Web-Technologien (HTML, CSS, JavaScript) zu verwenden. Sie bestehen aus einer Reihe von offenen Standards, die plattformunabhängig einsetzbar sind. Die BenutzerInnen benötigen nur einen Browser zur Nutzung einer Applikation.

Das Smartphone ist in diesem Kontext nicht nur zur Interaktion, sondern auch zur Informationsanreicherung und -mitnahme für interaktive Bildschirme im öffentlichen Raum geeignet. Das Gerät ist universell einsetzbar, hat einen geringen Formfaktor und wird ständig mitgetragen. Google befragte gemeinsam mit Ipsos MediaCT im ersten Quartal 2013 tausend Personen in Österreich im Alter von 18 bis 64 Jahren online nach deren Smartphone-Nutzung. Im Besitz eines Smartphones sind 48 Prozent der Befragten. Im Segment der 18- bis 24-Jährigen

ist diese Zahl allerdings weit höher und liegt bereits bei 85 Prozent. 71 Prozent aller Smartphone-BesitzerInnen gaben an, dass sie nicht mehr ohne das eigene Smartphone aus dem Haus gehen. Rund 80 % nutzen es bei einem Besuch in einem Café, Restaurant, in einem Geschäft und in öffentlichen Verkehrsmitteln (vgl. Google Inc. 2013, S. 1ff).

All diesen Überlegungen stehen aber auch Probleme gegenüber. Zwar existieren im öffentlichen Raum bereits viele Bildschirme. Sie werden aber zumeist für Werbung, selten auch zur Darstellung allgemeiner Nachrichten eingesetzt. PassantInnen haben sich so sehr an deren geringe Nützlichkeit gewöhnt, dass diese Bildschirme ignoriert werden (vgl. Davies et al. 2012, S. 58). Außerdem sind Bildschirme meist nicht interaktiv, sondern strahlen lediglich mediale Inhalte aus („broadcasting“). Es stellt daher eine große Herausforderung dar, PassantInnen zur Nutzung einer interaktiven Installation im öffentlichen Raum zu motivieren.

Des Weiteren kann der Einsatz von Web-Technologien auch Nachteile mit sich bringen. Im Gegensatz zu nativen Anwendungen ist über Web-Technologien der Zugriff auf Hardware bei mobilen Endgeräten nicht immer möglich. Außerdem ist die Standardisierung (etwa für JavaScript APIs[1], HTML5) noch nicht abgeschlossen, was zu unterschiedlichen Implementierungen bei Web-Browsern führen kann.

Der Autor dieser Diplomarbeit hat zum Ziel, die genannten Probleme und Herausforderungen näher zu behandeln und aufzuarbeiten.

1.2 Forschungsfragen

Zusammenfassend ergeben sich aus der Problemstellung folgende Forschungsfragen, die in der Arbeit untersucht und beantwortet werden sollen:

- Welche allgemeinen Herausforderungen und Anforderungen gilt es zu beachten, wenn eine interaktive Installation für den öffentlichen Raum entwickelt werden soll?
- Welche Herausforderungen und Anforderungen, aber auch Empfehlungen ergeben sich aus den für diese Arbeit durchgeführten Feldversuch?
- Für welche Zwecke ist ein Zusammenspiel von mobilen Endgeräten und öffentlicher Installation sinnvoll? Welche Beispiele dafür gibt es bereits in der Forschung und als marktreife Produkte?

[1] Abkürzung für Application Programming Interface

- Die händische Eingabe einer Internetadresse *(Uniform Resource Locator,* kurz *URL)* kann fehleranfällig sein. Welche Alternativen zur händischen Eingabe gibt es?
- Worin liegen die technischen Möglichkeiten und Einschränkungen, wenn moderne Web-Technologien in diesem Kontext eingesetzt werden?

1.3 Aufbau der Arbeit

In Kapitel 2 widmet sich der Autor den allgemeinen Herausforderungen interaktiver Installationen im öffentlichen Raum. Die Herausforderungen liegen etwa darin, vorbeigehende PassantInnen neugierig zu machen und sie zur Nutzung zu motivieren. Dafür spielen psychologische Aspekte im Kontext der Mensch-Computer-Interaktion eine Rolle, die es zu verstehen und zu beachten gilt. In diesem Zusammenhang wird auch das Thema „Gamification“ näher erläutert.

Anschließend werden in Kapitel 3 Anwendungsbeispiele – also Beispiele, wie mobile Endgeräte mit einer interaktiven Installation zusammenarbeiten können – beschrieben. Kategorisierungen sind in der Literatur bereits vorhanden, diese werden hier näher behandelt. Die genannten Beispiele setzen sich aus dem Bereich der wissenschaftlichen Forschung sowie aus marktreifen Produkten zusammen.

Die anschließende Technologiebewertung in Kapitel 4 gibt einen Überblick über die Einsatzmöglichkeiten moderner Web-Technologien im thematischen Kontext dieser Arbeit. Zu Beginn werden technische Alternativen zur händischen Eingabe einer URL in einem Web-Browser beschrieben. Das Kapitel zeigt dann hauptsächlich aber Möglichkeiten, Chancen und Limitationen im Einsatz moderner Web-Technologien auf. Dabei werden Themen wie Echtzeitkommunikation oder der Zugriff auf Hardware näher erläutert.

In Kapitel 5 wird eine prototypische Anwendung, die gemeinsam mit einem Kollegen des Autors konzipiert und entwickelt wurde, erläutert. Neben einer Beschreibung der Funktionalität, des Aufbaus und der technischen Umsetzung werden der durchgeführte Feldversuch und dessen Erkenntnisse erläutert.

Das Fazit in Kapitel 6 fasst die Erkenntnisse zusammen und gibt Empfehlungen für den Einsatz interaktiver Installationen im öffentlichen Raum.

Abschließend wird in Kapitel 7 ein Ausblick auf weitere Tätigkeiten des Autors dieser Diplomarbeit im thematischen Kontext, aber auch auf andere Forschungsprojekte in diesem Bereich gegeben.

1.4 Begriffsdefinition

Öffentlicher Raum

In der Norm EN ISO 9241-11 wird der Begriff „Nutzungskontext“ definiert durch die Parameter „BenutzerInnen“, „Aktivität“ (Aufgabe, Ziel), „Arbeitsmittel“ (Hard- und Software) und die „physische und soziale Umgebung“. Nagel und Fischer (2013) fassen den Begriff „physische und soziale Umgebung“ in „Umfeld“ zusammen und teilen diesen in mögliche Nutzungsräume auf: *privater, halb-öffentlicher, öffentlicher Raum* und *Unterwegs*. Der private Raum bezieht sich dabei auf die eigenen vier Wände, aber auch auf Räumlichkeiten, „zu denen man selbst einen exklusiven Zutritt hat (z.B. Hotelzimmer im Urlaub).“ (Nagel & Fischer 2013, S. 96). Zutritt zu diesen Räumen haben meist nur ausgewählte Personen wie FreundInnen, Bekannte und Verwandte. Unter dem Begriff *halb-öffentlicher Raum* verstehen die Autoren eine nur für eine eingeschränkte Gruppe von Personen zugängliche Räumlichkeit, wie etwa das Büro. Der eigene Arbeitsplatz kann aber auch genauso gut als privater Raum verstanden werden. Der öffentliche Raum dagegen ist für jede Person zugänglich. So befindet man sich beim Einkaufen, bei Behörden, öffentlichen Einrichtungen und Plätzen, bei Einkaufsstraßen in einem öffentlichen Umfeld (vgl. Nagel & Fischer 2013, S. 96f).

Interaktive Installation im öffentlichen Raum

In der Literatur beschäftigen sich unterschiedlichste Disziplinen – Architektur, Sozialwissenschaften, Hard- und Softwareentwicklung, um nur einige wenige zu nennen – mit dem Thema „interaktive Installation im öffentlichen Raum“. Dem zur Folge sind vielfältige Begriffe (Medienfassade, Medieninstallation, Public Display, Urban Screen, etc.) in der Literatur vorzufinden. Laut dem Autor dieser Arbeit hat der Begriff „öffentliche interaktive Installation“ folgende Charakteristika: Kernelement dieser Installation bildet die Informationsanzeige (das kann ein (Groß-)Bildschirm oder eine Projektion mittels Beamer sein), die öffentlich für eine große Anzahl an Personen zugänglich und ersichtlich ist. PassantInnen können mittels persönlichem mobilen Endgerät über einen Web-Browser Elemente und Inhalte auf der Informationsanzeige steuern oder beeinflussen, aber auch „mitnehmen“ (wie im weiteren Verlauf der Arbeit genauer beschrieben wird). Die Interaktion ist also sowohl einzeln (*Single User*) also auch gleichzeitig von mehreren Personen möglich (*Multi User*).

Dementsprechend sind Kiosksysteme aller Art und Ausprägung für diese Arbeit nicht relevant und werden nicht berücksichtigt.

2 Allgemeine Herausforderungen

Bei der Entwicklung einer interaktiven Installation für den öffentlichen Raum gilt es einige Herausforderungen zu beachten. Solch eine Anwendung steht in Konkurrenz mit anderen umliegenden Objekten. Sie muss erst von vorbeigehenden PassantInnen erkannt werden. Müller et al. (2010) und Müller et al. (2012) behandeln in ihren Forschungsarbeiten drei Herausforderungen: (1) PassantInnen müssen zuerst die Installation wahrnehmen und (2) erkennen, dass sie interaktiv ist. Anschließend müssen sie (3) motivierende Anreize erhalten, die Installation zu nutzen.

In diesem Zusammenhang haben sich einige ForscherInnen mit der Frage beschäftigt, welche Phasen PassantInnen vom ersten Sichtkontakt der Installation bis hin zur Nutzung durchlaufen und daraus ableitend Interaktionsphasen definiert. Im folgenden Kapitel werden die drei bekanntesten Arbeiten (in Anlehnung an die Recherchearbeit von Michelis (2009)) näher beschrieben. Anschließend werden die drei genannten Herausforderungen und Ansätze zur Bewältigung dieser näher erläutert.

2.1 Phasenmodelle der Interaktion

Brignull und Rogers (2003) gehen in ihrer Arbeit der Frage nach, wie sich Personen eigentlich vor einem öffentlichen interaktiven Display verhalten, welche Phasen der Interaktion erkennbar sind und vor allem auch, welche Ursachen zu Verunsicherung und Verlegenheit in der Nutzung führen können. Dafür wurden zwei Tests in geschlossenen Räumen – bei einer Buchpräsentation auf der CHI-Konferenz 2002 und auf einer Welcome Party einer Universität – durchgeführt. Auf einer Projektion war eine thematisch relevante Frage abgebildet, die die anwesenden Gäste zum Kommentieren einladen sollte. Das war über die Tastatur des Laptops, der mit dem Beamer verbunden war, möglich. Es konnte also immer nur eine Person mit der Installation direkt interagieren. Die AutorInnen identifizieren auf Basis der Erkenntnisse aus Beobachtung und Frageboden drei Phasen der Interaktion (vgl. Brignull & Rogers 2003, S. 2-6) (siehe Abbildung 1):

- **aufmerksamkeitsperiphere Aktivitäten:** Die Projektion wird von den anwesenden Personen nur peripher, also am Rande wahrgenommen. Die Haupt-Aufmerksamkeit der Personen gilt der Kommunikation untereinander.
- **aufmerksamkeitsfokussierte Aktivitäten:** Die Personen wenden sich der Projektion zu, tauschen sich untereinander aus und beobachten die aktiv

mit dem System interagierende Person. Durch die Bildung einer „Menschentraube“ rund um die Projektion wird auch das Interesse derjenigen geweckt, die noch nicht die Projektion gesichtet haben. Diesen Effekt nennen die AutorInnen „*honey-pot effect*“.

- **direkte Interaktion:** Die Person interagiert direkt mit dem Laptop und verfasst einen Kommentar.

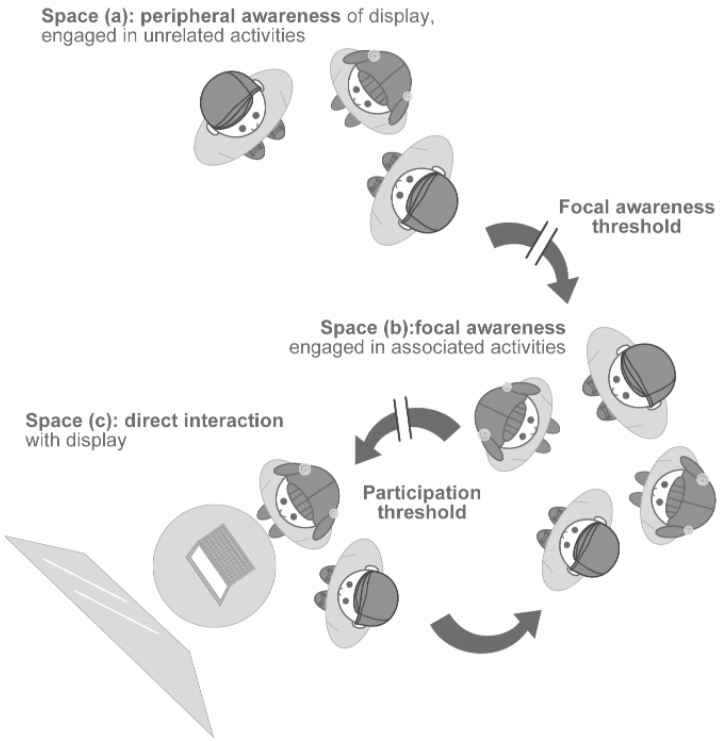

Abbildung 1: Phasen der Aktivitäten bei öffentlichen Displays nach Brignull und Rogers (Quelle: Brignull und Rogers 2003, S. 7).

Ziel dieser Anwendung ist es also, dass Personen direkt damit interagieren. Dazu müssen laut Modell der AutorInnen zwei Phasenübergänge bewältigt werden. Personen, die die Projektion nur peripher wahrnehmen, sollen innerhalb sehr kurzer Zeit grob erfassen können, welchen Zweck die Projektion erfüllt. Wird das Interesse geweckt und die Projektion näher betrachtet, sollen ein paar wesentliche aufkommende Fragen durch einfache Beobachtung beantwortet werden können, um etwaigen peinlichen Situationen zu entgehen: Wie lange wird die Interaktion dauern? Was ist das Ergebnis der Interaktion? Welche Schritte sind notwendig, um das Ergebnis zu erreichen? Wird die Benutzung angenehm sein? Ist der Abbruch der Interaktion jederzeit möglich? Um diese Fragen beantworten zu können, ist eine Beobachtung aus sicherer Entfernung notwendig. Damit durch Beobachtung diese Fragen beantwortet werden können, muss überhaupt erst einmal eine Person diesen ersten Schritt wagen, die Installation zu nutzen. Bei den Tests stellte sich das als Problem heraus. Hier empfehlen die AutorInnen den

Einsatz einer helfenden Person, die die anwesenden Gäste zur Interaktion motiviert und beim Umgang behilflich ist (vgl. Brignull & Rogers 2003, S. 6ff).

Nachdem nur ein Eingabegerät zum Kommentieren zur Verfügung stand, konnte die direkte Interaktion auch immer nur von einer Person durchgeführt werden. In einer begleiteten Umfrage gaben einige TeilnehmerInnen an, dass diese Tatsache Unbehagen auslöste, da sie sich bei der direkten Interaktion beobachtet fühlten und einem gewissen Druck ausgesetzt waren. Da die Eingaben in Echtzeit auf die Leinwand projiziert wurden, konnte jeder Rechtschreibfehler von den anwesenden Personen mitverfolgt werden. Laut Aussage der AutorInnen könnte dem entgegengewirkt werden, wenn die Eingabe auch per Mobiltelefon möglich ist, da die gesamte Aufmerksamkeit nicht mehr auf eine Person gerichtet ist. Das hat aber zur Folge, dass der Vorteil des *honey-pot effect* nicht optimal ausgenutzt werden kann (vgl. Brignull & Rogers 2003, S. 5ff).

Während sich die Arbeit von Brignull und Rogers (2003) rein mit der Verhaltensbeobachtung anwesender Personen beschäftigt und daraus die einzelnen Interaktionsphasen abgeleitet wurden, geben Streitz et al. (2003) in ihrer Arbeit diese Phasen selbst vor. Sie bedienen sich dabei dem Konzept kontextsensitiver Systeme, die auf die Umgebung reagieren und im Kontext dessen unterschiedliche Informationen anzeigen können. Im Zentrum ihres Test-Aufbaus steht ein Ambient Display in einer Größe von 1,80 x 2 Metern, bestehend aus einem LED-Cluster mit insgesamt 124 LEDs. Die Informationen werden also in abstrakter Form dargestellt, können aber mit einem eigens entwickelten Mobilgerät decodiert werden.

Die Autoren definieren drei Interaktionszonen (vgl. Streitz et al. 2003, S. 2ff) (siehe Abbildung 2):

- **Ambient Zone (Umgebungszone):** Eingebaute Sensorik im Ambient Display ermöglicht es, vorbeigehende Personen zu erkennen und darauf zu reagieren. Befindet sich eine Person in der Umgebungszone, werden allgemeine Informationen angezeigt. Diese Zone ist noch außerhalb des wahrnehmbaren Bereiches der Sensoren.
- **Notification Zone (Mitteilungszone):** Wechselt eine Person in die Mitteilungszone, wird sie von den Sensoren erkannt und es können relevante und spezifische Information angezeigt werden. In dieser Zone interagiert die Person bereits, wenn auch nur implizit. Die Person kann über das mitgeführte Mobilgerät die Informationen decodieren.
- **Interaction Zone (Interaktionszone):** Ist eine Person in der dritten und letzten Zone angelangt, kann sie auch direkt eingreifen und eigene Informationen publizieren. Das ist möglich, indem das mitgetragene Mobilgerät, welches mit einem RFID-Chip ausgestattet ist, auf eines der

LEDs gehalten wird. Dafür ist ein sehr geringer Abstand zum Ambient Display von Nöten.

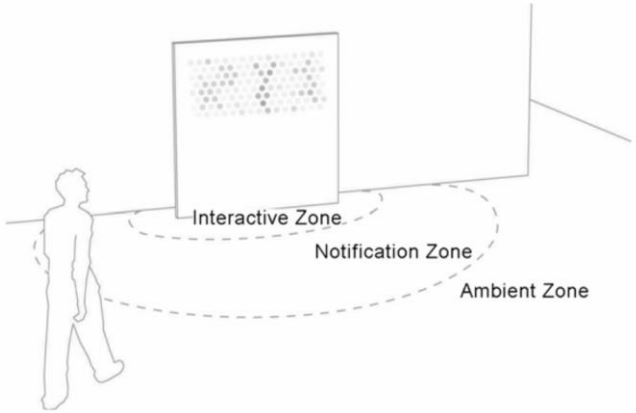

Abbildung 2: Drei Interaktionszonen (Quelle: Streitz et al. 2003, S. 3)

Vogel und Balakrishnan (2004) greifen die Interaktionsphasen nach Streitz et al. (2003) auf und verfeinern diese. Die dritte Phase (Interaktionszone) wird aufgeteilt in *Subtle Interaction* und *Personal Interaction*. Die *Notification Zone* wird umgewandelt in *Implicit Interaction Phase*. Der von den Autoren umgesetzte Prototyp ermöglicht sowohl implizite als auch explizite Interaktion und wurde für eine Büroumgebung konzipiert. Im Zentrum steht ein großer Bildschirm, der aktuelle Informationen zu Wetter, personalisierte Termine und Nachrichten anzeigen kann. Wie die Personalisierung genau erfolgen soll, lassen die Autoren in ihrem Prototyp offen, erwähnen aber mögliche Lösungsansätze wie Gesichtserkennung oder die Anmeldung über eine RFID-Karte. Die Interaktionsphasen sind wie folgt definiert (vgl. Vogel & Balakrishnan 2004, S. 139-144) (siehe Abbildung 3):

- **Ambient Display Phase:** Vorbeigehende Personen werden in dieser Phase beginnen. Sie werden noch nicht die Absicht haben, sich den Bildschirm näher anzusehen, sondern lediglich einen kurzen Blick darauf werfen. Daher ist es in dieser Phase wichtig, dass vorbeigehende Personen die notwendigsten Informationen erfassen können.
- **Implicit Interaction Phase:** Wurde das Interesse einer vorbeigehenden Person geweckt und betrachtet sie den Bildschirm länger, soll das System darauf reagieren. Der genannte Prototyp verwendet ein Kamera-Tracking System, das gewisse Haltungen – Blick der Person richtet sich auf den Bildschirm, der Körper ist dem Bildschirm zugewendet – erkennt und darauf reagiert. Je nach Position und Nähe zum Bildschirm erscheint ein vertikal zum Bildschirm ausgerichteter, halb-transparenter Balken, der auf Bewegungen der Person reagiert. Die Interaktion passiert implizit und soll

der Person vermitteln, dass mit dem Bildschirm interagiert werden kann. Außerdem soll dadurch Neugierde geweckt werden.

- **Subtle Interaction Phase:** Erreicht die Person diese Phase, ist explizite Interaktion durch Gestensteuerung mit dem Bildschirm möglich. Im Falle des Prototypen erweitert sich der vertikale Balken. Die interagierende Person kann einzelne Informationen auswählen und näher betrachten.
- **Personal Interaction Phase:** Tritt die Person noch näher an den Bildschirm, ist eine direkte Interaktion mittels Touch-Gesten möglich. Durch die geringe Distanz zum Bildschirm können persönliche Informationen (der nächste Termin) angezeigt werden. Wendet sich die Person dem Bildschirm ab, verschwindet die personalisierte Information auch gleich wieder.

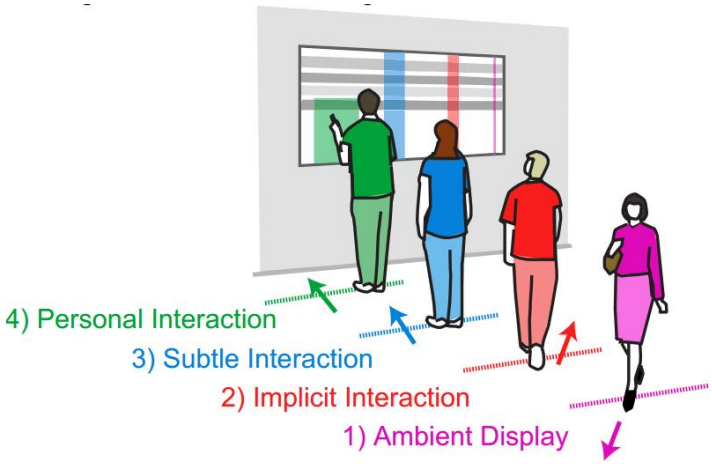

Abbildung 3: Interaktionsmodell nach Vogel und Balakrishnan (Quelle: Vogel & Balakrishnan 2004, S. 3).

2.2 Aufmerksamkeit und Interaktivität

2.2.1 Aufmerksamkeit erzeugen

Das menschliche Gehirn ist nicht in der Lage, alle eintreffenden Reize der Umwelt gleichzeitig zu interpretieren und zu verarbeiten. Ganz im Gegenteil, es ist einer ständigen Reizüberflutung ausgesetzt. Es muss daher entscheiden, welches der eintreffenden Informationen die volle Aufmerksamkeit geschenkt werden soll. In diesem Zusammenhang wird in der Psychologie der Begriff *Spotlight-Metapher* verwendet. Dabei richtet das Gehirn die visuelle Aufmerksamkeit einer gewissen sichtbaren Region für die genauere Informationsverarbeitung aus. Generell gesehen wird Aufmerksamkeit durch *Bottom-Up-* (Beispiel: das plötzliche Erscheinen einer Fehlermeldung) und *Top-Down-Prozesse* (Beispiel: Person sucht bewusst nach einem Wort in einer bestimmten Farbe) beeinflusst (vgl. Müller et al. 2010, S. 1286).

PassantInnen im öffentlichen Raum werden in erster Linie nicht nach interaktiven Installationen suchen, sondern ein ganz anderes Ziel verfolgen. Unter ForscherInnen herrscht daher Uneinigkeit darüber, wie viel Aufmerksamkeit eine interaktive Installation erregen soll. Einerseits wird argumentiert, dass sie unaufdringlich sein soll (vgl. Müller et al. 2010, S. 1286). Marc Weiser, ein ehemaliger Wissenschaftler im *Palo Alto Research Center (PARC)*, der den Begriff „Ubiquitous Computing" als Begriff für die dritte Generation des Computerzeitalters bekannt machte, spricht in diesem Zusammenhang von *Calm Computing*. Die zugrundeliegende Philosophie dahinter ist, dass die steigende Anzahl an Computersystemen dazu führen soll, dass sie im Alltag immer mehr verschwinden, in den Hintergrund und nur wenn notwendig in den Vordergrund treten sollen (vgl. Weiser und Brown 1996, o.S.). Demgegenüber steht aber die Ansicht, dass ein Computersystem zum Beteiligen animieren und vorbeigehende PassantInnen involvieren soll (vgl. Müller et al. 2010, S. 1286).

Das abrupte Erscheinen neuer Objekte kann Aufmerksamkeit erregen. Außerdem zeigen Ju und Sirkin (2010) in einem Feldversuch auf, dass ein physisches Objekt mehr Aufmerksamkeit erregt, als das gleiche Objekt in digitaler Form. Dazu wurde ein Arm angefertigt und rechts oben neben dem Bildschirm platziert (Variante A). In einer anderen Variante wurde der winkende Arm auf einem Bildschirm dargestellt (Variante B) (siehe Abbildung 4). Die Auswertung ergab, dass doppelt so viele Personen mit dem Bildschirm mit Variante A interagierten, als mit Variante B (vgl. Ju & Sirkin 2010, S. 9).

Abbildung 4: Vergleich physisches und digitales Objekt nach Ju und Sirkin (Quelle: Ju & Sirkin 2010, S. 4)

Der *honey-pot effect* wurde bereits in Kapitel 2.1 erwähnt. Er kann hohe Aufmerksamkeit erregen. Wenn es bereits eine Ansammlung von Personen vor einem Objekt gibt, ist es sehr wahrscheinlich, dass die in der Nähe befindlichen Personen schnell ihre Aufmerksamkeit auf die Menschenansammlung richten und näher treten werden (vgl. Müller et al. 2010, S. 1286).

2.2.2 Interaktivität kommunizieren

Müller et al. (2012) haben auf Basis einer Literaturrecherche sechs bekannte Techniken zur Anzeige von Interaktivität für öffentliche Bildschirme identifiziert (vgl. Müller et al. 2012, S. 298):

- **Call-To-Action:** Über ein Icon oder einen Text wird hingewiesen, dass das System interaktiv ist. Dabei kann es zu Problemen kommen, wenn nur eines von beiden eingesetzt wird. Es gibt wenige standardisierte Icons, daher kann es zu Unklarheiten in der Bedeutung kommen. Im schlimmsten Fall sind sich PassantInnen nicht bewusst, dass die Anwendung interaktiv ist. Bedford (2014) empfiehlt daher, bei Icons auch immer ein Text-Label zu verwenden: "A user's understanding of an icon is based on previous experience. Due to the absence of a standard usage for most icons, text labels are necessary to communicate the meaning and reduce ambiguity" (Bedford 2014, o.S.). Braucht es außerdem länger als fünf Sekunden, die Bedeutung des Icons zu erfassen, ist es nicht effizient und daher ungeeignet.
- **Attract Loop:** Darunter ist ein Video zu verstehen, dass in einer Endlosschleife abgespielt wird. Dabei wird die Funktionsweise des Systems vorgezeigt. Diese Technik findet häufig bei Spielautomaten Verwendung, wenn gerade keine Person spielt.
- **Analog Signage:** Eine Hinweistafel, die die Funktionsweise erläutert, wird in der Nähe der Installation aufgestellt.
- **honey-pot effect** (siehe Kapitel 2.1)**:** Wenn bereits Personen interagieren, können ZuseherInnen die Funktionsweise beobachten.
- **Person lädt Andere zur Interaktion ein:** Eine Person animiert nach Beendigung der Interaktion Bekannte und Freunde zur Nutzung des Systems. Zusätzlich wird eine helfende Person eingesetzt, die PassantInnen zur Nutzung einlädt und die Funktionsweise erklärt.
- **Vorwissen:** Die Funktionsweise des Systems ist der Person aus vergangenen Erfahrungen bereits bekannt.

In zwei Beispielen zu Interaktionsphasen in Kapitel 2.1 wurde bereits der Begriff implizite Interaktion erwähnt. Müller et al. (2012) sprechen in ihrer Forschungsarbeit auch von *unbeabsichtigter Interaktion*. Eine Person interagiert also zufällig mit einer Anwendung und schließt daraus, dass sie interaktiv ist. In einem Feldversuch haben eben genannte Autoren untersucht, inwiefern die Darstellung von Spiegelbildern der vorbeigehenden Personen als implizite, beziehungsweise unbeabsichtigte Interaktion geeignet ist. Dabei konnten PassantInnen an drei Standorten in Berlin mit eigenen Körperbewegungen Fußbälle am Bildschirm steuern. Die Forscher verwendeten unterschiedliche Darstellungsformen des Spiegelbildes: Das Spiegelbild im Original (mit einer

Hintergrund-Extraktion, sodass nur der Körper der Person sichtbar ist), eine Silhouette davon und kein Spiegelbild. Außerdem wurde beim Feldversuch zwischen rein impliziter Interaktion und Call-To-Action – Ein Hinweis "Step Close to Play" und ein Demo-Video einer spielenden Person wurde am Bildschirm dargestellt – gewechselt. Insgesamt wurden 502 Interaktionen registriert. Es stellte sich heraus, dass das Spiegelbild eine effektive Möglichkeit der impliziten Interaktion ist und zur Anzeige von Interaktivität sehr gut geeignet ist. Im Test war sie auch effektiver als die Variante „Call-To-Action“ (150 vs. 79 Interaktionen) (vgl. Müller et al. 2012, S. 302). Die Beobachtung zeigte auch, dass vorbeigehende Personen oft erst spät die Interaktivität des Bildschirmes wahrnahmen. Nachdem sie bereits am Bildschirm vorbeigegangen waren, gingen sie wieder ein paar Schritte zurück. Die Autoren nennen es *Landing Effekt*. Beim Design eines Systems sollte dieser Effekt beachtet werden, damit Personen Zeit haben, die Interaktivität auch rechtzeitig zu erkennen (vgl. Müller et al. 2012, S. 306).

2.3 Motivation

PassantInnen sollen dazu gebracht werden, eine interaktive Installation (bestenfalls wiederkehrend) zu nutzen. Dafür müssen Incentives, also Anreize geboten werden (vgl. Müller et al. 2010, S. 1287). In der Psychologie wird dabei zwischen *extrinsischer* und *intrinsischer Motivation* unterschieden. *Intrinsische Motivation* zeichnet sich dadurch aus, eine Sache aus eigenem Antrieb zu machen, während *extrinsische Motivation* auf äußeren Antrieben wie Geld oder anderen Belohnungen basiert. Intrinsisch motiviertes Verhalten hat also nicht das Ziel, eine Belohnung zu erhalten, es ist die Tätigkeit selbst, die motiviert (vgl. Michelis 2009, S. 69f).

Auf Basis der Arbeit von Thomas W. Malone, einem der Vorreiter in der Untersuchung motivierendes Verhalten in der Mensch-Computer-Interaktion beschreibt Michelis (2009) in seiner Dissertation fünf Kernfaktoren für intrinsisch motivierendes Verhalten. Vier davon werden in dieser Arbeit näher erläutert: *Herausforderung & Kontrolle*, *Neugier & Exploration*, *Auswahl* und *Kollaboration*.

2.3.1 Herausforderung & Kontrolle

Allgemein betrachtet streben Menschen nach einem optimalen Maß an Herausforderung. Bei einer Unterforderung suchen sie nach anderen, spannenderen Tätigkeiten. Bei Überforderung suchen sich nach Tätigkeiten, denen sie besser gewachsen sind. Das trifft auch bei der Nutzung von digitalen Anwendungen zu. Dabei gilt es laut Malone zu unterscheiden, ob es sich bei der Anwendung um ein Werkzeug oder Spielzeug handelt. Unter Spielzeugen versteht Malone "Anwendungen, die ihrer selbst wegen genutzt werden und kein externes

Ziel haben. Werkzeuge hingegen sind Anwendungen, die als Mittel genutzt werden, um ein externes Ziel zu erreichen." (Michelis 2009, S. 85). Aus diesem Grund ist die Herausforderung bei einem Werkzeug möglichst gering zu halten. Es soll zuverlässig, effizient und einfach zu bedienen sein. Das Spielzeug hingegen soll bewusst schwierig gehalten sein.

Um die Motivation zu erhalten, soll auch Raum für feste und emergente Ziele sein. Feste Ziele werden meist von der Anwendung vorgegeben, emergente Ziele setzt sich die spielende Person selbst (vgl. Michelis 2009, S. 87). Das Erreichen der festen Ziele soll nicht immer vorhersehbar, aber der Weg dorthin nachvollziehbar sein. Ist eine Reaktion der Anwendung auf eine Aktion der spielenden Person nicht nachvollziehbar, geht das Kontrollgefühl und damit auch das Interesse verloren (vgl. Michelis 2009, S.87).

Sind die genannten Anforderungen gegeben, kann es zu einer optimalen Herausforderung kommen. Der Psychologie-Professor Mihaly Csikszentmihalyi spricht von einer Flow-Zone, der optimalen Zone zwischen Angst und Langeweile unter Berücksichtigung der eigenen Fähigkeiten und dem Grad der Herausforderung über einen längeren Zeitraum (vgl. Zichermann & Cunningham 2011, S. 16ff) (siehe Abbildung 5).

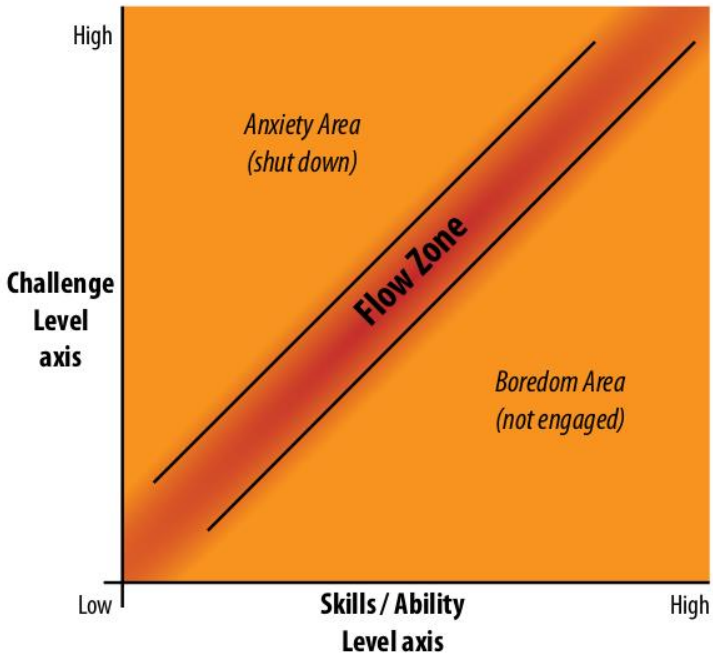

Abbildung 5: Grafische Darstellung der Flow-Zone (Quelle: Zichermann & Cunningham 2011, S. 18).

2.3.2 Neugier & Exploration

Neugier ist ein wesentlicher Antrieb für intrinsisch motiviertes Verhalten: "Neugier wird durch neuartige und unbekannte Stimuli in der Umgebung hervorgerufen, die

für die Person etwas Unklares, Unvollendetes oder Unsicheres darstellen. Das motivierende Element ist die Suche nach einer Erklärung." (Michelis 2009, S. 93). Im Kontext der Mensch-Computer Interaktion ist es wichtig, dass die Interaktion nicht zu komplex, aber auch nicht zu einfach ist. Damit bleibt die Neugierde erhalten. Daher macht es Sinn, Interaktionsmöglichkeiten einzusetzen, die überraschend, aber keinesfalls unverständlich sind. Auf Basis bereits vorhandener Erfahrungen sollten NutzerInnen erste Erwartungen haben, diese sollten aber nur zum Teil erfüllt werden. Für eine öffentliche Installation kann es daher hilfreich sein, nicht alle Inhalte (etwa nur den Titel einer Nachricht und nicht den gesamten Text) preiszugeben, sondern durch Unvollständigkeit die Neugierde zu wecken, damit BenutzerInnen dann die weiteren Informationen erkunden können. Hilfreich ist auch der Einsatz von Animation, Farbe und Klang, welche ebenfalls die Neugierde anregen können (vgl. Michelis 2009, S. 94-97).

2.3.3 Auswahl

Auswahlmöglichkeiten können ebenfalls eine motivierende Wirkung in der Mensch-Computer Interaktion haben, wie bereits in mehreren Forschungsarbeiten aufgezeigt wurde. So können bereits sehr einfache Auswahlmöglichkeiten wie die Wahl eines Spiel-Avatars oder die Eingabe des Namens zu einer Zunahme an intrinsischer Motivation führen (vgl. Michelis 2009, S. 99). Auswahl steht auch in einem Wechselspiel zu eben genannten Motivationsfaktoren wie zum Beispiel "Kontrolle". Können bei einem Spiel Parameter wie der Schwierigkeitsgrad, die Steuerung oder Farben verändert werden, vermittelt das wieder ein Gefühl der Kontrolle. Es zeigt sich aber auch, dass die Wahlmöglichkeiten nicht zu umfangreich sein sollen, da die Motivation dadurch wieder abnehmen kann (vgl. Michelis 2009, S. 101).

2.3.4 Kollaboration

Die Entwicklung des Internet erlaubt es Personen, über digitalem Wege mit anderen Personen sozialen Aktivitäten nachzugehen. Viele digitale Anwendungen sind mit sozialen Komponenten gekoppelt (beispielsweise das Teilen von Inhalten mit Freunden, um sich Meinungen einzuholen). Kollaboration kann dabei in Form von Wettbewerb oder Kooperation stattfinden. Öffentliche Installationen werden attraktiver, wenn Mechanismen zur Kollaboration vorhanden sind. So kann man sich bei einer Aufgabe gegenseitig messen. Man kann aber auch zusammenarbeiten, um ein Ziel schneller und effektiver zu erreichen. Mit solch einer Handlung kann die Anerkennung gestärkt werden, in dem man besser als eine andere Person ist, oder wesentlich zur Erreichung eines Ziels beigetragen hat. Für eine interaktive Installation im öffentlichen Raum ist es daher notwendig, im ersten Schritt auf jeden Fall mehrspielerfähig zu sein. Im zweiten Schritt sollten aber auch Möglichkeiten zum Wettbewerb, der Kollaboration oder beides geboten

werden, welche die Nutzungsmotivation der PassantInnen erhöhen werden (vgl. Michelis 2009, S. 106-110).

2.4 Gamification zur Steigerung der Motivation

Nicht immer wirkt eine Anwendung auf den ersten Blick motivierend, daher „kann der Einsatz von spielerischen Elementen sehr nützlich sein, um die Motivation zu fördern und den Spaß an der Interaktion zu steigern." (Michelis 2009, S. 86). In diesem Zusammenhang hat der Begriff „Gamification" in den letzten Jahren hohe Bedeutung erlangt. Darunter ist der Einsatz von bekannten und bewährten Spielelementen in einem spielfremden Kontext zu verstehen. Zichermann und Cunningham (2011) listen in ihrem Buch „Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps" sieben wichtige Spielelemente auf: *Points*, *Levels*, *Leaderboards* (Ranglisten), *Badges* (Abzeichen), *Challenges* (Wettbewerbe), *Onboarding* („Neue BenutzerInnen an Bord holen") und Social Engagement Loops (vgl. Zichermann & Cunningham 2011, S. 36). Die Elemente werden an dieser Stelle kurz beschrieben, da sie auch für interaktive Anwendungen im öffentlichen Raum wertvoll sein können.

2.4.1 Punkte

Im Kontext der Gamification ist die Vergabe von Punkten ein essentielles und wichtiges Instrument zur Erhaltung und Steigerung der Motivation[2]. Selbst die Anzahl der FreundInnen bei Facebook oder die Anzahl von „Follower" bei Twitter gleicht einem Punktestand. So wissen viele der Mitglieder bei Facebook die Anzahl der FreundInnen in der Freundesliste und auch darüber Bescheid, wer besonders wenige oder viele FreundInnen aufweisen kann (vgl. Zichermann & Cunningham 2011, S. 37). Zichermann und Cunningham (2011) unterscheiden zwischen folgenden unterschiedlichen Arten von Punktesystemen (vgl. Zichermann & Cunningham 2011, S. 38ff):

- **Experience Points (Erfahrungspunkte):** Diese Punkte können an BenutzerInnen für verschiedene Tätigkeiten vergeben werden, wo sie die Anwendung kennen lernen und dabei Erfahrung sammeln. Diese Punkte können niemals wieder weniger werden, können nicht eingelöst werden und erreichen nie ein Maximum.

[2] Ein Beispiel dazu bietet NikeFuel: Sämtliche sportliche Aktivitäten werden mittels eigens entwickelter Hardware wie ein Armband aufgezeichnet und in Punkte umgewandelt. Durch selbst gesteckte Ziele (die Erreichung eines gewissen Punktestandes) bleibt die Motivation über einen längeren Zeitraum aufrecht. Mehr Informationen zu diesem Beispiel gibt es unter: https://secure-nikeplus.nike.com/plus/what_is_fuel/

- **Redeemable Points (einlösbare Punkte):** Wie der Name bereits verrät, können diese Punkte gegen Sonderkonditionen oder Objekte eingelöst werden. Sie werden häufig bei Kundenbindungsprogrammen eingesetzt, wo KundInnen die gesammelten Punkte gegen ein Produkt eintauschen können.
- **Skill points (Punkte für besondere Fähigkeiten):** Diese Punkte bleiben Personen vorbehalten, die durch besondere Leistungen oder Tätigkeiten auffallen. Zichermann und Cunningham (2011) nennen hier folgendes Beispiel: Bei einer Photo-Sharing Webseite könnten Punkte dafür vergeben werden, wenn die Qualität der Bilder besonders hochwertig ist.
- **Karma points:** „The main purpose of karma points in your design is to create a behavioral path for altruism and user reward." (Zichermann & Cunningham 2011, S. 40). Karma-Punkte haben zum Beispiel den Zweck, an andere Personen weitervergeben zu werden, um Anerkennung auszudrücken. Eine Anwendung kann Karma-Punkte auch für selbstloses Verhalten vergeben. Bei einer interaktiven Installation könnte das System Karma-Punkte an Personen vergeben, die eine weitere Person zur Nutzung motiviert haben und beim Einstieg behilflich waren.

2.4.2 Ranglisten

Mittels Ranglisten sind Leistungen vergleichbar. Sie sind weitreichend bekannt und bedürfen unter den BenutzerInnen keiner besonderen Erklärung. Ranglisten sollen zum Verbessern und Weitermachen anspornen und dürfen auf keinen Fall entmutigen. Daher macht es bei schlechten oder mittelmäßigen Platzierungen Sinn, den Rang einer Person mittig zu platzieren und naheliegende Plätze darunter und darüber anzuzeigen. So sieht die Person die unmittelbaren KontrahentInnen, die ihr einerseits auf den Fersen und andererseits in erreichbaren Plätzen davor angesiedelt sind. Diese Ansicht soll zum Weitermachen motivieren. Eine andere Strategie ist, Ranglisten thematisch zu gruppieren: So könnten in einer Anwendung nur die Platzierungen unter FreundInnen angezeigt werden. Des Weiteren können Ranglisten nach Regionalität oder Schwierigkeitsstufe dargestellt werden (vgl. Zichermann & Cunningham 2011, S. 50f).

2.4.3 Abzeichen

Abzeichen haben bereits eine sehr lange Tradition, sie signalisieren einen gewissen Status. So geben sie beim Militär den Dienstgrad und damit den Status innerhalb der Organisation wieder. Alle Autohersteller platzieren ihr Unternehmenslogo auf deren Produkte. Dadurch wird die Preisklasse und die damit verbundene Qualität signalisiert. Im digitalen Zeitalter haben Abzeichen ebenso Einzug gefunden. So vergibt der Online-Dienst *Foursquare* Badges für

besonders häufige oder regelmäßige Check-Ins in Lokalitäten. Mittlerweile haben dieses System viele weitere Online-Dienste adaptiert.

Abzeichen zeigen nicht nur einen gewissen Status an, sie markieren erreichte Ziele und den Fortschritt innerhalb eines laufenden Prozesses. Dabei ist eine gute Mischung aus überraschenden und vorhersehbaren Abzeichen sinnvoll, um so die Motivation aufrecht zu erhalten. Somit können mehrere Spieltypen bedient werden: Während die eine Person das Ziel hat, möglichst viele zu sammeln, freut sich die andere über ein überraschend erhaltenes Abzeichen (vgl. Zichermann & Cunningham 2011, S. 55-58).

2.4.4 Levels

Levels sind ein klassisches Element bei Spielen. Auch bei anderen Anwendungen, kann es hilfreich sein, dass der Schwierigkeitsgrad etwas gesteigert wird, um so die Motivation zu erhalten. Es ist sinnvoll, vorgegebene Ziele zu stecken, die BenutzerInnen erreichen können und sollen, damit sich dann der Schwierigkeitsgrad steigert. Dieser Grad verläuft dabei nicht linear, sondern kurvenförmig. Typische Beispiele dafür sind etwa Online-Lern-Portale, die den Grad steigern, um so die Herausforderung hoch zu halten. Die Autoren nennen zudem einen Fortschrittsbalken als Beispiel zum Erreichen eines Levels. Facebook nutzt dieses System, um die Mitglieder zu motivieren, persönliche Informationen einzugeben. Ein guter Fortschrittsbalken erreicht dabei niemals 100 Prozent (vgl. Zichermann & Cunningham 2011, S. 45-48).

2.4.5 Challenges und Quests

Kurze Aufgaben für Zwischendurch können die Nutzungsmotivation hoch halten. Sie können in unterschiedlicher Ausprägung eingesetzt werden. Die Luxus-Marke Jimmy Choo hat einst einen Wettbewerb auf Foursquare gestartet: Personen, die es schafften, zur richtigen Zeit in eines der Geschäfte einzuchecken, konnten eines von nur sechs Paar produzierten Schuhen gewinnen. Diese Exklusivität war ein zusätzlicher Motivationsfaktor, bei dieser Aufgabe mitzumachen (vgl. Zichermann & Cunningham 2011, S. 64-67).

2.4.6 Onboarding

Ein wichtiger Grundsatz ist, dass Neuankömmlinge nicht gleich in den ersten Momenten des Spieleintritts wieder verloren gehen. Denn in der ersten Minute des Eintritts werden die meisten Entscheidungen getroffen, vor allem auch, ob es sich lohnt, die Anwendung weiter zu nutzen. Die Prämisse lautet: "Train and engage, but don't overwhelm." (Zichermann & Cunningham 2011, S. 59). Dabei sollen folgende Aspekte beachtet werden (vgl. Zichermann & Cunningham 2011, S. 60ff):

- Eine Registrierung gleich zu Beginn soll vermieden werden. Die Person soll das System erkunden können. Erst im notwendigsten Fall soll eine Registrierung angeboten werden. Eine Ausnahme stellen hier Dienste dar, die gleich zu Beginn Berechtigungen benötigen, die nur über einen Login abzuwickeln sind.
- Die erste Interaktion mit der Anwendung darf nicht fehlschlagen. Das kann im schlechtesten Fall zu Frustration und zum schnellen Verlassen der Anwendung führen. Sinnvoll ist, ein Tutorial-Level anzubieten, wo es nur eine Möglichkeit zur Interaktion gibt, die nicht fehlschlagen kann.
- Personen im Online-Zeitalter sind beschäftigt und wollen nicht aufgehalten werden. Sie lesen generell keine langen Instruktionen. Schon gar nicht, wenn sie die Anwendung noch nicht kennen und wissen, ob es Sinn macht, diese zu nutzen.
- Die Komplexität der Anwendung soll nur langsam gesteigert werden. Da es sich bei Neuankömmlingen um AnfängerInnen handelt, werden sie zu Beginn nur einen Bruchteil der Funktionalität benötigen.
- Es kann sehr wertvoll sein, Neuankömmlingen einfache Fragen zu stellen, um sie kennenzulernen. Netflix, ein Online-Video Portal stellt zu Beginn ein paar Fragen zu Interessen. Auf Basis dessen kann die Anwendung passende Medieninhalte bereitstellen.

2.4.7 Social Engagement Loops

Als EntwicklerIn einer öffentlichen Installation wird man das Ziel verfolgen, wiederkehrende BesucherInnen zu gewinnen. Die Frage stellt sich daher: Mit welchem Gefühl verlassen sie die Anwendung? Was kann die Motivation sein, die Anwendung wieder zu nutzen?

Ein einfaches Beispiel mit dem Kurznachrichten-Dienst *Twitter* soll die Technik des *Social Engagement Loops* erklären, der sich aus vier Teilen *Motivating Emotion, Player reengagement, Social call to action und Visible progress/reward* zusammensetzt:. Eine Person, die diese Anwendung zum ersten Mal nutzt, wird das Ziel haben, sich mit anderen Leuten zu verbinden und die eigene Meinung auszudrücken *(Motivating Emotion)*. Nachdem sie einen Tweet geschrieben hat, verlässt sie womöglich die Anwendung wieder. Es kann aber passieren, dass eine andere Person sie in einem Tweet erwähnt (@mention) *(Player reengagement)*. Sie wird wieder zur Anwendung zurückkehren und antworten. Das wiederrum lockt weitere Personen an, die die Kommunikation interessant und wertvoll finden, und folgen ihr (*Visible progress/reward*). Das wiederrum motiviert sie, weitere Tweets zu verfassen *(Social call to action)* (vgl. Zichermann und Cunningham 2011, S. 68f).

3 Anwendungsbeispiele

Mobile Endgeräte und interaktive Installation können in unterschiedlicher Weise zusammenspielen. Clinch (2013) teilt die Anwendungsmöglichkeiten in vier Kategorien ein: *Personalisierung*, *Interaktion*, *Co-Display & Cyberforaging* und *Information Take-Away* (Informationsmitnahme). Das folgende Kapitel beschreibt die Begrifflichkeiten näher und erläutert Beispiele aus Forschung und von marktreifen Anwendungen.

3.1 Personalisierung

Zeigt ein öffentlicher Bildschirm personalisierten Inhalt, erhöht das die Aufmerksamkeit der vorbeigehenden PassantInnen. Personalisierter Inhalt steht aber im Spannungsfeld mit Aspekten der Privatsphäre. Problematisch ist in diesem Zusammenhang, wenn unpassender oder enthüllender Inhalt angezeigt, oder permanent die Position der BenutzerInnen zum Zweck der Personalisierung mitgezeichnet wird (vgl. Clinch 2013, S. 92).

Davies et al. (2012) haben in einem Proof-Of-Concept die Beispielanwendung *Tactica* entwickelt. Sie erlaubt anonyme, personalisierte Interaktion mit einem öffentlichen Display. Die Autoren skizzieren dabei ein offenes Bildschirm-Netzwerk: Die EigentümerInnen stellen die Bildschirm-Infrastruktur für verifizierte Anwendungen von Dritt-AnbieterInnen zur Verfügung. Über eine öffentlich einsehbare Karte, die alle Bildschirme und deren unterstützte Anwendungen zeigt, können PassantInnen selbst entscheiden, welches Display personalisierten Inhalt auf Basis der angegebenen Präferenzen zeigen soll. Diese werden über die zugehörige *Tactica* Smartphone Anwendung angegeben. Will ein Benutzer/ eine Benutzerin das aktuelle Wetter eines bestimmten Ortes auf einem öffentlichen Display angezeigt bekommen, wählt er/sie diese Funktion über die Smartphone-Anwendung aus. Die Anwendung informiert dann auf Basis der GPS-Position automatisch den nächstgelegenen öffentlichen Bildschirm, dass die Wetterinformationen angezeigt werden sollen. Zum Schutz der Privatsphäre zeigt die Bildschirm-Anwendung das Wetter für drei Orte an: Für den Standort, an dem sich der Bildschirm befindet, für einen zufällig ausgewählten Standort und für den Ort, den der Benutzer/die Benutzerin angegeben hat (vgl. Davies et al. 2012, S. 60f).

Tactica ist ein gutes Beispiel impliziter Interaktion. Die Entwickler lassen aber die Frage offen, wie das System mit mehreren BenutzerInnen umgeht, die gleichzeitig unterschiedliche Informationen auf einem Bildschirm anfordern.

3.2 Interaktion

Das Smartphone eignet sich gut als Steuerungsgerät für große öffentliche Installationen. Ballagas et al. (2006) haben in ihrer Arbeit die Taxonomie von Foley et al. (1984) aufgegriffen und für den Kontext, das Smartphone als ubiquitäres Steuerungsgerät zu verwenden, angepasst. Es kann dabei mehrere Funktionen übernehmen, die im Folgenden näher erläutert werden.

3.2.1 Positionierung

Über das Smartphone kann ein Objekt im zwei-dimensionalen Koordinatenraum verschoben werden. Dabei wird zwischen kontinuierlicher (Veränderung der Position in „Echtzeit“) und diskreter Positionierung (Veränderung der Position am Ende der Aktion) unterschieden. Positionierungsaufgaben können über mehrere Wege durchgeführt werden:

3.2.1.1 Joystick und Tasten

Auch wenn mittlerweile Tasten und ein Joystick bei modernen mobilen Endgeräten kaum mehr Verwendung finden, sind sie für Positionierungsaufgaben gut geeignet. Im Projekt Blinkenlight[3] konnten PassantInnen auf einer Medienfassade in Berlin das Arcade-Game Pong spielen. Jedes Fenster des Gebäudes stellte ein Pixel dar. Um spielen zu können, musste eine Telefonnummer angerufen werden. Um den Balken nach oben zu bewegen, tippte man während des aktiven Gespräches die Taste „5“. Beim Tippen der Taste „8“ bewegt sich der Balken nach unten (vgl. Ballagas et al. 2006, S. 71).

Viele ältere mobile Endgeräte waren mit einem analogen Joystick ausgestattet, der immer in die Ursprungsposition zurückkehrt, wenn er ausgelassen wird („return-to-zero joystick“). Dieser eignet sich zum Beispiel für die Steuerung eines Cursors am zweiten Bildschirm. Zusätzlich ist es möglich, die Geschwindigkeit der Bewegung über den Joystick zu beeinflussen (vgl. Ballagas et al. 2006, S. 71).

3.2.1.2 Trackpad und Touchscreen

Zwar ist der Begriff *Trackpad* schon etwas veraltet, dennoch sind alle modernen mobilen Endgeräte mit einem berührungsempfindlichen Bildschirm ausgestattet, welcher sich zum Beispiel ähnlich dem Trackpad eines Laptops zur Bedienung eines Cursors auf einem öffentlichen Display eignet.

[3] Mehr Informationen zum Projekt auf der Webseite: http://blinkenlights.net/blinkenlights. Das Projekt lief insgesamt 23 Wochen und fünf Tage und startete am 11. September 2001.

Bei der interaktiven Installation *Tentacles*[4] von Shea und Longford (2011) nimmt das Smartphone die Funktion des Remote Controllers ein. Die native Applikation ist für iOS- sowie Android-Geräte entwickelt worden. Beim Start der Applikation bekommen die BenutzerInnen ein sogenanntes *Tentakel*, ein kleines Lebewesen, zugewiesen, dass vom Smartphone aus dann gesteuert werden kann. Mit einer einfachen Geste kann dem eigenen Tentakel die Richtung vorgegeben werden, wo es hinschwimmen soll (siehe Abbildung 6). Des Weiteren gibt es sogenannte Tenticules. Jedes Mal, wenn das eigene Tentakel damit in Berührung kommt, wächst es. Im Gegensatz dazu gibt es die Urchins, die das Tentakel bei Berührung schrumpfen lassen. Verlässt ein/e BenutzerIn das Spiel, wird das Tentakel zum *Tenticules* und so zu Nahrung für andere Tentakel. Außerdem gibt es sogenannte *Tentacle Toys*, mit denen man spielen kann. Kommt man mit ihnen in Berührung, werden zusätzlich Klänge erzeugt. Die Installation ist so konzipiert, dass der Ein- und Ausstieg jederzeit möglich ist. Jegliche spieltypischen Elemente wie Level oder Highscores fehlen (vgl. Shea & Longford 2011, S. 1-4).

Abbildung 6: Die Smartphone-Anwendung (links) und die Installation „Tentacles“ (rechts) (Quelle: www.tentacles.ca).

3.2.1.3 Accelerometer

Ein Accelerometer (Beschleunigungssensor) ist in den meisten derzeit am Markt verfügbaren Smartphones eingebaut. Damit kann etwa die Orientierung des mobilen Endgerätes ermittelt werden und ist somit auch für Positionierungsaufgaben interessant. Durch Bewegungen kann ein Cursor auf einem entfernten Bildschirm gesteuert werden (vgl. Boring et al. 2009, S. 163) (siehe Abbildung 7).

[4] Ein Video ist auf der Projektwebseite zu finden, hier ist auch die Steuerung über das Smartphone zu sehen: http://www.tentacles.ca/

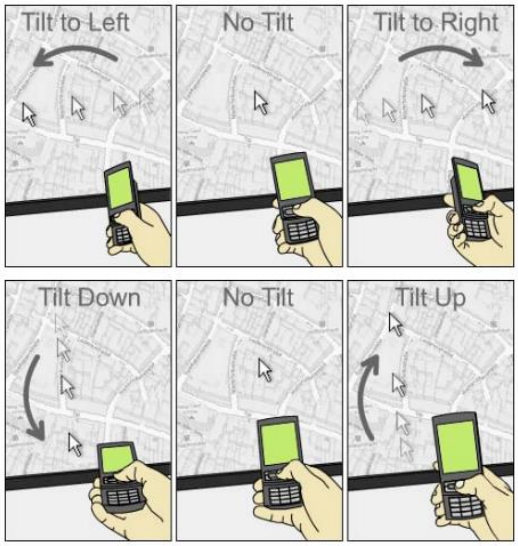

Abbildung 7: Anwendung im Beispiel „Tilt“ (Quelle: Boring et al 2009, S. 163).

3.2.1.4 Kamera-basierte Steuerung

In der Forschungsarbeit von Madhavapeddy et al. (2004) wird das Kamerabild des mobilen Endgerätes verarbeitet, um Bedienelemente (in der Forschungsarbeit *Tags* genannt) wie Slider oder Drehknöpfe auf einem großen Bildschirm zu steuern. Die Person richtet ihr Mobilgerät mit der Kamera-Anwendung auf das gewünschte Bedienelement. Die Anwendung decodiert die Bildinformation und initialisiert eine Bluetooth-Verbindung zu dem PC, der am großen Bildschirm angeschlossen ist (Abbildung 8). Der Datenstrom, der den Tag-Namen sowie X- und Y- Koordinaten, Größe und Orientierung enthält, wird permanent an den PC übermittelt, der wiederum die Anzeige am großen Bildschirm aktualisiert. Die Datenverarbeitung findet also am mobilen Endgerät statt, was sehr ressourcenintensiv sein kann (vgl. Madhavapeddy et al. 2004, S. 1f).

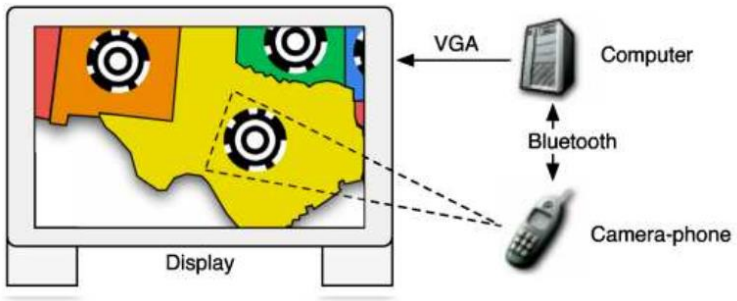

Abbildung 8: Funktionsweise der Anwendung von Madhavapeddy et al. (2004) (Quelle: Madhavapeddy et al. 2004, S. 1).

In einem anderen Anwendungsbeispiel wird die Bildverarbeitung nicht vom mobilen Endgerät sondern von einem entfernten Rechner übernommen. Die Installation *Moving Shapes* kommt ganz ohne mobiler Webseite oder nativer App

aus, hier wird nur die Smartphone-Kamera benötigt. Auf einer Projektion sind verschiedene geometrische Formen zu sehen. Der/Die SpielerIn sucht sich eine der Formen aus, fotografiert diese mit dem Smartphone ab und zeigt das Foto am Gerät an (siehe Abbildung 9, Punkt 2). Anschließend wird das Smartphone umgedreht, sodass das Smartphone-Display zur Projektion gerichtet ist. Bewegt nun der/die SpielerIn das Smartphone, bewegt sich das ausgesuchte Objekt auf der Projektion mit. Möglich ist dies durch Farb-Tracking einer handelsüblichen Webcam, die oberhalb der Projektion angebracht ist. So wird die aktuelle Position des Smartphones relativ zur Projektion ermittelt und das Objekt entsprechend auf der Projektion verschoben. Ziel ist es, möglichst viele automatisiert generierte Kreise, die sich auf der Projektion bewegen, einzufangen. Der- oder diejenige mit den meisten eingefangenen Kreisen liegt in Führung und kann Spieltempo und Hintergrundmusik durch Handbewegungen beeinflussen. Technisch gesehen kommt das Java-Framework *Processing* zum Einsatz (vgl. Matthies 2013, S. 558ff).

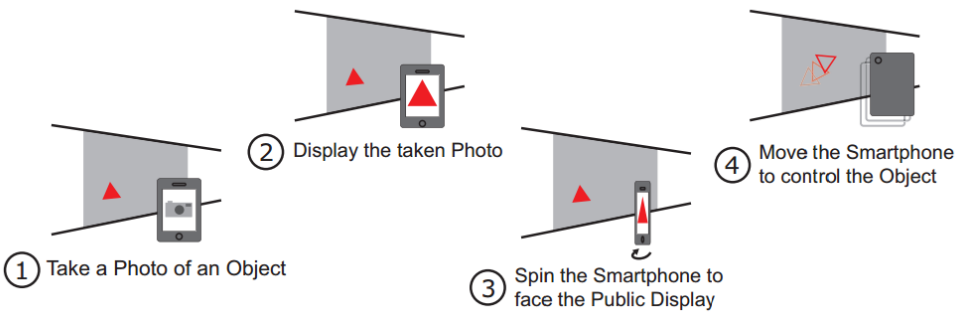

Abbildung 9: Die Funktionsweise von „Moving Shapes“ (Quelle: Matthies et al. 2013, S.559)

3.2.2 Selektierung (Auswahl)

Es kann notwendig sein, Elemente am entfernten Bildschirm auszuwählen. Das kann zum Beispiel über Buttons oder über die Eingabe eines gewissen Befehls am mobilen Endgerät erfolgen. Die Autoren nennen aber noch weitere Optionen, Elemente direkt am entfernten Bildschirm zu selektieren (vgl. Ballagas et al. 2006, S. 73):

3.2.2.1 Direkte Auswahl via Kamera

Die „Point & Shoot“-Technik macht es möglich, über die Kamera des mobilen Endgerätes ein Objekt am entfernten Bildschirm zu selektieren. Die Anwendung zeigt ein Kamera-Bild mit einem Fadenkreuz in der Mitte des Bildschirms (Abbildung 10 links). Der/Die BenutzerIn visiert mit dem Fadenkreuz ein Element, das selektiert werden soll, am entfernten Bildschirm an. Zur Selektierung drückt die Person den Joystick. Das Drücken gleicht einem Event, welches das öffentliche

Display über WLAN erhält. Auf dem entfernten Bildschirm wird dann für einen Augenblick ein Raster mit visuellen Codes angezeigt, das dem mobilen Endgerät zur Verarbeitung der Position dienen soll (siehe Abbildung 10, Mitte). Ist die Selektierung erfolgreich, wird das Element auf dem großen Bildschirm mit einem Rahmen hinterlegt (siehe Abbildung 10 rechts) (vgl. Ballagas et al. 2005, S. 1201f).

Abbildung 10: Funktionsweise von „Point & Shoot“
(Quelle: Ballagas et al. 2005, S. 1202)

Die Autoren merken an, dass das Erscheinen des Code-Rasters störend für andere Personen sein kann. Das könnte sich aber dadurch lösen lassen, dass die Codes nur im Infrarot-Bereich erscheinen. Diese Codes kann die Kamera des mobilen Endgerätes verarbeiten, für den Menschen sind sie unsichtbar.

3.2.2.2 Direkte Auswahl von markierten Objekten über einen Tag

Die Selektierung kann auch über Tags (QR, NFC, RFID) erfolgen, die Objekte eindeutig identifizieren. Ein Beispiel dazu liefert die koreanische Supermarktkette „Homeplus“. Sie hat laut eigenen Angaben bereits 22 sogenannte „Virtual Stores“ auf U-Bahn- und Bus-Stationen in ganz Südkorea im Einsatz. Wie in Abbildung 11 zu sehen ist, sind Plakatwände im Stil einer Regalreihe eines Supermarktes gleich in der Nähe des Einstiegs einer U-Bahn aufgebaut. Jedes einzelne abgebildete Produkt ist mit einem QR-Code versehen. Mittels nativer Homeplus-App können Produkte gescannt und in den Warenkorb gelegt werden.

Abbildung 11: Plakatwand im Stil eines Supermarkt-Regals
(Quelle: http://2d-code.co.uk/tesco-qr-code-store).

Nach Abschluss des Kaufs werden die Produkte an die angegebene Wohnadresse am selben oder am nächsten Tag zugestellt. Dieses Konzept ist sehr simpel und laut Angaben des Unternehmens erfolgreich. Einerseits ist die Eröffnung neuer Geschäfte dadurch nicht notwendig, andererseits ist es für Personen, die täglich mit öffentlichen Verkehrsmitteln unterwegs sind, überaus praktisch, in der Wartezeit einen Einkauf zu tätigen. Laut Angaben des Unternehmens wirkte sich dieses Konzept seit dem ersten Aufbau eines „Virtual Store“ im August 2011 sehr positiv auf die Online-Käufe aus. Der Umsatz stieg in den ersten 3 Monaten um 130%, die Anzahl an registrierten Personen bei Homeplus stieg um 76% (o.V., 2011, o.S.). Mittlerweile gibt es eine Reihe weiterer Marken weltweit, die dieses Konzept aufgegriffen haben und eigene „Virtual Stores“ errichten[5].

3.2.2.3 Weitere Möglichkeiten der Selektierung

Als weitere Möglichkeiten zur Selektierung geben Ballagas et al. (2006) Sprach- und Gestensteuerung an. So könnten per Spracheingabe bestimmte Objekte ausgewählt werden. Im Kontext einer öffentlichen Installation könnte nach Meinung des Autors dieser Diplomarbeit diese Option womöglich auf Ablehnung stoßen, da BenutzerInnen nicht immer gewillt sein werden, im öffentlichen Raum Sprachkommandos über das Smartphone zu tätigen.

Mittels vorher genannten Optionen (Touchscreen, Accelerometer) kann ebenfalls eine Selektierung vorgenommen werden. So könnten eine Geste am Touchscreen oder eine Bewegung des mobilen Endgerätes zur Auswahl eines Elements am entfernten öffentlichen Bildschirm führen.

3.2.3 Weitere Möglichkeiten der Interaktion

Ballagas et al. (2006) führen noch eine weitere Möglichkeiten der Interaktion an, die der Vollständigkeit halber erwähnt werden. Ohne zusätzlicher mobiler Anwendung könnte eine Interaktion per *Short Message Service (SMS)* möglich sein. Damit kann theoretisch eine sehr große BenutzerInnen-Gruppe erreicht werden, da jedes mobile Endgerät diese Funktionalität besitzt. Mittels gewisser Text-Befehle könnte etwa eine Animation am entfernten Bildschirm ausgelöst werden.

3.3 Co-Displays und Cyberforaging

Bei der Kombination zweier Technologien – in diesem Fall der öffentliche Bildschirm und das Smartphone – ist es sinnvoll, dass eine Technologie jeweils

[5] Eine Liste an Beispielen ist hier zu finden: http://www.shop2mobi.com/blog/virtual-qr-code-store-examples/

die Schwächen der anderen ausgleicht. Der öffentliche Bildschirm eignet sich aufgrund der Größe als bessere Anzeige, speziell im Fall einer kollaborativen Nutzung. Über den Bildschirm können Inhalte mehrerer mobiler Geräte zusammengeführt werden. Der gesamte Inhalt des Smartphones kann zur besseren Ansicht aber auch am entfernten Bildschirm gespiegelt werden[6]. Das kleine Display des mobilen Endgerätes eignet sich wiederum besser zur Anzeige sensibler Informationen (vgl. Clinch 2013, S. 93).

Ein praktisches Beispiel, wo das Smartphone zur Erstellung eines Inhaltes und eine entfernte Projektion zur Darstellung verwendet werden, sei an dieser Stelle erwähnt. Bei der Installation *flourish* gibt es zwei Komponenten: Eine Rückprojektion auf ein Schaufenster und eine für Mobilgeräte optimierte Webseite. Die Projektion zeigt eine unscheinbare Landschaft in abstrakter Form, die sich in eine Blumenwiese verwandelt, sobald PassantInnen vorbeigehen (Abbildung 12). Die Körperbewegungen vorbeigehender Personen werden dabei von einer Microsoft Kinect registriert. Die auf der Projektion dargestellten Blumen stammen von BenutzerInnen, die über das Mobilgerät eine mobile Web-Applikation aufgerufen haben. Die Web-Applikation wurde mit HTML5 und JavaScript umgesetzt. Unter anderem kam das JavaScript 3D-Framework *three.js* zum Einsatz (vgl. Heuger et al. 2011, o.S.).

Abbildung 12: Ansicht der Rückprojektion im Projekt flourish (Quelle:Heuger et al., 2011)

[6] Hier ist vor allem Miracast bekannt. Der offene Standard erlaubt die Bildübertragung über WiFi vom Basisgerät zu einem entfernten Anzeigegerät (TV-Bildschirm). Mehr Informationen sind hier zu finden: http://www.wi-fi.org/discover-wi-fi/wi-fi-certified-miracast

3.4 Information Take-Away

Aus der analogen Welt sind Papier-Plakate mit „Abreißer“ weitreichend bekannt. Die relevanteste Informationseinheit (etwa eine Telefonnummer) kann sich eine Person einfach mitnehmen. Im digitalen Zeitalter verwenden Personen gerne die Kamera des mobilen Endgerätes zum Festhalten von Informationen, die sie nicht vergessen möchten (etwa die Informationen zu einer Veranstaltung auf einem Plakat). Die Informationen sind dann aber nur in Form eines Bildes verfügbar und müssen selbstständig ausgewertet werden. Von dieser Problematik ausgehend, haben Boring et al. (2007) in ihrer Forschungsarbeit einen Prototypen entwickelt, der die Informationsverarbeitung eines Bildes selbst vornimmt. Die Funktionsweise schaut folgendermaßen aus: Eine Person macht ein Foto von einem Ausschnitt des Bildschirmes. Das Bild wird dann über WLAN oder Bluetooth an den Host-Computer des Bildschirms gesendet. Dieser wertet die Bildinformation aus und identifiziert die Region am Bildschirm, die das Bild zeigt. Die Anwendung sucht dann nach dem nächstgelegenem Referenz-Icon, welches auf einen Medieninhalt (Audio- oder Videodatei oder Text und Bild) verweist. Die Anwendung sendet dann das Ergebnis dem Mobilgerät. Das Ergebnis ist entweder eine URL, die auf den Medieninhalt referenziert, oder die Mediendatei direkt (vgl. Boring et al. 2007, S. 26).

Dieses Konzept der Mitnahme von Informationen einer öffentlichen Installation ist auch für den Handel interessant: Stoßen PassantInnen nach Geschäftsschluss vor einem Schaufenster auf ein interessantes Produkt, das sie sich am besten auch gleich kaufen wollen, bleibt ihnen nur der Besuch über einen Online-Shop, oder das Warten auf den nächsten Geschäftstag übrig. Von dieser Problematik ausgehend entwickelte die Agentur TBWA\Helsinki für Adidas einen Prototyp für den Adidas NEO Store in Nürnberg. Dieser kam bei einem sechswöchigen Pilottest Ende 2012 zum Einsatz[7]. Die interaktive Installation setzt sich aus mehreren berührungsempfindlichen Bildschirmen zusammen, die im Verbund in einem Rahmen vor dem Schaufenster des NEO Stores aufgestellt wurden (Abbildung 13). Die Bildschirme zeigen ausgewählte Produkte in Originalgröße. Mithilfe von Touch-Gesten können PassantInnen die verfügbaren Produkte auf interaktiver Weise erkunden und mehr Informationen darüber erhalten.

[7] Diese Informationen wurden einer Presseaussendung der Agentur entnommen. Sie ist unter folgendem Link abrufbar: http://www.tbwa.com/news/adidas-tests-new-window-shopping-experience/

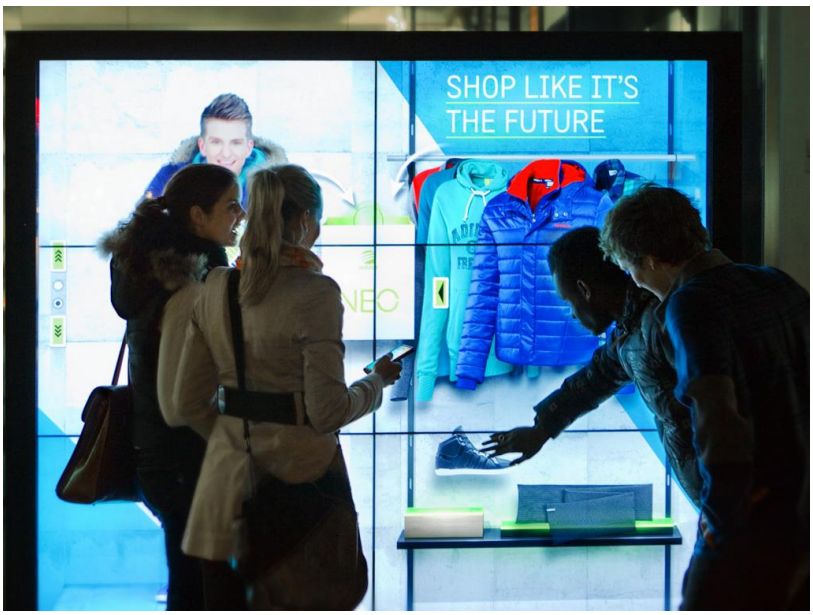

Abbildung 13: Ansicht der berührungsempfindlichen Bildschirme (Quelle: http://www.tbwa.com/news/adidas-tests-new-window-shopping-experience/)

Auf den Bildschirmen wird außerdem ein Warenkorb angezeigt, der durch Hineinziehen von Produkten befüllt werden kann. Sind im Warenkorb Produkte enthalten, wird ein vierstelliger PIN und ein Link zur Webseite *ws.adidas.com* angezeigt. Durch Eingabe des PINs wird eine bidirektionale Verbindung über WebSocket zwischen interaktivem Bildschirm und mobilem Endgerät aufgebaut. Wird der Warenkorb mit weiteren Produkten befüllt, werden diese in Echtzeit am mobilen Endgerät angezeigt. Die Produkte können für den späteren Kauf vorgemerkt, über soziale Medien geteilt oder auch per E-Mail versendet werden. Aus der PIN wird eine eindeutige URL generiert, die als Lesezeichen im Browser zum späteren Betrachten gespeichert werden kann (vgl. Mohr et al. 2014, S. 1-4).

Mittlerweile hat die Agentur TBWA\Helsinki das Konzept beim Europäischen Patentamt eingereicht.

4 Technologiebewertung

Auf Basis der genannten Beispiele in Kapitel 3 wird in diesem Kapitel eine Technologiebewertung vorgenommen, die aufzeigen soll, welche Möglichkeiten und Grenzen es mit Web-Technologien gibt. Dazu dienen folgende Fragestellungen als Grundlage dieser Bewertung:

- Eine mobile Web-Anwendung erfordert die Eingabe einer URL. Welche Alternativen gibt es, damit die händische Eingabe obsolet wird?
- Zur Selektierung in Kapitel 3.2.2.2 wurden QR-Codes verwendet. Des Weiteren ist auch eine Selektierung über ein NFC-Tag möglich. Ist ein Zugriff auf den NFC-Sensor über ein JavaScript API möglich, um Inhalte auszulesen?
- Interaktive Installationen im öffentlichen Raum, die Kollaboration und Wettbewerb erlauben, regen die Motivation an (Kapitel 2.3.4). Dafür wird Echtzeitkommunikation zwischen dem Client und Server notwendig sein. Welche Methoden zur Echtzeitkommunikation stehen mit Web-Technologien zur Verfügung?
- In den genannten Beispielen in Kapitel 3.2.1.4 und 3.2.2.1 wurde über das Kamerabild die Steuerung und Selektierung eines Objektes vorgenommen. Welche Möglichkeiten gibt es dazu im Bereich der Web-Technologien, auf die Kamera, aber auch das Mikrofon zuzugreifen?

4.1 Alternativen zur händischen URL-Eingabe

Der erstmalige Aufruf einer Webseite erfordert die Eingabe einer URL. Durch den kleinen Formfaktor und den daraus resultierenden kleinen Tastaturen eines mobilen Endgerätes erhöht sich bei längeren Texteingaben die Fehlerrate, was zugleich zu Frustration bei den BenutzerInnen führen kann. Wigdor et al. (2007) identifizieren zudem zwei weitere Probleme: Das Element, dass eine Person drückt, ist zu dem Zeitpunkt des Drückens verdeckt (*occlusion problem*). Außerdem sind interaktive Flächen (zum Beispiel Buttons) in vielen Fällen kleiner als die Fläche des Fingers (*fat finger problem*) (vgl. Wigdor et al. 2007, S. 269).

Das Fehlerrisiko kann durch möglichst kurze URLs[8] oder durch die Verwendung von Barcodes oder Tags als gänzlichen Ersatz einer händischen Eingabe

[8] Wenn trotzdem Links sehr lange sind, kann ein Short-URL Dienst zur Kürzung verwendet werden. Ein Beispiel ist etwa: http://tinyurl.com/

vermindert werden. Im folgenden Abschnitt werden daher zwei Alternativen zur händischen Eingabe behandelt: 2D-Barcodes (QR-Code) und Near Field Communication (NFC).

4.1.1 Quick-Response (QR) Code

Der *QR-Code* ist eine Weiterentwicklung des *Barcode* und kam anfänglich in der Automobil-Industrie zur Bestandsverfolgung zum Einsatz. Im Gegensatz zum Barcode können Informationen sowohl horizontal, als auch vertikal gespeichert werden (zweidimensionaler Barcode), was die Speicherung größerer Datenmengen erlaubt. Außerdem ist das Lesen dieses Codes wesentlich schneller möglich, als bei herkömmlichen Barcodes. Mittlerweile findet der QR-Code in verschiedenen Branchen weltweit Verwendung (vgl. Vazquez-Briseno et al. 2012, S. 222).

4.1.1.1 Funktionsweise eines QR-Codes

Der Aufbau eines QR-Codes ist standardisiert und in der ISO-Norm *ISO/IEC 18004:2006* festgehalten. Der Code setzt sich aus schwarzen und weißen Quadraten zusammen und kann mit einem geeigneten Scanner ausgelesen werden. Ein Quadrat trägt den Namen „Modul“. Der Standard definiert 40 Versionen (also die Größe eines Codes). Der kleinste Code (Version 1) besteht aus 21x21 Modulen, der größte aus 177x177 Modulen. Je höher die Version, desto mehr Daten können gespeichert werden (vgl. International Organization for Standardization 2011, S. 19).

Der Code teilt sich in zwei Bereiche: Das *Function Pattern* enthält keine Nutzdaten, sondern allgemeine Informationen für den Scanner. Die *Encoding Region* enthält die eigentlichen Nutzdaten mit einer Fehlerkorrektur (vgl. ebd., S. 8). Die beiden Bereiche werden im folgenden Abschnitt näher erläutert:

Function Pattern (mit visueller Darstellung eines Version 2-Codes in Abbildung 14) (vgl. ebd., S. 17):

- **Finder Pattern:** Drei große Quadrate an drei Ecken dienen dem Scanner zur Definition von Position und Orientierung des Codes. Der Scanner liest als erstes diese Bereiche aus. Das Muster besteht aus 7x7 Modulen (vgl. Vazquez-Briseno et al. 2012, S. 223).
- **Separator:** Ein Platzhalter in der Größe eines Moduls in weißer Farbe dient zur besseren Trennung zwischen *Finder Pattern* und *Encoding Region*.
- **Timing Pattern:** Dieses ist sowohl horizontal als auch vertikal zwischen den Finder Patterns platziert. Es besteht aus einem abwechselnd schwarzen und weißen Modul und dient dem Lesegerät zur Bestimmung der Größe und somit der Version des Codes.

- **Alignment Pattern:** Dieses Muster ist erst ab Version 2 zu finden. Es besteht aus 5x5 Modulen und dient dem Scanner zum Korrigieren möglicher Verzerrungen des Codes, beispielsweise wenn dieser auf einer gewölbten Oberfläche platziert ist (vgl. Vazquez-Briseno et al. 2012, S. 223).
- **Quiet Zone:** Unter diesem Begriff ist der Abstand rund um den gesamten QR-Code zu verstehen. Dieser soll zur besseren Trennung zwischen QR-Code und angrenzender Elemente mindestens vier Module breit sein.

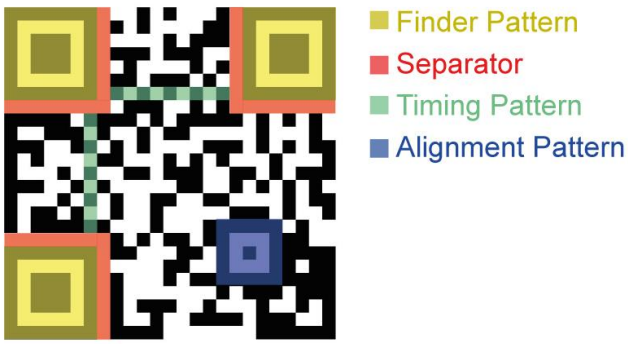

Abbildung 14: Aufbau eines QR-Codes
(Quelle: Eigene Abbildung)

Encoding Region: Diese Region enthält Informationen zu Format und Version (1 bis 40). Außerdem sind in dieser Region dann die eigentlichen Nutzdaten und die Fehlerkorrektur gespeichert (siehe Abbildung 14, alle restlichen schwarzen und weißen Module, die nicht eingefärbt sind). Die Fehlerkorrektur dient zum Ausgleichen von beschädigten Stellen am QR-Code. Insgesamt stehen vier Fehlerkorrektur-Levels zur Verfügung: L mit 7%, M mit 15%, Q mit 25% und H mit 30%. Die Prozentangabe gibt die maximale Fehlerquote an, die ein QR-Code haben kann, um trotzdem richtig gelesen werden zu können. Ist ein QR-Code zu 30% beschädigt, können mit Level H trotzdem alle Daten richtig ausgelesen werden. Möglich macht das der Algorithmus *Reed-Solomon Codes*, der auch für die Fehlerkorrektur bei Speichermedien Verwendung findet[9] (vgl. Vazquez-Briseno et al. 2012, S. 223).

Ein QR-Code kann maximal 7.089 Ziffern speichern (bei Fehlerkorrektur-Level L). Außerdem erlaubt er eine Speicherung von alphanumerischen Daten im Umfang von maximal 4.296 Zeichen (vgl. Vazquez-Briseno et al. 2012, S. 222). Der Speicherumfang hängt vom Fehlerkorrektur-Level ab. Je höher das Level, desto niedriger der Speicherplatz für die eigentlichen Nutzdaten.

[9] Die Funktionsweise wird an dieser Stelle ausgespart und kann unter folgendem Link nachgelesen werden:
http://www.cs.cmu.edu/~guyb/realworld/reedsolomon/reed_solomon_codes.html

4.1.1.2 QR-Codes und mobile Endgeräte

Das Beispiel „Homeplus Virtual Store“ in Kapitel 3.4 hat bereits einen passenden Einsatzbereich dieser Codes aufgezeigt. In dieser Variante ist in den Codes eine Produkt-ID hinterlegt, die die Scanner-Anwendung interpretieren und weiterverarbeiten kann. In einem QR-Code kann auch eine URL hinterlegt werden, die der Scanner dann als Internetadresse interpretiert und die entsprechende Aktion vorschlägt. Beim Einsatz von QR-Codes mit mobilen Endgeräten sollten folgende Aspekte beachtet werden:

- **Erkennung:** Das Scannen eines Codes soll für BenutzerInnen möglichst einfach sein, daher gilt es ein paar äußere Einflüsse zu beachten: Da eine Kamera zum Scannen verwendet wird, spielen die Lichtverhältnisse und die Qualität der Kamera eine wichtige Rolle. Je mehr Daten der Code enthält, desto aufwändiger die Decodierung für den Scanner. Daher sollen möglichst niedrige Versionen verwendet werden, um eine schnelle Decodierung zu gewährleisten. Schwarze und weiße Module gewährleisten den größtmöglichen Kontrast. Beim Einsatz eines farbigen QR-Codes soll auf ein größtmögliches Kontrastverhältnis geachtet werden (vgl. Vazquez-Briseno et al. 2012, S. 223).
- **Open Source:** Der Name „QR-Code“ ist eine eingetragene Marke des japanischen Unternehmens *Denso Wave Corporation*. Wie bereits erwähnt, ist die Funktionalität in einem ISO-Standard festgehalten, die Technologie kann frei verwendet werden. Für EntwicklerInnen stellt dies einen Vorteil dar, da keine Lizenzgebühren für die Verwendung gerichtet werden müssen. Dementsprechend groß und vielfältig ist das Angebot an QR-Code Scannern am Markt, was die Suche nach einem guten Scanner erschwert (vgl. Tarantola 2012, o.S.).
- **Sicherheit:** Da der QR-Code ein visuelles Objekt ist, kann es einfach modifiziert werden. QR-Codes auf Werbetafeln können daher mit einem böswilligen Code überklebt werden. Außerdem kann er einfach von Personen mit einem schwarzen Stift übermalen werden, sodass er unter Umständen nutzlos wird (vgl. Kieseberg et al. 2012, S. 25).

4.1.2 Near Field Communication (NFC)

„*Near Field Communication* (NFC) ist eine kontaktlose Technologie zum Austausch von Nachrichten über kurze Distanzen. NFC wurde 2002 von NXP Semiconductors (ehemals Philips Semiconductors) und Sony entwickelt.“ (Langer & Roland 2010, S. 1). Sie basiert auf der Technologie *Radio Frequency Identification (RFID)*. Das besondere an NFC ist, dass die strikte Trennung in Lesegerät und kontaktloser Chipkarte (Transponder) aufgehoben wird. Während es bei RFID-Systemen immer eine aktive und passive Komponente (die aktive

Komponente versorgt die passive immer mit Energie) gibt, kann ein NFC-Gerät beide Funktionen übernehmen[10] (vgl. Langer & Roland 2010, S. 12).

4.1.2.1 NFC Data Exchange Format (NDEF)

Für die Standardisierung zwischen dem Nachrichtenaustausch wurde im Jahr 2004 das NFC-Forum von NXP, Sony und Nokia gegründet. Die *NFC Data Exchange Format (NDEF)*-Spezifikation definiert „das Format und die Regeln für den Aufbau der Datenstrukturen zum Austausch von Informationen zwischen NFC-Geräten bzw. zwischen NFC-Geräten und NFC-Forum-Tags." (Langer & Roland 2010, S. 120). Unter dem Begriff „NFC-Forum-Tags[11]" sind passive Speicherelemente, wie zum Beispiel ein NFC-Sticker, gemeint.

Die Daten werden in sogenannte *NDEF-Records* verpackt. Jeder *Record* besteht aus einem Header, der zum Beispiel Flags wie „*Message Begin"* und Informationen über den Typ der Nutzdaten enthält, sowie dem Datenteil. Ein oder mehrere *Records* lassen sich dann zu einer *NDEF-Message* zusammenfassen. Zur Angabe des Typs der Nutzdaten werden die *MIME Media Types*, die aus dem Internet bekannt sind, verwendet. Liegt beispielsweise eine Bilddatei im JPEG-Format vor, wird als MIME Media Type „image/jpeg" angegeben. In der NDEF-Spezifikation tragen sie den Namen „NFC Record Type". Für den Kontext dieser Arbeit ist der „URI Record Type" von Bedeutung. Damit können Daten wie Internetadressen, E-Mail-Adressen oder Telefonnummern hinterlegt werden (Tabelle 1 zeigt die verschiedenen URIs dazu) (vgl. Langer & Roland 2010, S. 120-126).

Tabelle 1: Beispiele für URIs (Quelle: Langer & Roland 2010, S. 128)

Zweck	**URI**
WWW-Adresse	http://www.nfc-research.at/
E-Mail-Adresse	mailto:Max.Mustermann@nfc-research.at
Telefonnummer	tel:+43987654321
SMS-Nachricht	sms:+43987654321?body=Text
E-Mail-Nachricht	mailto:Max.Mustermann@nfc-research.at?subject=Betreff&body=Text

[10] Damit ist der Datenaustausch zwischen zwei Mobilgeräten mit NFC-Funktionalität leicht möglich. Ein Beispiel dafür ist *Android Beam:* *http://developer.android.com/guide/topics/connectivity/nfc/nfc.html#p2p*

[11] Tags können mit einer entsprechenden App beschrieben werden. Dafür bietet zum Beispiel NXP Semiconductors eine App im Google Play Store für Android an: https://play.google.com/store/apps/details?id=com.nxp.nfc.tagwriter&hl=de

Auch wenn der *Record Type* bereits ein gewisses Format definiert, ist damit noch keine auszuführende Aktion verknüpft. Die Lese-Anwendung muss also selbst darüber entscheiden, wie mit diesem Format umzugehen ist. Der *URI Record Type* ist aber Teil des *Smart Poster Record Type*. Abbildung 15 zeigt den Aufbau dieses Typs. Neben der URI (*URI Record*) können auch ein Titel (*Title Record*), ein Icon (*Icon Record*, nicht in Abbildung 15 enthalten) und die bevorzugte Aktion (*Recommended Action Record*) angegeben werden (vgl. Langer und Roland 2010, S. 127-131).

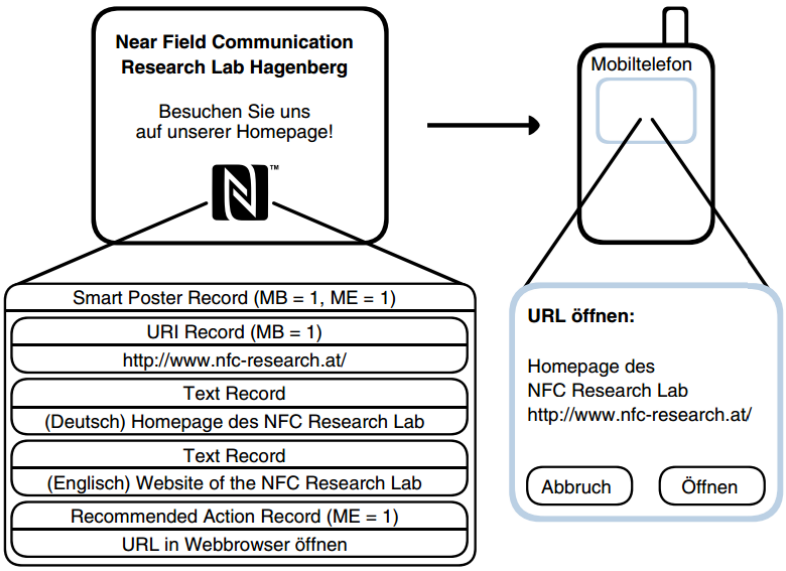

Abbildung 15: Aufbau des Smart Poster Record
(Quelle: Langer & Roland, 2010, S. 131)

4.1.2.2 NFC in mobilen Endgeräten

Bereits 2005 wurden die ersten Mobiltelefone mit NFC-Technologie von Samsung und Nokia entwickelt. In Österreich wurde der erste große kommerzielle Einsatz dieser Technologie im Jahr 2007 gestartet. Die Mobilkom Austria, die Wiener Linien und die österreichischen Bundesbahnen (ÖBB) statteten U-Bahn Stationen, Bahnhöfe und Getränkeautomaten zum Zweck des Ticket- und Getränkekaufs mit mehr als 1000 NFC-Tags aus (vgl. Langer & Roland 2010, S. 6).

Zum Zeitpunkt der Verfassung dieser Diplomarbeit sind bereits viele Android-, Windows Phone- und Blackberry-Geräte[12] mit einem NFC-Chip ausgestattet und können zugleich als aktive, als auch passive Komponente eingesetzt werden. Wird beispielsweise ein Android-Gerät auf einen NFC-Tag gehalten, der eine URL hinterlegt hat, öffnet das Betriebssystem den Web-Browser mit der

[12] Eine aktualisierte Liste von NFC-Geräten ist unter http://www.nfcworld.com/nfc-phones-list/ zu finden.

entsprechenden Webseite. Außerdem kann das mobile Endgerät selbst als Eintrittskarte oder Fahrschein dienen (vgl. Langer & Roland 2010, S. 5f).

Mobile Endgeräte der Firma Apple verfügen zurzeit nicht über NFC.

4.1.2.3 Direkt-Zugriff auf NFC mittels Web-Technologien

Es stellt sich an dieser Stelle die Frage, ob ein JavaScript-API den direkten Zugriff auf den NFC-Chip möglich macht. Das *World Wide Web Consortium* (W3C) hat eine Arbeitsgruppe eingerichtet, die an einem standardisierten API für die Verwendung der NFC-Funktionalität über den Web-Browser arbeitet. Dazu wurde mit 14. Jänner 2014 ein erster „Public Working Draft" veröffentlicht. Das API soll dabei folgende Funktionalität unterstützen (vgl. Yriarte 2014, o.S.):

- Lesen und Schreiben von NDEF-Nachrichten auf NFC-Tags
- Senden und Empfangen von NDEF-Nachrichten zwischen zwei NFC-Geräten
- Bluetooth oder WiFi-Pairing über NFC

Dem Autor dieser Diplomarbeit ist bekannt, dass es in Firefox OS[13] möglich ist, das API zu nutzen. Es bleibt aber nur zertifizierten Anwendungen, die vom Geräte-Hersteller oder Netzbetreiber genehmigt werden müssen, vorbehalten. Dritt-EntwicklerInnen können diese Funktionalität also noch nicht verwenden[14].

4.2 Echtzeitkommunikation

Das *Hypertext Transfer Protocol (HTTP)* wurde dafür entwickelt, Daten über ein Netzwerk zu übertragen. Es funktioniert nach dem Prinzip „*Request-Response*": Ein Client (der Web-Browser) stellt eine Anfrage (*HTTP-Request*) an einen Server, dieser antwortet (*Response*) mit der entsprechenden Ressource. Das Protokoll ist grundsätzlich zustandslos. Der Server erkennt bei einer Anfrage eines Clients also nicht, ob diese neu oder wiederkehrend ist. Das erspart zwar die Speicherung von Client-Daten, bedeutet aber wiederum, dass der Client bei mehreren Anfragen am gleichen Server redundante Informationen übermittelt. War es in Protokoll-Version HTTP/1.0 als Client notwendig, bei jeder Anfrage auch eine neue Verbindung zum Server aufzubauen, kann in HTTP/1.1 die Verbindung wiederverwendet werden. Hinzu kommt aber auch noch, dass HTTP nur Halbduplex unterstützt und daher

[13] Firefox OS ist ein offenes Betriebssystem auf Basis von Linux, dessen Benutzeroberfläche mithilfe von Web-Technologien umgesetzt wird. Damit soll die Entwicklung von Web APIs weiter vorangetrieben werden. Mehr Informationen zum Projekt gibt es unter: https://developer.mozilla.org/de/Firefox_OS/Einf%C3%BChrung

[14] Mehr unter: https://developer.mozilla.org/en-US/Marketplace/Options/Packaged_apps#Types_of_packaged_apps

Daten nur abwechselnd, aber nicht gleichzeitig in beide Richtungen geschickt werden können (vgl. Wang et al. 2013, S. 3ff).

Aufgrund der Tatsache, dass dieses Protokoll nicht die besten Voraussetzungen für Echtzeitkommunikation bietet, haben EntwicklerInnen Übergangslösungen entworfen: *polling*, *long polling*[15] und s*treaming*. Diese Methoden dienen vor allem dazu, beim Server anzufragen, ob neue Informationen vorhanden sind oder nicht. Beispiele dazu können sein: Abrufen aktualisierter Wetterinformation, Veränderungen eines Aktienkurses oder neue Nachrichten (vgl. Wang et al. 2013, S. 5).

Beim *polling* stellt der Client in regelmäßigen Intervallen eine Anfrage an den Server, dieser sendet eine Antwort, unabhängig davon, ob neue Informationen vorhanden sind oder nicht. Diese Methode kann sich gut eignen, wenn der Zeitpunkt der Aktualisierung von Information bekannt ist und der Client das Anfrage-Intervall danach richten kann. Echtzeitinformationen sind aber selten vorhersehbar. Somit sendet der Client auch viele unnötige, erfolglose Anfragen. Beim *long polling* hingegen wird die Verbindung so lange aufrechterhalten, bis der Server neue Informationen zurücksenden kann oder bis das vordefinierte Timeout erreicht ist. In beiden Fällen wird die Verbindung wieder getrennt und der Client muss eine neue Anfrage stellen. Bei der Methode *streaming* stellt der Client einen *Request*, und der Server sendet einen *Response*, der dann bei neuer Information aktualisiert wird („open response“). Dieser Response ist entweder unendlich lange oder für eine gewisse Zeit offen (vgl. Wang et al. 2013, S. 6).

Auch wenn die genannten Methoden Echtzeitkommunikation möglich machen, gibt es dennoch einige Nachteile: Es ist kein bidirektionaler Daten-Austausch möglich, es fehlt an Standard-Implementierungen und der HTTP-Header stellt redundante Information dar, was wiederum zu Latenz führen kann (vgl. ebd. 2013, S. 6).

4.2.1 WebSocket-Protokoll und API

Das WebSocket-Protokoll wurde im Dezember 2011 von der *Internet Engineering Task Force (IETF)* veröffentlicht. Es ist ein Netzwerkprotokoll, das auf dem *Transmission Control Protocol (TCP)* basiert und bidirektionalen Datenaustausch ermöglicht. Das zugehörige JavaScript API hat das W3C entwickelt und veröffentlicht[16]. In diesem Kapitel wird auf die Funktionsweise des APIs näher eingegangen.

[15] *Long Polling* tritt auch unter weiteren Synonymen auf: Comet oder Reverse AJAX,

[16] Der aktuelle Status des API ist unter folgenden Seite des W3C zu finden: http://www.w3.org/TR/websockets/

4.2.1.1 Verbindung zu einem WebSocket-Server

Um eine Verbindung zu einem Server herzustellen, muss das WebSocket-Objekt instanziiert werden. Der zugehörige Konstruktor nimmt zwei Argumente an: `url` und `protocols` (entweder ein einzelnes Protokoll oder ein Array mit mehreren). Das WebSocket-Protokoll definiert zwei URI Schemata: `ws` (WebSocket) und `wss` (WebSocket Secure, ähnlich einer verschlüsselten HTTP-Verbindung). Danach wird der Endpunkt eines WebSocket-Servers angegeben. Der Konstruktor gibt die WebSocket-Instanz zurück, die dann in einer Variable gespeichert wird (siehe Listing 1) (vgl. Wang et al. 2013, S. 14f).

Listing 1: Verbindung zu einem WebSocket-Server (Quelle: vgl. Wang et al. 2013, S. 15)

```
var socket = new WebSocket('ws://example.com:12010');
```

Wird das Objekt instanziiert, wird ein sogenannter *Opening Handshake* durchgeführt: Dabei kommt die Verbindungsanfrage vom Client noch über HTTP, aber mit einer Anfrage auf ein Upgrade auf das WebSocket-Protokoll (siehe Request-Header in Listing 2). Ist die Anfrage erfolgreich, antwortet der Server mit dem *Response-Code 101 (Switching Protocols)* (siehe Response-Header in Listing 3) (vgl. Wang et al. 2013, S. 40f).

Listing 2: Client Opening Handshake (Quelle: http://tools.ietf.org/html/rfc6455)

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Origin: http://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
```

Listing 3: Server Opening Handshake (Quelle: http://tools.ietf.org/html/rfc6455)

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
Sec-WebSocket-Protocol: chat
```

4.2.1.2 Events

Das WebSocket-API sowie das Protokoll sind Event-getrieben. Es stehen vier Events zur Verfügung: `Open`, `Message`, `Error`, `Close`. Mittels JavaScript ist es möglich, auf diese Events zu horchen. Tritt ein Event auf, kann die Anwendung die zugehörige *Callback-Funktion* aufrufen.

Der zugehörige Callback des Events `open` ist `onopen`. In diesem Zusammenhang ist es wichtig, dass die Anwendung immer zu Beginn auf das Event horcht. Denn erst wenn eine erfolgreiche Verbindung zum Server besteht, wird das Event gefeuert. Erst dann ist gewährleistet, dass eine fehlerlose Verbindung zwischen Client und Server besteht. Listing 4 fasst diese Funktionsweise in einem Code-Beispiel zusammen: Zuerst wird das WebSocket-Objekt instanziiert und ein Parameter (URL) übergeben. Die Rückgabe (*return*) des Konstruktors wird in der Variable `socket` gespeichert. Wird der Callback `onopen` gefeuert, sendet der Client mit der Funktion `send` eine Nachricht („Hello Server“) an den Server (vgl. Wang et al. 2013, S. 17).

Listing 4: Verbindung zu WebSocket-Server, Aufruf des Callback „onopen“ (Quelle: in Anlehnung an Wang et al. 2013, S. 17)

```
var socket = new WebSocket('ws://example.com:12010');
socket.onopen = function(e) {
    var text = "Hello Server!";
    socket.send(text);
};
```

Das Event `message` (Callback-Funktion `onmessage`) wird dann gefeuert, wenn entweder der Client oder der Server Text (String) oder Binärdaten an den jeweils anderen verschicken will. Große Binärdaten können Audio-, Video- oder Bilddateien sein und als *Binary Large Object (Blob)* oder *ArrayBuffer* verschickt werden. Es ist sinnvoll, zu überprüfen, welche Daten der Sender verschickt hat. Die Callback-Funktion des Events `message` liefert ein Objekt, das neben allgemeinen Informationen unter der Variable `data` die geschickten Daten enthält. In Listing 5 wird das Objekt der Variable `e` zugeordnet und `e.data` nach dessen Dateityp abgefragt (vgl. Wang et al. 2013, S. 17f).

Listing 5: Aufruf des Callback „onmessage“, Abfrage nach dem Datentyp (Quelle: in Anlehnung an Wang et al. 2013, S. 18)

```
socket.onmessage = function(e) {
    if (typeof e.data === "string") {
        console.log("String received!");
    }
    if (e.data instanceof Blob) {
        console.log("Blob message received!");
    }
};
```

Tritt ein Fehler auf, wird das Event `error` gefeuert. Die dazugehörige Callback-Funktion lautet `onerror`. Ein Fehler veranlasst auch die Trennung der WebSocket-Verbindung zwischen Client und Server. Wie in Listing 6 zu sehen, kann die Callback-Funktion die Art des Fehlers übermitteln.

Das letzte Event `close` (siehe Listing 6, Funktion `onclose`) wird dann gefeuert, wenn ein Fehler die Trennung der Verbindung veranlasst, oder wenn sie absichtlich geschlossen wurde (vgl. Wang et al. 2013, S. 19).

Listing 6: Aufruf der Callbacks „onerror“ und „onclose“ (Quelle: in Anlehnung an Wang et al. 2013, S. 19)

```
socket.onerror = function(e) {
    console.log("Error: ", e);
};
socket.onclose = function(e) {
    console.log("Connection closed: ", e);
};
```

Neben dem Horchen auf Events bietet das WebSocket-Objekt zwei Funktionen an: `send` und `close`. Über die Funktion `send` können Daten an den Verbindungspartner gesendet werden, die Funktion `close` bewirkt die Trennung der Verbindung (vgl. Wang et al. 2013, S. 20ff).

4.2.1.3 Verfügbarkeit

Nicht jede Browser-Version hat das WebSocket-API implementiert. Die Unterstützung ist aber mittlerweile sehr gut[17]. Die Verfügbarkeit kann über `window.WebSocket` abgefragt werden. Gibt die Abfrage das WebSocket Konstruktor-Objekt zurück, wird dieses Protokoll unterstützt. Wird nichts oder ein `undefined` ausgegeben, unterstützt der Browser es nicht nativ. Dann bedarf es einer Bereitstellung einer Fallback-Lösung, da sonst BenutzerInnen einer Anwendung nicht in der Lage sein werden, diese vollständig und fehlerfrei zu nutzen. Ein weiteres Problem kann eine Firewall oder ein Proxy darstellen, die es unter Umständen nicht erlauben, andere Protokolle als HTTP zuzulassen (vgl. Wang et al. 2013, S. 26f). Eine JavaScript-Bibliothek, die auf eine Reihe von Fallback-Lösungen zurückgreift, ist Socket.IO. Diese wird in Kapitel 4.2.2.2 näher erläutert.

4.2.2 Node.js und Socket.IO

Für eine WebSocket-Verbindung braucht es auch einen Server. Serverseitige Implementierungen sind für unterschiedliche Plattformen und Programmiersprachen verfügbar[18]. In diesem Zusammenhang wird auch oft Node.js genannt. Warum, wird im Folgenden erläutert:

[17] Eine Übersicht unter mobilen Browsern bietet: http://mobilehtml5.org/

[18] Eine Übersicht von verfügbaren Implementierungen liefert die Webseite HTML5Rocks: http://www.html5rocks.com/de/tutorials/websockets/basics/

4.2.2.1 Node.js

Node.js ist eine serverseitige JavaScript-Umgebung, die die V8 JavaScript-Engine, welche von Google entwickelt und für den Chrome Web-Browser verwendet wird, nutzt, um JavaScript-Code in nativen Maschinen-Code zu kompilieren und auszuführen (vgl. Hughes-Croucher und Wilson 2012, S. 3). Außerdem macht sich Node.js das Event-Model von JavaScript als eines der Kernkonzepte zu Nutze. Hughes-Croucher und Wilson (2012) erklären das mit der Analogie eines Fast-Food Restaurants: Person A bestellt bei Bedienung A ein Essen, dieses dauert aber ein paar Minuten. Person A erhält daher eine Nummer. Bedienung A kann in der Zwischenzeit Person B bedienen. Ist das Essen für Person A fertig, wird diese benachrichtigt (Event) und sie kann sich ihr Essen abholen. So wird eine lange Wartezeit von Person B und eine Blockierung von Bedienung A verhindert. Dadurch verkürzt sich die durchschnittliche Wartezeit (Latenz) (vgl. Hughes-Croucher & Wilson 2012, S. 36f). Die Funktionsweise soll mit einem Code-Beispiel illustriert werden. In Listing 7 wird eine Datei aus dem Dateisystem geladen. Das kann Zeit in Anspruch nehmen. Da Node.js „non-blocking“ (nicht-blockierend) ist und nur in einem einzigen Prozess läuft, horcht ein *EventListener* auf ein Event, das anzeigt, dass die Datei fertig geladen ist. Ist das der Fall, wird eine Callback-Funktion aufgerufen, die den Inhalt der Datei ausgibt. Während also das Laden der Datei im Hintergrund erfolgt, kann Node.js den weiteren Code abarbeiten (also den zweiten `console.log` ausgeben) und erst dann den Datei-Inhalt ausgeben, wenn dieser fertig geladen ist (siehe Ausgabe in Abbildung 16).

Listing 7: Laden einer Bild-Datei in Node.js (Quelle: Eigener Code)

```
var fs = require('fs');
fs.readFile('./bild.jpg', function(err, data) {
    console.log(data);
});
console.log("hi!");
```

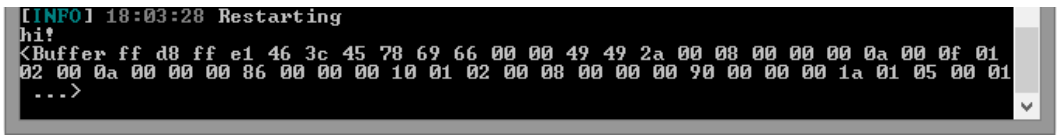

Abbildung 16: Ausgabe der Server-Konsole des Beispiels aus Listing 7 (Quelle: Eigene Abbildung).

Die genannte Methode zeigt, dass der Fokus dieser Technologie eindeutig auf der Performance liegt. Jeff Harrell, ein Entwickler bei *PayPal*, beschreibt in einem Blogbeitrag einen durchgeführten Performance-Vergleich zwischen Java und Node.js. Nachdem PayPal sich für einen teilweisen Umstieg auf Node.js entschied, parallel aber die neue Anwendung auch in Java mit dem Framework *Spring*

entwickelte, war ein Vergleich zweier Anwendungen mit gleicher Funktionalität möglich. Durchsatz und Antwortzeiten wurden mitprotokolliert und ausgewertet. Es stellte sich heraus, dass die Node.js-Anwendung doppelt so viele Anfragen in einer Sekunde schaffte, als die Java-Anwendung. Und das, obwohl die Java-Anwendung auf einem Fünfkern-Prozessor lief, die Node.js-Anwendung auf einem Einkern-Prozessor. Außerdem war die Antwortzeit von Node.js im Schnitt um 35% schneller (vgl. Harrell 2013, o.S.).

4.2.2.2 Socket.IO

Socket.IO ist eine JavaScript-Bibliothek, die als Paket für Node.js über den *Node Package Manager (NPM)* installiert werden kann. Auf Seiten des Clients muss die zugehörige JavaScript-Datei eingebunden werden (siehe Listing 8, erste Zeile).

Das API bietet eine einheitliche Schnittstelle für EntwicklerInnen an, egal welche darunterliegende Methode dann letztendlich verwendet wird. Wenn bei einem Browser das WebSocket-Protokoll nicht zur Verfügung steht, probiert Socket.IO in abfallender Reihenfolge folgende weitere Methoden zur Verbindungsherstellung aus: *FlashSocket* (benötigt ein Flash Plug-In), *XHR long polling, XHR multipart streaming, XHR polling, JSONP polling, iframe.* Die Verbindungsherstellung erfolgt ähnlich wie zuvor bei WebSocket beschrieben: Der Client unternimmt einen Handshake mit dem Server. Unterstützt der Browser WebSocket, wird auf das Protokoll umgestellt. Andernfalls wird aus den eben genannten Alternativen gewählt (vgl. Rai 2013, S. 87f).

Socket.IO bietet im Vergleich zum WebSocket-API folgende erweiterte Funktionalitäten (vgl. Rai 2013, S. 91-96):

Custom Events

Neben der Verwendung reservierter Events (`message`, `connect`, `disconnect`, `open`, `close`, `error`, `retry`, `reconnect`) können selbst-erstellte Events verwendet werden, auf die sowohl Client, als auch Server horchen können (mittels der Funktion `on`, siehe Event `news` in Listing 8).

Listing 8: Verbindung zu einem WebSocket-Server (Quelle: http://socket.io/docs/)

```
<script src="/socket.io/socket.io.js"></script>
<script>
  var socket = io('http://localhost');
  socket.on('news', function (data) {
    console.log(data);
    socket.emit('my other event', { my: 'data' });
  });
</script>
```

Broadcasting:

Socket.IO ermöglicht auch das serverseitige, gleichzeitige Versenden einer Nachricht an alle verbundenen Clients. Die zugehörige Funktion zum Senden lautet `emit` (siehe Listing 9). Mittels dem `broadcast` Flag kann der Client, der das Event ausgelöst hat, ausgenommen werden (siehe Listing 10).

Listing 9: Nachricht an alle verbundene Clients aussenden, aus Sicht des Servers (Quelle: http://socket.io/docs/)

```
var app = require('http').createServer(handler);
var io = require('socket.io')(app);
io.on('connection', function(socket) {
    io.emit('this', { will: 'be received by everyone'});
});
```

Listing 10: Nachricht an alle verbundene Clients aussenden, mit Ausnahme des Clients, der das Event ausgelöst hat. (Quelle: http://socket.io/docs/)

```
io.sockets.on('connection', function(socket) {
    socket.broadcast.emit('user connected');
});
```

Datentypen

Üblicherweise werden die Nachrichten mittels JSON-Objekte mit Key/Value-Paaren (siehe Listing 9) übermittelt. Die Bibliothek vollzieht die Serialisierung und Deserialisierung der Objekte automatisch. Socket.IO ab Version 1 unterstützt das Senden binärer Daten mittels `Blob, ArrayBuffer` und `File`.

Reconnection

Der Client sendet in regelmäßigen Abständen einen sogenannten *Heartbeat* zur Kontrolle einer aufrechten Verbindung. Erhält er keine Antwort vom Server, wird die Verbindung getrennt und ein neuer Verbindungsaufbau automatisch initiiert.

4.3 Medienzugriff

Grundsätzlich sind zwei Entwicklungen im Bereich des Hardware-Zugriffs auf Kamera oder auch Mikrofon von Bedeutung. Der Zugriff war zwar bereits auch vor den beiden Initiativen möglich, erforderte aber den Einsatz separater Plug-Ins (Flash, Silverlight). Die erste Initiative des W3C schlug die Erweiterung des Input-File-Tag (`<input type="file">`) um die Attribute `accept` und `capture` vor. Es war der erste Versuch der Standardisierung der Medienaufnahme, dieser

mündete in die Spezifikation *HTML Media Capture*[19]. Der erste Parameter soll dabei die Medienauswahl eingrenzen und zum Beispiel nur Bilder erlauben. Der zweite Parameter erlaubt EntwicklerInnen eine bevorzugte Quelle, woher der Medieninhalt kommen soll, anzugeben (vgl. Bidelman 2013, o.S.).

Das `<device>` Element ist bis dato in keinem Browser implementiert und wird an dieser Stelle ausgespart. Im Jahr 2011 wurde die Initiative *Web Real-Time Communications (WebRTC)*[20] ins Leben gerufen, an deren Standardisierung *Google Inc.,* die *Mozilla Foundation* und *Opera Software* maßgeblich beteiligt sind. Überwacht wird sie von der *W3C WebRTC Working Group*[21] (vgl. Bidelman 2013, o.S.). Die beiden Techniken werden im Folgenden anhand eines Beispiels näher erläutert.

4.3.1 HTML Media Capture

In folgendem Beispiel soll ein Foto mit der Kamera gemacht werden. Listing 11 zeigt den erweiterten Input-Tag mit den Attributen `accept` und `capture`. Wird als Type `file` angegeben, sollte der Browser automatisch einen Button zum Auswählen einer Datei anzeigen, was auch bei beiden Testgeräten[22] in Abbildung 17 der Fall ist. Wird auf den Button getippt, öffnet sich ein Auswahl-Fenster, das mehrere Optionen für die Bildauswahl zeigt. Beim Test mit den beiden Endgeräten war eine Auswahl eines vorhandenen Fotos aus der Bildergalerie möglich. Außerdem war die Option, ein Foto mit der Kamera aufzunehmen, auswählbar.

Listing 11: Input Tag mit erweiterten Tags `accept` und `capture` (Quelle: http://www.w3.org/TR/html-media-capture/)

```
<input type="file" name="image" accept="image/*" capture>
```

Listing 12 zeigt einen EventListener, der auf Änderungen im `input`-Tag horcht. Die Callback-Funktion `onchange` wird daraufhin aufgerufen. Im ersten Eintrag des Arrays `files` ist die Datei in Form eines `File`-Objektes gespeichert. Dieses Objekt beinhaltet neben Name, Aufnahmedatum, Dateigröße und Typ auch die Binär-Daten der Datei. Vorsicht ist unter iOS geboten: Safari vergibt für jedes Foto, das mit der Kamera aufgenommen wurde, automatisch den Datei-Namen `image.jpg`. Unter Android wird eine Ziffernreihe generiert, die dann den Datei-Namen ausmacht.

[19] Webseite der Spezifikation: http://dev.w3.org/2009/dap/camera/

[20] Webseite der Initiative: http://www.webrtc.org/

[21] Webseite der Arbeitsgruppe: http://www.w3.org/2011/04/webrtc/

[22] Der Code in Listing 12 wurde vom Autor dieser Diplomarbeit mit folgenden mobilen Endgeräten getestet: iPhone 4 mit iOS 7.1.1 sowie Samsung Galaxy S3 Mini mit Android 4.1 und dem Android Browser in der Version 4.1.2.

Listing 12: Abgreifen des Inhaltes im Input-Tag (Quelle: Code in Anlehnung an Kostiainen et al. 2014, o.S.)

```
var input = document.querySelector('input[type=file]');
input.onchange = function() {
    var file = input.files[0];
    drawOnCanvas(file);
};
```

Mit dem HTML `canvas`-Element kann dann das Bild dargestellt werden. Ein Canvas ist eine zweidimensionale Zeichenfläche, auf der mittels JavaScript Grafiken und Bilder gerendert werden können (siehe Abbildung 17) (vgl. Pilgrim o.J.a, o.S.). Das Bild kann dann mittels Socket.IO direkt als File-Objekt an einen entfernten Server übermittelt werden.

Abbildung 17: Test mit Android-Gerät (links) und iOS-Gerät (rechts) (Quelle: Eigene Abbildung)

Das HTML Media Capture API wird von folgenden Browsern unterstützt[23]:

- Mobile: iOS Safari ab Version 6, Android mit dem Android Browser ab Version 3, Chrome Mobile ab Version 4.0 und Blackberry 10.
- Das API wird nicht im Internet Explorer (sowohl unter Windows Phone, als auch in der Desktop-Version) unterstützt.

4.3.2 WebRTC

Der WebRTC-Standard wird mit dem Ziel einer Echtzeitkommunikation ohne zusätzlicher Plug-Ins entwickelt und besteht aus drei APIs: *MediaStream* (auch unter dem Namen *getUserMedia* bekannt), *RTCPeerConnection* und *RTCDataChannel*. Für den Kontext dieser Arbeit ist MediaStream interessant, da es den Datenstrom der Kamera abgreift und im Browser darstellen kann.

[23] Die genaue Browser-Unterstützung ist unter http://mobilehtml5.org/ zu finden.

Notwendig dazu ist das HTML-Tag `<video>`. Die Funktion `getUserMedia` wird in Listing 13 mit dem Präfix `webkitGetUserMedia` dargestellt (dieses wird für Google Chrome benötigt). Der erste Parameter gibt an, welche Ressourcen benötigt werden (in diesem Fall Kamera und Mikrofon), der zweite gibt die Callback-Funktion an, die dann das Objekt `localMediaStream` (also den Datenstrom) übernimmt. Der dritte Parameter dient der Fehlerbehandlung. Danach muss das `video`-Element selektiert und eine `src` übergeben werden. Mit der Funktion `video.play()` wird dann der Datenstrom in Echtzeit im Browser angezeigt.

Listing 13: Code-Beispiel mit der Funktion getUserMedia (Quelle: in Anlehnung an https://developer.mozilla.org/en-US/docs/NavigatorUserMedia.getUserMedia)

```
navigator.webkitGetUserMedia(
    // constraints
    {
        video: true,
        audio: true
    },
    // successCallback
    function(localMediaStream) {
        var video = document.querySelector('video');
        video.src = window.URL.createObjectURL(localMediaStream);
        video.play();
    },
    // errorCallback
    function(err) {
        console.log("The following error occured: " + err);
});
```

Derzeit wird `getUserMedia` in den Desktop-Browsern Chrome, Firefox und Opera (mit Präfixen) und unter den mobilen Browser nur von Blackberry 10 (mit Präfix) unterstützt[24].

[24] Die genaue Übersicht zu den unterstützten Browsern findet sich unter: http://caniuse.com/#search=getUserMedia

5 Prototypische interaktive Installation für den öffentlichen Raum

Der Autor dieser Diplomarbeit hat in Zusammenarbeit mit dem Master-Studenten Roman Reiter interdisziplinär eine interaktive Installation mit dem Namen „Sights“ für den öffentlichen Raum konzipiert und entwickelt. In Folge dessen wird der Autor dieser Arbeit den Wortlaut „die Entwickler“ verwenden, um beide Personen gemeinsam zu nennen. Das folgende Kapitel beschreibt die Idee, den Spielablauf und die technische Umsetzung der Installation. Außerdem wird der Feldversuch, der im Rahmen dieser Arbeit durchgeführt wurde, und dessen Ergebnisse näher erläutert.

5.1 Aufbau

Ein auf einer Traverse montierter Beamer projiziert in einigen Metern Höhe vertikal auf ein dreidimensionales Modell, bestehend aus einzelnen Tetraedern mit jeweils einem gleichseitigem Dreieck als Grundfläche. Die Tetraeder wurden aus drei weiß-beschichteten, verleimten Spanplatten angefertigt. Diese wurden in drei Grundgrößen (40, 60 und 80 Zentimeter) hergestellt. Das Modell liegt auf dem Boden, dadurch ist es von allen Seiten betracht- und begehbar. Auf jedem Tetraeder werden drei unterschiedliche Bildausschnitte, im Fall dieses Projektes von einer Sehenswürdigkeit aus St. Pölten dargestellt (siehe Abbildung 18).

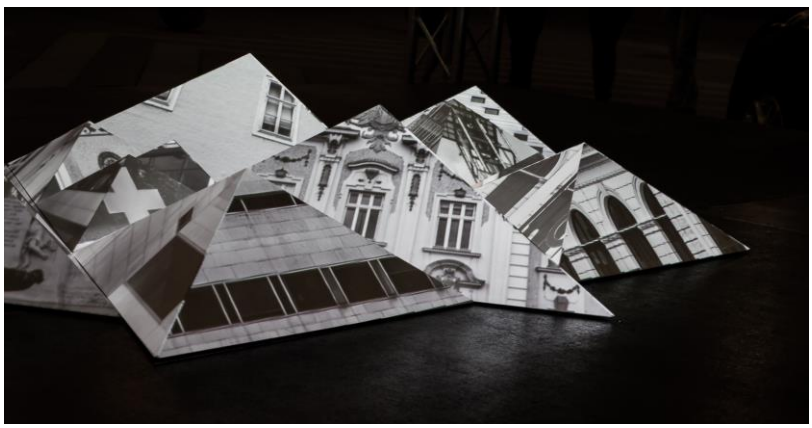

Abbildung 18: Tetraeder mit der Projektion der Bildausschnitte (Eigene Abbildung)

5.2 Spielprinzip

5.2.1 Web-Anwendung für PassantInnen

Inhaltlich gesehen stellt die Anwendung ein einfaches Quiz dar. Ziel ist es, Sehenswürdigkeiten richtig zu erraten. Die Anzahl, die erraten werden muss, ist von einem/einer AdministratorIn frei konfigurierbar. Aufgrund der Tatsache, dass die Installation für einen Feldversuch in der Stadt St. Pölten aufgestellt wird, haben die Entwickler Sehenswürdigkeiten aus diesem Umfeld ausgewählt. Die Sehenswürdigkeiten sind nicht als Ganzes, sondern nur in kleinen Ausschnitten zu sehen. Pro Tetraeder ist eine fix zugewiesene Sehenswürdigkeit abgebildet.

Das Spiel ist mehrspielerfähig, die Anzahl der gleichzeitig spielenden Personen kann konfiguriert werden. Der Spieleinstieg erfolgt entweder über NFC, QR-Code oder durch händische Eingabe einer URL. Dafür ist eine Tafel vor der Projektion angebracht, die diese drei Optionen bereitstellt (siehe Abbildung 19).

Abbildung 19: Tafel mit drei Einstiegsmöglichkeiten zur mobilen Web-Applikation (Eigene Abbildung).

Ruft eine Person die mobile Web-Applikation auf, wird ihr vom System automatisch ein Tetraeder zufällig zugewiesen und reserviert. Die Reservierung wird durch einen Farbrahmen auf einem Tetraeder gekennzeichnet (siehe Abbildung 20). Die Farbe, im Spiel „Spielfarbe“ genannt, wird ebenfalls zufällig gewählt und kann von der spielenden Person zu Beginn verändert werden. Bevor dies aber möglich ist, wird die Person gebeten, den Vornamen einzugeben.

Abbildung 20: Tetraeder mit Farbrahmen (Eigene Abbildung)

Zur Auswahl stehen neun vordefinierte Farben, davon wählt die Person eine aus (siehe Abbildung 21). Wird eine Farbe bereits von einer anderen spielenden Person verwendet, ist diese in der Farbauswahl deaktiviert und nicht anwählbar (siehe Abbildung 21 rechts, die Farbe Gelb ist nicht mehr ersichtlich, da bereits in Verwendung). Die Farbauswahl ist optional und kann auch übersprungen werden.

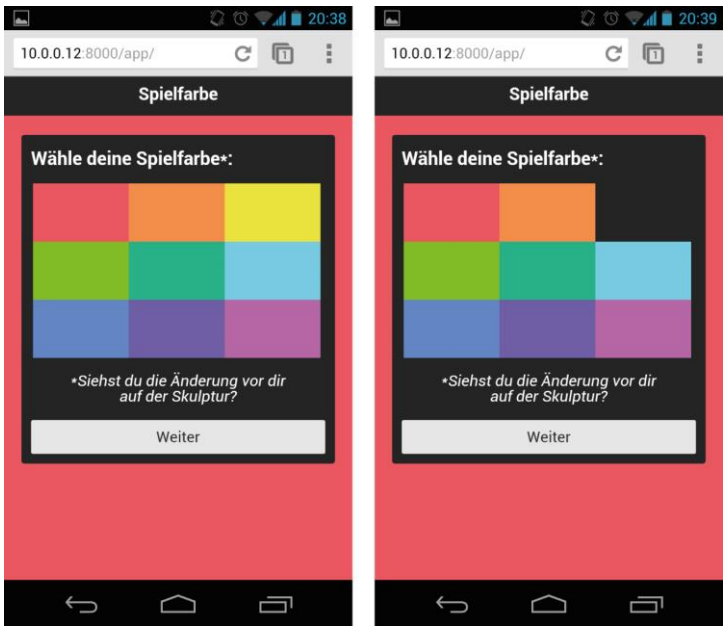

Abbildung 21: Farbauswahl und deaktivierte Farbe (rechts) (Eigene Abbildung)

Eine Spielerklärung wurde bewusst ausgespart, die spielende Person bekommt nach der Farbauswahl gleich die erste Frage präsentiert. Diese lautet immer gleich: „Welche Sehenswürdigkeit ist auf deiner Spielfarbe?“. Darunter sind vier Buttons mit einer Antwortmöglichkeit platziert (siehe Abbildung 22 links). Abbildung 22 zeigt außerdem im Browser-Fenster im schwarzen Header auf der

rechten Seite einen Countdown. Die Person hat also nur eine gewisse Zeit, die ebenfalls frei konfigurierbar ist, zur Verfügung, um eine Antwort abzugeben. Pro richtiger Antwort gibt es 100 Punkte. Keine Punkte werden vergeben, wenn die Antwort falsch oder der Countdown abgelaufen ist. Bei der Punktevergabe spielt auch die Zeit eine Rolle. Je schneller eine richtige Antwort abgegeben wurde, desto mehr Punkte gibt es. Nach erfolgter Beantwortung erscheint der Hinweis, ob die Antwort richtig oder falsch ist. Darunter wird eine Kurzbeschreibung zur Sehenswürdigkeit angezeigt (siehe Abbildung 22 rechts).

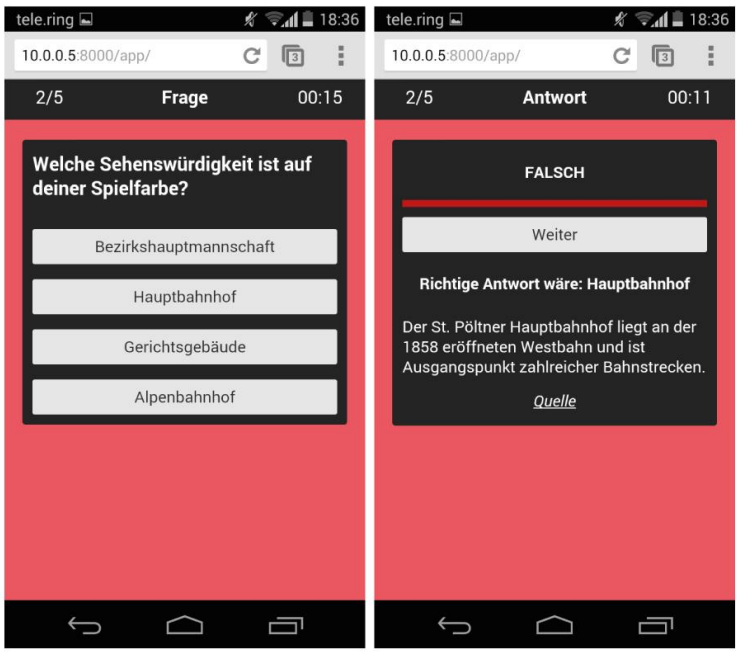

Abbildung 22: Bildschirm mit Frage und Antwort (links), Antwort-Bildschirm (rechts) (Eigene Abbildung)

Drückt die Person auf den Button „Weiter“, erscheint die nächste Frage. Dieser Vorgang wiederholt sich, bis eine gewisse Anzahl an Fragen erreicht wurde. Danach erscheint das Rundenresultat und zeigt an, wie viele richtige und falsche Antworten abgegeben wurden und welcher Punktestand sich daraus resultiert (siehe Abbildung 23 links). Im abschließenden Highscore werden alle Personen, die bereits ein Spiel gespielt haben, sortiert nach Punktestand aufgelistet (siehe Abbildung 23 rechts). Die Person kann danach wieder eine neue Runde spielen, indem sie den Button „Neue Runde starten“ drückt.

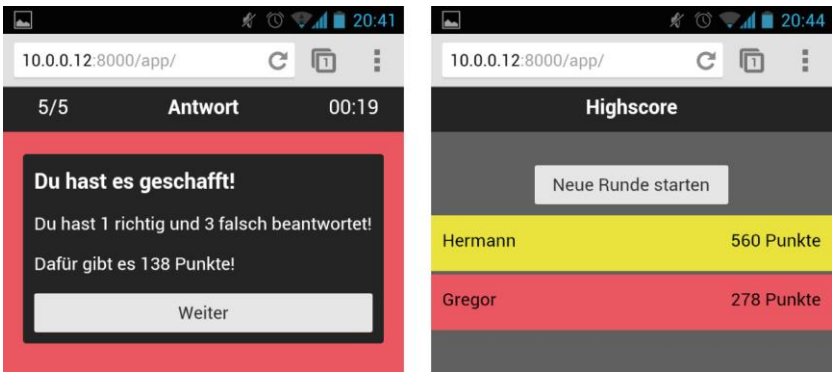

Abbildung 23: Bildschirmausschnitt Rundenübersicht (links) und Highscore (rechts) (Eigene Abbildung)

Es gibt zudem die Möglichkeit, einen Gutschein-Modus zu aktivieren. Dafür wird die zu erreichende Punktezahl angegeben. Nach jeder Runde wird dann den SpielerInnen angezeigt, wie viele Punkte noch nötig sind, um einen Gutschein zu erhalten (siehe Abbildung 24).

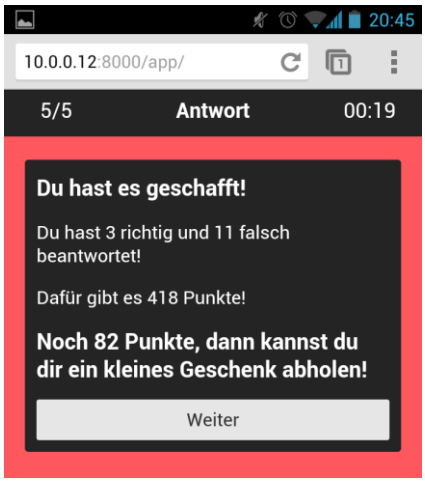

Abbildung 24: Anzeige der fehlenden Punkte bis zum Erhalt eines Gutscheins (Eigene Abbildung)

5.2.2 Web-Anwendung für AdministratorIn

Zusätzlich hat ein/e AdministratorIn die Möglichkeit, über eine eigene Oberfläche Aktionen während des laufenden Betriebes auszulösen (siehe Abbildung 25). Zum einen kann das Spiel über die Oberfläche gestartet und gestoppt werden. Zum anderen kann eine Animation ein- und ausgeschalten werden. Dazu werden die Sehenswürdigkeiten auf den Tetraedern ausgeblendet und stattdessen abwechselnd unterschiedliche Farben ein- und ausgeblendet (vorstellbar ähnlich einem Bildschirmschoner). Durch Drücken des Buttons „Animation Signal“ blinkt die gesamte Projektion dreimal in kurzen Abständen. Diese Funktion ist vom Autor dafür gedacht, implizite Interaktion zu „simulieren“. Vorbeigehenden PassantInnen

soll so vermittelt werden, dass sie es sind, die das Blitzen auslösen. Sie sollen somit erkennen, dass das System interaktiv ist.

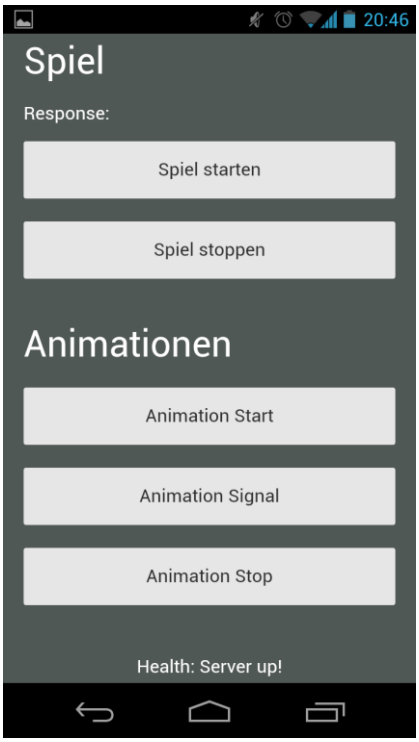

Abbildung 25: Web-Anwendung für AdministratorIn (Eigene Abbildung)

5.2.3 Argumentation für dieses Konzept

Bei der Konzeption der interaktiven Installation orientierten sich die Entwickler an den Erkenntnissen aus den Interaktionsmodellen (Kapitel 2.1) und aus den drei wesentlichen Herausforderungen: Aufmerksamkeit erzeugen (Kapitel 2.2.1), Interaktivität kommunizieren (Kapitel 2.2.2) und zur Nutzung motivieren (Kapitel 2.3):

5.2.3.1 Aufmerksamkeit erhalten

Hier setzen die Entwickler auf den intrinsischen Motivationsfaktor Neugier: Die Tetraeder zeigen Bildausschnitte von Gebäuden, die auf den ersten Blick nicht klar erkennbar sind. Dadurch soll erreicht werden, dass PassantInnen näher treten. Außerdem grenzt sich die verwendete Projektionstechnik von herkömmlichen Bildschirmen und Projektionen ab, ist daher etwas Neuartiges und soll zusätzlich Neugierde und Interesse wecken. Ist das Interesse erst einmal vorhanden, werden PassantInnen das Objekt näher betrachten und dadurch automatisch eine Menschentraube bilden, die weitere Personen anlockt (*honey-pot effect*).

5.2.3.2 Interaktivität kommunizieren

Streitz et al. (2003) sowie Vogel und Balakrishnan (2004) setzen implizite Interaktion in ihren Arbeiten ein, um die Interaktivität des Systems zu kommunizieren. Im einen Fall zeigt das mobile Endgerät auf Basis der Entfernung speziellen Inhalt an, im anderen Fall reagiert das System auf Bewegungen der vorbeigehenden PassantInnen. Beim ersten Fall besteht aber bereits eine aktive Verbindung zur Installation, was in diesem Projekt nicht möglich wäre. Weitere Methoden wie das Aufstellen einer Infotafel oder der Einsatz einer helfenden Person wurden in Kapitel 2.2.2 erwähnt. Die beiden Entwickler haben zum einen eine Hinweistafel zum Spieleinstieg gewählt und zum anderen durch die in Kapitel 5.2.2 beschriebene Animation eine Möglichkeit zur „gefälschten" impliziten Interaktion geschaffen. Zusätzlich soll auch ein/e HelferIn die Personen informieren, dass die Installation interaktiv ist.

5.2.3.3 Zur Nutzung motivieren

Hier soll wieder der intrinsische Motivationsfaktor Neugier der treibende Faktor zur Nutzung der Installation sein. Die Installation stellt etwas Unvollständiges dar und lädt dazu ein, sie näher zu erkunden. Zudem haben die beiden Entwickler bewusst auf einen unkomplizierten Einstieg geachtet. Die ersten Interaktionen sollen auf keinen Fall schief gehen (siehe Aussagen in Kapitel 2.4.6). Deshalb sind bei den ersten beiden Bildschirmen nur eine Eingabe des Namens und das Auswählen einer Farbe notwendig. Die Farbauswahl stellt gleich ein positives Erlebnis dar, da PassantInnen das Aussehen der Installation sichtbar und ohne großen Aufwand beeinflussen können. Das Sammeln von Punkten soll zusätzlich motivieren.

Optional soll ein extrinsischer Motivationsfaktor (Gutschein) die Personen dazu anhalten, sich länger mit der Installation zu beschäftigen. Das kann aber nicht der alleinige Antrieb zur Nutzung sein, da die Ankündigung einer Belohnung erst nach einer fertig gespielten Runde zu sehen ist.

5.3 Mögliche Einsatzszenarien für diese Installation

Im Zuge der Konzeption von „Sights" stellte der Autor dieser Arbeit auch Überlegungen zu Einsatzmöglichkeiten an. Das bedeutet konkret: Für welche Zwecke kann diese Installation dauerhaft im öffentlichen Raum eingesetzt werden? Der Autor beschreibt zwei mögliche Szenarien mit vier Personas. Die Begriffe „Persona" und „Szenario" stammen aus dem Bereich des *User Centered Design (UCD).* Unter ersterem versteht man die Modellierung repräsentativer Archetypen aus dem Feld einer zuvor definierten Zielgruppe (vgl. Cooper et al. 2007, S. 75f). Unter einem Szenario wird eine fiktive Geschichte verstanden, in der

die zuvor definierten Personas mit einem zukünftigen Produkt interagieren. Damit wird der Funktionsumfang und die zugrundeliegende Idee dahinter von einem Produkt kommuniziert (vgl. Cooper et al. 2007, S. 110).

5.3.1 Szenario 1: Tourismus

Folgendes Szenario könnte für den Wiener Tourismusverband interessant sein, da es eine Attraktion mit Mehrwert darstellt und zudem Werbung für lokale Gastronomiebetriebe bedeutet. Die notwendigen Inhalte (Bilder und Texte) könnten automatisiert aus Online-Quellen wie der Wikipedia bezogen werden:

„Das junge Ehepaar Ana und Jorge aus Málaga/Spanien unternimmt mit ihren beiden Kindern zum ersten Mal eine Reise in die österreichische Bundeshauptstadt Wien. Sie wollen möglichst viele typische Sehenswürdigkeiten besuchen und haben sich deshalb im Internet etwas vorinformiert. Bei ihrer Ankunft am Wiener Westbahnhof am späten Nachmittag fällt ihnen eine Installation auf, die Bildausschnitte von lokalen Sehenswürdigkeiten zeigt. Als sie neugierig näher treten, verschwinden die Bilder und ein Hinweis zum Einstieg via Smartphone wird eingeblendet. Nachdem Jorge keinen Datentarif für das Ausland hat, wählt er sich in das offene WLAN ein, das auf einem Hinweis eingeblendet ist. Danach gibt er den Link in sein Smartphone ein. Gleich erscheint die Frage, in welcher Sprache der Text dargestellt werden soll. Jorge wählt "spanisch" aus. Nach der Farbauswahl erscheint schon die erste Frage, welche Sehenswürdigkeit er auf seiner Farbe erkennen kann. Er tippt auf „Karlskirche“ und die Antwort ist richtig. Eine Kurzinformation zur Sehenswürdigkeit wird eingeblendet. Jorge beantwortet noch vier weitere Fragen, ehe der Ergebnis-Bildschirm eingeblendet wird. Er hat zwei Fragen richtig und drei Fragen falsch beantwortet. Am Ergebnis-Bildschirm wird Jorge gefragt, ob er eine Tour zu den eben erratenen fünf Sehenswürdigkeiten vornehmen möchte. Bei allen richtig erratenen Sehenswürdigkeiten erhält er einen kleinen Rabattgutschein für den Eintritt beziehungsweise für Gastronomiebetriebe in der Nähe. Jorge ist erfreut und plant gleich für den nächsten Tag, diese Tour durchzuführen“.

5.3.2 Szenario 2: Werbung

Folgendes Szenario könnte für die Gastronomie interessant sein, da die Möglichkeit besteht, an einem zentralen Ort Werbung für Aktionen oder für Produkte zu machen. Dieses Konzept kann aber auch für viele andere Betriebe funktionieren, die zwar am Abend geschlossen haben, aber durch eine Rabattaktion PassantInnen am nächsten Tag in die Geschäfte locken:

„Katharina und Lisa besuchen Donnerstagabend gerne eine der unzähligen Bars und Pubs in der Innenstadt von St. Pölten. Sie wissen bereits von der Existenz einer interaktiven Installation vor dem Rathaus, wo es Gutscheine zu gewinnen

gibt. Auch diesmal suchen sie den Platz auf. Die Projektion zeigt Ausschnitte von Gastronomiebetrieben der Stadt. Diese gilt es alle richtig zu erraten. Da sich die beiden mittlerweile in der Stadt gut auskennen, erraten beide im zweiten Anlauf alle richtig. Auf dem Handy-Display erscheint die Meldung, dass beide einen Getränkegutschein für das Café Emmi erhalten. Außerdem gibt es für alle richtig erratenen Fragen Punkte, die pro Person in einem Highscore angeführt werden. Die fünf Personen, die am Monatsende die meisten Punkte erreicht haben, erhalten zusätzlich einen Gutschein. Deshalb spielen die beiden jungen Frauen noch eine Runde, ehe sie dann ihren Gutschein im Emmi einlösen“.

5.4 Technische Umsetzung

Innerhalb der technischen Umsetzung war der Autor dieser Arbeit für die Entwicklung und Kommunikation zwischen Server und der Software *vvvv*[25] (die für das *3D-Projection Mapping* verwendet wurde) sowie für die Umsetzung der mobilen Web-Applikation zuständig. Daher liegt der Schwerpunkt der technischen Beschreibung auf diese beiden Komponenten.

Die einzelnen Komponenten sind in Abbildung 26 ersichtlich. Eine Person tippt die Kurz-URL ein (oder nutzt den QR-Code oder NFC-Tag). Sie leitet zur Original-URL weiter. Hinter der Original-URL verbirgt sich ein DDNS-Dienst[26], der zur entsprechenden hinterlegten IP-Adresse des Routers weiterleitet. Da auf dem Router ein *Forwarding*[27] eingerichtet ist, leitet dieser die Anfrage an das Gerät mit der hinterlegten IP-Adresse weiter. Auf dem Gerät läuft dann der Server, der die Anfrage entgegennimmt und verarbeitet.

Für den Server setzt der Autor Node.js ein, der um die beiden Pakete *Socket.IO* zur Echtzeitkommunikation sowie *Express* zur Implementierung einer REST-Schnittstelle[28] erweitert wird. Als Datenbank kommt MongoDB[29] zum Einsatz. Da die Entwicklung von zwei Personen durchgeführt wurde und daher der Server

[25] vvvv ist eine flexible grafische Programmieroberfläche der Firma Meso. Projektwebseite: http://vvvv.org/

[26] Im Fall des Projektes „Sights“ war dies der Anbieter: http://www.noip.com/. Herkömmliche Router sind mit einer dynamischen IP ausgestattet, die sich regelmäßig ändert. Um den Router trotzdem weltweit permanent erreichbar zu machen, können DDNS-Dienste eingesetzt werden. Die meisten Router bieten eine entsprechende Option, um einen DDNS-Dienst anzugeben, an und informieren den Dienst bei Änderung der IP.

[27] Am Router können Regeln erstellt werden, die bei eingehenden Anfragen eines bestimmten Ports an den angegebenen Rechner im Netzwerk weiterleiten, auf dem dann beispielsweise ein Web-Server läuft.

[28] Representational State Transfer (REST)

[29] MongoDB ist eine NoSQL-Datenbank, die Daten als JSON-Objekte speichert. Webseite: http://www.mongodb.org/

permanent erreichbar sein musste, lief dieser auf einem Raspberry Pi[30]. Für den in Kapitel 5.5 beschriebene Feldversuch lief der Server und die Software vvvv auf einem Gerät, der Raspberry Pi fiel dadurch weg.

Die Kommunikation zwischen Node.js und der Software *vvvv* erfolgt über das *User Datagram Protocol (UDP)*, einem verbindungslosen Netzwerkprotokoll, das der Transportschicht der Internetprotokollfamilie angehört. Die Logik der mobilen Web-Applikation wurde mittels JavaScript unter Zuhilfenahme von *AngularJS* implementiert.

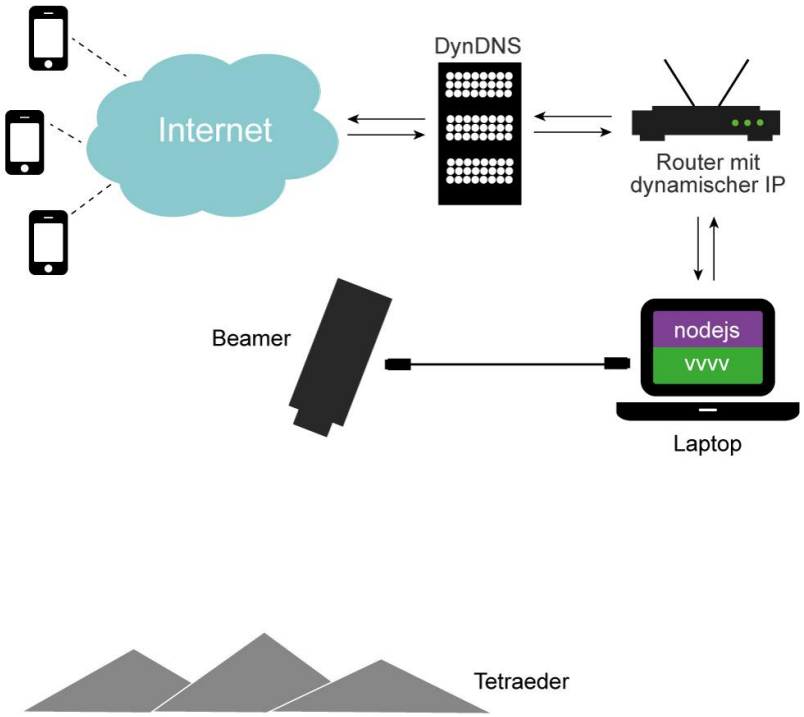

Abbildung 26: Die einzelnen Komponenten der Installation „Sights“ (Eigene Abbildung)

5.4.1 Mobile Web-Applikation

Für die in Kapitel 5.2.1 beschriebene Web-Applikation kommt das JavaScript-Framework *AngularJS*, das von Google entwickelt wurde, zum Einsatz. Es ist frei verfügbar und quelloffen (*open source*). Dieser Teil der Arbeit soll keine Dokumentation[31] von AngularJS darstellen, sondern anhand von praktischen

[30] Ein Raspberry Pi ist ein günstiger Einplatinencomputer im Scheckkarten-Format. Webseite des Projekts: http://www.raspberrypi.org/

[31] Der Autor dieser Diplomarbeit verweist auf die ausführliche Dokumentation des Frameworks, welche unter https://docs.angularjs.org/guide verfügbar ist.

Beispielen beschreiben, welche Aufgaben mit dem Framework leicht umsetzbar sind. Die Beispiel-Codes stammen von der beschriebenen Anwendung „Sights“.

AngularJS ist ein MVC-Framework. Die *Views* werden in HTML und die Steuerung (Controller) in JavaScript, meist in der Datei `controller.js` verfasst. Folgende Begrifflichkeiten sind in diesem Zusammenhang von Bedeutung:

- **Directives:** Mittels AngularJS ist es möglich, eine Reihe weiterer Attribute in der normalen HTML-Syntax zu verwenden, die eigentlich nicht Teil der Spezifikation sind, die aber AngularJS kennt und bearbeitet. Diese Attribute tragen den Namen *Directives*. Zum Beispiel weist das *Directive* `ng-controller` in Listing 14 auf die Controller-Klasse, die die Anwendungs-Logik implementiert. Weitere *Directives* sind zum Beispiel: `ng-show` oder `ng-style`. Das vorangestellte „ng“ zeigt den Namespace von AngularJS an (vgl. Green und Seshadri 2013, S. 5f).
- **Data Binding:** Da sich in einer modernen Web-Anwendung Inhalte in der HTML-Datei dynamisch verändern (Aktionen durch BenutzerInnen, oder durch die Verfügbarkeit aktuellerer Daten am Server), stellt AngularJS eine sehr einfache Möglichkeit zur Verfügung, diese Änderungen ohne viel Aufwand vorzunehmen. Möglich macht dies *Data Binding*. Im HTML-Code werden Variablen in zwei geschwungenen Klammern deklariert. Die Variablen sind dann im `$scope`-Objekt verfügbar (vgl. Green und Seshadri 2013, S. 3f). Ein Beispiel folgt in Kapitel 5.4.1.2.

Listing 14: Directive `ng-controller` (Quelle: Eigener Code)

```
<body ng-controller='TestController'>
<!-- Code here -->
</body>
```

5.4.1.1 Single Page Applikation

Die Web-Anwendung im Projekt „Sights“ ist eine Single Page Applikation. Neue Inhalte werden per AJAX oder über WebSocket nachgeladen. Es ist kein Neuladen der Webseite notwendig (vgl. Takada o.J., o.S). Die einzelnen Inhalte der Anwendung sind in DIVs zusammengefasst, die alle das *Directive* `ng-show`, welches die boolschen Werte `true` oder `false` annimmt, enthalten (siehe Listing 15, das einen Ausschnitt zeigt).

Listing 15: `div` mit Direktive `ng-show` (Quelle: Eigener Code)

```
<div ng-show="welcome_screen"><!-- Content here --></div>
<div ng-show="color_screen"><!-- Content here --></div>
```

Der Controller entscheidet dann, welches DIV sichtbar sein soll. Zuständig dafür ist die Funktion `switchScreens`, der die Variable `screen_id` übergeben wird.

Die Funktion holt sich die ID des aktuell sichtbaren DIVs aus der Variable `sights.activescreen`, setzt sie auf `false` (Zeile 3 in Listing 16) und das anzuzeigende Element auf `true`.

Listing 16: Die Funktion `switchScreens` (Quelle: Eigener Code)

```
$scope.switchScreens = function(screen_id) {
    var active_screen = sights.activescreen;
    $scope[active_screen] = false;
    $scope[screen_id] = true;
    sights.activescreen = screen_id;
};
```

5.4.1.2 Ändern von Inhalten

Durch das erwähnte *Data-Binding* können Inhalte sehr einfach mit JavaScript zur Laufzeit verändert werden. Listing 17 zeigt einen Code-Ausschnitt aus dem Projekt „Sights“. Dabei sind drei Variablen durch drei geschwungene Klammern definiert. Die Variable `gamestatus` gibt an, wie viele Tetraeder noch zu spielen sind. Die Variable `headertext` zeigt den Titel und `prop_countdown` die verbleibenden Sekunden zur Beantwortung der Frage an.

Listing 17: Data-Binding im Header in der Datei `index.html` (Quelle: Eigener Code)

```
<header>
    <div class="gamestatus"><p>{{gamestatus}}</p></div>
    <div class="heading"><h1>{{headertext}}</h1></div>
    <div class="timer"><p>{{prop_countdown}}</p></div>
</header>
```

Im Controller kann in der entsprechenden Controller-Funktion das `$scope`-Objekt übergeben werden, das alle Variablen, auch die oben genannten enthält. Diese können dann verändert werden (siehe Listing 18).

Listing 18: Aktualisieren der Daten in `controller.js` (Quelle: Eigener Code)

```
function($scope) {
   $scope.headertext = "Frage";
   $scope.gamestatus = sights.played + "/" + sights.toplay;
});
```

Besonders nützlich wird *Data Binding* dann, wenn mehrere Daten zu gleichen Zeit aktualisiert werden müssen und diese bereits in einem Array (etwa von einem Server) zur Verfügung stehen. In Listing 19 werden die vier Antwortmöglichkeiten per AJAX vom Server geladen, diese sind in einem Array jeweils in einem Objekt mit den Parametern `name` und `value` gespeichert. Das Array wird der Variable `$scope.answers` übergeben.

Listing 19: Übergabe der Daten vom Server (Quelle: Eigener Code)

```
var url = "../user/" + sights.user.uid + "/round/" +
sights.user.lastround + "/";
var getfreepyramid = $http.get(url);
getfreepyramid.success(function(pyramid, status, headers, config) {
    if (pyramid.data.pid != undefined) {
        $scope.answers = pyramid.data.answers;
    }
});
```

Ähnlich einem For-Loop (unter AngularJS mit dem Directive `ng-repeat`) iteriert AngularJS dann durch das Array und platziert die Elemente in der HTML-Struktur (siehe Listing 20). Diese Methode wird beim Anzeigen der Farben und der Einträge aus dem Highscore ebenfalls verwendet. In Listing 20 ist außerdem zu sehen, dass die Antworten als Button dargestellt werden. Über das Directive `ng-click` kann auch eine Funktion hinterlegt werden, die wieder über das `$scope`-Objekt zur Verfügung steht. Klickt man also auf einen Button, wird die Funktion `checkAnswer` aufgerufen und die Antwort als Parameter übergeben. Diese wird dann auf ihre Richtigkeit überprüft.

Listing 20: Der Einsatz von `ng-repeat` (Quelle: Eigener Code)

```
<h1>Welche Sehensw&uuml;rdigkeit ist auf deiner Spielfarbe?</h1>
<div ng-repeat="answer in answers" class="answer-button">
    <button type="button" ng-click="checkAnswer(answer.value)"
value="{{answer.value}}" class="pure-button">{{answer.name}}</button>
</div>
```

5.4.1.3 Wiederkehrende BenutzerInnen erkennen

Die Web-Anwendung soll erkennen, dass eine Person, die bereits zu einem früheren Zeitpunkt gespielt hat, wieder eingestiegen ist. Das hat folgende Gründe:

- Die Installation wird mehrmals an verschiedenen Tagen genutzt. Dafür soll nicht immer ein neuer User angelegt, sondern der bereits bestehende verwendet werden. Außerdem kann auf dieses Verhalten dann reagiert werden (Bonus, wenn Person mehrmals wiederkommt und die Installation benutzt, Auswertung des Nutzungsverhaltens).
- Die Person versucht möglicherweise durch ein Problem das Browser-Fenster neu zu laden. Automatisch würde ein neuer User angelegt werden, obwohl dieser bereits besteht.

HTML5 bietet durch das *localStorage-API* eine sehr einfache Möglichkeit, Daten im Web-Browser permanent zu speichern, auch wenn die Webseite geschlossen wird. Dieses API wird mittlerweile von allen bekannten mobilen und Desktop-

Browsern (außer Opera Mini[32]) unterstützt. Im Projekt „Sights“ wurde die User-ID im *localStorage* gespeichert (siehe Listing 21). Über die Funktion `setItem` kann ein Element gespeichert, über `getItem` wieder aus der Datenbank geholt werden (vgl. Pilgrim, o.J.b, o.S.).

Listing 21: Anwendungsbeispiel von LocalStorage (Quelle: Eigener Code)

```
//Element aus localStorage laden
sights.user.uid = localStorage.getItem("uid");

//Element in localStorage speichern
localStorage.setItem("uid", sights.user.uid);
```

5.4.1.4 Farbauswahl

Zu Beginn des Spiels wird jedem/jeder SpielerIn vom Server zufällig eine von neun vordefinierten Farben zugewiesen. Allerdings kann diese verändert werden (wie bereits in Abbildung 21 dargestellt). Um zu vermeiden, dass mehrere Personen mit der gleichen Farbe spielen, wird im Projekt folgende Lösung angewendet, die anhand eines Beispiels erläutert wird:

- Person A öffnet die Web-Anwendung und kommt zum Bildschirm mit der Farbauswahl. Bevor dieser Bildschirm erscheint, werden über Socket.IO alle verfügbaren Farben vom Server abgerufen und dargestellt.
- Person B steigt ein und kommt zur Farbauswahl.
- Person A wählt eine Farbe (gelb) aus. Die Farbauswahl wird dem Server mitgeteilt. Socket.IO teilt dann allen verbundenen Clients mit, dass die Farbe Gelb bereits reserviert ist. Der Client von Person B blendet daher die Farbe Gelb aus, sie ist nicht mehr anwählbar.

5.4.2 Node.js-Server

5.4.2.1 Einführung

Diese Arbeit erläutert nicht die gesamte Funktionsweise des Servers, sondern zeigt exemplarisch die wichtigsten Aufgaben[33]. Der Server koordiniert den Datenfluss zwischen Web-Anwendung und der Software vvvv. Ein Beispiel dazu:

- Person A bekommt Tetraeder X zugewiesen. Der Server muss dafür sorgen, dass *vvvv* die Information erhält, dass auf Tetraeder X die

[32] Die Unterstützung kann auf folgender Seite nachgelesen werden: http://caniuse.com/#search=localstorage

[33] Der gesamte Quellcode des Projekts „Sights“ ist auf der beiliegenden CD im Ordner „sights“ zu finden.

Spielfarbe von Person A angezeigt werden soll. Ansonsten kann Person A die Frage nicht beantworten.

- Person A beantwortet eine Frage und sendet die Antwort an den Server. Der Server teilt Person A einen neuen Tetraeder Y zu. Jetzt muss der Server der Software *vvvv* mitteilen, dass Tetraeder X keine Farbe mehr anzeigen soll, dafür aber Tetraeder Y.

Die Einstiegsdatei ist `server.js`, diese wird über die Kommandozeile gestartet. Hier sind *Express* und *Socket.IO* implementiert. Die meiste Kommunikation zwischen Client und Server wird über REST durchgeführt. Jede Funktion, die „Echtzeitkommunikation" erfordert (zum Beispiel die Farbauswahl) wird über Socket.IO abgewickelt. Das REST-Framework *Express* reagiert auf die Anfragen und ruft die entsprechende Funktion auf. Die wichtigsten davon sind in Tabelle 2 dargestellt.

Tabelle 2: Die wichtigsten REST-Anfragen (Eigene Abbildung)

Aktion	**URL (server.js)[34]**	**Funktion (routes/index.js)**
Spiel starten	GET /game/start	routes.startGame
Spiel stoppen	GET /game/stop	routes.stopGame
BenutzerIn anlegen	GET /user	routes.newUser
BenutzerIn-Daten anfordern	GET /user/:uid	routes.getUser
Antwort übermitteln, Tetraeder erhalten	POST /user/:uid/round/:rid/	Routes.postPyramid

5.4.2.2 Spiel starten und stoppen

Beim Server-Start wird auch automatisch das Spiel gestartet. Alternativ kann der Start über die URL `/game/start` manuell vollzogen werden. Folgende Aktionen werden dabei ausgeführt:

- Alle Tetraeder-Informationen (im Spiel Pyramiden genannt) werden aus der Datenbank geholt und in ein Array gespeichert. Die Informationen sind: ID, Name, Bild-Pfad, richtige Antwort, Informations-Text und zugehörige Quelle, Pyramiden-Zustand (gesperrt oder nicht).

[34] Die Basis-URL der Anwendung ist: http://diplo13.zapto.org:8000/

- Alle bisher angelegten User werden aus der Datenbank geholt und in ein assoziatives Array gespeichert (Index ist die User ID).
- Pyramiden-Daten werden an vvvv übermittelt (mehr dazu in Kapitel 5.4.2.6)

Wird das Spiel gestoppt, werden alle Pyramiden in der Datenbank zurückgesetzt.

5.4.2.3 Neue BenutzerInnen anlegen

Hat ein Client bereits eine gültige User ID, holt sich dieser nur die entsprechenden Daten über den AJAX-Call `/user/:uid` (`:uid` wird durch die User ID ersetzt). Hat der Client noch keine User ID, fordert er das Erstellen eines neuen Users durch den Aufruf der URL `/user` an. Der Server prüft, ob die maximale Anzahl an SpielerInnen noch nicht erreicht ist. Danach legt er einen neuen User an, speichert diesen in der Datenbank und schickt die Daten als Antwort dem Client zurück.

5.4.2.4 Tetraeder für BenutzerInnen reservieren

Der Client fordert immer einen freien Tetraeder an. Der Server überprüft im zugehörigen User-Objekt, welche Tetraeder der Client bereits gespielt hat und schließt diese von der möglichen Auswahl aus. Danach wird überprüft, welche Tetraeder gerade nicht von anderen Clients besetzt (gelockt) sind. Alle möglichen Tetraeder-IDs werden in einem Array zusammengefasst (`to_play_and_free`), aus dem die Anwendung dann zufällig eine auswählt (siehe Listing 22). Die zugehörigen notwendigen Daten werden dann dem Client übermittelt.

Listing 22: Array, aus dem eine zufällige ID ausgewählt wird. (Quelle: Eigener Code)

```
//Zufällige ID auswählen
var random_id = to_play_and_free[Math.floor(Math.random() *
to_play_and_free.length)];
```

5.4.2.5 Runde abschließen

Nachdem der Client die vorgegebene Anzahl an Tetraedern beantwortet hat, wird die Runde fertig gestellt. Dabei werden die richtigen und falschen Antworten zusammengezählt und daraus die Punkte ermittelt. Die Punkte werden zum Punktestand der letzten Runden, wenn vorhanden, addiert und das Runden-Resultat dem Client zurückgeliefert. Außerdem werden die Punkte in den Highscore eingetragen.

5.4.2.6 Nachrichten an vvvv über UDP senden

Der Server übernimmt, wie bereits erwähnt, die gesamte Logik für *vvvv* und sendet der Software über UDP den entsprechenden Datenstring. Dafür wird in Node.js das Paket `dgram`, das automatisch mitgeliefert wird, verwendet.

Im Projekt „Sights“ wurde diese Funktionalität in ein eigenes Modul ausgelagert, die das Erstellen eines UDP-Sockets und das Senden der Daten übernimmt (Modul ist im Ordner unter /mymodules/vvvv/index.js zu finden).

Listing 23 zeigt die Funktion `send`, die den String vorbereitet und diesen dann über UDP versendet. Die Software vvvv horcht auf den angegebenen Port auf einkommende Daten und kann sie so weiterverarbeiten. Die Struktur des Strings haben beide Entwickler gemeinsam definiert.

Listing 23: Array, aus dem eine zufällige ID ausgewählt wird (Quelle: Eigener Code)

```
this.send = function(type, data) {
    var string = "pyramidID:" + data.pid + "; image:" + data.image +
";";
    //Connection-Objekt holen
    var client = this.connection;
    var buff = new Buffer(string);
    //Host und Port zum Senden notwendig
    client.send(buff, 0, buff.length, this.port, this.host,
function(err, bytes) {});
};
```

5.4.2.7 Spielausstieg einer Person erkennen

Es kann passieren, dass eine Person die Anwendung während des Spiels verlässt, aber noch ein Tetraeder reserviert ist. Damit wird dieser blockiert und kann nicht von anderen SpielerInnen verwendet werden. Diese Situation gilt es zu vermeiden. Da der Client beim Aufruf der Seite automatisch immer eine Socket.IO-Verbindung aufbaut, kann das Verlassen des Spielers/ der Spielerin somit abgefangen werden. Das Event `disconnect` wird dann ausgelöst, wenn der/die BenutzerIn die Webseite verlässt (durch Schließen des Browser-Fensters oder durch Aufruf einer neuen Webseite). Listing 24 zeigt, wie der Server auf ein `disconnect` reagiert: Zu jeder Socket-ID wird die User-ID des Spielers/der Spielerin in einem assoziativen Array (`connected_sockets`) gespeichert. Löst das Event `disconnect` aus, wird die Funktion `clearUser` aufgerufen und die User-ID als Parameter übergeben. Die Funktion entzieht dem/der SpielerIn den Tetraeder und schließt die Runde automatisch ab.

Listing 24: Aktion bei `disconnect`-Event (Quelle: Eigener Code)

```
socket.on('disconnect', function() {
    console.log('User with ID: ' + connected_sockets[socket.id] +
'disconnected');

    //Aktives Tetraeder deaktivieren, Runde abschließen
    routes.clearUser(connected_sockets[socket.id]);
});
```

5.4.3 Projektion

Aufgrund der Komplexität der Technik des *3D-Projection Mapping* war es zum Zeitpunkt dieser Diplomarbeit nicht möglich, sie mittels Web-Technologien zu realisieren. Dafür wurde die Software *vvvv* eingesetzt. In dieser müssen die einzelnen Tetraeder nachgebaut und der Inhalt (die Bilder der Sehenswürdigkeiten) auf das Echt-Modell gemappt werden, damit die Projektion den Eindruck vermittelt, die Bilder würden auf den einzelnen Tetraedern „kleben". Die genaue technische Funktionsweise wird in der Diplomarbeit von Roman Reiter erläutert (vgl. Reiter 2014, in Bearbeitung).

5.5 Feldversuch

5.5.1 Aufbau

Die Installation „Sights" wurde an drei Tagen an zwei Standorten in der Stadt St. Pölten aufgebaut (siehe Tabelle 3).

Tabelle 3: Datum, Ort und Zeitraum des Feldversuchs

Datum	**Ort**	**Testzeitraum**
10. Juli 2014	Kremser Landstraße 20	21:30 bis 23:00 Uhr
16. Juli 2014	Rathausplatz St. Pölten	21:30 bis 23:00 Uhr
17. Juli 2014	Rathausplatz St. Pölten	21:00 bis 23:00 Uhr

Die Installation am Standort „Kremser Landstraße 20" wurde am Eingang eines Innenhofes platziert. Dafür wurde eine eigene Vorrichtung aus Holz angefertigt, damit der Beamer aus einem Fenster, das zum Innenhof gerichtet war, projizieren konnte (siehe Abbildung 27).

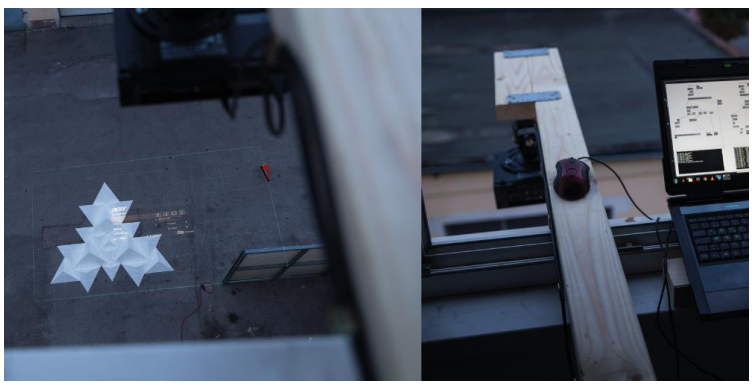

Abbildung 27: Projektionsaufbau Standort Kremser Landstraße (Eigene Abbildung)

Am Standort „Rathausplatz St. Pölten“ war eine Aufstellung im Rahmen einer Veranstaltung (Sommerfestival St. Pölten) auf einer Bühne möglich. Da hier eine Traverse zur Verfügung stand, konnte der Beamer an dieser einfach montiert werden. Die einzelnen Tetraeder wurden dann auf der Bühne platziert (Abbildung 28).

Abbildung 28: Aufbau der Installation auf einer Bühne am Rathausplatz St. Pölten (Eigene Abbildung)

5.5.2 Fragestellung und Methodik

Folgende Fragestellungen sollen mit dem Feldversuch beantwortet werden:

- Welche Verhaltensweisen der PassantInnen vor der Installation können beobachtet werden? Stimmen diese mit den bereits vorhandenen Literatur-Beispielen aus Kapitel 2 überein. Welche neuen Verhaltensweisen sind erkennbar?
- Welche Methode zum Spieleinstieg (QR, NFC, URL) findet die meiste Verwendung?
- Wie wirkt sich der Einsatz extrinsischer Belohnung (Gutscheine) auf die SpielerInnen aus? Spielen sie auch nach Erhalt des Gutscheines weiter?
- Usability-Kriterien: Wie klar ist die Bedienung unter den SpielerInnen? Wie wird der Schwierigkeitsgrad der Fragen bewertet?

Zur Datenerhebung kamen drei Methoden zum Einsatz: teilnehmende Beobachtung, Online-Umfrage und Data Logging. Die teilnehmende Beobachtung wurde vom Autor beziehungsweise von einer vorher eingeschulten Person aus ein paar Metern Entfernung verdeckt durchgeführt. Es wurde keine Ton- oder Videoaufzeichnung vorgenommen, sondern die wichtigsten Beobachtungen schriftlich festgehalten.

Die Online-Umfrage war vollständig in die mobile Web-Applikation integriert, und bestand aus vier Fragen (siehe Abbildung 29). Diese wurden nur einmal pro Smartphone nach dem Rundenergebnis eingeblendet:

- **Wie bewertest du den Schwierigkeitsgrad der Fragen?**
  Antwortmöglichkeiten: „leicht“, „mittel“ und „schwer“.
- **War dir immer klar, was zu tun ist?**
  Antwortmöglichkeiten: „ja“, „naja“ und „nein“.
- **Wie findest du die Darstellung der Bilder?**
  Antwortmöglichkeiten: „interessant“, „naja“, „langweilig“.
- **Hast du etwas Ähnliches schon einmal gesehen oder gespielt?**
  Antwortmöglichkeiten: „ja“, „weiß ich nicht“ und „nein“.

Die Online-Umfrage kam an beiden Testtagen beim Sommerfestival St. Pölten zum Einsatz.

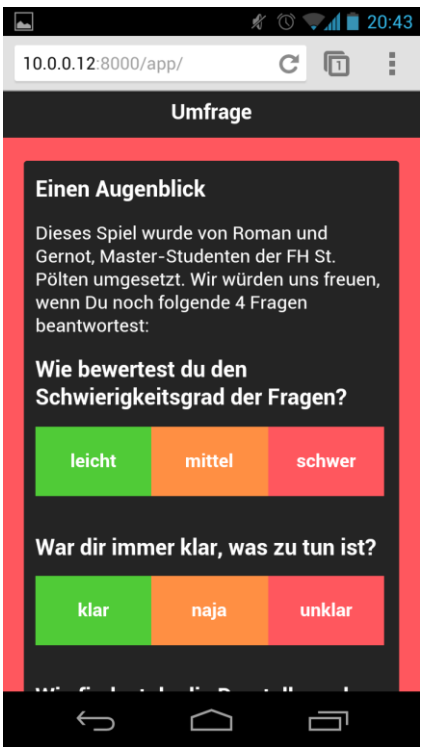

Abbildung 29: Bildschirmausschnitt der Online-Umfrage (Eigene Abbildung)

Zusätzlich zu den beiden genannten Methoden wurden folgende Daten automatisiert pro BenutzerIn mitprotokolliert:

- Spieleinstieg: Verwendete Einstiegsmöglichkeit (NFC, QR, URL) sowie Datum und Uhrzeit des Einstiegs.
- User Agent: Das verwendete mobile Endgerät samt Browser-Typ

- Vorname: Zu Spielbeginn gab es die (optionale) Möglichkeit zur Eingabe des Vornamens.
- Rundeninformation: Gespielte Anzahl an Runden, Dauer für Beantwortung einer Frage.

5.5.3 Testablauf

Für jeden Testaufbau wurden die Tetraeder circa eine Stunde vor Testbeginn aufgestellt und die Projektion eingerichtet. Das bedeutete vor allem für den Standort „Rathausplatz St. Pölten“, dass die anwesenden BesucherInnen beim Aufbau bereits zusehen konnten. Es wurden aber seitens der Entwickler keine Hinweise gegeben, um was es sich beim Aufbau handelt. Zuvor wurde keinerlei Werbung für die Installation geschaltet. Es wurden lediglich ein paar Personen eingeladen, die Veranstaltung am Rathausplatz in St. Pölten zu besuchen. Für alle drei Testtage wurde eine Person dafür abgestellt, vorbeigehende PassantInnen, die bereits die Projektion beobachten, zur Nutzung mit dem Smartphone zu motivieren. Diese helfende Person wurde immer 20 Minuten nach Testbeginn eingesetzt, um zu beobachten, wie sich Personen verhalten, die keine Hilfe bekommen. Sie wurde instruiert, Personen, die bereits vor der Installation standen anzusprechen, ob sie das Spiel probieren wollen.

Beim Standort „Rathausplatz St. Pölten“ erhielten diejenigen Personen, die mehr als 1500 Punkte erspielten, einen Gutschein. Die Gutschein-Aktion kam beim Standort „Kremser Landstraße 20“ nicht zum Einsatz.

5.5.4 Ergebnisse

5.5.4.1 teilnehmende Beobachtung

Zusammenfassend kam es beim Feldversuch zu folgenden Beobachtungen:

Intrinsischer Motivationsfaktor Neugier

Vor allem beim Standort „Rathausplatz St. Pölten“ zog die Installation viele neugierige Blicke an. Der Standort war gut gewählt und die vorbeigehenden Personen hatten auch Zeit, sich die Installation genauer anzusehen. Am Standort „Kremser Landstraße 20“ war die Neugier weit geringer. Vorbeigehende Personen blieben zwar kurz stehen, gingen dann aber wieder weg. Die Personen, die von einem Helfer angesprochen wurden, waren dann aber kurz gewillt, die Installation näher zu betrachten und auch 1-2 Runden zu spielen.

Extrinsischer Motivationsfaktor Gutschein

Es konnte beobachtet werden, dass der Gutschein Antrieb dafür war, ein paar Runden mehr zu spielen. War das Ziel dann erreicht, verließen die meisten

Personen wieder die Installation. Ein junger Bub fiel damit auf, besonders viele Punkte zu sammeln (über 11000 Punkte). Auf die Nachfrage, warum er das macht, gab er an, dass er der Beste im Highscore sein möchte und ihn keiner schlagen soll.

honey-pot effect sehr wirksam

Die Aussage von Brignull und Rogers (2003), dass bei einer Nutzung von Smartphones zur Interaktion der *honey-pot effect* nicht optimal ausgenutzt werden könnte (siehe Kapitel 2.1), erwies sich im Kontext des Feldversuchs als nicht richtig. Immer wieder bildeten sich Personengruppen, die weitere Personen anzogen. Das liegt laut Meinung des Autors vor allem daran, dass zum Erraten der Sehenswürdigkeiten ein geringer Abstand zur Installation notwendig war und somit die Bildung von Personengruppen begünstigte.

Schwieriger Übergang von PassantIn zu SpielerIn

Generell konnte beobachtet werden, dass Personen längere Zeit vor der Projektion standen, aber nur sehr wenige Personen den Spieleinstieg von sich aus wagten, sondern durch die helfende Person zur Nutzung animiert wurden. Das hat nach Meinung des Autors folgende Gründe: Am Rathausplatz in St. Pölten war der Zugang zur Bühne von allen vier Seiten möglich, die Tafel für den Einstieg war aber nur an einer Seite der Bühne stationiert. Dadurch haben einige Personen die Tafel nicht wahrgenommen. Außerdem scheint manchen Personen der Zusammenhang zwischen Projektion und Tafel nicht klar gewesen zu sein. Bei weiteren Personen konnte ein großes Interesse wahrgenommen werden, sie haben sich aber dann gescheut, mit dem Smartphone mitzuspielen. Eine weitere Erklärung kann auch sein, dass durch Beobachtung aus sicherer Entfernung nicht ganz klar war, wie das Zusammenspiel zwischen mobilem Endgerät und interaktiver Installation funktioniert.

Spiel in Zweierteams oder mehr

Viele BesucherInnen waren zu zweit oder auch in kleineren Gruppen unterwegs. In vielen Fällen spielten sie gemeinsam mit einem mobilen Endgerät (siehe Abbildung 30). Dabei konnte eine Aufgabenteilung beobachtet werden: Eine Person interagierte mit dem Smartphone, die andere konzentrierte sich eher auf die Projektion und die Beantwortung der Fragen.

Gut Mischung aus Jung und Alt

Die SpielerInnen wurden nicht nach dem Alter gefragt. Dennoch geht aus der Beobachtung hervor, dass Personen unterschiedlichen Alters die Installation nutzten (In der geschätzten Altersspanne von 14 bis 60 Jahren).

Implizite Interaktion ohne große Wirkung

Die Animation zur Kommunikation der Interaktivität (Kapitel 5.2.2) erfuhr keine große Wirkung. Der Autor dieser Arbeit gegründet dies damit, dass der Zusammenhang, warum dieses Signal ausgelöst wurde, nicht klar genug war.

Mitspielen auch ohne Smartphone

In einem konkreten Fall hielt sich eine Gruppe von vier Personen sehr lange vor der Installation auf, um die Sehenswürdigkeiten gemeinsam zu erraten, allerdings ohne ein mobiles Endgerät zu benutzen. Sie hatten sichtliche Freude am Erraten aller Sehenswürdigkeiten. Das kann sich damit erklären lassen, dass sich diese Personen ein emergentes Ziel (siehe Kapitel 2.3.1) (alle Sehenswürdigkeiten auf der Projektion erraten) gesetzt haben und so motiviert waren, das Spiel einige Minuten zu nutzen (siehe Abbildung 30).

Fehlende Mehrsprachigkeit der Web-Anwendung

An beiden Standorten waren ProbandInnen anwesend, die kein Deutsch sprachen, aber die Installation begutachteten. Da die Web-Anwendung auf Deutsch ist, war es für diese Personen schwierig, diese erfolgreich zu nutzen.

Kein offenes WLAN angeboten

In einem konkreten Fall kam eine Gruppe von Personen zur Installation, um sie näher zu betrachten. Auf Nachfrage der Helferin, ob sie teilnehmen möchten, gaben sie an, Engländer zu sein und keinen Auslands-Datentarif zu besitzen. Sie fragten nach einem offenen WLAN.

Abbildung 30: Gruppe von Personen vor der Installation am Rathausplatz in St. Pölten (Eigene Abbildung)

5.5.4.2 Online-Umfrage

Insgesamt haben 33 von 37 möglichen ProbandInnen an der Online-Umfrage teilgenommen. Abbildung 31 und Abbildung 32 zeigen die Ergebnisse der einzelnen Fragen. Die Auswertung von Frage 1 zeigt, dass rund 66% der befragten Personen als Schwierigkeitsgrad „mittel“ angegeben haben. Das ist insofern positiv zu bewerten, da es keine Anzeichen von Unter- oder Überforderung gab und die ProbandInnen sich möglicherweise in einer optimalen Flow-Zone befanden (siehe Kapitel 2.3.1).

Aus Usability-Sicht ist Frage 2 sehr positiv ausgefallen. Unter den ProbandInnen herrschte durchwegs Klarheit, wie das Spiel funktioniert. Die Antworten bei Frage 4 lassen auf die Neuartigkeit schließen, die dieses Projekt hatte: Fast alle Probandinnen haben so ein Projekt noch nicht gesehen oder gespielt.

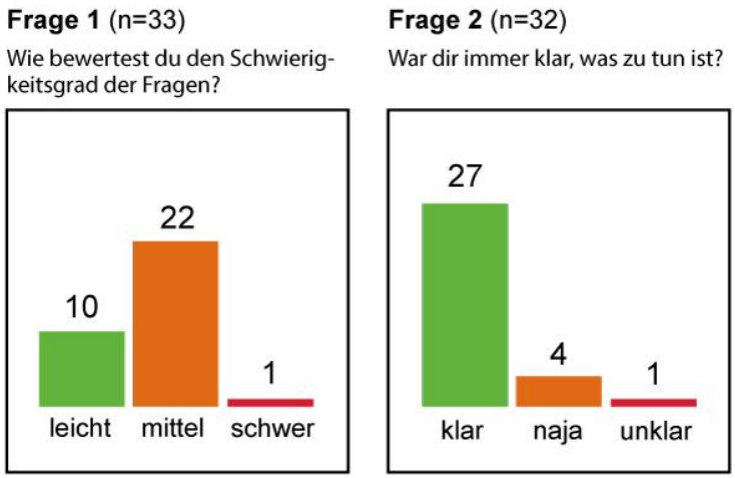

Abbildung 31: Auswertung Fragen 1-2 (Eigene Abbildung)

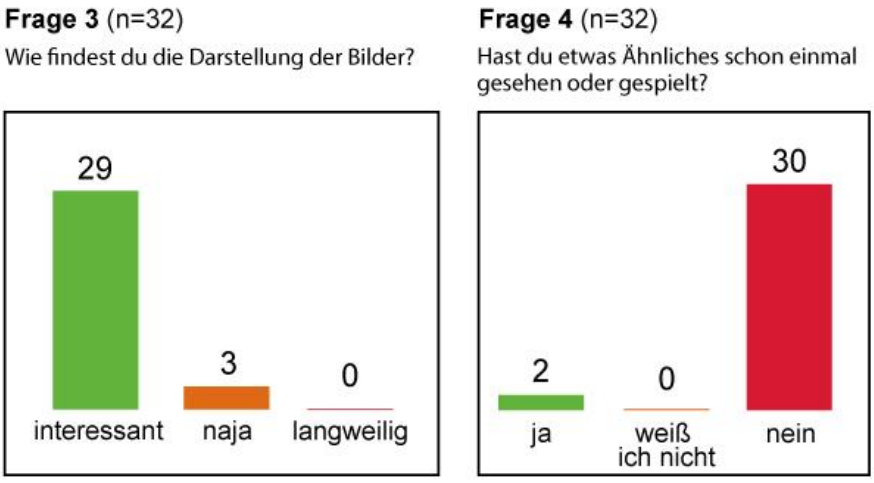

Abbildung 32: Auswertung Fragen 3-4 (Eigene Abbildung)

5.5.4.3 Demografische Daten der ProbandInnen

In der Datenbank wurden 45 Personen registriert. 30 Personen davon gelangten mittels händischer URL-Eingabe zur mobilen Web-Applikation. 12 Personen scannten erfolgreich den QR-Code, lediglich eine Person verwendete NFC als Spieleinstieg. Zwei Personen wurden von der Auswertung ausgenommen, da sie kein eigenes Smartphone hatten und das Gerät eines Helfers nutzten. 25 Personen stiegen über ein Endgerät mit Android-, 20 Personen über ein Endgerät mit iOS-Betriebssystem ein.

Vierzig Personen gaben eindeutige Vornamen an, die Aufteilung nach Geschlecht mit je 20 Personen ist genau gleichmäßig. Von einer Person konnte kein eindeutiger Vorname ermittelt werden, da es sich wahrscheinlich um einen Spitznamen handelt. Bei vier Fällen sind dem Autor technische Probleme bekannt, hier konnte das Spiel nicht erfolgreich abgeschlossen werden (daher erschien auch die Umfrage nicht), beziehungsweise gab es beim Aufruf der mobilen Web-Applikation Probleme.

Nimmt man die Ergebnisse aus der Beobachtung hinzu, lässt sich eine heterogene BenutzerInnen-Gruppe erkennen. Die Installation benutzten sowohl Frauen, als auch Männer sehr unterschiedlichen Alters.

6 Fazit

Zu Beginn dieser Diplomarbeit stellte der Autor folgende Forschungsfragen:

- Welche allgemeinen Herausforderungen und Anforderungen gilt es zu beachten, wenn eine interaktive Installation für den öffentlichen Raum entwickelt werden soll?
- Welche Herausforderungen und Anforderungen, aber auch Empfehlungen ergeben sich aus den für diese Arbeit durchgeführten Feldversuch?
- Für welche Zwecke ist ein Zusammenspiel von mobilen Endgeräten und öffentlicher Installation sinnvoll? Welche Beispiele dafür gibt es bereits in der Forschung und als marktreife Produkte?
- Die händische Eingabe einer Internetadresse *(Uniform Resource Locator, kurz URL)* kann fehleranfällig sein. Welche Alternativen zur händischen Eingabe gibt es?
- Worin liegen die technischen Möglichkeiten und Einschränkungen, wenn moderne Web-Technologien in diesem Kontext eingesetzt werden?

Es zeigte sich, dass es im Bereich der allgemeinen Herausforderungen rege Forschungstätigkeit gibt. Die Herausforderungen lassen sich in drei Bereiche gliedern (siehe Kapitel 2): Aufmerksamkeit erhalten, Interaktivität kommunizieren und zur Nutzung motivieren.

Da PassantInnen nicht auf der Suche nach interaktiven Installationen im öffentlichen Raum sind, sondern andere Ziele verfolgen, müssen sie erst darauf aufmerksam gemacht und zur Nutzung motiviert werden. Dabei zeigte der in Kapitel 5.4.3 beschriebene Feldversuch, dass der Kontext des Aufbaus eine wesentliche Rolle spielt: Wann, wo und wofür wird die Installation aufgebaut? Bei einer großen Abendveranstaltung im öffentlichen Raum ist mit BesucherInnen zu rechnen, die für jede Attraktion offen sind und sich die Zeit nehmen, diese auch in Ruhe anzusehen. Hier zeigte der intrinsische Motivationsfaktor Neugier eine große Wirkung. Das Projekt „Sights“, das für den Feldversuch verwendet wurde, ist auf den ersten Blick etwas Unvollständiges und Neuartiges. Sehr viele der vorbeigehenden PassantInnen suchten nach einer Erklärung für den Sinn dieses Objektes und blieben stehen. Dieser Aufbauort hat nach Ansicht des Autors sehr gut funktioniert. Der gleiche Aufbau hat bei einer normalen Straße zu einem Hofeingang weniger gut funktioniert, hier herrschte weniger Interesse und Bereitschaft, etwas Neues auszuprobieren, vor.

Eine weitere Anforderung war im Zuge des Feldversuchs schnell ersichtlich: Eine interaktive Installation, die mit dem Smartphone zu bedienen ist, muss diesen

Zweck auch verständlich kommunizieren. Verstehen vorbeigehende PassantInnen nicht, wofür das Smartphone gebraucht wird, gehen potenzielle BenutzerInnen verloren. Bestehende erwähnte Beispiele aus der Literatur wie „Adidas Window Shopping“ (Kapitel 3.4) oder „Homeplus Virtual Store“ (Kapitel 3.2.2.2) definieren den Nutzungszweck des Smartphones deutlich: Das Smartphone dient zur „digitalen“ Mitnahme von Produkten. Der Feldversuch zeigte, dass der Zusammenhang zwischen Installation und Smartphone zu Beginn nicht immer klar war und es einen Helfer/ eine Helferin brauchte, um die Leute zusätzlich zu motivieren, den Spieleinstieg über das Smartphone zu wagen. Ist der Zusammenhang nicht klar oder ist der/die PassantIn gar nicht im Besitz eines Smartphones, sollte er/sie sich nicht frustriert abwenden, sondern trotzdem ein positives Erlebnis mitnehmen. Ein Beispiel dazu aus dem Feldversuch: Eine Gruppe von vier Personen konnte die Tafel zum Spieleinstieg nicht sehen, da sie sich auf der anderen Seite der Installation befand (siehe Abbildung 30). Die Personen verbrachten aber trotzdem einige Minuten damit, die Sehenswürdigkeiten auf der Projektion zu erraten. Sie hatten sichtliche Freude, obwohl sie das Spiel ohne Smartphone spielten.

Ist der Zweck, dass ein Smartphone zur bestmöglichen Nutzung notwendig ist, klar kommuniziert, soll der Spieleinstieg für PassantInnen möglichst unkompliziert sein. Werden mobile Web-Technologien eingesetzt, erfordert das den Aufruf eines Web-Browsers. Die händische Eingabe ist nicht immer die beste Lösung (siehe Kapitel 4.1). Daher wurden in der Arbeit zwei alternative Möglichkeiten beschrieben: QR-Code (Kapitel 4.1.1) und NFC (Kapitel 4.1.2). Nach Ansicht des Autors stellt NFC die intuitivste Lösung dar, wäre es auf jedem Smartphone verfügbar. Das mangelnde Wissen über die Verfügbarkeit spiegelt sich in der Auswertung aus dem Feldversuch wieder: Von 45 teilnehmenden Personen haben 30 die URL zum Spieleinstieg verwendet, 12 den QR-Code und nur eine Person NFC[35]. Technische Begriffe (QR-Code und NFC) an nicht-technikaffine Personen zu vermitteln, ist eine bekannte Schwierigkeit. Um möglichen Fehlern zu entgehen, wird die Variante gewählt, die einem am ehesten etwas sagt.

Sind PassantInnen erst einmal zu BenutzerInnen der Installation geworden, gilt es die Nutzungsmotivation möglichst aufrecht zu erhalten und ein positives Nutzungserlebnis zu schaffen. Strategien dafür werden ausführlich in Kapitel 2.4 erläutert. Methoden aus dem Bereich der Gamification können hier nützlich sein (siehe Kapitel 2.4). Aus dem durchgeführten Feldversuch lassen sich weitere Anforderungen ableiten: Vielfach spielten mehrere Personen gemeinsam mit einem mobilen Endgerät, diesen Vorteil soll sich das System zu Nutze machen und Anreize bieten, damit alle anderen auch mit ihrem eigenen Gerät spielen

[35] Zwei Personen wurden von der Auswertung ausgenommen, da sie kein eigenes Smartphone hatten und das Gerät eines Helfers nutzten.

(vorausgesetzt, sie haben auch eines). Zudem sollen Anreize geboten werden, dass Personen, die bereits gespielt haben, weitere Bekannte und Freunde zur Nutzung motivieren. Das Smartphone der Person, die bereits gespielt hat, könnte dabei als Startpunkt dienen.

Die Kombination von Smartphone und interaktiver Installation im öffentlichen Raum bietet viele Vorteile, die in Kapitel 3 mit einer Reihe an Beispielen erläutert wurden. Das Smartphone eignet sich neben der Steuerung einer Installation auch bestens zur Informationsanreicherung und -mitnahme, aber auch zum Anzeigen sensibler Informationen. Der Feldversuch zeigte zudem auch, dass das Smartphone dafür geeignet ist, auf individuelle Bedürfnisse von BenutzerInnen einzugehen. Im öffentlichen Raum treffen Menschen unterschiedlicher Herkunft aufeinander. So kann Text in verschiedenen Sprachen verfügbar gemacht werden. PassantInnen können dann für sich selbst wählen, welche Sprache sie zum Benutzen der Installation brauchen. Nützlich kann das vor allem dann sein, wenn es sich bei der Installation um eine Informationsanzeige handelt. Beim Feldversuch war die fehlende Mehrsprachigkeit ein Problem, da sich Personen mit wenig Deutschkenntnissen für die Installation interessierten, diese aber schwer nutzen konnten. Eine Adaptierung dahingehend ist leicht möglich. Ohne der Nutzung eines Smartphones wäre dieses Problem weit schwieriger zu lösen.

Diese Arbeit behandelte außerdem auch Herausforderungen und Einschränkungen moderner Web-Technologien. Da der Einsatz von Echtzeitkommunikation ein wesentliches Kriterium für eine gut funktionierende Installation sein kann, wurde dieses Thema ausführlich in Kapitel 4.2 behandelt. Es zeigt sich, dass mit dem WebSocket-Protokoll eine gute, standardisierte Möglichkeit bereits zur Verfügung steht und diese Technologie sehr gut von der vorhandenen Browser-Landschaft unterstützt wird. Für das Projekt „Sights“ kam die Bibliothek Socket.IO zum Einsatz, da der Autor auch alle Browser auf Android-Geräten unter der Version 4.4 (hier steht WebSocket nicht zur Verfügung) unterstützen wollte. Socket.IO bietet dann anhand von Fallback-Lösungen „simulierte“ Echtzeitkommunikation an.

Es zeigt sich aber auch, dass es im Bereich des Medien-Zugriffs (Kamera und Mikrofon) große Einschränkungen gibt und die entsprechenden APIs aus dem WebRTC-Standard bei Mobilgeräten zum Zeitpunkt der Verfassung der Diplomarbeit nicht zur Verfügung stehen. Es ist aber trotzdem möglich, dass Personen, die im Besitz eines iOS- oder Android-Gerätes sind, eigene Bilder oder Videos zu einem fremden Server hochladen können (wie das Kapitel 4.3.1 beschreibt).

Das Themengebiet der interaktiven Installation im öffentlichen Raum ist ein umfangreiches, komplexes, aber auch sehr spannendes. Das zeigen die vielen Herausforderungen, aber auch Möglichkeiten, die es im Kontext dieses Themas

gibt. Abschließend fasst der Autor dieser Diplomarbeit die Erkenntnisse aus der Literatur und des Feldversuchs in folgende allgemeine Empfehlungen zusammen:

1. Eine interaktive Installation soll durch Unvollständigkeit und Neuartigkeit Neugierde erzeugen und die Kreativität der PassantInnen anregen.
2. Der Nutzungszweck, also warum ein Smartphone zur Nutzung der interaktiven Installation notwendig ist, soll klar ersichtlich und verständlich sein.
3. Die interaktive Installation soll auch ohne Smartphone zumindest in eingeschränkter Form benutzbar sein. Eine Installation im öffentlichen Raum sollte die Gruppe derjenigen Personen, die kein Smartphone besitzen, nicht ausgrenzen und ihnen ein positives Nutzungserlebnis bieten.
4. Der Spieleinstieg soll von jeder Position aus möglich und erkennbar sein. Die URL muss möglichst kurz gehalten werden. Alternativ können auch mehrere Einstiegsmöglichkeiten geboten werden (durch einen QR-Code oder NFC).
5. Durch intrinsische Motivationsfaktoren soll ein positives Nutzungserlebnis gefördert werden. Die BenutzerInnen sollen nach dem Ausstieg eine positive Erinnerung mitnehmen, dadurch weitere Personen zur Nutzung einladen und selbst gerne wieder zurückkehren.

7 Ausblick

Das Projekt „Sights“, das in Kapitel 5 ausführlich beschrieben wird, kann auf Basis der Erkenntnisse dieser Arbeit erweitert werden. So kann die Anwendung gänzlich in Englisch übersetzt werden, um so auch Personen ohne Deutschkenntnisse das Spielen zu ermöglichen. Die Installation ist zwar mehrspielerfähig, allerdings kann sie dahingehend erweitert werden, dass auch Kollaboration und Wettbewerb untereinander möglich sind, um so die Nutzungsmotivation zusätzlich zu steigern.

Das Projekt wird für einen weiteren Feldversuch bei der European Researchers‘ Night in der Aula der Wissenschaften in Wien am 26. September 2014 aufgebaut[36]. Diese Veranstaltung findet europaweit gleichzeitig in 300 verschiedenen Städten statt und dient der Präsentation aktueller Forschungsaktivitäten an der Fachhochschule St. Pölten für die Öffentlichkeit. Im Rahmen dieser Veranstaltung wird sich der Autor unter anderem genauer ansehen, welche Methode zum Spieleinstieg verwendet wird, und nachfragen warum sich die BesucherInnen gegen die anderen vorhandenen Möglichkeiten entschieden haben. Da in den ersten Feldversuchen keine qualitative Befragung durchgeführt wurde, wäre das eine gute Möglichkeit für neue Erkenntnisse in dieser Thematik.

Diese Diplomarbeit behandelt eine sehr aktuelle Fragestellung, das zeigen auch die vielen bereits genannten Forschungsarbeiten in diesem Umfeld. Das derzeit laufende Forschungsprojekt ATREUS am Forschungszentrum Telekommunikation Wien (FTW) untersucht das Potenzial von HTML5, um Smartphones als Universal-Fernbedienung für öffentliche Bildschirme einsetzen zu können[37]. Die dafür entwickelte Anwendung soll dann öffentlich verfügbar sein.

[36] Mehr Informationen zur Aufstellung unter: http://www.fit-for-future.at/mobile-corner-vielfaeltige-nutzungsmoeglichkeiten-der-digitalen-begleiter

[37] Mehr Informationen auf der Projektwebseite unter: http://matthiasbaldauf.com/atreus/

8 Literaturverzeichnis

Ballagas, R., Borchers, J., Rohs, M., & Sheridan, J. G. (2006). The Smart Phone: A Ubiquitous Input Device. *IEEE Pervasive Computing*, *5*(1), 70–77. doi:10.1109/MPRV.2006.18

Ballagas, R., Rohs, M., & Sheridan, J. G. (2005). Sweep and point and shoot: phonecam-based interactions for large public displays (p. 1200). ACM Press. doi:10.1145/1056808.1056876

Bedford, A. (2014). Icon Usability. Retrieved from http://www.nngroup.com/articles/icon-usability/

Bidelman, E. (2013). Capturing Audio & Video in HTML5 - HTML5 Rocks. Retrieved August 8, 2014, from http://www.html5rocks.com/de/tutorials/getusermedia/intro/

Boring, S., Altendorfer, M., Broll, G., Hilliges, O., & Butz, A. (2007). Shoot & copy: phonecam-based information transfer from public displays onto mobile phones (p. 24). ACM Press. doi:10.1145/1378063.1378068

Boring, S., Jurmu, M., & Butz, A. (2009). Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays (p. 161). ACM Press. doi:10.1145/1738826.1738853

Brignull, H., & Rogers, Y. (2003). Enticing people to interact with large public displays in public spaces. In *In Proceedings of the IFIP International Conference on Human-Computer Interaction (INTERACT 2003* (pp. 17–24).

Clinch, S. (2013). Smartphones and Pervasive Public Displays. *IEEE Pervasive Computing*, *12*(1), 92–95. doi:10.1109/MPRV.2013.16

Cooper, A., Reimann, R., & Cronin, D. (2007). *About Face 3: The Essentials of Interaction Design* (Auflage: 3. Auflage.). Indianapolis, IN: John Wiley & Sons.

Davies, N., Langheinrich, M., Jose, R., & Schmidt, A. (2012). Open Display Networks: A Communications Medium for the 21st Century. *Computer*, *45*(5), 58–64. doi:10.1109/MC.2012.114

Foley, J. D., Wallace, V. L., & Chan, P. (1984). The human factors of computer graphics interaction techniques. *IEEE Computer Graphics and Applications*, *4*(11), 13–48. doi:10.1109/MCG.1984.6429355

Google Inc. (2013). Unser mobiler Planet: Österreich. Retrieved from http://services.google.com/fh/files/misc/omp-2013-at-local.pdf

Green, B., & Seshadri, S. (2013). *AngularJS* (Auflage: 1.). Beijing ; Sebastopol, CA: O'Reilly & Associates.

Harrell, J. (2013). Node.js at PayPal | PayPal Engineering Blog. Retrieved from https://www.paypal-engineering.com/2013/11/22/node-js-at-paypal/

Heuger, R., Sadowski, S., & Schulz, F. (2011). flourish. An interactive installation in public space. Retrieved from http://flourish.ahoi.in/

Hughes-Croucher, T., & Wilson, M. (2012). *Node: Up and Running: Scalable Server-Side Code with JavaScript* (Auflage: 1.). O'Reilly Media.

International Organization for Standardization. (2011). ISO/IEC 18004:2006 - Information technology - Automatic identification and data capture techniques - QR Code 2005 bar code symbology specification. Retrieved from http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43655

Ju, W., & Sirkin, D. (2010). Animate Objects : How Physical Motion Encourages Public Interaction. Retrieved from http://wendyju.com/publications/animateobjects.pdf

Kieseberg, P., Schrittwieser, S., Leithner, M., Mulazzani, M., Weippl, E., Munroe, L., & Mayank, S. (2012). Malicious Pixels Using QR Codes as Attack Vector. In *Trustworthy Ubiquitous Computing*. Paris, France: Atlantis Press.

Kostiainen, A., Oksanen, I., & Hazaël-Massieux, D. (2014). HTML Media Capture - W3C Last Call Working Draft 19 June 2014. Retrieved from http://dev.w3.org/2009/dap/camera/

Langer, J., & Roland, M. (2010). *Anwendungen und Technik Von Near Field Communication (NFC)*. Springer-Verlag.

Madhavapeddy, A., Scott, D., Sharp, R., & Upton, E. (2004). Using Camera-Phones to Enhance Human-Computer Interaction. In *Sixth International Conference on Ubiquitous Computing (Adjunct Proceedings: Demos)*. Nottingham, UK.

Matthies, D. J. C., Nguyen, N. D. H., Lucas, S. J., & Botz, D. (2013). Moving shapes: a multiplayer game based on color detection running on public displays (p. 558). ACM Press. doi:10.1145/2493190.2494433

Michelis, D. (2009). *Interaktive Großbildschirme Im Öffentlichen Raum: Nutzungsmotive und Gestaltungsregeln*. Springer DE.

Mohr, M., Reunanen, T., & Tikkanen, J. (2014). A mechanism for distributing browsing results from a public electronic retailing system. Retrieved from http://worldwide.espacenet.com/publicationDetails/biblio?CC=EP&NR=2706498A1&KC=A1&FT=D&ND=&date=20140312&DB=&locale=en_EP

Müller, J., Alt, F., Michelis, D., & Schmidt, A. (2010). Requirements and design space for interactive public displays (p. 1285). ACM Press. doi:10.1145/1873951.1874203

Müller, J., Walter, R., Bailly, G., Nischt, M., & Alt, F. (2012). Looking Glass: A Field Study on Noticing Interactivity of a Shop Window. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 297–306). New York, NY, USA: ACM. doi:10.1145/2207676.2207718

Nagel, W., & Fischer, V. (2013). *Multiscreen Experience Design: Prinzipien, Muster und Faktoren fur die Strategieentwicklung und Konzeption digitaler Services fur verschiedene Endgeräte* (Auflage: 1.). Schwäbisch Gmünd: digiparden.

Pilgrim, M. (o.J.a). Let's call it a draw(ing surface). Retrieved from http://diveintohtml5.info/canvas.html

Pilgrim, M. (o.J.b). The Past, Present & Future of Local Storage for Web Applications. Retrieved from http://diveintohtml5.info/storage.html

Rai, R. (2013). *Socket.io Real-time Web Application Development*. Packt Publishing.

Reiter, R. (2014). 3D Projection Mapping am Beispiel einer interaktiven öffentlichen Installation. In Bearbeitung.

Shea, G., & Longford, M. (2011). Identity Play in an Artistic, Interactive Urban Projection. CHI 2011 Workshop Paper. Retrieved from http://largedisplaysinurbanlife.cpsc.ucalgary.ca/PDF/shea_final.pdf

Streitz, N. A., Streitz, N., Röcker, C., Prante, T., Stenzel, R., & Alphen, D. van. (2003). *Situated Interaction with Ambient Information: Facilitating Awareness and Communication in Ubiquitous Work Environments*.

Takada, M. (o.J.). Single page apps in depth. Retrieved from http://singlepageappbook.com/goal.html

Tarantola, A. (2012). How QR Codes Work and Why They Suck So Hard. Retrieved August 11, 2014, from http://gizmodo.com/5969312/how-qr-codes-work-and-why-they-suck-so-hard

Tesco builds virtual shops for Korean commuters. (2011). Retrieved from http://www.telegraph.co.uk/technology/mobile-phones/8601147/Tesco-builds-virtual-shops-for-Korean-commuters.html

Vazquez-Briseno, M., I., F., Dios Sanchez-Lopez, J. de, Jimenez-Garcia, E., Navarro-Cota, C., & Ivan, J. (2012). Using RFID/NFC and QR-Code in Mobile Phones to Link the Physical and the Digital World. In I. Deliyannis (Ed.), *Interactive Multimedia*. InTech. Retrieved from http://www.intechopen.com/books/interactive-multimedia/using-rfid-nfc-and-qr-code-in-mobile-phones-to-link-the-physical-and-the-digital-world

Vogel, D., & Balakrishnan, R. (2004). Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users (p. 137). ACM Press. doi:10.1145/1029632.1029656

Wang, V., Salim, F., & Moskovits, P. (2013). *The Definitive Guide to HTML5 WebSocket* (Auflage: 2013.). Berkeley, Calif.; Birmingham: Apress.

Weiser, M., & Brown, J. S. (1996). The coming Age of Calm Technolgy. Retrieved from http://www.ubiq.com/hypertext/weiser/acmfuture2endnote.htm

Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., & Shen, C. (2007). Lucid touch: a see-through mobile device (p. 269). ACM Press. doi:10.1145/1294211.1294259

Yriarte, L., Ortiz, S., & Coleman, D. (2014). Web NFC API - W3C First Public Working Draft 14 January 2014. Retrieved from http://www.w3.org/TR/nfc/

Zichermann, G., & Cunningham, C. (2011). *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps* (Auflage: 1.). Sebastopol, Calif: O'Reilly & Associates.

9 Abbildungsverzeichnis

Abbildung 1: Phasen der Aktivitäten bei öffentlichen Displays nach Brignull und Rogers (Quelle: Brignull und Rogers 2003, S. 7). 13

Abbildung 2: Drei Interaktionszonen (Quelle: Streitz et al. 2003, S. 3) 15

Abbildung 3: Interaktionsmodell nach Vogel und Balakrishnan (Quelle: Vogel & Balakrishnan 2004, S. 3)..... 16

Abbildung 4: Vergleich physisches und digitales Objekt nach Ju und Sirkin (Quelle: Ju & Sirkin 2010, S. 4) 17

Abbildung 5: Grafische Darstellung der Flow-Zone (Quelle: Zichermann & Cunningham 2011, S. 18). 20

Abbildung 6: Die Smartphone-Anwendung (links) und die Installation „Tentacles“ (rechts) (Quelle: www.tentacles.ca). 28

Abbildung 7: Anwendung im Beispiel „Tilt“ (Quelle: Boring et al 2009, S. 163).. 29

Abbildung 8: Funktionsweise der Anwendung von Madhavapeddy et al. (2004) (Quelle: Madhavapeddy et al. 2004, S. 1). 29

Abbildung 9: Die Funktionsweise von „Moving Shapes“ (Quelle: Matthies et al. 2013, S.559) 30

Abbildung 10: Funktionsweise von „Point & Shoot“ (Quelle: Ballagas et al. 2005, S. 1202) 31

Abbildung 11: Plakatwand im Stil eines Supermarkt-Regals (Quelle: http://2d-code.co.uk/tesco-qr-code-store). 31

Abbildung 12: Ansicht der Rückprojektion im Projekt flourish (Quelle:Heuger et al., 2011) 33

Abbildung 13: Ansicht der berührungsempfindlichen Bildschirme (Quelle: http://www.tbwa.com/news/adidas-tests-new-window-shopping-experience/) 35

Abbildung 14: Aufbau eines QR-Codes (Quelle: Eigene Abbildung) 38

Abbildung 15: Aufbau des Smart Poster Record (Quelle: Langer & Roland, 2010, S. 131) 41

Abbildung 16: Ausgabe der Server-Konsole des Beispiels aus Listing 7 (Quelle: Eigene Abbildung). 47

Abbildung 17: Test mit Android-Gerät (links) und iOS-Gerät (rechts) (Quelle: Eigene Abbildung) 51

Abbildung 18: Tetraeder mit der Projektion der Bildausschnitte (Eigene Abbildung) 53

Abbildung 19: Tafel mit drei Einstiegsmöglichkeiten zur mobilen Web-Applikation (Eigene Abbildung). 54

Abbildung 20: Tetraeder mit Farbrahmen (Eigene Abbildung)..... 55

Abbildung 21: Farbauswahl und deaktivierte Farbe (rechts) (Eigene Abbildung) 55

Abbildung 22: Bildschirm mit Frage und Antwort (links), Antwort-Bildschirm (rechts) (Eigene Abbildung) 56

Abbildung 23: Bildschirmausschnitt Rundenübersicht (links) und Highscore (rechts) (Eigene Abbildung) 57

Abbildung 24: Anzeige der fehlenden Punkte bis zum Erhalt eines Gutscheins (Eigene Abbildung) 57

Abbildung 25: Web-Anwendung für AdministratorIn (Eigene Abbildung)..... 58

Abbildung 26: Die einzelnen Komponenten der Installation „Sights“ (Eigene Abbildung) 62

Abbildung 27: Projektionsaufbau Standort Kremser Landstraße (Eigene Abbildung) 70

Abbildung 28: Aufbau der Installation auf einer Bühne am Rathausplatz St. Pölten (Eigene Abbildung) 71

Abbildung 29: Bildschirmausschnitt der Online-Umfrage (Eigene Abbildung).... 72

Abbildung 30: Gruppe von Personen vor der Installation am Rathausplatz in St. Pölten (Eigene Abbildung) 75

Abbildung 31: Auswertung Fragen 1-2 (Eigene Abbildung)..... 76

Abbildung 32: Auswertung Fragen 3-4 (Eigene Abbildung)..... 76

10 Tabellenverzeichnis

Tabelle 1: Beispiele für URIs (Quelle: Langer & Roland 2010, S. 128)..... 40

Tabelle 2: Die wichtigsten REST-Anfragen (Eigene Abbildung) 67

Tabelle 3: Datum, Ort und Zeitraum des Feldversuchs 70

11 Listingverzeichnis

Listing 1: Verbindung zu einem WebSocket-Server (Quelle: vgl. Wang et al. 2013, S. 15) 44

Listing 2: Client Opening Handshake (Quelle: http://tools.ietf.org/html/rfc6455) . 44

Listing 3: Server Opening Handshake (Quelle: http://tools.ietf.org/html/rfc6455) 44

Listing 4: Verbindung zu WebSocket-Server, Aufruf des Callback „onopen“ (Quelle: in Anlehnung an Wang et al. 2013, S. 17) 45

Listing 5: Aufruf des Callback „onmessage“, Abfrage nach dem Datentyp (Quelle: in Anlehnung an Wang et al. 2013, S. 18) 45

Listing 6: Aufruf der Callbacks „onerror“ und „onclose“ (Quelle: in Anlehnung an Wang et al. 2013, S. 19) 46

Listing 7: Laden einer Bild-Datei in Node.js (Quelle: Eigener Code) 47

Listing 8: Verbindung zu einem WebSocket-Server (Quelle: http://socket.io/docs/) 48

Listing 9: Nachricht an alle verbundene Clients aussenden, aus Sicht des Servers (Quelle: http://socket.io/docs/) 49

Listing 10: Nachricht an alle verbundene Clients aussenden, mit Ausnahme des Clients, der das Event ausgelöst hat. (Quelle: http://socket.io/docs/) 49

Listing 11: Input Tag mit erweiterten Tags `accept` und `capture` (Quelle: http://www.w3.org/TR/html-media-capture/) 50

Listing 12: Abgreifen des Inhaltes im Input-Tag (Quelle: Code in Anlehnung an Kostiainen et al. 2014, o.S.) 51

Listing 13: Code-Beispiel mit der Funktion getUserMedia (Quelle: in Anlehnung an https://developer.mozilla.org/en-US/docs/NavigatorUserMedia.getUserMedia) 52

Listing 14: Directive `ng-controller` (Quelle: Eigener Code) 63

Listing 15: `div` mit Direktive `ng-show` (Quelle: Eigener Code) 63

Listing 16: Die Funktion `switchScreens` (Quelle: Eigener Code) 64

Listing 17: Data-Binding im Header in der Datei `index.html` (Quelle: Eigener Code)....64

Listing 18: Aktualisieren der Daten in `controller.js` (Quelle: Eigener Code)64

Listing 19: Übergabe der Daten vom Server (Quelle: Eigener Code) 65

Listing 20: Der Einsatz von `ng-repeat` (Quelle: Eigener Code) 65

Listing 21: Anwendungsbeispiel von LocalStorage (Quelle: Eigener Code) 66

Listing 22: Array, aus dem eine zufällige ID ausgewählt wird. (Quelle: Eigener Code)....68

Listing 23: Array, aus dem eine zufällige ID ausgewählt wird (Quelle: Eigener Code)....69

Listing 24: Aktion bei `disconnect`-Event (Quelle: Eigener Code) 69

12 Anhang

Auf der beiliegenden CD befindet sich der Sourcecode vom Projekt „Sights“, das in Kapitel 5 beschrieben wurde, im Ordner „sights“.

Alle Literaturquellen, die aus dem Internet stammen, sind im Ordner „literatur“ zu finden.