FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Investigating Browser-based Remote Controls for Interactive Applications on Public Displays

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Medieninformatik

eingereicht von

Florence Adegeye , BSc
Matrikelnummer 0825147

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Univ. Prof. Geraldine Fitzpatrick , PhD
Mitwirkung: Univ. Ass. Dr. Wolfgang Reitberger

Wien, 15. April 2015

Florence Adegeye	Geraldine Fitzpatrick

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Investigating Browser-based Remote Controls for Interactive Applications on Public Displays

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Media Informatics

by

Florence Adegeye , BSc
Registration Number 0825147

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Univ. Prof. Geraldine Fitzpatrick , PhD
Assistance: Univ. Ass. Dr. Wolfgang Reitberger

Vienna, 15th April, 2015

Florence Adegeye | Geraldine Fitzpatrick

Erklärung zur Verfassung der Arbeit

Florence Adegeye , BSc
Satzingerweg 8A/4/3, 1210 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 15. April 2015

Florence Adegeye

Dedication

To my beloved father Mag. Dr. Olayeni Adegeye who died unexpectedly and could not live to see my graduation at the university. He was a great man I will always look up to and I am deeply grateful that he was always there for me in every condition of life.

Acknowledgements

First of all I would like to express my gratitude to my advisor *Univ. Prof. Geraldine Fitzpatrick, PhD* for her engagement, the valuable feedback and continuous advise through the development process of this master thesis. Furthermore, I would like to thank *Univ. Ass. Dr. Wolfgang Reitberger* for initiating this exciting topic for my master thesis and providing the opportunity to work on a research project in cooperation with the research center *Forschungszentrum Telekommunikation Wien (FTW).*

Thanks to all my colleagues at *FTW* for sharing their knowledge and experience to help me in conducting this research project. I wish to thank for the good teamwork with *Dr. Peter Fröhlich* who manages the *ATREUS* project and *Mag. Stefan Suette* who spent a lot of time in preparing and performing the user study with all the participants. Above all, I would like to thank *Dr. Matthias Baldauf* who initiated the *ATREUS* research project and continuously supported me throughout the framework development and investigation process. I learned a lot from his expertise and it was a pleasure to work in this team. I am really thankful that he gave me the opportunity to contribute to this project and be part of this interesting scientific work.

Moreover, I would like to thank my friends who have willingly shared their precious time for participating in the focus group workshop. I am also thankful to my all relatives, friends and colleagues who have supported me throughout my studies with their enthusiasm and encouraging words. Special thanks goes to all current and former members of the *e-Business* team at my working place for their patience, the possibility to take my time to finish my study and for providing the devices for the framework tests.

My deepest gratitude goes to my parents *Mag. Gertrude Adegeye* and *Mag. Dr. Olayeni Adegeye* for always believing in me and supporting me to achieve everything I have aimed at throughout my life. They have paved the way for studying in Vienna and have always encouraged me in any situation. I will be grateful forever for giving me strength and support.

Kurzfassung

Als Vorzeigemodell des modernen und futuristischen Informationszeitalters werden immer häufiger große Displays in öffentlichen Räumen eingesetzt. Sie kommen nahezu überall in städtischen Gebieten vor, wie zum Beispiel in U-Bahnstationen, Einkaufszentren, auf öffentlichen Plätzen oder anderen Treffpunkten. Obwohl öffentliche Displays immer mehr an Bedeutung gewinnen, sind die meisten nicht für interaktive Anwendungen ausgelegt. Doch mit der Weiterentwicklung von mobilen Geräten kann man Information nicht nur passiv abrufen, sondern es werden auch neue Interaktionsmöglichkeiten für zum Beispiel Multiplayer Games oder interaktive Produktpräsentationen geschaffen. Jedoch sind die meisten mobilen Anwendungen systemabhängig und setzen einen aufwendigen Download- und Installationsprozess voraus. Dies stellt eine große Hürde für Passanten dar, welche spontan eine interaktive Anwendung mit ihrem eigenen Gerät an einem öffentlichen Display nutzen wollen.

Als Lösung für diese Herausforderungen wird im Rahmen dieser Diplomarbeit die Fähigkeit von Smartphones als web-basierende Eingabegeräte für interaktive Anwendungen auf öffentlichen Bildschirmen näher untersucht.

Das Ziel ist die Realisierbarkeit von herkömmlichen Interaktionstechniken mit aktuellen mobilen Technologien zu erforschen und die *User Experience* und die Benutzerakzeptanz für den web-basierten Lösungsansatz zu evaluieren. Im Gegensatz zu bisherigen Forschungsarbeiten beschäftigt sich diese Diplomarbeit mit systemunabhängigen Ansätzen, um die Nachteile von *nativen Apps* zu umgehen. Dafür wurde ein rein web-basiertes Framework entwickelt und mehrere Interaktionstechniken mit verschiedenen Smartphones getestet. Dadurch konnte die Unterstützung derzeitiger Webtechnologien mit aktuellen mobilen Browsern untersucht werden. Weiters wurde auch die Handhabung von web-basierten Smartphone-Steuerungen in einer Benutzerstudie analysiert. Der Schwerpunkt wurde dabei auf Spiele-Anwendungen gelegt, um das Benutzerverhalten und die Benutzervorlieben mit verschiedenen web-basierenden Steuerungen aufzuzeigen. Um noch weitere Anwendungsmöglichkeiten zu identifizieren, wurden unterschiedliche Anwendungsfälle in einer Fokusgruppe diskutiert. Wichtige Erkenntnisse daraus wurden als Leitfaden zusammengefasst und im Framework-Konzept mitberücksichtigt, um ein reibungsloses Benutzererlebnis zu garantieren.

Mit dieser Forschungsarbeit wurde gezeigt, dass eine Realisierung von herkömmlichen Interaktionstechniken mit Smartphones durch die Verwendung von aktuellen Webtechnologien effizient möglich ist.

Im Allgemeinen wurden web-basierte Eingabegeräte gut aufgenommen und von den Benutzern als sehr komfortabel wahrgenommen. Die empfundene Leistungsfähigkeit ist jedoch von den zu lösenden Aufgaben abhängig und die Anforderungen können sich aufgrund unterschiedlicher Anwendungsfälle verändern. Da die Entwicklung von mobilen Technologien immer weiter voranschreitet, sind somit anspruchsvollere Interaktionstechniken für eine verbesserte *User Experience* bei verschiedenen Anforderungen – vor allem im öffentlichen Raum – möglich. Die Verwendung von web-basierten Steuerungen für öffentliche Displays eröffnen uns somit neue Möglichkeiten in der urbanen Kommunikation unseres Informationszeitalters.

Abstract

The showpieces of a modern and futuristic information age are large public displays which decorate our urban landscape nearly everywhere e.g. in metro stations, shopping malls, at urban squares or many other meeting points. Although they have gained more and more popularity in our environment, most public screens are not designed for interactive applications. However, besides achieving information passively, with the further development of mobile devices new possibilities for interactions come up such as urban multi-player games or interactive product presentations. But most mobile applications are platform-dependent solutions which need an inconvenient download and installation process. These main issues are a major barrier for passers-by to spontaneously use an interactive application on public displays with the own mobile device.

To address these problems this master's thesis explores the potential of smartphones as browser-based input devices for interactive applications on public displays. The aim is to investigate the feasibility of present interaction methods with latest mobile technologies and to evaluate the user experience and user acceptance of this web-based approach. In contrast to previous researches this thesis provides platform-independent solutions to overcome the drawbacks of native applications. Therefore, a framework purely based on latest web technologies was developed and several interaction methods were tested on different smartphones to investigate the support of current web technologies by mobile browsers. Furthermore, the usage of web-based smartphone controllers was evaluated in a user study. It revealed the user behaviour and user preference of different web-based remote controls in a gaming context. For identifying more application contexts different use cases were discussed in a focus group. All findings relevant for achieving a seamless user experience were summarized in short guidelines and further considered in the framework design to ensure a great user acceptance.

This study showed that realizing present interaction techniques for using smartphones as input devices is efficiently possible with today's web technologies. Furthermore, the usage of browser-based remote controls turned out to be very comfortable and widely accepted by most users. The perceived performance for such applications depended on the certain task while the requirements can vary according to the use case. Since the support of mobile technologies are further improving, more advanced interaction methods are possible to achieve a seamless user experience for different challenges especially in public spaces. The usage of browser-based remote controls for public displays presents a new technical opportunity to enable new ways of urban communication in our digital age.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Motivation and background

Different kinds of digital signage technologies have become prevalent in today's information age to enhance our urban life. Particularly large public displays accompanies us nearly everywhere. We can find them in metro stations, airports, shopping malls, restaurants, museums, at urban squares or many other meeting points. Even in Austria this trend does not stop as we can already notice by new display installations projects called *Infoscreen*, *Digilight* and *Railscreen*.[1] For example, according to *Gewista Werbegesellschaft mbH* (an Austrian advertising company) 2.026 *Infoscreens* have been installed at public transportation hotspots in Austria (November 2014).[16] These examples show how public displays have already become an integral part of our environment and have gained more and more popularity especially for advertising or public relation purposes. This led me to delve more deeply into this topic and motivated me to investigate the potential of interactive applications on public displays in my master thesis.

Up to now most public displays are used to replace conventional information kiosks and can only provide information passively. But research shows that people are interested in interacting with urban screens by exploring information, manipulating or sharing content. Interactive displays encourage social collaboration and are useful to draw people's attention also known as the *honeypot* effect.[26] Especially the commercial industry benefits from the growing attractivity of large public displays and various marketing campaigns are emerging.[2] One interactive campaign from *DDB Stockholm*[3] in 2011 used the capabilities of today's smartphones as a remote control for a digital paddle game on a public display. With a web-based application called *"McDonald's: Pick N' Play"* passers-by were invited to control the paddle on a big screen by using the

[1] More information on: www.infoscreen.at, www.digilight.at, www.railscreen.at, Accessed: 2014-24-01

[2] Video examples of interactive marketing campaigns on http://www.awwwards.com/8-brilliant-interactive-marketing-campaigns.html, Accessed: 2014-27-03

[3] http://ddb.se/, Accessed: 2014-25-01

touchscreen interface of their phone.[19] But using mobile devices for interaction with large screens are not a promising technology for marketing purposes only. In my scientific work many other use cases and interaction methods with mobile devices were identified. According to statements from ABI Research (2013) mobile devices will become the main access point for applications at home, in the office or on-the-go over the next five years. 2.1 billion smartphones and tablets will include a functionality to connect to external displays through 2018.[41] Furthermore, today's mobile devices have enough processing power and with new sensor technologies they can be used as sophisticated input devices for displays. Because of the growing popularity and capabilities, mobile devices have become indispensable in the vision of ubiquitous computing. Especially smartphones are pervasive. For instance, 70,1% of the people in Austria are using the mobile internet via a mobile device outside the household or the working place.[4]
One main benefit in using smartphones as input devices is that many people are already familiar with their own device. The smartphone evolves into a personal and handy remote control for your environment that is always carried along. Because now everybody has its own device, an efficient support of multiple users interacting with the display will be possible. A rich set of applications with multiple users enables new user experiences and also enhances social interaction with new technologies in public spaces.[7, 43]

The provided interaction methods with smartphones as input devices can take on several forms. Remote systems can use traditional functionalities like SMS or voice communication for sending text-based control languages and cursor interactions or they can support more complex interaction operations with new technologies such as Near Field Communication (NFC). Further on they can use sensors for tilting or use cameras to create a pointing device.[13] Based on all this many other different interaction techniques were discussed in my thesis.

While most researches focus on remote controls developed for native smartphone applications [3] [13], I want to analyse the possibilities for cross-platform applications. One promising approach is the usage of browser-based remote controls. Through the further development of the current web standard *HyperText Markup Language 5 (HTML5)* by the *Worldwide Web Consortium (W3C)* [47], no installation of platform-dependent systems is necessary to access enhanced functionalities of a mobile device. One example mentioned by ABI Research for supporting touchscreen devices with a mobile website was already reported by *Taptu* in 2010 to investigate the new capabilities of HTML5. They predicted the following tendency of mobile app development in their report:

[4] Statistik Austria - the usage of ICTs in households, 2014, Last modified: 2014-21-10, Accessed: 2015-24-01

> *"Proponents of Mobile Web development cite a number of reasons why developers will increasingly favour Web-based services over apps. Unlike apps, which only run on single device platforms, Mobile Touch Web sites run under any mobile browser which is built on the open source Webkit components. With support for HTML 5 features already being rolled out in these browsers, it's getting easier and easier to create rich touch screen user experiences with the browser without having to create platform-specific apps."*[38]

The use of these web technologies allows a larger audience to control public displays and enables new interaction methods to enhance the user experience through our personal device. Interested people only need to access a web page through e.g. Quick Response (QR)-Codes, NFC tags or other short URL services and the remote display is controlled by the pre-installed browser on the smartphone. But there are still issues with the quality of new web standards which have to be further investigated to prove if they are suitable for interaction methods in remote systems.

All these interesting aspects encouraged me to study this topic in my master thesis and discuss several challenges and findings in using browser-based remote controls with interactive applications on public displays.

1.2 Problem statement

Many different input mechanisms and interaction models like touch screens or keyboards have been investigated but they are still too limited to achieve a seamless user experience. Moreover, not every interaction technique is suitable for open surroundings such as public spaces where a high pedestrian rate, big bustle, physical access, security and noise must be taken into account.
Direct interaction with widespread touch screens leads to usability problems with the large display surface. People struggle in finding the appropriate distance to the screen. It is difficult to achieve the right balance in getting close enough to the display for a comfortable interaction, and getting away from the display far enough to catch everything in the field of vision. Additionally the interface must always be directly accessible to the user which is sometimes not possible due to architectural, security or sanitary reasons. Other approaches like using gestures and speech to interact with the display are not effective and difficult to implement in public spaces which tend to be an audibly and visually noisy environment. Only a device-based interaction technique can ensure a reliable interaction between users and display. But providing publicly accessible devices is expensive and we still have to conquer physical security, sanitation and maintenance issues.[7, 26, 25]

A solution for these problems is allowing people to interact with remote screens by using their personal mobile devices e.g. smartphones or tablets. But other challenges are: offering an easy installation of a software on the mobile device and finding a reliable pairing method to establish a connection between the display and the device.

Therefore, a complicated and time-consuming configuration process is typically necessary.[37] Although today's development of native apps makes it easier to setup a software on the mobile device, the users still have to download and install an app on their own and this increases the threshold of use primarily. Furthermore, to maintain a working interaction with every device, different app versions have to be implemented to support several mobile operating systems e.g. Android, Apple's iOS or Windows phone. Installing the right app version is another hurdle for the user and requires higher affinity of mobile technologies. For facing all these challenges new concepts for enabling interactive applications on large display by using smartphones have to be developed.

1.3 Research aim

The central aim of my research is investigating and evaluating new web technologies to exploit the potential of today's smartphones acting as input devices for public displays. Therefore, a web-based framework to support different interaction techniques with the mobile device is designed and developed. The development of this framework helps to consider some technical aspects of possible interaction methods with smartphones which are further explored to evaluate the user experience of browser-based remote controls.

The following main research questions will be answered within this master thesis:

1. **What existing interaction techniques can be realized with state-of-the-art web technologies on smartphones?**
   In this part related work about interaction techniques with mobile devices is compared to analyse diverse controller approaches and today's device capabilities. Based on this literature research a web-based and platform-independent framework is created to primarily support common interaction techniques like touch and multi-touch events, accessing accelerometers and gyroscope and streaming media from a camera device. It will demonstrate how a web-based system for an interaction between large remote screens and a mobile device can be provided. Then all features are tested with different smartphones and browser versions to learn about the strengths and limits of web technologies used for smartphones acting as remote controls. The main goal is to evaluate all experiences with the framework to specify particular requirements and considerations for further development of web-based remote controls. After all the final framework should serve as a basic tool for creating one's own web-based remote control which can be used in more research projects or experiments.

2. **How do people use their smartphone as a web-based remote control for interactive applications?**
   This question focuses on the handling of smartphones as web-based remote controls and the users controlling a large screen with their own mobile device. With a prototype developed with this framework a user study was conducted to get a

better understanding of the user experience. The results should reveal which interaction techniques are preferred by users and what are their benefits of the web-based approach. Therefore, the interaction with four different gamepad designs for smartphones were observed and compared. In addition the learnings from the user study and the feedback of the participants are evaluated and relevant results are considered for reviewing the framework. The findings will provide significant aspects for further development and guidelines for different application issues.

3. **What use cases can be identified for using smartphones as web-based remote controls with public displays?**
   This study part deals with possible use cases for interactive applications on public displays which can be controlled with the own mobile device. The aim is to give an overview of many different fields of application in the public and to figure out a preferred use case. Within a focus group several use cases were discussed to gain an insight into people's need and their behaviour by using smartphones with large screens. Considering that the most favoured use case is taken as an example to analyse the application context. Furthermore, it will be evaluated how users can benefit from a web-based remote control in comparison to common approaches.

For answering these research questions all issues are analysed and elaborated in the following chapters of the master thesis. It considers both: the practical aspect in developing interaction methods with web technologies and theoretical issues about use cases and user experiences.
Parts of this thesis were conducted within the research project *Advanced Web technologies for Remotely Controlling Ubiquitous Screens (ATREUS)*[1] of the *Forschungszentrum Telekommunikation Wien (FTW)* and funded by *netidee*.[5]

[5]Team members: Dr. Matthias Baldauf (Principal investigator ATREUS and user study), Dr. Peter Fröhlich (Project manager FTW) and Mag. Stefan Suette (Test manager user study).

1.4 Structure of the work

This master thesis is organized as follows:

First of all in chapter 2 related work and literature is discussed and evaluated. It covers the topics about state-of-the art interaction methods with smartphones, general issues about controlling public displays with a mobile device and a review of latest web technologies. Afterwards in chapter 3 the methodological approach for answering the research questions is explained in detail. Chapter 4 focuses on the framework development. It explains used technologies, the properties of the framework and the technical test results with mobile browsers. Following this in chapter 5 the procedure and the results of the user study are elaborated. In chapter 6 possible use cases and the method including the evaluation of the focus group are described. The overall results and lessons learned from the framework, the user study and the focus group are summarized in chapter 7. Finally chapter 8 provides a short conclusion and future outlook about this topic.

CHAPTER 2

State of the art

2.1 Coupling of smartphones and public displays

Large displays and mobile devices have become increasingly part of our environment and our everyday life. In combining these sophisticated technologies new opportunities in ubiquitous computing are opened up to enhance public infrastructure. Beyond acting as passive, digital advertising posters, public displays can provide more interactive and freely adjustable content for entertaining or engaging passers-by. But due to their individual characteristics like different size, shape, form and purpose as well as public factors like location and audiences, various challenges have to be considered. The main issues are different input capabilities and the viewing distance which require adequate interaction techniques.[13, 9] One example for interactive displays are the *BBC Big Screens* projects launched in 2012 for about 20 towns across the United Kingdom. LED screens with a square of approximately 26 to 35 meters should encourage the audience to participate in public events like news, sports, music and movies. They could text comments, submit photos or play with interactive applications. Though every BBC display includes an on-screen camera for interactive content, for more challenging activities additional equipments and an expensive installation are still required.[1] [2]

A highly promising and more efficient way for interacting with public displays are smartphones because of their varied features. With their computational resources and storage they can wirelessly manipulate the screen's content even for interaction at a distance. In return smartphones can benefit from the large display size for tasks which are not well-suited for small displays and keyboards.[9, 37] It shows that both technologies perfectly complement each other – Pering et al. even talks about *"the spontaneous marriages of mobile devices and interactive spaces"*.[37]

---

[1] `http://www.capitalnetworks.com/clients/case-study/bbc-big-screen`, Accessed: 2015-25-01

[2] `http://downloads.bbc.co.uk/london/pdf/interactive_doc_v2.pdf` BBC Big Screen Interactive Applications - a technical guide, Spring 2009, Accessed: 2015-25-01

In [25] a user interface concept called *"dual display"* is investigated to combine the advantages of the input and output abilities of public displays and mobile devices. It pursues the approach to place some elements of the user interface on the large screen and some elements on the mobile device. For example, detailed information about a selected place on a map are only shown on the personal device while the navigation on the map can be followed by the audience on the public screen.
Basically we can differentiate between an *overt* or *covert* interaction: The overt interaction reveals the user who is currently controlling the display and shows other passers-by how the content is manipulated. This is common for most interactive displays that work with touch-based, gesture, mouse and keyboard interfaces. But mobile interactions with smartphones can also support covert interaction techniques because users can hide their identity with the personal device even in crowded environments. For analysing the user requirements for the interoperation between smartphones and large screens the audience of an interaction scenario in public places can be classified in the following groups:[25]

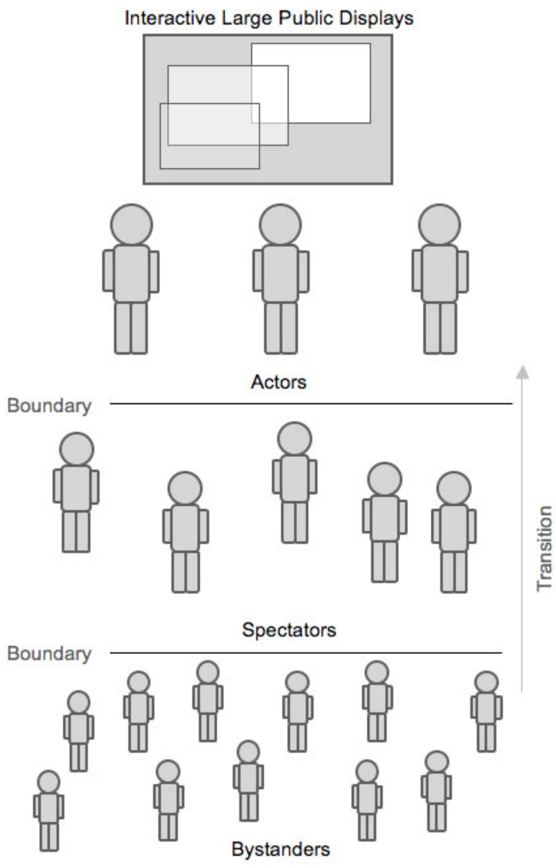

Figure 2.1: Actors, spectators and bystanders as users in a public space. [25, p.2]

- **Actors:**
  Persons who are playing an active role and are controlling the content on the display.
- **Spectators:**
  Passive observers who are following the interaction on the display but are not manipulating the content.
- **Bystanders:**
  People who are not interested in what is currently going on the display.

All needs of these stakeholders have to be already considered in the design process of interactive applications for public displays. The goals are to keep the *actors* engaged with the interaction and motivate the *spectators* and the *bystanders* to participate in the interactive play. Kaviani et al. summarized it as follows:

> *"It is thus important to encourage users to get engaged in interaction with these displays and provide them with a satisfactory experience so as to encourage continual use. This type of engagement usually requires special considerations for calm aesthetic, comprehension, notification, short-duration fluid interaction, immediate usability, shared use, and privacy"*[25]

Further design considerations that are relevant for interaction techniques with public displays and can be easily obtained with smartphones as remote controls are:
portability, *information security*, *dexterity*, *interruptability* and *social acceptability.*[7]
Moreover, to achieve a seamless interoperation between smartphones and public displays also a reliable and easy pairing process is essential. Basically we can distinguish between an *intentional* and an *unintentional* initiation process by the user. An unintentional interaction requires the public display to sense the presence of a user and starts a pairing process.[7] But in this thesis the focus are on methods where passers-by have to actively initiate the interaction. For instance, Pering et al. developed the *Elope system* for initiating a connection between devices and for configuring the interactive application. Before connecting with the system the user has to scan tagged objects in the environment with the Radio-frequency identification (RFID) reader embedded in the mobile phone. This object contains information about the user's intention (e.g. "show photo album"- object for presenting a photo album on a large screen) and triggers the corresponding interaction for importing data from the mobile device. This approach is suitable for any presentation purposes because no complicated integration tasks are needed on start-up.[37]
But in case of public spaces many other pairing issues such as distance, locally infrastructure, performance and acceptance for connecting by passers-by have to be taken into account. Only a very low threshold of use leads an interactive application to success – also referred as *"serendipity": the user's ability to spontaneously interact with a large display.*[7] In the comparative study of Baldauf et al. the potential of different pairing techniques for public displays were investigated in a user study. The participants had to retrieve a screen identifier with their smartphone which then was sent to a computer over

Wi-Fi to establish a connection with the desired display. This screen identifier could be recognized by the smartphone through the following pairing methods: [4]

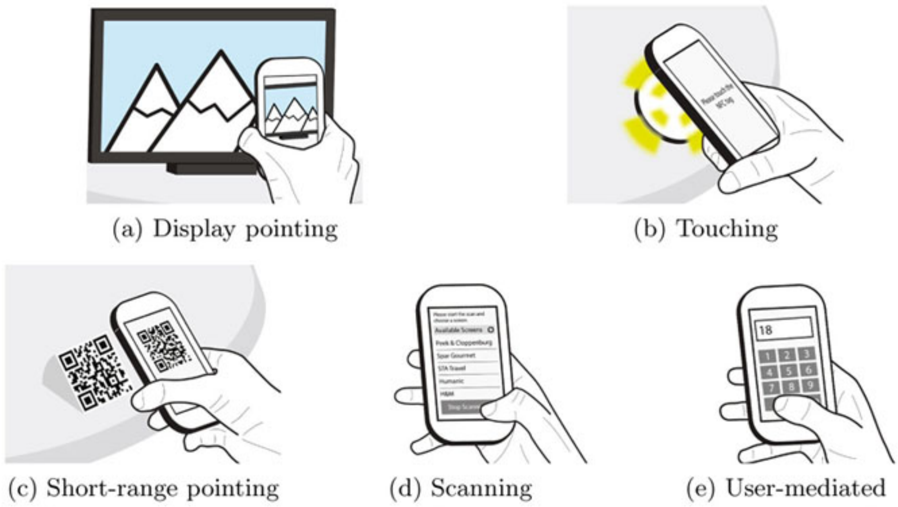

Figure 2.2: Overview of different pairing methods. [4, p.33]

- **Display pointing:**
The connection is established by a pointing gesture where passers-by target the public display with the mobile camera (cf. figure 2.2 a). For this task the augmented reality toolkit *"Qualcomm's Vuforia"* for mobile devices was used. It has markerless image recognition abilities based on natural image features to identify the background images on the screen. Each predefined image is associated with a screen identifier for the connection establishment.
The observation demonstrated that display pointing is robust to varying distances and angles, so no precise pointing to the target is needed. It also showed a high responsive speed and was stated to be efficient and more fun than conventional techniques. But many participants claimed this pairing method as too fast and inaccurate. This is because of the perceived lack of control during the interaction and the fear to make a wrong selection. Thus further improvements e.g. the integration of an additional confirmation step must be developed to address this problem.

- **Touching:**
The smartphone has to be brought into close proximity of about three centimeters to a smart object e.g. a NFC tag to read the screen identifier (cf. figure 2.2 b). It is similar to the concept with the RFID tag described by Pering et al. but with the new extended NFC technology. Such NFC modules are already integrated in latest smartphones, so a simple implementation is possible with e.g. the standard Android framework as done in [4].

In comparison to display pointing the gesture of this method was not sensed as very intuitive by the users. It required a special hand gesture to position the smartphone over the NFC tag. But due to the growing omnipresence of mobile services (e.g. mobile payment with contactless NFC [3]), many people are already familiar with this procedure. Although this technique is also considered as very fast, the main drawback remains that in a public context the passers-by are forced to walk to the display.

- **Short-range pointing:**
  Unique visual markers e.g. QR-Codes at the public display are captured with the smartphone camera (cf. figure 2.2 c). The screen identifier is gathered by decoding the QR code. The solution for the user study operated with the *ZXing* software library [4].
  The test revealed that many participants struggled with capturing the QR code because it was difficult to target the visual marker. Thus this method needs a high accuracy in pointing which lowers the learnability and fun of the interaction.

- **Scanning:**
  With the Bluetooth feature of mobile devices the environment is searched for screens nearby to create a list with the names of available displays. To establish a connection, the user can select the desired display by touching the name on the smartphone (cf. figure 2.2 d).
  Whereas this method is easy to learn and use, it needs several interaction steps in comparison to the other approaches. But it is very useful in situations where many displays are placed in long distances.

- **User-mediated pairing:**
  The user has to enter a numerical code on the smartphone to access the desired public display (cf. figure 2.2 e). It is a trivial manual input process with traditional on-screen keyboards which is very easy to learn. However many test persons regarded this method as less innovative and funny to use. Furthermore, it requires properly reading and knowing of the corresponding code which leads to the problem how to optimally inform passers-by about the pairing method.

Summarized the key factors for motivating passers-by to start an interaction are: the ease of connection, speed, fun and convenience. According to Baldauf et al. [4] *display pointing* with some modifications regarding the too high sensitivity would be a suitable technique for public displays because it shows the best results in learnability, perceived performance and distance issues. In context of mobile interaction long-distance techniques such as display pointing, scanning and manual input are generally preferred over pairing methods which require immediate proximity to the display. People are not forced to break through a crowd to reach the public display and they can control the interactive

[3] Mobile Contactless payment with Visa (http://www.visa.co.uk/products/visa-contactless/mobile-contactless), Accessed: 2015-07-02

application from a position in the background. It is also possible to establish a connection while heading towards the screen. This shortens the walking distance what seems to be a faster connection process for some users.
Another aspect for the interoperation between smartphone and public display is the combination of different pairing and interaction techniques. In the work of Baldauf et al.[4] some combinations like e.g. pairing with manual inputs in portrait mode and interacting with "Mini Video" in landscape mode was one point of criticism during the user study. The main problem was switching the device orientation from the connection phase to the interaction phase which is uncomfortable for a seamless user experience. As a consequence an appropriate interaction technique depending on the use case is needed and affects the choice for the proper pairing technique in a specific application field. But what new scenarios can be enabled by using personal mobile devices as remote-controls for public displays?

Generally a rich set of applications has emerged through the new sophisticated and further developing features of smartphones. In the work of Sarah Clinch several use cases where smartphones can play an important role in combination with public displays are classified as follows: [13]

- **Personalization:**
  The content shown on a public display depends on personal interests and user preferences. Passers-by can manipulate the published content on the screen with their mobile device and communicate with displays nearby. Early approaches used Bluetooth as a main communication technology but to support more flexible personalization features new systems with a combination of Wi-Fi fingerprinting, GPS and Bluetooth were developed. For instance, the *Tacita*[13] application collects user preferences from the smartphone and sends the personalized information to a cloud-based display application on a nearby screen. With this the user can e.g. request a weather update based on personal data and show it on the screen.

- **Co-Displays and Cyberforaging:**
  To overcome the limited screen size of smartphones, the content is transferred to nearby displays for a better overview or for sharing information. A commercial example is the system *Miracast*[4] which allows users to share content over Wi-Fi from any mobile device with a TV, display or projector nearby. Other use cases for smartphones used as private Co-Displays are "individual views". While some parts are hidden on the public display, the user can simultaneously obtain private and sensitive data on the personal mobile device.

[4] `http://www.wi-fi.org/discover-wi-fi/wi-fi-certified-miracast`, Accessed: 2014-08-02

- **Information Take-Away:**
  Information from a public screen can be stored on the smartphone with e.g. QR codes, RFID tags or Bluetooth. In [26] different interaction models for Bluetooth-based "information pick-up" with mobile devices were analysed and live tested with the prototype *BlueInfo* in a campus setting. As an example of an Android application is *Digifieds* [5] where advertisements are transferred from a display to the smartphone via QR codes.

- **Interaction:**
  Smartphones act as remote controls for interactive applications on public displays. In several scientific works different kind of interaction techniques are explored from basic communication functions e.g. SMS to complex interaction patterns with integrated sensors whereby most research focuses on the mobile as input device and the display as output device. More Details about interaction techniques relevant for the topic of this master thesis are discussed in the next chapter 2.2.

This classification mentioned in [13] shows the wide range of possible use cases for the interoperation between smartphones and public displays. On the basis of all these aforementioned examples different application areas were collected and elaborated in a brainstorming process. As a result different use case scenarios were summarized in a mind-map shown in figure 2.3. Principally for the interaction with large screens we can distinguish between three application domains: personal (displays referring to a single user), semi-public (displays accessible to a limited amount of people e.g. office building) and public (displays in open or crowded environments e.g. theme parks, train stations).[7] For further considerations in this master thesis we focus on use cases relevant for interaction with public displays.

[5] http://digifieds.ubioulu.fi, Accessed: 2015-31-03

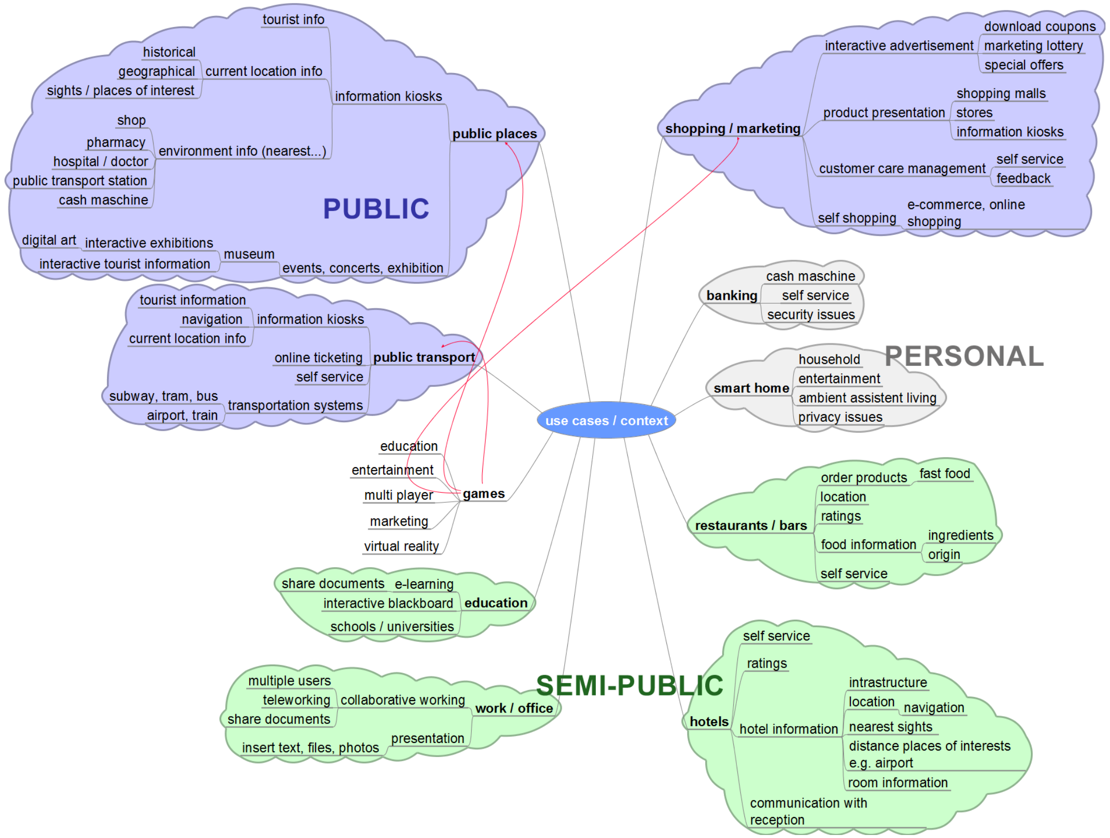

Figure 2.3: Mind-map with different use cases based on the literature of S. Clinch.

The topic *games* is particularly suitable as a representative use case because it covers several areas and presents different challenges for interaction regarding speed and accuracy. As well games are very popular in general and can provide entertainment to both active or passive participants in public spaces. That's why most related work focuses on scenarios with games like the concept for a vision about interaction with mobile devices created by Ballagas et.al [7]. The storyboard shown in figure 2.4 illustrates a gaming scenario at a subway station where the user *Tom* challenges an unknown person with the nickname *ACE* to play a *PacMan* game on the large screen. Therefore, *Tom* simply takes his mobile phone and joins the game by scanning the visual marker attached next to the public display.[7]

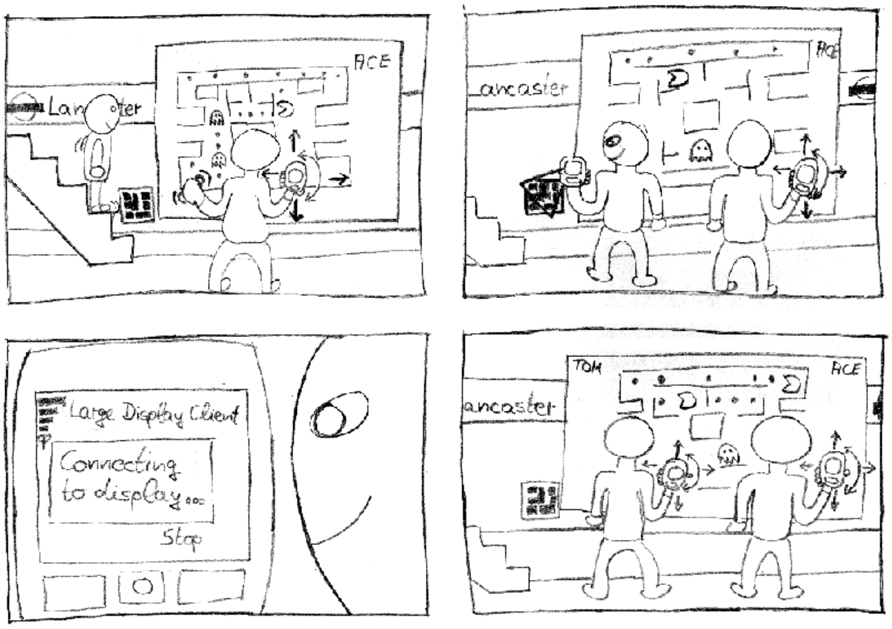

Figure 2.4: A storyboard illustrating envisioned interactions between mobile phones and large public displays. [7, p.6]

Referring to this vision the main research questions mentioned in chapter 1.3 will be investigated in this master thesis. Besides identifying different use cases, the selection of an appropriate interaction technique for mobile devices acting as remote-controls is also essential for a seamless user experience.

> *"Prototype tabs, pads, and boards are just the beginning of ubiquitous computing. The real power of the concept comes not from any one of these devices; it emerges from the interaction of all of them." (Mark Weiser)*[5]

2.2 Interaction techniques with mobile devices

In scientific literature we can find a lot of different approaches and discussions about interaction with public displays by using a mobile device. What mainly differentiates them are the use of different phone features, the interaction techniques depending on the supported sensor technologies and the pairing methods for communicating with the remote display. But so far no approach for web-based remote controls interacting with public screens have been taken into account. Currently various native apps acting as remote controls are emerging[6] but they all need a platform-specific installation and are not purely based on latest web-technologies. With the evolution of mobile technologies new possibilities in developing interaction techniques have opened up and require further investigations.

In previous research various experiments with the simple features of first mobile phones reveal the high potential of mobile interactions with large screens. Ballagas et al.[5] analyse smart phones as ubiquitous input devices for large displays and define a categorization of different interaction styles and tasks. Interactions can be divided into the following key tasks: position, orient, select, path, quantify and text entry. For every task further distinctions are made between: dimensionality, direct versus indirect interaction styles, relative versus absolute measurements and continuous versus discrete feedback from the environment. This taxonomy of Ballagas et al. is summarized as follows:[5]

Position:
The aim of this task is to place an element at a specific position on the screen. The interaction technique can be continuous if the object's position is changing continually. Or discrete if the position is changing at the end of the task.

Orient:
The orientation of an object (e.g. a security camera) is defined in the coordinate system of the display. This interaction can also be continuous (e.g. changing orientation with a velocity controlled joystick) or discrete (e.g. changing orientation by entering an angle).

Select:
A certain object or command is chosen from a set of alternatives e.g. the selection of a command icon from a menu by using a cursor. Possible interaction techniques for selection are: direct picking with cursors, pointers or cameras, typing in a command prompt, pushing a key button and voice or gesture recognition.

Path:
This task is a combination of position and orientation series which are always continuous and in a closed loop.

[6] 5 Apps to turn your phone into a universal remote (http://gizmodo.com/5982909/5-apps-to-turn-your-phone-into-a-universal-remote), Accessed: 2014-23-03

Quantify:
A task to specify a value or number within a range of numbers e.g. Graphical User Interface (GUI) elements for numeric input.

Text:
Traditional interaction task with simple text inputs.

With this segmentation different interaction methods can be compared to evaluate which techniques are most efficient for particular interaction scenarios. A short overview about the classification of mobile phone interactions is given in figure 2.5.

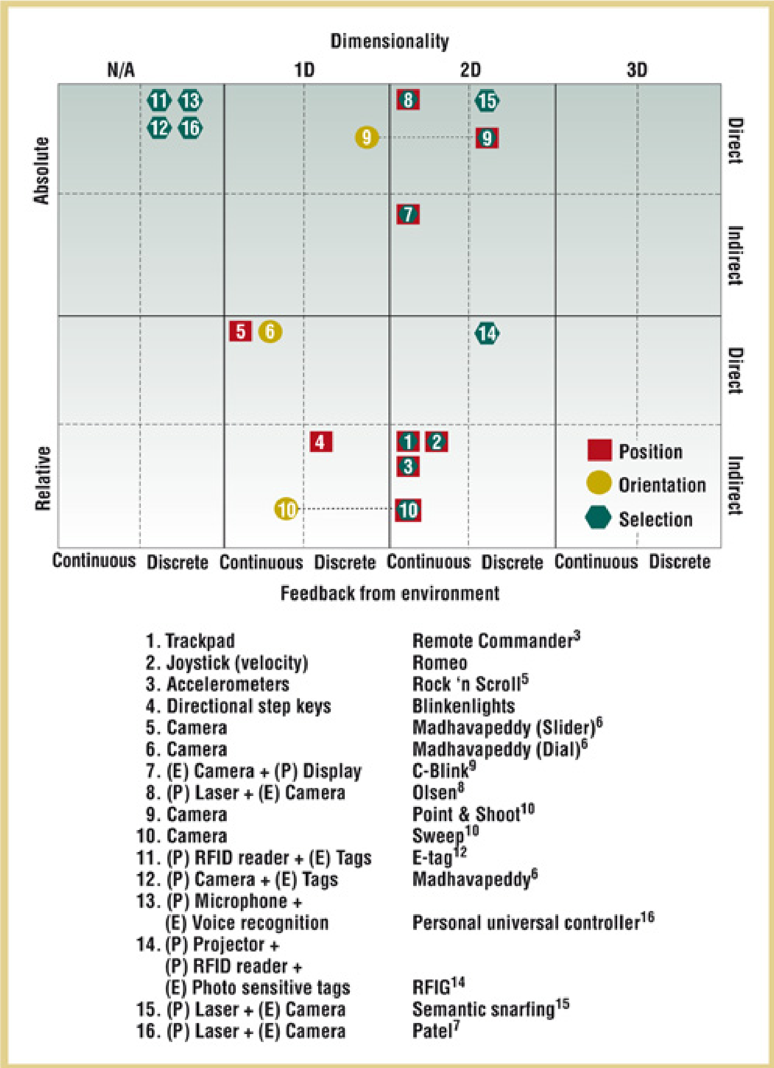

Figure 2.5: Classification of mobile phone interactions according to Ballagas et al.[5, p.76]

Considering example number 3 (cf. diagram 2.5), interactions using *accelerometers* allow a *continuous indirect* translation for positioning tasks such as users can scroll the screen by tilting the device. This indirect and relative technique is further evaluated within the project *Rock 'n Scroll* in [5]. Similar tasks are very common for motion sensors in mobile devices even for very early developments like Samsung's SCH-S310 phone[7] as a representative of first gesture phone models. Today's modern smartphones provide more sophisticated technologies which are useful for serendipitous mobile interactions. Based on the following literature review we can divide the technologies in:
motion gesture-based, *surface* or *touchscreen gesture-based*, *camera-based*, *touch-based* and *Bluetooth-based* interaction approaches.

For *motion gesture-based* interaction methods the movement of a mobile device can be easily tracked with new integrated sensor technologies like accelerometers, gyroscopes and orientation sensors. This allows gestures in three dimensions e.g. rotating or translating the device to mimic a more natural usage behaviour instead of interacting on a screen with a finger or button. Depending on a specific application some tasks can benefit from motion gesture interaction. For example, motion can be mapped to a tilting operation for selecting menu items or manipulating 3D objects. In this way interaction can be simplified and also enables interaction for users who are not able to interact with the touchscreen of the device e.g. when wearing gloves.[42] According to Ruiz et al. the social acceptability of motion gestures is fairly high although people still have concerns how bystanders would interpret the intention of certain gestures. The results of the guessability study by Ruiz et al. showed that 82% of the participants would use motion gesture at least occasionally.[42] They have defined possible motion gestures in a user-defined motion gesture set which is summarized in figure 2.6. Relevant gestures for interacting with a public screen are "flicking" movements like zooming, panning and direction movements (e.g. right/next or left/previous). The main problem of using gestures is to differentiate from everyday movements with low kinematic impulse. Furthermore, depending on the sensor sensitivity of the smartphone it is often difficult to distinguish from one gesture to another. This causes a high false positive rate which further leads to user frustration and poor user experience.[42]

Interaction solutions with *surface gestures* can overcome these problems. Modern smartphones acting as mobile surface computers allow users different *touchscreen* interaction techniques on the personal device in two dimensions. With the widespread familiarity in using touchpads on several devices touchscreen gestures have become intuitive and are very easy to learn. The multi-touch capability of smartphones enables several on-screen operations like panning, zooming, rubbing and tapping whose properties are often evaluated in many studies regarding touchscreen displays e.g. in [34]. But the key challenge beside a precise and rapid interaction is finding the appropriate Control-Display gain (CD gain) to avoid clutching while moving an object.

[7] (Video) Samsung SCH-S310 Demonstration, Accessed: 2014-23-03

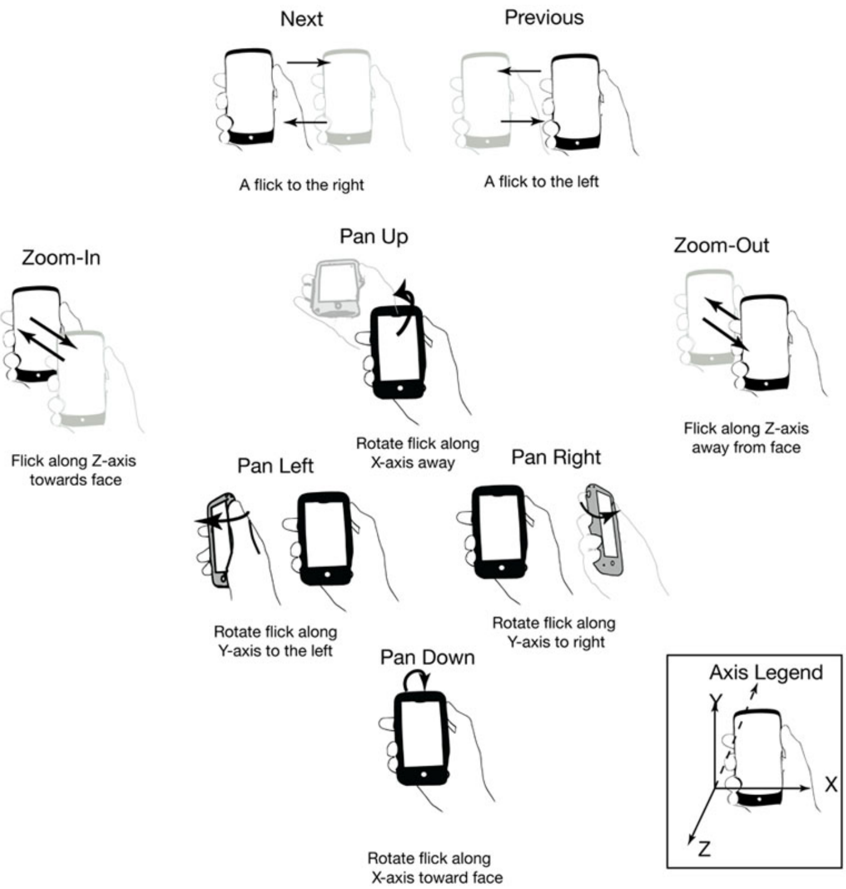

Figure 2.6: A user-defined motion gestures set with flicking movements. [42, p.204]

The value of the CD gain should amplify the finger movements so that small movements on the phone's display result in large motions of an object on the public display. With the *Absolute and Relative Cursor - Pad (ARC-Pad)* McCallum et al. [30] developed a technique for mobile touchscreens where absolute and relative cursor positioning are combined. By tapping on the pad the cursor instantly jumps to the corresponding location on the screen (absolute mode) while sliding the finger along the touchscreen invokes a relative motion (cf. figure 2.7). Their results show that absolute positioning reduces the clutching by half and users were faster in selecting targets in comparison to cursor acceleration methods.[30] In the comparative study of Baldauf et al. [3] different interaction methods including "*Touchpad*" are investigated for the tasks: targeting, drag 'n' drop and drawing. The touchscreen approach revealed a high accuracy especially in selecting or drag 'n' drop small objects but was significantly slower than direct interaction techniques for targeting and drawing. As a result touchpad mechanisms are well-suited for controlling small user elements on the screen or for accurate dropping tasks e.g. precisely placing tiles in a puzzle game.[3]

Figure 2.7: (a): Cursor is initially on the top right corner, (b): Tapping anywhere causes cursor to jump across the screen (absolute mode), (c): For accurate positioning the user can clutch and slide the finger (relative mode). [30, p.1]

Other methods which are interesting for controlling a remote display are *camera-based* interaction techniques. For comparing different approaches further work of Ballagas et al. [6] developed two new techniques *Sweep* and *Point and Shoot* which use the embedded camera on the mobile device.

- **Sweep:** The user can control e.g. a cursor on the display like moving an optical mouse with multiple degrees of freedom. The relative motion is determined by using optical flow image processing. (cf. figure 2.8)

Figure 2.8: Sweep interaction [5, p.72]

- **Point & Shoot:** This method works with visual codes which set up an absolute coordinate system on the display. For selecting an object the user has to aim with the cross hair of the camera image at the target on the public screen and press a joystick to confirm the selection. Then the phone can compute the target coordinates in the captured image. (cf figure 2.9)

Figure 2.9: Point & Shoot interaction. [5, p.72]

An other camera-based interaction with a marker-based attempt is described in the paper of Pears et al. [36] where the geometric mapping between the smartphone's images and the display is computed. Therefore, the public screen must show four distinctively coloured reference targets as markers for detecting the image with the smartphone. By computing the homography[8] the pixels of the public display can be mapped to the smartphone and vice versa.[36] Although marker-based approaches can provide accurate results, the dependence of this tracking signs limits a fluently interaction. People have to take care that the camera is always facing the markers which restricts their mobility. Another disadvantage is the placement of the markers. The markers must be placed around the display or are shown on the display thus valuable space on the screen is wasted.[9] An advancement of camera-based approaches with markerless computer vision

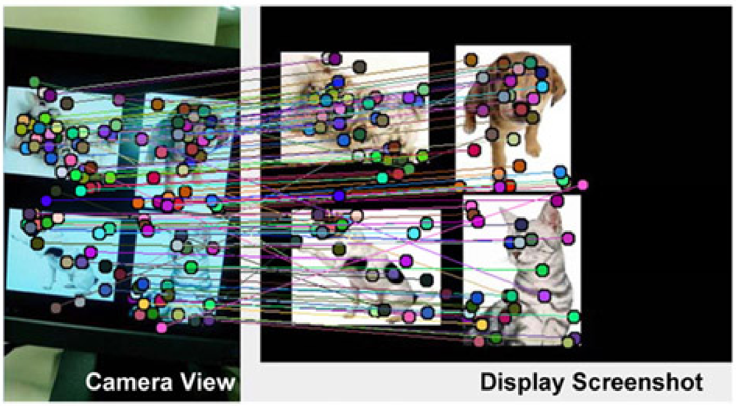

Figure 2.10: Interaction with an image matching algorithm. [20, p.455]

techniques is introduced in the article of Hu et al. [20] They have developed a software where the user interacts with the remote screen by capturing the target with a live video endued with augmented information. This cross-screen interaction is based on a visual, feature-based object matching and tracking algorithm.(cf. figure 2.10)

[8] Wikipedia - Homography (Computer vision) (http://en.wikipedia.org/wiki/Homography_%28computer_vision%29), Accessed: 2015-16-02

A similar real-time markerless tracking system is developed by Boring et al. [9] to use *virtual projection* as an interaction technique for public spaces. With the *Fast Retina Keypoint (Freak)* algorithm feature points are extracted from the template image (the display's content) and the captured image (a frame of the phone's live video) (cf. figure 2.11). Then the features of both images are matched to calculate a transformation matrix for deriving the spatial relationship between the screen and the device. This attempt with image features also works with subregions, so the camera does not need to capture the full display view. This tracking system was realized in the *Touchable Facades* application for the Apple devices iPhone 4, 4S and 5. It was tested on the media facade of the *Ars Electronica Center*[9] in Linz, Austria. Passers-by could change the color of the facade by aiming their smartphone at the building and "touch" it in the live video on their device. (cf.figure 2.12)[9]

Figure 2.11: The tracking system based on the display's content. [9, p.30]

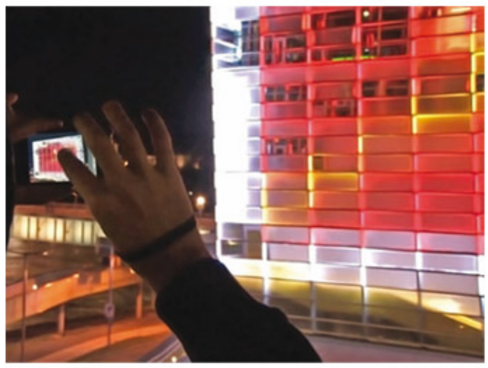

Figure 2.12: "Touchable Facades" application, [9, p.33]

With the increasing equipment of sensor technologies in smartphones *touch-based* interaction techniques are becoming more important. Several explorations with NFC or RFID tags are described in the research of Hardy et al. [17, 18]. They have realized a new interaction technique called *Touch & Interact* where an NFC phone is used as a direct input device for a large display. To perform an action, the person just have to touch the corresponding point of interest on the screen with the mobile device. Therefore, the display is equipped with a grid of NFC tags to recognize the part of the display which was touched by the user. The information stored in the NFC tags (e.g. all information about a specific region on a tourist map, cf. figure 2.13) can be read with the NFC phone.

All previously mentioned interaction methods are realized as native applications which have to be installed on the mobile phone or need extra sensors placed at the display. Facing this problem in the study of Dearman et al. [15] the authors have developed and evaluated a system called *BlueTone*.(cf. figure 2.14) The application requires a

[9] Ars Electronica Center (http://www.aec.at), Accessed: 2015-16-02

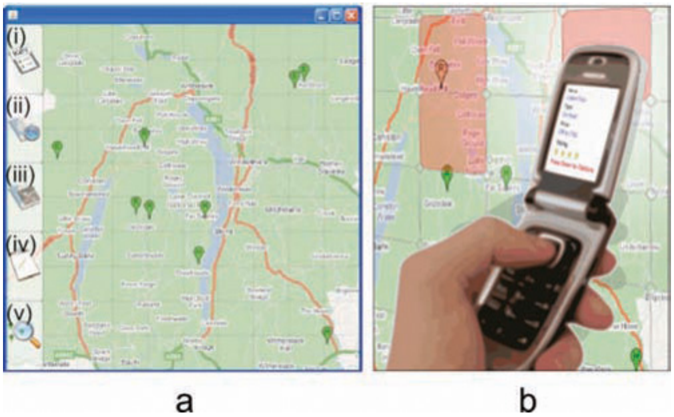

Figure 2.13: Touch & Interact: interaction technique with NFC. Example of a tourist guide prototype:(a): information showed on the public display, (b): mobile phone selecting point of interest. [17, p.253]

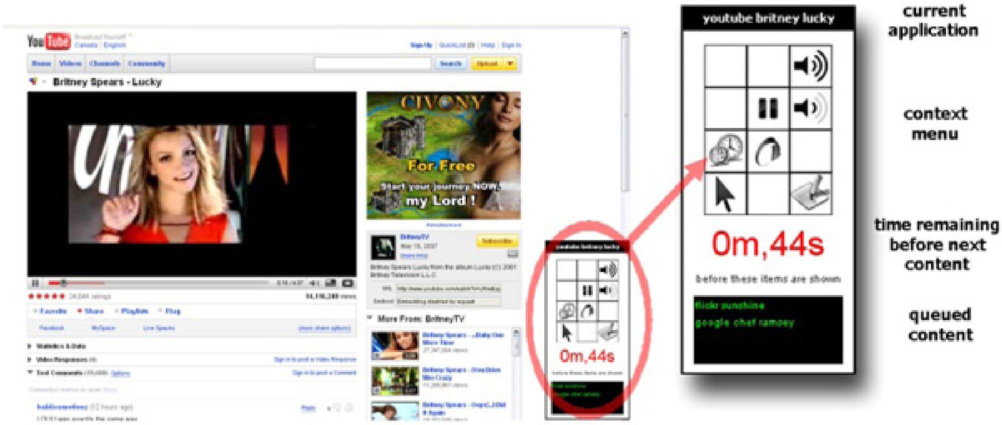

Figure 2.14: BlueTone: interacting with a YouTube video e.g. the user will press '3' to increase the volume and '5' to pause playback. [15, p.99]

Bluetooth-enabled phone but no installation of an extra software for e.g. text entry or cursor manipulation is necessary. A predecessor of this *Bluetooth-based* interaction technique is the *E-Campus* system in [14] where people can interact with a remote screen by manipulating the Bluetooth device name. However, *BlueTone* complements this concept by using the mobile phone's Bluetooth headset profile as an audio gateway to transmit predefined Dual-tone multi-frequency (DTMF)[10] sounds. The user generates the DTMF sounds by pressing buttons on the numeric keypad of the device.

[10] http://en.wikipedia.org/wiki/Dual-tone_multi-frequency_signaling, Accessed: 2015-01-04

The sound of the user's keypad actions will be interpreted by the display application and enables different interactions by reacting to certain audio patterns.
Although the usage of Bluetooth can avoid software installation problems, there are still complicated configuration and pairing issues left such as renaming the phone on startup or scanning Bluetooth devices. In the field study with *BlueInfo* [26] only 11,5% of the users claimed to have Bluetooth turned on while market reports suggest that more than 90% of devices would support Bluetooth. The main reasons for disabling this function were security and power consumption concerns.[26]
It follows from all these explorations that new web technologies are promising to achieve better and easier solutions for interacting with public displays by using the own mobile device.

2.3 Game interaction using smartphones

Since gaming is a very popular and challenging use case for interacting with public displays, related interaction techniques have been realized for a variety of gaming scenarios. Several commercial products are already emerging such as *GestureWorks Gameplay*[11] – a software tool for creating customized virtual game controllers on the own mobile device. Currently it provides various touch gestures and it only supports games running on Windows 8 and remote controls only available for Android devices. But these requirements restrict the number of potential users for interacting in public spaces. Therefore, a platform-independent solution such as web-based remote controls is recommended. For example, Luojus et al. [28] developed the mobile multiplayer game *Wordster* for public displays to evaluate the gaming interaction with mobile phones in a user study. It is an advancement of the word game *Boggle*[12], the aim is to find as many words as possible from a grid of characters within a limited time. In contrast to most mobile game applications *Wordster* uses purely web-based technologies. The game logic is implemented in Java and uses a standard client-server architecture. The user interfaces of the public display and the mobile phone are realized as websites using the new web technologies HTML5, Cascading Style Sheets Level 3 (CSS3) and JavaScript or jQuery. To start the game, passers-by have to establish a connection with their smartphone by scanning the QR code displayed on the public screen. The QR code contains information about the game session and the URL of the game server which is opened by the mobile browser to retrieve the corresponding gaming controller on the phone.[28] This approach enables a public game which works with most standard desktop and mobile web browsers and thus is available for a wide range of passers-by with different mobile devices. But due to enhanced smartphone technologies like touchscreens, accelerometers, GPS and cameras, more new sophisticated game interactions than picking some words on the display are possible.

The aim of game developers is to achieve a full immersion to improve gaming experience also with mobile devices. This is strongly influenced by the naturalness of the game controls depending on the chosen interaction mechanism.

[11] `https://gameplay.gestureworks.com` , Accessed: 2015-17-02
[12] `http://en.wikipedia.org/wiki/Boggle`, Accessed: 2015-17-02

While many related studies have focused on the investigation of latest control devices e.g. Nintendo's Wii, Guitar Hero guitar, Microsoft Kinect or Sony Move etc., in the work of Cairns et. al [12] the gaming experience arising from mobile devices has been further investigated. Their experiments dealt with the steering wheel metaphor for tilting controls in a racing game and they compared the three interaction styles touch, tilt and slip in a second study with a jumping game. The results show that tilting and slipping are more natural mappings than the simple touch interaction for this kind of game.[12] A further survey by Oshita et al. [35] compares the classical gamepads with touchscreen interfaces. Latest game consoles such as *Nintendo Wii U*[13] – a classic gamepad with an integrated touchscreen and inbuilt accelerometers – already combine the advantages of both technologies which enables new interaction styles (cf. figure 2.15)

The explorations in [12] stated that touchscreen interfaces are better or similar to the gamepad interfaces regarding the selection time and number of errors during the interaction task. The design of the interface (touchscreen with many, small buttons in a 6x4 grid versus a touchscreen with big buttons in a 3x2 grid) and the action methods (pushing a single button, pushing two buttons simultaneously or in a sequence) affect the interaction performance.[12] Based on these insights similar interactions can also be achieved with smartphone game controllers which are more available and well-known for more people as part of their everyday life. Katzakis et al.[24] realized three interaction techniques for 3D cursor positioning by using the smartphone's accelerometers, magnetic and gyroscopic sensors, and touchscreen. Hence, it is a controller comparable with the Wii U but with more personal data and always carried along by many users. The following smartphone control techniques shown in figures 2.16 - 2.18 were evaluated in [24]:

Plane-Casting:

A plane is cast inside the game scene and follows the orientation of the device. The user can translate the cursor or object on the plane by moving the finger on the touchscreen. In doing so the moved object is always snapped to the plane during the translation. With this approach accurate curved motions in a defined path on the touchscreen or positioning occluded objects are possible. In comparison to the other two techniques the performance was considered the slowest but was the most relaxed method with less physical fatigue.

Fixed-Point Extended Ray-Casting:

The orientation of the ray is controlled by the orientation of the smartphone which allows to move objects along the ray in the depth axis. The origin of the ray is fixed below the camera position outside the viewpoint. The additional gesturing on the touchscreen enables a comfortable one-handed operation with the thumb.

[13] `https://www.nintendo.at/Wii-U/Wii-U-344102.html`, Accessed: 2015-19-02

Point-and-Move:

The object or cursor is moved with a constant speed towards the direction in which the smartphone is pointing when somebody touches the phone's screen. Most participants preferred this interaction method to complete the experiment task in steering an UFO through some gates. *Point-and-Move* was regarded as the quickest technique with the highest accuracy.

Figure 2.15: Wii U Touch controller.[33]

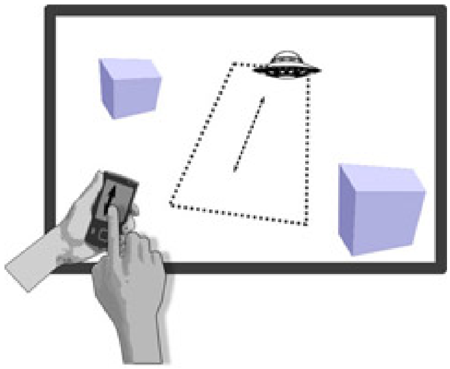

Figure 2.16: Plane-Casting. [24, p.3]

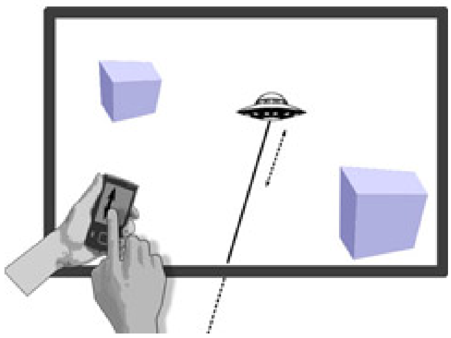

Figure 2.17: Fixed-Point Extended Ray-Casting. [24, p.3]

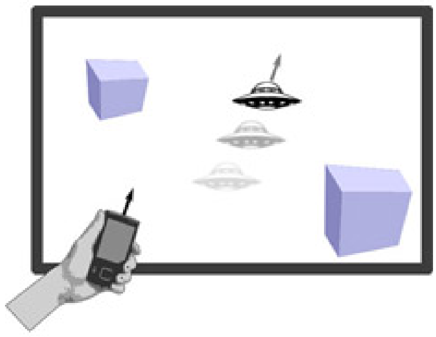

Figure 2.18: Point-and-Move. [24, p.4]

Based on these insights the provided interaction method has a significant influence on the performance and the user experience – in this context especially on the gaming experience.

To facilitate many different interaction techniques, a framework with a well designed architecture is essential for creating smartphone controllers in an efficient way. In many prior works frameworks were developed to evaluate mobile phones as input devices for games on large displays. Vajk et al [46] created the *Poppet framework* to imitate a “Wii-like” Controller with Nokia 5500 – one of the first mobile devices which supports tilting operations with inbuilt accelerometers. For accessing the 3D motion sensors the proprietary Symbian Sensor API was used to establish a low level socket connection. The retrieved sensor data from the phone was then forwarded to any J2ME application. The framework works with a wireless client-server communication over Bluetooth to control the motions of objects on the display. But the *Poppet* system has one main disadvantage: the client and server applications are strong platform dependent. The mobile application runs only on Symbian-based devices like old mobile phones whose manufacturers e.g. Nokia have already terminated the production of devices with this operating system. [14] In addition the game server was implemented in `C#`, as a result all applications only works on Windows machines. A further drawback is that the framework only supports a one-way communication using mobile phones as merely input devices.

To solve these problems, Malfatti et al. [29] realized the *BlueWave* framework. It is a similar system working with Bluetooth technology but was implemented entirely with Java and it supports a feedback mechanism by using the phone’s display as an output device. However, this framework uses the key layout of specific mobile phones to control the input which can vary from one phone to another.

A better and more modern architecture for today’s smartphone is discussed in [22]: The framework is also built with a client-server application but the communication takes place over Wi-Fi and the connection is realized with a Socket using the TCP/IP protocol. With this framework Joselli et al. developed the three input methods key touch, motion scheme and gesture touch and evaluated these interaction styles in a usability test with regard to ergonomic, fun factor, feedback, ease of learning and the player score. They found out that people with any level of experience prefer interaction techniques that are as natural as possible e.g. the motion scheme for a space shooter game. Another important factor for enhancing the immersion in a game is the possibility of feedback. Therefore, the architecture provides functions for sending data from the server to the client application on the mobile phone. Depending on the received data the device can render image or text on the phone’s screen or process in-built sound or vibration features e.g. sending a vibration command if the player is hit by an enemy. While the server of the framework in [22] is platform-independent, the client software for mobile interactions only runs on iOS devices like an iPhone, iPad and iPod Touch.

This means a completely platform-independent solution for accessing all built-in hardware resources is still missing. Currently a highly promising attempt is the usage of latest web technologies to provide browser-based interaction methods.

[14] `http://www.zdnet.com/article/nokia-says-final-sayonara-to-symbian-and-meego-apps-as-store-freezes-updates/`, Accessed: 2015-19-02

2.4 Web technologies for browser-based interaction

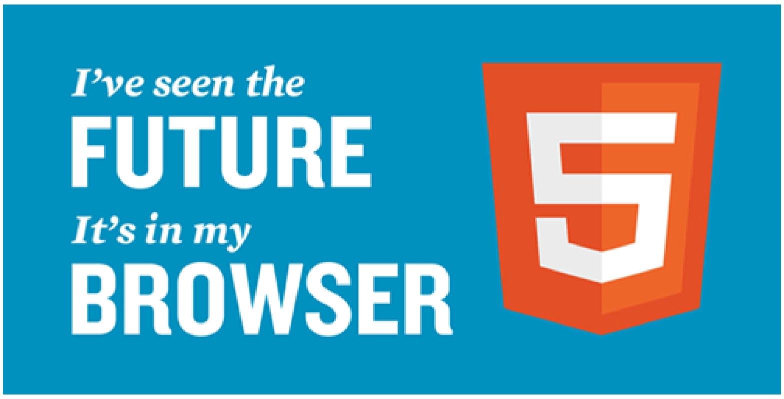

Figure 2.19: Official HTML5 logo sticker. [48]

> *"One of our goals is to move the Web away from proprietary technologies."*
> (Ian Hickson, Google researcher)

> *"HTML 5 is still a markup language for webpages, but the really big shift that's happening here – and, you could argue, what's actually driving the fancy features – is the shift to the Web [supporting applications]."*
> (Tim Berners-Lee, W3C director and inventor of the World Wide Web) [47]

With the further development of the HTML5 standard started since 2009 by the *W3C*[15] and *Web Hypertext Application Technology Working Group (WHATWG)*[16] a big discussion about the role of web technologies in mobile devices came up. HTML5 is not only a standard itself, it also defines web technologies in the combination of HTML5 (used for content), CSS3 (responsible for presentation) and JavaScript (defining the behaviour). Many researches and articles are investigating the potential of HTML5 apps in comparison to platform-specific applications for e.g. Apple iOS or Google Android - so called native apps. In [21] different technologies for mobile apps are evaluated concerning development criteria (e.g. programming language, debugging, deployment etc.) and functionality (e.g. hardware interface, web service interaction). Particularly interesting for browser-based interaction with smartphones is the hardware interface criteria which describes how to easy access the device's hardware features like camera, accelerometers, GPS etc. According to the results of Huy and Van Thanh HTML5 apps are with 4 from 5 points in the category hardware interface the second best choice for mobile app development after platform-based apps.

[15] `http://www.w3.org`, Accessed: 2015-20-02
[16] `https://whatwg.org`, Accessed: 2015-20-02

The main reason for this rating was that some important device features e.g. media capture were still a working draft by W3C in 2011. So until now the API was too limited and did not work in most browsers.[21] Critics are the opinion that the standardization process is too slow whereas other companies are already extending the HTML5 features in a way that the standard will never support.[47]

> *"In fact, while HTML 5 recently became a working draft, it's not expected to become even a W3C candidate recommendation until 2012 or a final W3C standard until 2022."*[47]

But lately advances have already exceeded the community's expectations and pave the way for new applications with HTML5 apps. 1.4 billion HTML5-capable mobile devices were predicted to be in circulation at end-2013 with an annual increase of 87% [40]. Previous forecasts from 2011 estimated a grow to 2.1 billion mobile devices with HTML5 browsers by 2016.[39] Additionally several experts and hardware vendors e.g. Intel and Samsung assured a deeper HTML5 support already in the chip and operating system level to achieve a higher performance of web apps like native solutions.[40]

> *"We expect HTML5 features in categories such as graphics, multimedia, user interactions, data storage, and others to be widely adopted sooner rather than later. A significant number of HTML5 features will be adopted in the mass market in the next three to five years"*(Mark Beccue, senior analyst 2011)[39]

Finally on the 28th of October 2014 the specification process was completed and HTML5 was released as a stable W3C Recommendation.[17] [18] Further developments are still in progress and current draft versions can be found on the official W3C Website (`http://www.w3.org/html/wg/drafts/html/master/`). The next steps are optimizations and an extended HTML 5.1 specification with a Recommendation released by the end of 2016. Another project "*HTML5 Apps*" with the explicit focus on mobile web applications funded by the European Union arose in January 2014 with the goal *"to close the gap between native and HTML5 apps through the standardisation of missing HTML5 functionalities"*.[19] The current versions of the standards for mobile web applications are published on `http://www.w3.org/Mobile/mobile-web-app-state/`. Primarily relevant technologies for web-based interactions with mobile devices are: (Accessed: 2015-20-02)

- **"Media and Real-Time Communications"**: Integration and manipulation of multimedia content e.g. capturing audio, video, canvas.

- **"Usability and Accessibility"**: Handling of different interaction and feedback mechanisms e.g. touch events, vibration, voice.

[17] `http://www.w3.org/2014/10/html5-rec.html.en`, Accessed: 2015-21-02

[18] `http://techcrunch.com/2014/10/28/w3c-declares-html5-standard-done/`, Accessed: 2015-21-02

[19] `http://html5apps-project.eu/about/`, Accessed: 2015-21-02

- **“Device Interaction”**: Interacting with the real world via sensors e.g. geolocation, motion sensors, camera, NFC.
- **“Network Integration”**: Provision of network connectivity and real-time communication e.g. WebSocket.

A complete list and API guide of the new HTML5 technologies are described in the Mozilla Developer Network[32], a short summary of features interesting for mobile devices is given in figure 2.20

Figure 2.20: Overview HTML5 Technology classes. [50]

2.4.1 Overview HTML5 features for smartphone controllers

Based on the previous technology review only some HTML5 specifications are discussed in more detail which are useful for web-based interaction methods with mobile devices and thus are relevant for the further investigation in this master thesis. Detailed code documentation and examples can be found on [32]. Implementation examples of browser-based interaction techniques are realized within the framework described in chapter 4.

Media Capture and Streams:

Feature	Maturity	Stability	Not supported by mobile browser
Capturing audio or video	Working Draft	Stabilizing	iOS Safari, IE

Table 2.1: API available on `http://www.w3.org/TR/mediacapture-streams/`
Last updated December 2014 [49]

This feature provides APIs for manipulating camera and microphones (MediaStream API), recording streams into files (Media Recorder API) and for accessing the camera to take pictures (Image Capture API).

SYNTAX

```
navigator.getUserMedia(constraints, successCallback,
errorCallback);
```

Table 2.2: More information on `https://developer.mozilla.org/en-US/docs/NavigatorUserMedia.getUserMedia` , (Accessed: 2015-22-02)

With this method 2.2 a *MediaStream* object is created which uses the camera or the microphones as source input. Some browsers need a prefix for supporting this function (e.g. Chrome, Opera Mobile: `webkit-`, Mozilla Firefox: `moz-`). Before accessing the media devices the browser always requests permission from the user. The Media Capture and Streams specification is a promising approach for *camera-based* interaction techniques with mobile devices.

Touch Events:

Feature	Maturity	Stability	Not supported by mobile browser
Touch-based interactions	Recommendation	Finished	—*

Table 2.3: API available on `http://www.w3.org/TR/touch-events/`
* Largely deployed, Last updated July 2014 [49]

For supporting *surface gesture-based* interaction this API offers the ability to interpret the finger activity on the touchscreen. It provides three interfaces: *Touch* (describes the touch point), *TouchEvents* (defines for which action the event is triggered), *TouchLists* (contains all touches associated with the event). An event is triggered when the state of the touches on the surface changes. The touch events can be handled by adding specific event listeners on the corresponding DOM-Element (cf. table 2.4). Possible event types are: “touchstart”, “touchend”,“touchcancel”,“touchleave” and “touchmove’. The touch feature is widely supported among all mobile browsers that’s why it is a very stable method for mobile interaction.

SYNTAX

```
EventTarget.addEventListener(type, handleEventFunction,
useCapture)
```

Table 2.4: More information on https://developer.mozilla.org/en-US/docs/Web/Guide/Events/Touch_events, (Accessed: 2015-22-02)

Vibration:

Feature	Maturity	Stability	Not supported by mobile browser
Vibration	Proposed Recommendations	Mostly stable	iOS Safari, IE

Table 2.5: API available on http://www.w3.org/TR/vibration/
Last updated November 2014 [49]

With this function the integrated vibration hardware in mobile devices can be accessed. A vibration pulse is triggered one time with the following command in table 2.6. The number of milliseconds defines how long the device should vibrate. Several vibration patterns can be created by using an array with more values instead of just one. With vibration users can get some haptic feedback while interacting with their mobile device. Especially for interactions with smartphones and large screens it can allow the users to stay focused on the display application without changing the glance.

SYNTAX

```
window.navigator.vibrate(pattern in ms);
```

Table 2.6: More information on https://developer.mozilla.org/en-US/docs/Web/Guide/API/Vibration, (Accessed: 2015-22-02)

DeviceOrientation:

Feature	Maturity	Stability	Not supported by mobile browser
Motion sensors	Last Call Working Drafts	Stabilizing, but planned updates	—*

Table 2.7: API available on http://www.w3.org/TR/orientation-event/
* Well deployed, Last updated August 2014 [49]

This specification allows to access the orientation and movement data from the mobile device. The information is obtained from integrated sensors such as accelerometers, gyroscopes and compasses. The API defines two JavaScript Events: "DeviceOrientationEvent" (triggered if the accelerometer detects a change to the device's orientation) and "DeviceMotionEvent" (triggered if a change in the acceleration has happened). Changes in motion can be received by registering the appropriate event listener (cf. table 2.8).

SYNTAX

```
window.addEventListener("deviceorientation",
handleOrientationFunction, useCapture);

window.addEventListener("devicemotion",
handleMotionFunction, useCapture);
```

Table 2.8: More information on https://developer.mozilla.org/en-US/docs/Web/API/Detecting_device_orientation, (Accessed: 2015-22-02)

The events contain the following properties which can be used as a condition to start a certain action: (cf. figure 2.21 - 2.23)

- `DeviceOrientationEvent.alpha`:
  The motion around the z-axis in degrees $[0°, 360°]^{\star}$; the device is twisted.
- `DeviceOrientationEvent.beta`:
  The motion around the x-axis in degrees $[-180°, 180°]^{\star}$; the device is tilted from front to back.
- `DeviceOrientationEvent.gamma`:
  The motion around the y-axis in degrees $[-90°, 90°]^{\star}$; the device is tilted from left to right.

- `DeviceMotionEvent.acceleration`:
  The amount of acceleration recorded by the device in m/s^2 upon the x- (west to east), y- (south to north), or z- (down to up) axis.

- `DeviceMotionEvent.accelerationIncludingGravity`:
  Same values as DeviceMotionEvent.acceleration but without compensating the influence of gravity. This is for devices that do not have a gyroscope integrated and therefore cannot remove gravity from the acceleration data.

- `DeviceMotionEvent.rotationRate`:
  The rotation rate of a device around each of its axis in degrees per second. The values are defined in *alpha* (rotating around the z-axis), *beta* (rotating around the x-axis) and *gamma* (rotating around the y-axis).

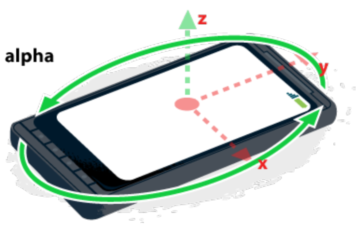

Figure 2.21: Alpha - rotation around the z axis. [31]

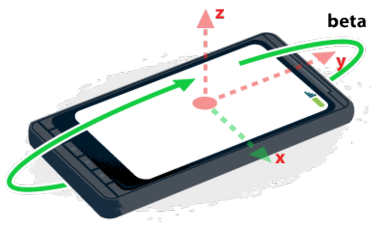

Figure 2.22: Beta - rotation around the x axis. [31]

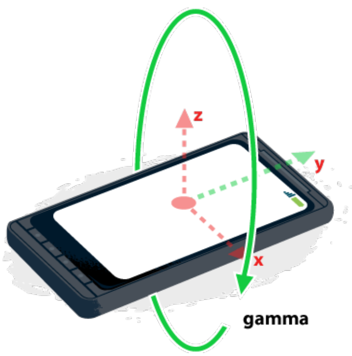

Figure 2.23: Gamma - rotation around the y axis. [31]

(* degrees based on Mozilla Firefox)

With this feature various *motion-based* interaction techniques are possible and offers greater opportunities for achieving a more natural user experience. But currently different browsers do not interpret the coordinates in the same way e.g. the values of `DeviceOrientationEvent.beta` on iOS mobile Safari are between $-90°$ and $90°$ but on Firefox $-180°$ and $180°$. These deviating values could cause some troubles during interaction with different devices and have to be considered in the mobile application.

The WebSocket API:

Feature	Maturity	Stability	Not supported by mobile browser
Bidirectional connections	Candidate Recommendations	Stable	—*

Table 2.9: API available on `http://www.w3.org/TR/websockets/`
* Good deployment, Last updated June 2014 [49]

This API enables mobile applications to use the *WebSocket protocol* for a bidirectional (full duplex) and more flexible communication than standard HyperText Transfer Protocol (HTTP) requests. HTTP is only one directional (half duplex) thus only the client can send a request to the web server. Until now this restriction was handled with the two methods *short polling* (making requests to the server every few seconds to detect a change) or *long polling* (making a single request to the server but keeping the connection open until the data has changed). But these workarounds cause unnecessary network traffic with excess server requests and an overhead in CPU usage on the web server.[45] In the article of Swamy and Mahadevan[45] the performance of web socket communication and the polling mechanism are compared. Figure 2.24 shows three different use cases with increasing numbers of clients receiving one message per second. The results confirm that long polling has a higher network throughput with the growing number of clients than a WebSocket connection e.g. 100.000 clients needs about 697 Mbps with polling, whereas WebSockets only needs 1,6 Mbps. This significantly reduces the network traffic and latency, so WebSockets have a better performance regarding the bandwidth and scalability. Furthermore, WebSockets do not require to sent headers for every request and response. This enables an efficient two-way communication where the client is talking to the server, while the server is talking to the client [45] (cf. figure 2.25).

A WebSocket communication is established by opening a standard HTTP connection to the server first and then the HTTP protocol is switched to the WebSocket protocol. This process is also known as the "WebSocket handshake" which automatically sets up a tunnel to pass through all network agents (e.g. proxies, routers and firewalls) and builds a persistent connection between client and server. The WebSocket connection works over the same TCP/IP connection only with a different URL scheme "*ws://*" or for secure connections over SSL/TLS with the "*wss://*" scheme. [45, 23]

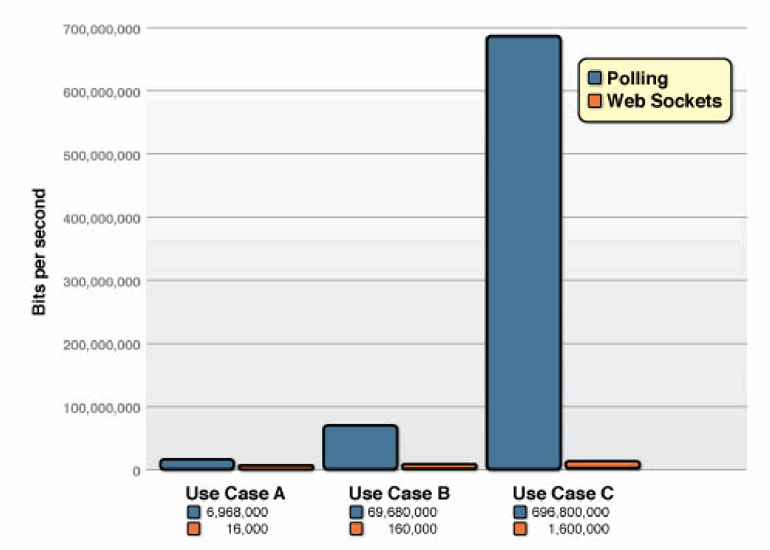

Figure 2.24: Comparison of network throughput overhead between the polling and the WebSocket applications. [27]

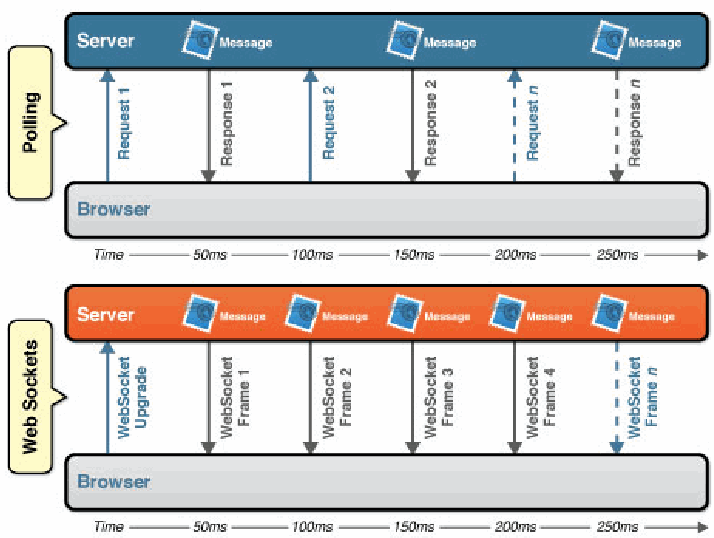

Figure 2.25: Latency comparison between the polling and WebSocket applications. [27]

The following basic methods and events are used for implementing a WebSocket connection (cf. table 2.10). For interacting with WebSockets the JavaScript API provides the events: *onopen*, *onclose*, *onmessage* and *onerror*. The browser can send or receive binary (Blob or ArrayBuffer objects) or text data (String objects) which have to be encoded to UTF-8 for transfer. This HTML5 feature lays the foundation for the interoperation

SYNTAX

```
WebSocket WebSocket(
        in DOMString url, //started with "ws://" or "wss://"
        in optional DOMString protocols
);

/*EventListeners for connection*/
WebSocket.onopen = function(event){
                alert("Connection open...");};
WebSocket.onmessage = function(event){
                alert("Received message " + event.data);};
WebSocket.onclose= function(event){
                alert("Connection closed.");};

/*sending data as string, Blob or ArrayBuffer to server*/
WebSocket.send(in DOMString data);

/*closing a connection*/
WebSocket.close(in optional unsigned long code,
                 in optional DOMString reason);
```

Table 2.10: More information on `https://developer.mozilla.org/en-US/docs/WebSockets`, (Accessed: 2015-23-02)

between smartphones and a public display. A reliable and a well performing connection is a basic requirement for supporting web-based interactions.

All these HTML features demonstrate that HTML5 apps are getting closer to mimic the behaviour of native apps and have access to different hardware features. However, some specifications are still in the draft stage. The feature support cannot be guaranteed to be consistent across all mobile browsers. Different rendering engines (e.g. WebKit vs. Gecko) and varying HTML5 and CSS3 skills can cause different behaviour in layout and performance.[44] An overview about HTML5 compatibility on mobile browsers is summarized on `http://mobilehtml5.org` (Last updated on December 2014, Accessed: 2015-24-02).
But the key advantage remains that no manual installation is necessary and this can be utilized for a seamless user interaction. The next chapters are going to analyse interaction approaches and user preferences for smartphones as remote controls based on purely web technologies.

CHAPTER 3

Methodological approach

This chapter gives a short overview about the study process and the methods used to analyse the topic of this thesis in more detail. Basically the elaboration of this work is divided into three main parts: the framework development, the user study and the focus group. The research process and the relation between the results of the study parts are shown in figure 3.1.

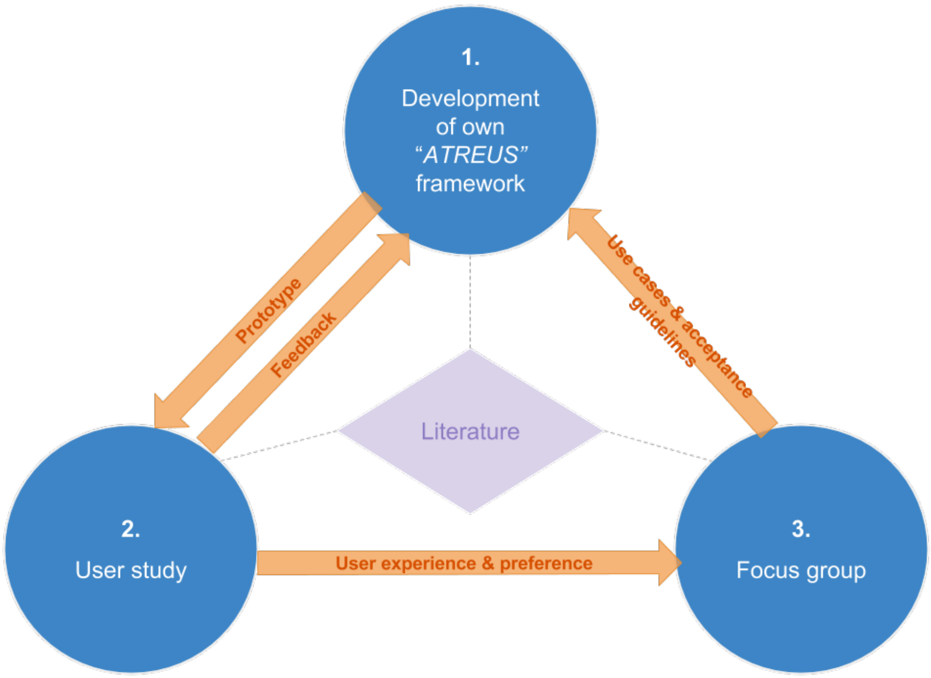

Figure 3.1: Overview research methodology in three main steps.

First of all a comprehensive literature research of scientific work was done for obtaining profound insights in the research field and to establish the methodological approaches for the research aim. The focus was on related work which addresses interaction techniques with smartphones controlling large screens and covers first approaches with latest mobile web technologies. All important findings and current applications were discussed and analysed in the previous chapter 2. Based on the learnings from literature the research strategies are pursued as follows:

- **Development of web-based interaction techniques**:

  Based on the knowledge of similar approaches discussed in chapter 2 a web-based framework was designed and developed. The mobile framework for smartphones utilizes latest web technologies only; all supported interaction techniques can be found in section 4 in more detail. Principally the features of the framework focus on classic interaction methods like surface or touch screen gestures e.g. discrete button pushes, picking objects, zooming, etc. and on motion gestures e.g. continuous indirect translation with accelerometers, tilting, rotating etc. as mentioned in [5]. But also camera-based approaches to support advanced interactions like "Mini Video" and "Smart Lens" described by Baldauf et al. [3] were implemented but not further evaluated in the user study.
  The framework concept is based on related earlier studies [46, 29, 22] but with the focus on using modern web technologies. This system should provide a platform-independent and easy adaptable development tool for realizing smartphones as remote controls for large displays. Furthermore, the potential and behaviour of HTML5 technologies for interaction purposes with today's mobile browsers will be tested and embedded in the system. Due to the requirements derived from the discussed challenges in the sections 2.2 and 2.1 a self-developed HTML5 mobile framework was preferred to existing tools such as *Sencha Touch*[1] or JavaScript libraries like *hammer.js*[2]. Even though some system extensions would be possible, these tools are still too limited to experiment with a wide set of web-based interaction techniques. Additionally the developed framework contains beside the mobile web library, a server platform written in Java for hosting an application on a large screen. The final software is published on the research project page [1]; more about the framework architecture and some implementation details are given in section 4.

[1] `http://docs.sencha.com/touch/2.3.1/`, Accessed: 2014-10-02

[2] `https://github.com/hammerjs/hammer.js/wiki/Getting-Started`, Accessed: 2014-05-03

- **User experience analysis with smartphone controllers**:

  For gaining experiences about the user's acceptance and behaviour in dealing with web-based interaction techniques on smartphones a user study was realized at the research center in Vienna called *FTW* [3] in July 2014. Therefore, some prototypes based on the aforementioned framework was created to compare different gamepad designs. As described in section 2.3 games are an important use case most people are interested in and where speed and accuracy are essential for completing a given task. This suggests that tests with game applications can reveal the suitability for other applications as well. Hence, the user study focused on interaction with mobile on-screen gamepads for controlling remote video games with the own mobile device. The main aim was to identify which gamepad design is preferred by the users and which interaction method is suitable for gaming purposes. Similar related works are summarized in the previous chapter 2.3.

  During the user study relevant information for further investigations were collected with a structured interview, a corresponding questionnaire and logging data. Relevant documents are attached in the appendix C; important parts are translated to English for the evaluation in this master thesis. On the basis of user feedback and observations the advantages and disadvantages of using a web-based remote control are studied to reveal the user experience in a gaming context. The experiment setup and the lessons learned important for the topic of this thesis are described in chapter 5. More details will be published with a comparative study in the article *"Game Over? Comparing On-Screen Gamepad Designs for Smartphone-Controlled Video Games"*. [2]

  In conclusion the results of the user study were considered for a review of the first framework version. In a redesign phase the features of the framework were improved and extended to enable a user-centred development of remote controls. In a conclusive guidance relevant issues derived from the user study were summarized which should be considered for providing web-based remote controls.

- **Use case analysis for smartphone controllers in public spaces**:

  As reviewed in chapter 2.1 many different use cases have emerged for controlling interactive applications with the own smartphone on public displays. A great popularity increases for marketing and gaming purposes – but in which application areas web-based remote controls can play an important role too? Based on several projects and researches from chapter 2 a collection of various fields was conceptualized for further investigation (cf. figure 2.3). Then four use case representatives (public places, public events/exhibitions, shopping and public transport) were discussed with a focus group during a workshop at *FTW* in September 2014.

[3] `http://www.ftw.at`, Accessed: 2014-18-03

In two interview phases a group of five smartphone-affine participants evaluated the presented use cases and were talking about their personal experiences with similar applications. As basis for the discussion a questionnaire and some idea cards related to the design research methods of Buskermolen et al. [11] and Beck et al. [8] were used. All documents including a transcription of the recorded conversations are attached in the appendix D; relevant parts are translated into English for further analysis.

As a result the most preferred use case discussed by the focus group is further studied in chapter 6. By this example the user preference and acceptance are reviewed for using smartphones as a web-based remote control. It should also demonstrate the user's benefit of using the web-based approach in this certain context. Afterwards the key aspects which have crucial impact on the user acceptance for web-based remote controls were summed up.

All results of the three main research parts were reviewed in chapter 7 for answering all research questions. Finally all lessons learned are summarized in the conclusion of this thesis for future work and further improvement of browser-based remote controls.

CHAPTER 4

Development of web-based interaction techniques

In this chapter the development of a system for browser-based remote controls for interactive applications on large displays is discussed and further investigated. The main goals are to provide a platform-independent research framework for realizing smartphone controllers and to test the strengths and limits of interaction methods with latest web technologies. Therefore, a system called *ATREUS* was developed for enabling smartphone-controlled applications on large displays.
This framework was conducted within the research project of the *FTW* and funded by *netidee*. The project is published on the research website[1] under the *GNU General Public License* and the source code is available in the BitBucket Repository on `https://bitbucket.org/matbal/atreus` (Accessed: 2014-19-11).

4.1 Requirements and system concept

An overview about the key characteristics of the system to meet all requirements based on related work from chapter 2 is summarized as follows:

- **Platform-independence:** The remote control on the smartphone is purely web-based as a HTML5 website received by the web server. Thus every mobile device with a browser installed – regardless of operating system or manufacturer – is able to interact with a large screen. The platform for hosting an interactive application on the display is written with the cross-platform Java technology. The communication between the mobile device and the display is handled by a *Jetty*[2] server which is also implemented in Java and embedded in the system. Therefore, the display platform is also suitable for any operating systems like Windows, Mac or Linux.

[1] `http://atreus.ftw.at`, Last updated November 2014, Accessed: 2014-19-11
[2] `http://www.eclipse.org/jetty/`, Accessed: 2014-17-02

- **Easy installation:** For running the framework no specific installation or configuration is necessary, neither on the smartphone nor on the computer hosting the display. This is a major benefit for public spaces where an additional installation of sensors or other devices would be problematic, expensive or even impossible.

- **Easy development & expandable:** The framework design is easy extendable and customizable because of using standard web technologies (e.g. HTML5, JavaScript) and open source libraries instead of integrating a prefabricated and rigid framework (e.g. Sencha Touch) which often uses proprietary technologies and programming languages. For implementing one's own interaction technique the appropriate *"atreus-JavaScript"* libraries of the framework can be added in any HTML5 template. The remote control commands can be received realizing a *MobileActionListener* in the display application. As example some demonstrators are added to serve as orientation.

- **Seamless user experience:** Due to the familiarity with the own mobile device, the threshold of use is very low to check out new interaction technologies. Moreover, the framework is not limited in supporting any interaction methods. Every approach which can be realized with today's web and device technologies now or in future are possible to enhance the user experience.

- **User attractiveness:** With the client-server architecture and the communication over a Wi-Fi network several smartphone users can interact simultaneously with various devices in front of a public display. This overcomes the restricted visual capacity of the screen and encourages to participate in multi-user applications. For easy pairing the framework supports the generating of QR codes which can be shown with the display application on the screen.

To achieve all these properties, a suitable and expandable framework concept was created as a foundation for supporting various interaction features. Basically the framework consists of two parts: the *ATREUS platform* for hosting the interactive application on the display and the *ATREUS mobile library* for providing web-based remote controls on the smartphone. The system architecture is shown in figure 4.1.

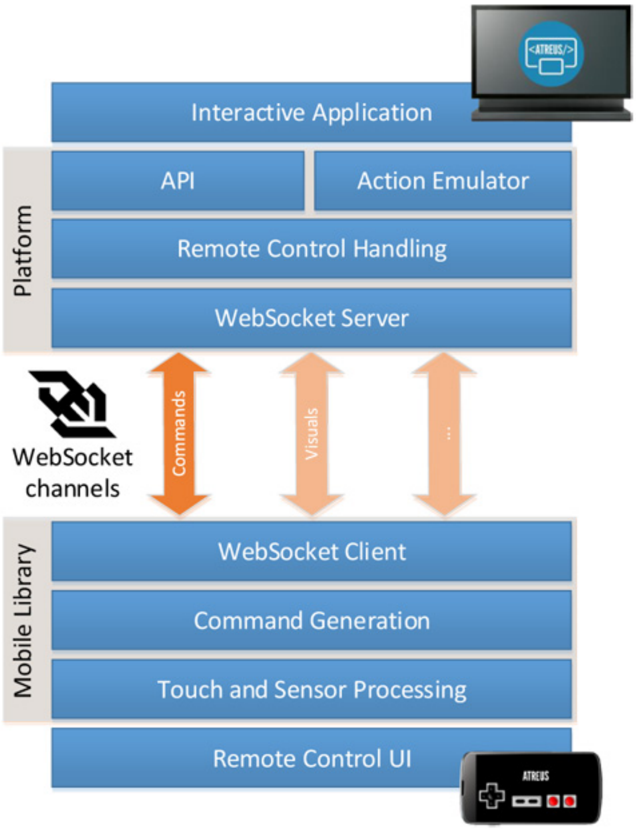

Figure 4.1: ATREUS architecture overview: the platform on the public display and the mobile library on the smartphone. (Figure by Baldauf et al.[3]).

The communication between these two parts takes place over a WebSocket connection by exchanging the data with JSON[4] protocols. The framework supports a bidirectional communication: the client sends all data (e.g. sensor values, interaction type, coordinates etc.) with the remote control protocol and the server can response with a feedback protocol which can contain simple text or special commands for triggering an action on the smartphone. Table A.1 in the appendix defines the protocol structure and all possible parameters used for the supported interaction techniques.

[3]From article *ATREUS – An Open Source Framework for Web-Based Smartphone Remote Controls for Public Screens* by Baldauf, M.; Adegeye, F.; Alt, F. & Harms, J., 2015, page 3.

[4]`http://json.org`, Accessed: 2014-26-03

For example, if the user presses a button “A” on the touchscreen the following JSON protocol 4.1 will be sent to the server platform. With this information the platform can emulate the corresponding key presses for the interactive application.

```
JSON EXAMPLE

{"interactionType":0,                                        (a)
 "action":1,                                                 (b)
 "elementId":"buttonA",                                      (c)
 "coords":{                                                  (d)
  "x": 585,
  "y": 62,
  "z": null
 },
 "sensorData":["ATREUS_mobile/images/buttonA.png"]           (e)
}

(a) indicates the type of interaction e.g. button
(b) what event is triggered e.g. button pressed (touchstart)
(c) the ID of a HTML element e.g. <div id="buttonA">...</div>
(d) the position of the touchevent
(e) other information (optional) e.g. filepath of the image
```

Table 4.1: JSON remote control protocol.

In figure 4.2 the basic principle of the communication process between the smartphone and the interactive application is further explained. The first step is to couple the personal device with the display by using the website URL or the IP address of the server platform. Then the mobile browser requests the website as the remote control application and opens a WebSocket connection for exchanging further interaction or feedback commands. Every time the user triggers an event (e.g. touchevent or motion) with the smartphone, the included JavaScript functions (`atreus-controls.js`, `atreus-connections.js`) of the remote control create a message in JSON format and send it to the web server. The *MobileActionListener* of the platform receives the data and starts an action in the interactive display application by emulating key events. As a result the user can interact with the large screen.

But the framework can also allow the user to receive information from the display application. The platform provides a feedback function that creates a JSON feedback protocol and sends it back to the smartphone client. The feedback protocol contains defined action commands for vibration, text or sound to trigger a certain event on the mobile device. Therefore, the JavaScript library `atreus-feedback.js` has to be included in the provided remote control website.

More details about the technical implementation are described in the next section; a short documentation is available in the BitBucket Repository.

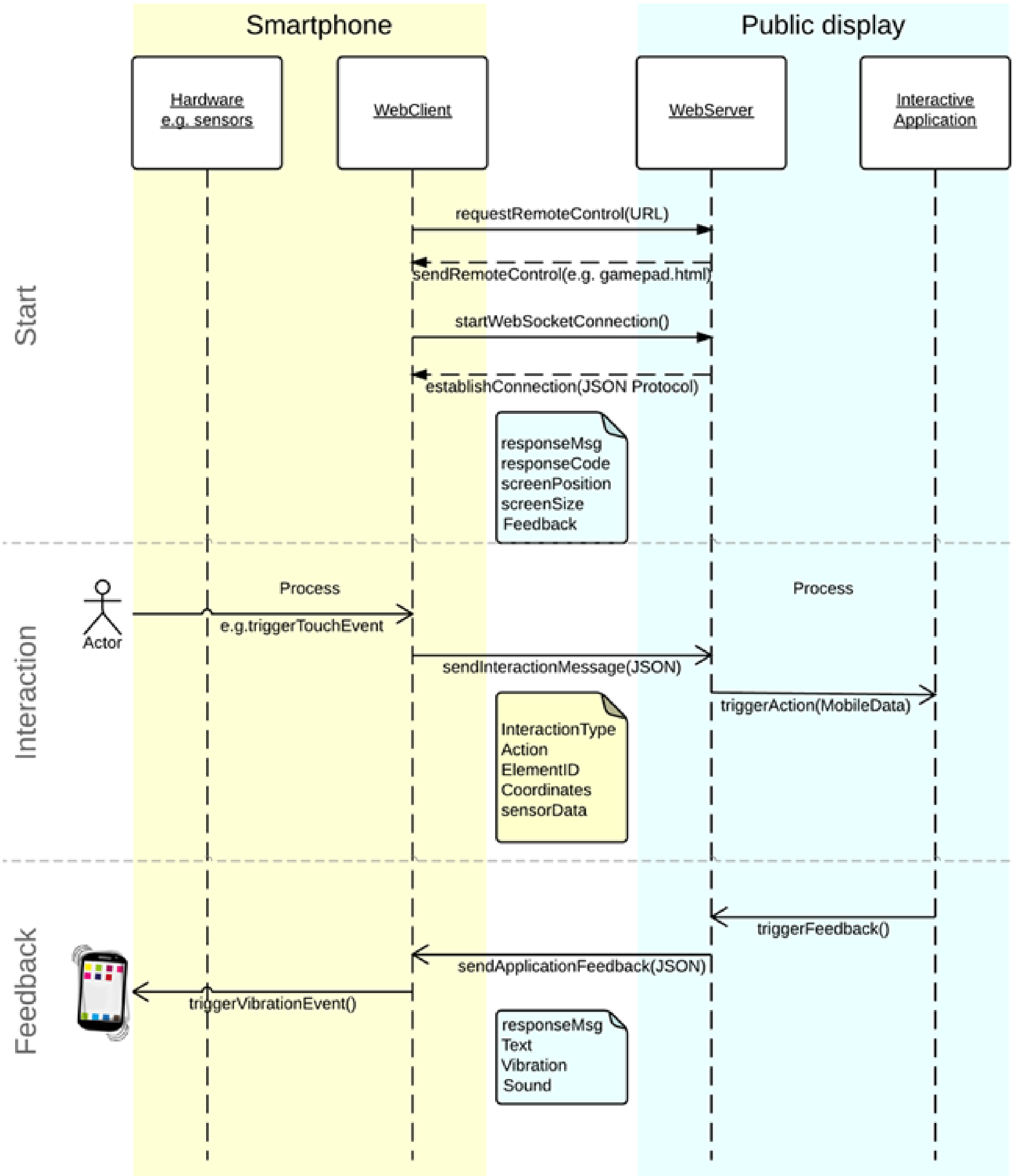

Figure 4.2: Illustrating the communication process between smartphone and display application.

4.2 Implementation of the framework

Only relevant components of the ATREUS system are discussed in this master thesis; the complete source code can be looked up in the repository.
Based on the framework architecture the system is divided into the following main parts:

4.2.1 ATREUS Platform

This platform part is used by the computer for hosting the interactive application on the large screen. It is written in Java and contains an embedded Jetty server to avoid any system dependencies. The main tasks are managing the communication with the smartphone and reacting on received remote control commands. An overview about the general structure of the system and the packages is shown in figure 4.3.

Managing interaction events
To react on incoming remote control commands, a *MobileActionListener* has to be implemented. For the general handling of connected or disconnected devices an optional *MobileEventListener* is provided. The *MobileActionListener* contains a *triggerAction* method which receives a *MobileData* object with all control parameters to emulate input events through the Java Robot[5] class. Some exemplary listeners are included in the `at.ftw.atreus.io` package; every listener has a corresponding HTML-file in `\htdocs` for acting as remote control. The functions for connection (`mobileClientConnected(String id, Session session)`, `mobileClientDisconnected(String id)`) and remote control events (`triggerAction(MobileData)`) are provided as *Java interfaces*, so the framework is easy extendable with own listeners for implementing new interaction techniques. Before starting the server the customized listener must be registered in the *AtreusServer* class as demonstrated in 4.2.

SET LISTENER

```
AtreusServer as = AtreusServer.getInstance();
as.setActionListener(myMobileActionListener);
as.setEventListener(myMobileEventListener);
as.start(SettingsManager.getIP(), SettingsManager.getPort(),
AtreusConstants.HTML_PATH,SettingsManager.getRemoteControl());
```

Table 4.2: Example how to register a listener for the remote control.

[5] `http://docs.oracle.com/javase/7/docs/api/java/awt/Robot.html`, Accessed: 2014-27-02

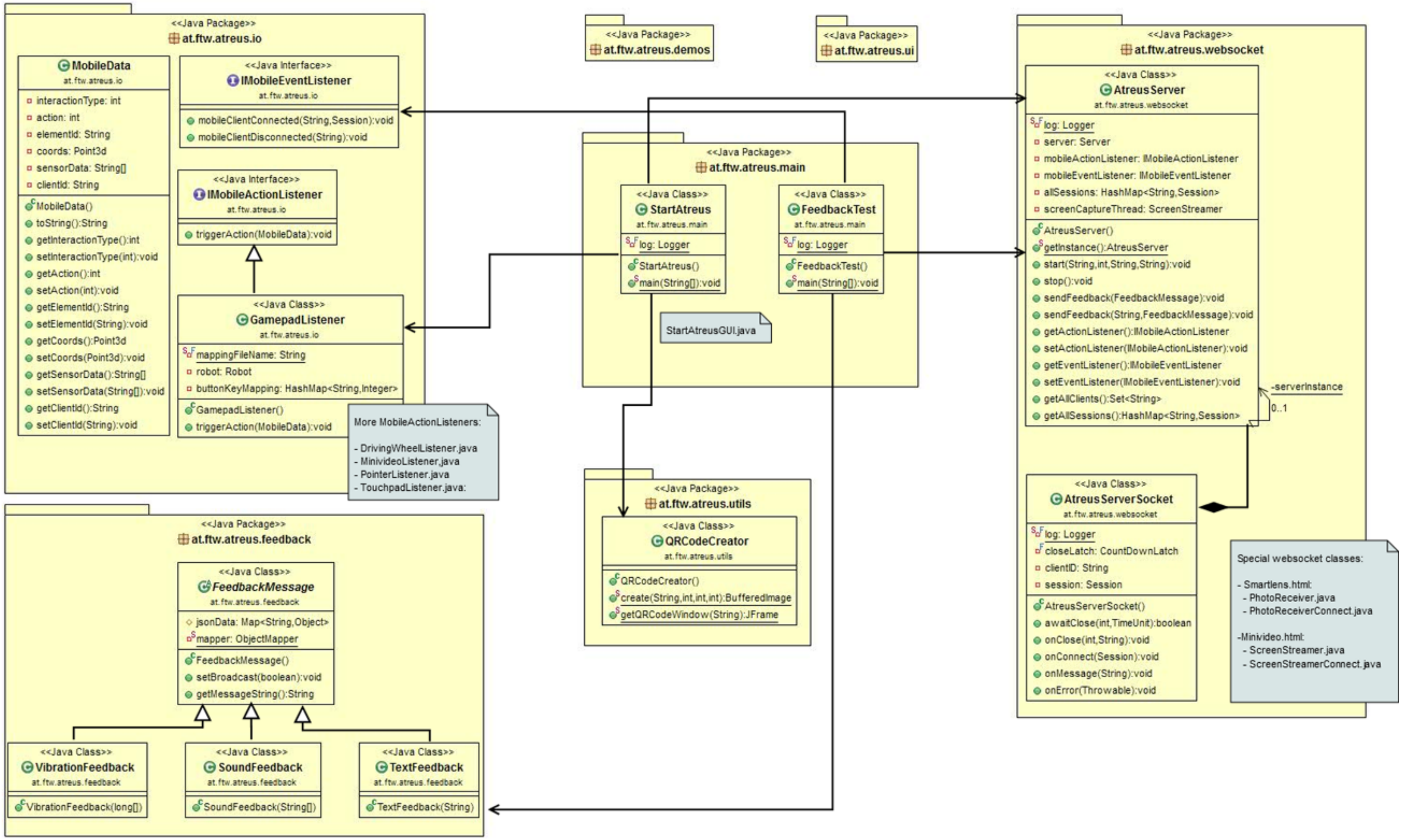

Figure 4.3: Class diagram of the ATREUS platform.

As an example for creating a simple touchscreen-based interaction with a smartphone gamepad the *GamepadListener.java* is implemented as follows:

```java
public class GamepadListener implements IMobileActionListener {

    private final static String mappingFileName = "data/gamepad.map";

    private Robot robot;
    private HashMap<String, Integer> buttonKeyMapping;

    public GamepadListener() {

        // read button/key mapping from file
        buttonKeyMapping = new HashMap<>();
        Properties props = new Properties();
        try {
            props.load(new FileInputStream(new File(mappingFileName)));
        } catch (IOException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }

        // run through properties, check and add to hashmap
        for (Object p : props.keySet()) {
            String propkey = p.toString();
            String propvalue = props.getProperty(propkey);

            Integer keycode = null;
            try {
                // check whether numeric keycode is specified
                keycode = new Integer(Integer.parseInt(propvalue));
            } catch (NumberFormatException e1) {
                // if not try to interpret value as keyevent constant
                try {
                    keycode = (Integer) KeyEvent.class.getDeclaredField(
                            propvalue).get(Integer.class);
                } catch (Exception e2) {
                }
            }

            // if parsing was successful add to mapping table
            if (keycode != null)
                buttonKeyMapping.put(propkey, keycode);
        }

        // create robot instance
        try {
            robot = new Robot();
        } catch (AWTException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

```

```java
    @Override
    public void triggerAction(MobileData mobiledata) {

        if (mobiledata != null) {
            if (mobiledata.getInteractionType() == AtreusConstants.
    INTERACTION_BUTTON) {

                int actionType = mobiledata.getAction();
                String elementId = mobiledata.getElementId();

                // do we have a mapping for this gamepad button?
                if (buttonKeyMapping.containsKey(elementId)) {

                    int keycode = buttonKeyMapping.get(elementId).intValue
    ();

                    // simulate keyboard action
                    if (actionType == AtreusConstants.ATREUS_DOWN) {
                        robot.keyPress(keycode);
                    } else if (actionType == AtreusConstants.ATREUS_UP) {
                        robot.keyRelease(keycode);
                    }
                }
            }
        }
    }

}
```

Listing 4.1: GamepadListener.java

The `GamepadListener` class is extended with the interface *IMobileActionListener* to implement the *triggerAction* method which defines what should happen with the received mobile data. The *MobileData* class contains all information relevant for interaction retrieved from the received JSON protocol (cf. table 4.1). For simulating an interaction with simple buttons on the touchscreen corresponding key presses are emulated with `robot.keyPress(keycode)`. The codes for triggering a certain key are mapped in a configuration file *gamepad.map* (cf. table A.1) which contains the ID of the touched HTML element and the KeyEvent values (e.g. pressing the HTML element with id="buttonA" corresponds to pressing "A" on the keyboard).

Using the feedback functionality

For enabling an interactive application to send data back to the mobile client the *AtreusServer* class provides the methods:
`sendFeedback(FeedbackMessage feedback)` for broadcasting a message to all clients and
`sendFeedback(String client, FeedbackMessage feedback)` for broadcasting to a specific client. In the package `at.ftw.atreus.feedback` the abstract class *FeedbackMessage* (cf. A.2) can be used to create a corresponding JSON feedback protocol for the web client. The framework already offers three class extensions for sending feedback

messages with parameters used for text, sound or vibration. To start the corresponding feedback event on the mobile device, the JavaScript library *atreus-feedback.js* must be added to the remote control client and the feedback parameter must be set to “true” in the configuration file of the platform application. As an example for sending feedback information a demo application *FeedbackTest.java* (cf. A.3) comes with the framework. Whenever a new client is connected, the program sends out messages to trigger a vibration, sound and text event on the smartphone.

Server-side WebSocket communication
The platform uses the Jetty [6] library for creating a WebSocket server which manages the communication between the display application and the mobile device. All classes regarding WebSocket communication are in the `at.ftw.atreus.websocket` package. The main methods are allocated in *AtreusServer.java* A.4 and *AtreusServerSocket.java* A.5. The other classes *PhotoReceiver* and *ScreenStreamer* are extra implementations relevant for the interaction with *smartlens.html* and *minivideo.html.* These special cases open up a second WebSocket connection on a further port for sending or receiving images without blocking the main connection stream. On start up an instance of the *AtreusServer.java* is created and after setting the selected *MobileActionListeners* the server is started with the parameters IP address, port, the path of all supported remote controls and the chosen remote control file being delivered to the client (cf. table 4.2). For building a WebSocket client with the same connection values the IP address and the port are set in the mobile library *atreus-connection.js* with `initWebsocketSettings(String ip, int port, String path)`. Then the *AtreusServerSocket* class is registered at the *AtreusServer* and the WebSocket connection is established waiting for a client to connect. With the socket method `onConnect(Session session)` every new client gets its own session and receives a notification message with further parameters. Meanwhile, the server socket is listening for incoming client messages with `onMessage(String msg)` which are then extracted from the JSON protocol and wrapped in new *MobileData* objects to trigger an action. If a client gets disconnected, the session will automatically close until the connection is resumed and the client wants to connect again. But the communication channel between the WebSocket server and the WebSocket client stays open until the *AtreusServer* is terminated.

Starting and configuring the platform
Before starting the ATREUS platform the following parameters have to be set and are stored in JSON format in the configuration file `atreus_config.json`. The *SettingsManager.java* contains methods for accessing all configuration values.

[6] `http://www.eclipse.org/jetty/documentation/9.2.6.v20141205/jetty-websocket-server-api.html`, Accessed: 2014-13-02

```json
{"port":8083,"ip":"http://192.168.0.12","screenx":0,"screeny":0,"
    screenwidth":1280,"screenheight":800,"remotecontrol":"gamepad.html","
    maxuser":1,"feedback":false,"showqr":false}
```

Listing 4.2: Example atreus_config.json

- *port:* defines the port for the communication.
- *ip:* the ip address or url (starting with "http://") of the platform.
- *screenx:* the left coordinate of the display for determining the correct position during interaction.
- *screeny:* the right coordinate of the display for determining the correct position during interaction.
- *screenwidth:* the width of the large screen used for interaction.
- *screenheight:* the height of the large screen used for interaction.
- *remotecontrol:* the filename of the remote control being sent to the mobile device.
- *maxuser:* the maximal amount of connected users.
- *feedback:* a boolean value whether the platform can sent feedback messages.
- *showqr:* a boolean value whether the QR code window should be shown on the display.

The application can be started with *StartAtreus.java* A.6 using the settings edited directly in the configuration file. It also provides a graphical user interface *StartAtreusGUI.java* for setting all these parameters (cf. figure 4.4). If the QR code mode is enabled, the platform will generate a QR code containing the predefined IP address and show it in a separate window on the display. The QR code is generated with the *QRCodeCreator.java* helper class using the ZXing[7] software library. After the platform has been started, the user can now connect the smartphone with the display by scanning the QR code. Then the browser automatically opens the preset URL with the corresponding remote control.

[7] https://github.com/zxing/zxing, Accessed: 2015-28-02

Figure 4.4: Graphical user interface of the ATREUS platform with QR code window.

4.2.2 ATREUS Mobile library

The mobile library consists of JavaScript libraries stored in `\htdocs` which are already included in the HTML5 remote control demonstrators. They provide all functions for sending interaction commands or optionally receiving feedback information. Therefore, the following JavaScript files and the recent jQuery library have to be added in the HTML head:

```
<script type="text/javascript" charset="utf-8" src="ATREUS_mobile/js/
    atreus-connection.js"></script>
<script type="text/javascript" charset="utf-8" src="ATREUS_mobile/js/
    atreus-controls.js"></script>
<script type="text/javascript" charset="utf-8" src="ATREUS_mobile/js/
    atreus-feedback.js"></script>
```

Listing 4.3: ATREUS JavaScript library

- *atreus-connection.js:* creates a WebSocket client and manages the communication to the platform with `WebSocket.send(JSON msg)`.
- *atreus-controls.js:* contains all methods for interaction with the remote controls.
- *atreus-feedback.js:* optional component which contains methods for receiving commands that trigger a text, sound or vibration feedback event.

As soon as the website is loaded a WebSocket connection is established with the *atreus-connection.js* and the web client is now able to send or receive messages from the platform. With the included *atreus-controls.js* the required interaction events are attached with the library's function `addControlEvent(eventObj, eventType, eventFunction)`. The value "eventFunction" defines what interaction method should be used with the remote control. The mobile library can be easily extended with new interaction methods by writing own functions or using the provided event functions in the JavaScript file. An example how to implement a simple remote control with the JavaScript files is given in *gamepad.html* A.7. More samples can be found in `\htdocs` and can be selected with in graphical user interface of the platform application *StartAtreusGUI*.

4.3 Implementation of web-based interaction techniques

Experiments with the demonstrators gave us more insight in the possibilities of smartphone interactions with latest web technologies. Relevant HTML5 specifications used for interaction techniques provided by this framework were already described in 2.4.1; more implementation details can be gathered from the source code. Based on the classifications derived from chapter 2.2 the sample remote controls are allocated in the following interaction criteria:

Surface gesture-based interaction:

- **gamepad.html:**
  This remote control mimics an interaction with a classic gamepad controller. It uses the Touch Events mapped on images to emulate actions like button pressed or released. The samples *gamepad_btn*, *gamepad_dpad*, *gamepad_joystick* and *gamepad_tilt.html* are further modifications which were tested in the user study. In contrast to the classic approach they already recalculate the touch coordinates on the client-side or combine them with other interaction methods before sending the position to the platform. In this way the surface gestures get more accurate and can imitate physical controllers, e.g. a joystick.

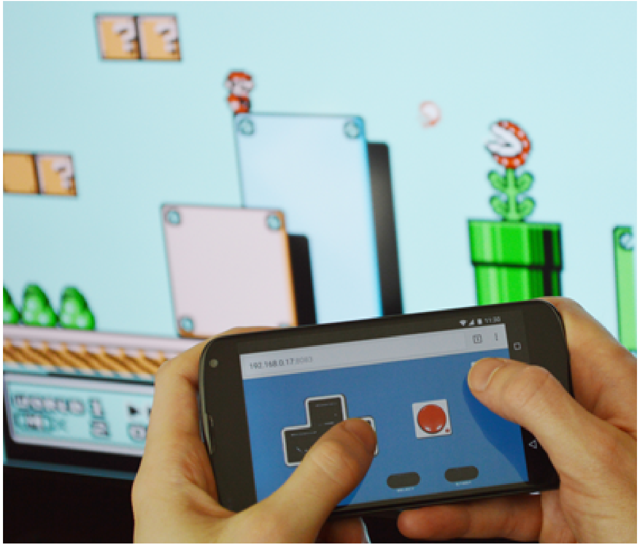

Figure 4.5: Interaction with *gamepad.html*. User can control the movement by touching the gamepad buttons. (Figure by Baldauf et al.[8]).

- **touchpad.html:**
  The interaction with this remote control is similar to the trackpad known from notebooks. The motion on the touchscreen area results in corresponding movements of the cursor on the display. It also makes use of the Touch Event API but extended with additional determinations for recognizing special gestures like long-pressed, zooming, scrolling and swiping. The protocol of these gestures contains further data information which can be used for more sophisticated actions on the display platform (cf. *TouchpadListener.java* emulates a scrolling action with `Robot.mouseWheel(int data)`).

[8]From article *ATREUS – An Open Source Framework for Web-Based Smartphone Remote Controls for Public Screens* by Baldauf, M.; Adegeye, F.; Alt, F. & Harms, J., 2015, page 1.

Figure 4.6: Interaction with *touchpad.html*. User can move an item with touch gestures similar to a trackpad with special gestures: zooming, swiping and scrolling.

- **minivideo.html:**
  The corresponding *MinivideoListener* continuously captures screenshots of the large display and sends it to the connected mobile device (based on the *Mini Video* concept in [3]). Then it is scaled to the size of the mobile screen reflecting the display's content. The coordinates of the touches on this miniature version are recalculated with the original screen size and sent back to the platform. As a result all touch events are simulated as corresponding mouse clicks on the display.

Figure 4.7: Interaction with *minivideo.html*. User can select an item by touching the received screenshot of the remote screen on the mobile device. (Figure by Baldauf et al.[9]).

Motion gesture-based interaction:

- **drivingwheel.html:**
  This example continuously transmits the sensor information to the platform by using the DeviceOrientation Event. The received data can be further processed by the platform and mapped to analogue joystick movements, mouse or keyboard events.

Figure 4.8: Interaction with *drivingwheel.html*. User can control an object by tilting the mobile device. (Figure by Baldauf et al.[9]).

- **pointer.html:**
  This remote control mimics the behaviour of a laser pointer by using the smartphone as pointing device (based on the *Pointer* concept in [3]). The pointer positions are calculated from the DeviceMotion Events and transmitted to the platform. Thereby the movement of the device can be mapped to the absolute cursor positions on the display.

[9]From article *ATREUS – An Open Source Framework for Web-Based Smartphone Remote Controls for Public Screens* by Baldauf, M.; Adegeye, F.; Alt, F. & Harms, J., 2015, page 1.

Figure 4.9: Interaction with *pointer.html*. Figure illustrates the mobile screen with the orientation coordinates after pressing the start button. User can move the pointer by tilting the mobile device (cf. *Point-and-Move* 2.18).

Camera-based interaction:

- **smartlens.html:**
  This interaction makes use of the media capture abilities of the mobile device. Elements on the large display can be selected by targeting and touching them through the smartphone's camera view. (based on the *Smart Lens* concept in [3]). When the user touches the mobile touchscreen the current camera frame is captured and an image is sent to the server platform. With the *PhotoReveiverConnect.java* using the computer vision library *BootCV*[10] the image is matched with the current screen content. If the matching process is successful, the touch coordinates will be mapped to the corresponding screen coordinates where the mouse cursor is moved to.

[10] `http://boofcv.org/`, Accessed: 2014-12-10

[11] From article *ATREUS – An Open Source Framework for Web-Based Smartphone Remote Controls for Public Screens* by Baldauf, M.; Adegeye, F.; Alt, F. & Harms, J., 2015, page 1.

Figure 4.10: Interaction with *smartlens.html*. User can select an item by targeting it with the camera and touching the mobile screen. (Figure by Baldauf et al.[11]).

4.4 Evaluation of latest web technologies as remote control

This chapter analyses the realization of web-based remote controls and evaluates the support of the demonstrated interaction techniques by current mobile browsers. With the development of the framework demonstrators it is confirmed that existing interaction techniques can be realized by using latest web technologies only. Although some HTML5 features are still in draft stage, they can already keep up with similar applications implemented with native apps.

During the interaction over the WebSocket connection no significant deficiencies in performance was perceptible. The communication between the display and the smartphone proceeded fluently without any delay in response time. Because of establishing a second WebSocket connection for bigger data, even sending camera images (cf. Mini Video or Smart Lens) did not cause any noticeable delay which would affect a seamless interaction. Only the size of the transmitted data chunk from the client must be taken into account in the implementation. If the binary data exceeds the defined maximum binary message size of the WebSocket protocol[12], the connection will be closed and result in an “1004 - Frame too large” error.[13]

[12] `http://tools.ietf.org/html/draft-ietf-hybi-thewebsocketprotocol-17#section-5`, Accessed: 2014-07-02

[13] `http://www.lenholgate.com/blog/2011/07/websockets-is-a-stream-not-a-message-based-protocol.html`, Accessed: 2014-07-02

This problem can be solved on the client-side by reducing the image size and splitting the image in smaller data chunks for sending them bit by bit to the platform (e.g. smartlens.html sends message chunks with the maximum size = 65535 bytes). Afterwards the web server has to buffer the incoming chunks and merge them together before continuing with further image processing steps (cf. PhotoReceiverConnect.java).

Generally speaking the HTML5 WebSocket functionality as a fundamental basis for supporting web-based interaction techniques works flawlessly and is already widely supported by all common mobile browsers. (Global usage [14]: 84,54%, Last updated January 2015). These results show that the functionality is completely independent of operating system and device manufacturer.

But the main issues – the mobile browser support and the browser behaviour – still remains: As already mentioned in chapter 2.4 the HTML5 support and new browser technologies are still growing, nevertheless, some measures must be taken in the implementation to achieve the desired browser behaviour throughout all mobile devices: First of all to avoid troubles during interaction, some default browser features have to be suppressed or must be handled with a workaround. For example, double tapping on the touchscreen results in zooming into the browser's view but this default behaviour disturbs the interaction with a remote control. One solution is the definition of the *viewport meta tag*[15] that tells the browser how to behave when rendering the website. That's why every demonstrator contains the following command in the document head:

```
<!-- prevent zooming in -->
<meta name="viewport" content="width=device-width, initial-scale=1.0, user
    -scalable=no" />
```

Listing 4.4: gamepad.html

Another issue what could cause problems during interaction are the default actions provoked by JavaScript events. But this can be suppressed by using the command `Event.preventDefault()` before calling other event functions as done in *atreus-controls.js*. Without this statement a long-press action would invoke the standard browser behaviour showing a context menu for e.g. saving the touched picture from the website.

Most browsers still have problems to define the screen orientation. Currently the view in portrait or landscape mode is overruled by the smartphone's system settings and is determined by the accelerometers which recognize the position of the device. As a consequence the layout of the remote control changes when the user is rotating the device from portrait to landscape format. This can change the coordinates of relevant website components or lead to problems during interaction and thus must be considered in the implementation. The best solution would be to freeze the remote control in landscape mode but for now this does not work in most browsers.

[14] `http://caniuse.com/#feat=websockets`, Accessed: 2015-03-03

[15] `https://developer.mozilla.org/en-US/docs/Mozilla/Mobile/Viewport_meta_tag`, Accessed: 2015-01-03

The latest working draft since 23^{th} of October 2014 of a *Screen Orientation API*[16] [17] promises the ability to lock the screen orientation to a specific state but this is still experimental and only supported by the mobile browser Chrome for Android version 40 (Browser support last updated in January 2015).[18] Also the tests during implementation could not overcome this screen orientation problem. That's why the current version of the framework does not use the Screen Orientation API. As common for most gamepads it is assumed that the users are going to hold the device in landscape mode during interaction and this does not affect the user experience. For future works the ability of freezing the screen orientation by the browser can make web-based remote controls more reliable and ensures a stable user interaction. As we can see there are some browser issues which have to be taken into account but there are many different mechanisms to overcome these problems.

To learn about the remote control capabilities with web technologies in more detail, all demonstrators were tested with different browser versions and mobile devices. The complete test plan with all results is attached in the appendix B. The framework tests were completed with one iOS device - the latest *iPhone 6* with the corresponding Safari Browser, one latest Android device *Samsung Galaxy Alpha* and an older Android representative *Samsung Galaxy SII* with four different browsers: Standard Android Browser 4.4.4, Google Chrome, Opera Mobile and Firefox Mobile.

The main focus was to identify differences between various browser versions. The usage of mixed mobile devices with similar hardware technologies demonstrates the platform-independence. As the results show the web-based remote controls are primarily affected by the used browser version, whereas the operating system on the smartphone only plays a role for pre-installed default browsers e.g. Standard Android Browser or Safari iOS. The smartphone type does not matter for using the remote control functions if they contain comparable hardware features. In summary the most suitable browsers with the best results that already support all key features are *Google Chrome* and *Opera Mobile*. In contrast, Firefox Mobile has a limited support of latest HTML5 features and is more susceptible to bugs during implementation (e.g problems in using canvas with window.innerWidth/innerHeight for all *gamepad.html* controls as shown in B). [19]

Although the Safari iOS browser supports most latest web technologies, the main disadvantage is the missing support of the HTML5 feature Media Capture and Streams. This function is necessary for accessing the smartphone's camera or other media data. For example, the framework test with the remote control *smartlens.html* completely failed in using the iPhone with the Safari browser. It was not possible to access the mobile camera and therefore the browser could not start the mobile application and resulted in an error.

[16] `http://www.w3.org/TR/screen-orientation/`, Accessed: 2015-01-03

[17] `https://developer.mozilla.org/en-US/docs/Web/API/CSS_Object_Model/Managing_screen_orientation`, Accessed: 2015-01-03

[18] `http://caniuse.com/#feat=screen-orientation`, Accessed: 2015-03-03

[19] `https://bugzilla.mozilla.org/show_bug.cgi?id=1071620`, Accessed: 2014-21-10

Furthermore, the selection of the video source was empty because the browser could not recognize the front or back camera of the iPhone. As detailed in appendix B the main reasons were problems with the *MediaStreamTrack* and the *Media Capture Stream* HTML5 specifications. As a consequence the remote control *smartlens.html* cannot be used with Apple devices. The test result concurs with current browser specifications found on popular information websites about HTML5 support.[20] [21]

For browsers which can make use of the mobile camera the accessing behaviour between front and back camera must be further taken into account. For instance, new versions of Opera Mobile offer the users a selection mask as default to decide which camera to use. But most browsers (e.g. Chrome) automatically access the front camera facing the user defined as default camera. But the *smartlens.html* remote control needs the back camera, for that reason a separate mechanism for selecting the video source has to be implemented by using `MediaStreamTrack.getSources()`. Otherwise the usage of this web remote control is not possible despite supporting Media Capture and Streams (cf. test results of Firefox Mobile in B).

Further relevant for camera-based interaction methods is the specified image format returned by the mobile client. As default several browsers submit different data formats (e.g. .png, .jpeg, .webp) when converting a HTML canvas element to an image data object. The wrong format could cause problems in sending image data over WebSockets or in matching the images on the server platform and therefore should be considered in the implementation.

Using a motion-based remote control like *pointer.html* or *drivingwheel.html* the issue with different angle values depending on the browser have to be handled by the mobile application. For testing issues the provided remote controls were optimized for Opera Mobile and Chrome. But the detailed test results point out that an interaction with other user agents fails due to different sensor values. As shown in appendix B the browsers Safari and Firefox would basically support this feature but due to the different angle values the calibration value of *pointer.html* is changing as well. As a result the movement of the controlled cursor on the remote screen was too quick or not fluently by rotating the device. The Android Browser 4.4.4 did not support the *DeviceMotion* event and therefore the interaction method with *pointer.html* was not possible at all.

With the demonstrator *feedbackdemo.html* the possibility of providing feedback for a seamless user experience with latest web technologies were also investigated. Basically every browser can retrieve text information from the server and show it on the smartphone's display. But focusing on interaction with public displays the support of feedback methods without looking at the device like sound or vibration features are more interesting approaches.

[20] `http://caniuse.com/#feat=stream`, Accessed: 2014-13-02
[21] `http://mobilehtml5.org`, Accessed: 2014-13-02

While at least one of these two features works in most tested browsers, the iOS Safari and Android 4.4.4 have problems with both. (cf. appendix B no support of vibration and sound) Because of the missing support of the HTML5 Vibration API up to now haptic feedback is not possible with Apple devices. According to many studies already mentioned in chapter 2 feedback is relevant for an immersive user experience.

But what other factors are important for using the smartphone as a web-based remote control? The next chapter examines how people are interacting with a web-based gamepad controller. Considering that a comparative user study with prototypes developed with the ATREUS framework was conducted for further evaluation.

CHAPTER 5

User experiences with smartphone controllers

The user study for comparing smartphone gamepad designs was conducted within the research project ATREUS in cooperation with *FTW*. All details and results will be published in the article *"Game Over? Comparing On-Screen Gamepad Designs for Smartphone-Controlled Video Games"*[2]. In this master thesis we focus on the findings relevant for interaction with web-based remote controls in general. As a representative use case for the user study we chose gaming applications because this context combines many different aspects to derive awareness for other applications as well. Figure 2.3 illustrates the popularity of games in several application areas especially in public spaces. But to achieve comparable conditions for investigating the usage of interaction techniques, the user study was carried out in a lab. The study design was based on previous research from chapter 2.3 where the gaming context is also used for evaluating mobile devices as input devices for interactive applications.

5.1 Method and experiment setup

The evaluated gamepad prototypes were developed with the aforementioned ATREUS framework to achieve purely web-based remote controls for every mobile device. For the user study we used simple gamepad designs with one conformation button and four different navigation controls to compare the interaction techniques as shown in figure 5.1. All four gamepads are attached in the ATREUS mobile library as demonstrator remote controls. The participants had to hold the Android test device *Google Nexus 4* in landscape mode and were observed and interviewed during completing three tasks with the variable on-screen gamepads. Therefore, a *Sony* flat screen TV was connected with a notebook running the ATREUS platform and the corresponding test application; the connection with the smartphone was established over Wi-Fi (cf.figure 5.2).

For easily receiving the suitable gamepad after completing a test phase, the test manager touched a NFC tag with the smartphone and started the study console to switch between the different controller designs. We investigated the following interaction techniques with the mobile device:

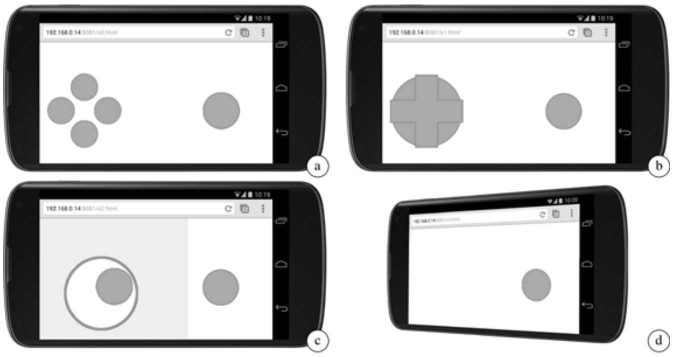

Figure 5.1: Four different gamepad designs for touchscreen smartphones: (a): directional buttons, (b): 8-way d-pad supporting swiping, (c): virtual joystick and (d): gestural tilt control. [2, p.2]

Figure 5.2: A participant controlling the Super Mario Bros. game with the joystick technique. [2, p.6]

- **Directional Buttons:** (cf. figure 5.1a)
  This controller presents a simple design with four buttons placed in a cross pattern similar to traditional hardware gamepads such as the first *Nintendo* series. By touching a circular button a pressing action is evoked and the corresponding movement command is sent to the platform. This technique is a representative for a surface gesture-based interaction where the user had to actively press and leave the buttons. Sliding the finger from one button area to another is not supported.

- **Directional Pad (d-pad):** (cf. figure 5.1b)
  Another demonstrator for a surface gesture-based controller is the directional pad which features four main directions plus four diagonals. This realization is more similar to the classic hardware counterparts because it supports sliding the finger over the touch-sensitive cross. The corresponding direction command sent to the platform is determined by dividing the touch area in pie slices of 120 degrees.

- **Joystick:** (cf. figure 5.1c)
  With this remote control the directional commands are sent by dragging a graphical knob inside a big circle by analogy with a hardware joystick. If the user touches the grey area, the joystick will appear and work relatively to the first touch point until the user raises the finger again. Then the movement stops and the joystick circle disappears. This on-screen joystick is a special implementation of a surface gesture-based interaction method. It works with the same angle configuration like the d-pad but without cancelling the movement commands when the finger is removed from the joystick circle.

- **Tilt Controller:** (cf. figure 5.1d)
  As example for a motion gesture-based interaction approach a simplified tilt controller was tested. The movement commands are sent by tilting the mobile device to the corresponding direction e.g. tilting to the right corresponds to pressing the "right" button. The movement can be stopped by bringing the smartphone into a defined neutral position – in our test case we used 30 degrees in landscape mode towards the user's face.

With each of these web gamepads every participant had to perform several tasks in two major test phases: the *training phase* and the *gaming phase* (cf.figure 5.3). At first the goal of the training phase was to move a white circle into the center of a blue circle of varying size as quick and precise as possible. At the same time our test application was logging several data in respect of completion time, accuracy in centre the circle, selection success and key presses, while the test manager was recording the number of gazes and noting relevant behaviour or commends of the test person. After completing the training test block the participant were interviewed and asked to rate the gamepad with a questionnaire as attached in C.

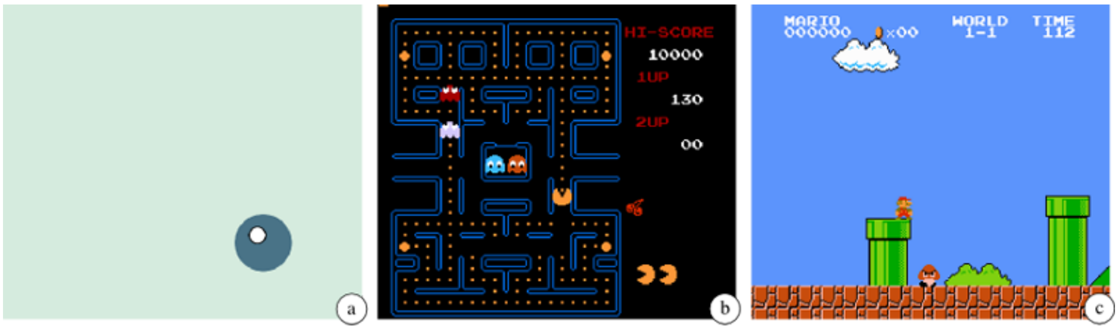

Figure 5.3: We used three tasks to compare the gamepads: (a): a formal multi-dimensional pointing test consisting of targets with varying distance and size, (b): Pac-Man game and (c): Super Mario Bros. game. [2, p.7]

Then the user could start with the gaming phase and were asked to play the two well-known video games *Pac-Man*[1] and *Super Mario Bros.*[2] (executed by an emulator for PC running on the notebook) with the remote control used in the training phase. During each game every interaction command was logged by the platform. After the participant had solved the gaming task, spent all lives or lost the interest the test manager recorded the game score (amount of lives), the level completion time and made some notes from a short follow-up questioning. Finally every user was asked to rate all tested gamepad techniques with regard to the two games in a final interview. The last three questions referred to the acceptance of web-based interaction techniques in general (cf. questionnaire in C).

Overall we conducted the user study with 30 game-affine participants (15 male and 15 female) in the age between 19 and 50 years. Every entire test procedure took about two hours per test person.

[1] `http://en.wikipedia.org/wiki/Pac-Man`, Accessed: 2015-13-03
[2] `http://en.wikipedia.org/wiki/Super_Mario_Bros.`, Accessed: 2015-13-03

5.2 Analysis of web-based smartphone gamepads

In this section all collected testing material in form of logging data, questionnaires, interview and observation notes are examined in more detail to get better insight into the usage of web-based remote controls. The aim is to evaluate the four gamepad prototypes in terms of *user behaviour*, *user experience* and *user acceptance* in a gaming context.

Therefore, all collected data was analysed with the thematic analysis method [10] to cluster the results and find patterns and relationships in using the different gamepads. The user experience and acceptance were analysed with data from various comments during interacting, the interviews and the answers of the questionnaire by classifying all data into the following discussed topics: "gazes", "solving tasks and performance", "gamepad preferences", "interaction familiarity " and "web usage". Additionally the user behaviour was mainly deduced from the logging data and from observations during the user study and assigned to the respective theme. The logging data served as a control parameter for recognizing differences or similarities in perceived and measured user behaviour (e.g. tilt was perceived as too quick due to the inaccuracy but in contrast to the logging data it had the highest completion time). Then the results of all four gamepads were discussed and compared referring to each topic. All positive and negative experiences with the different interaction methods were collected and similar, repetitive or remarkable properties were summarized for preparing a guideline.
The results should provide guidance in developing or improving web-based interaction techniques with smartphones for interacting on public displays. Finally relevant learnings, which are also reasonable for other application issues, are summarized and incorporated in the review of the framework.

Gazes on remote control:
Especially interesting for gaming immersion on public displays is the analysis of the number of gazes on the mobile device which are needed during interaction with a remote screen. The observations reveal that the buttons required the highest amount of gazes followed by the d-pad than the joystick and tilt gamepad (cf. figure 5.4a). This behaviour corresponds to the interaction experience of the participants. In answering the question "*ease of usage without looking at the gamepad*" the button and the d-pad got significantly worse scores (e.g. 13 participants chose "very easy" and "easy" without looking with button) than joystick and tilt (e.g. 26 participants chose "very easy" and "easy" without looking with tilt, nobody stated "very difficult").

Most users reported that they need control gazes for interacting with button because they often lost track in (re-)hitting the button especially for opposite directions. The swiping feature of the d-pad was very useful for quickly correcting the direction without looking but users still complained to drift beyond the sensitive area. Although the joystick got better scores, it was observed that several participants tend to slide to the edge of the touchscreen while dragging the joystick knob which leads to an interruption of the current action. Only the tilt gamepad enables the user to stay fully focused on the remote display.

To reduce the amount of gazes, users stated that they would accept an improvement of the gamepad layout through larger buttons or higher colour contrast. It was in particular desirable to recognize the borders of sensitive areas by using a feedback mechanism e.g. vibrational patterns. The main drawback of surface gesture-based methods was the missing haptic feedback in comparison to a classic gaming console. *"You just have the grip with a Playstation controller - so that you don't have to look at it that often."* (User 1, Commented on: 2014-30-06)

Solving tasks and performance:
According to the logging data the directional pad is the winner in performance with the shortest completion time and highest *Pac-Man* score followed by the button. In contrast, the tilt gamepad had the lowest Pac-Man score (cf.figure 5.4b) and the highest completion time particularly by selecting smaller targets. Another disadvantage of tilt was the high inaccuracy which was caused by the significant higher cursor offset than button, d-pad and joystick. Although the accuracy of the joystick was not significant worse than button and d-pad, it could only achieve the second lowest *Pac-Man* score. The button got higher scores but was more sensitive to target orientation. Thus it had a longer selection time than d-pad especially for selecting targets in non-orthogonal directions. This direction dependence was reflected in the user experience.

In general most participants had positive comments for solving the different tasks with the button and the d-pad but the slightly difference was the ranking between the two games *Pac-Man* and *Super Mario Bros.* While 11 participants stated to solve the task playing *Pac-Man* "very good" and "good" with the d-pad, only 5 users claimed the same for the button. Because the button is restricted to only four directions, the d-pad is more appropriate for moving tasks. Even during the training test many participants appreciated the possibility to move diagonally with the d-pad.

Although the joystick were positively accepted by the users in general, succeeding in the gaming tasks seemed to be difficult. The experience in playing *Pac-Man* was mostly negative, 15 participants stated problems in controlling the character. Also the ratings for playing *Super Mario Bros.* was very low. The main reason was the unusual combination of pressing the action button on the right and dragging the joystick on the left simultaneously for jumping sideways. Similar problems in gaming were stated by using the tilt interaction. Mainly playing *Pac-Man* was completely mentioned negative because the direction commands could not be executed timely and then the users had often missed junctions. Additionally the combination of pressing the action button while tilting the device caused an unintentional movement of the finger and led to errors in the training phase or *Super Mario Bros.* Some participants suggested that other games like a flight simulator would be more appropriate for the tilt gamepad.

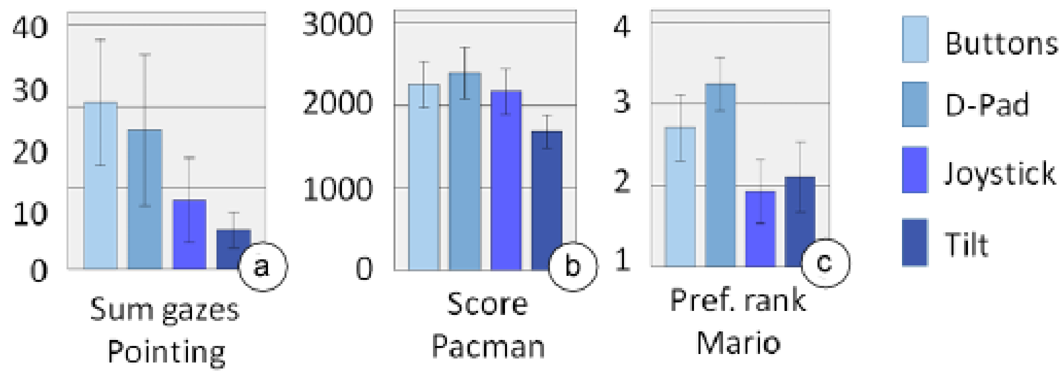

Figure 5.4: (a): mean sum of gazes per pointing task in training phase, (b): mean score per Pac-Man session and (c): mean preference rank with regard to playing Super Mario Bros. (4 = most preferred, 1 = least preferred). [2, p.10]

Gamepad preferences:
Based on the findings from the final interview it is confirmed that generally the preference of a gamepad mostly depends on a certain task. In this user study the directional pad was clearly preferred throughout all tasks in comparison to the others. The preference of the tilt gamepad was ranked the lowest except for playing the *Super Mario Bros.* game, there it was slightly less negative rated than the joystick. (cf.figure 5.4c) Though the joystick seemed not suitable for both gaming tasks in the user study, it was very well-received by most participants for general interaction purposes. 16 test persons stated very positive commands like "*easier for corrections*“, "*intuitive*“ and "*fun to use*“. In answering the question “*what kind of gamepad is preferred for games in general*” the joystick was ranked slightly better than the button.

After solving a certain task every participant had to estimate their user experience with the current gamepad regarding mental and physical effort, accuracy, speed, comfort, usage difficulty, fun and learn effort. In the final interview the user was asked to compare all four gamepads in general and give a overall ranking (1-4 points from worst to best) about the perceived gamepad properties. (cf.figure 5.5c) Despite these answers are more subjective, we can see a connection between gamepad characteristics and the user preference. For example, the d-pad succeeded in all categories and therefore was rated with the highest preference in all tasks while the tilt was mostly low ranked. This evaluation gives a good indication that the more gamepad requirements are well-solved, the greater the user preference for one gamepad. But which property has a significant main effect on the user's choice strongly depends on the certain task as derived from the previous results. For instance, although tilt got higher ratings for “comfort” and “fun to use” than the joystick, the overall evaluation in solving the tasks was mainly negative which also resulted negative for the general preference.

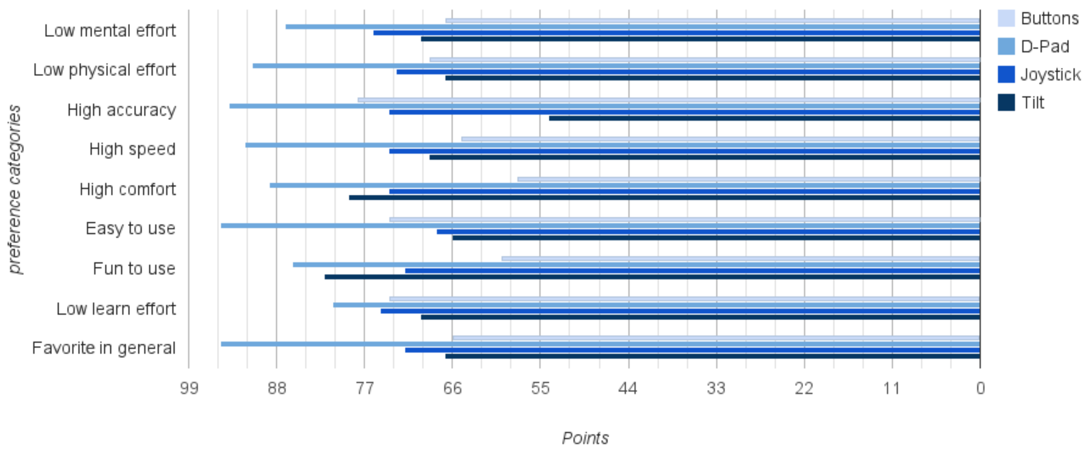

Figure 5.5: Summary of user preferences regarding the categories low mental effort, low physical effort, high accuracy, high speed, high comfort, ease to use, fun to use, low learn effort and favorite in general.

Familiarity with smartphone interactions and web usage:
The results of the user study reveal that there is a clear difference in familiarity with interaction styles known from gamepad controllers and those known from today's smartphones (cf. figure 5.6). While most participants were very familiar in using button and directional pad designs with classic gamepads, these styles are more unknown for smartphone interaction in gaming applications. Because of the integrated accelerometers in nearly every mobile device, tilting have become a more popular interaction style for smartphone users. Nevertheless, this did not show any positive effect on the participants' feedback regarding tilt. Due to the positive comments the joystick turned out to be a promising interaction method but is currently unusual for many users.

Regarding the usage of web-based remote controls 21 participants did not recognize that they are controlling the display application with a web page loaded in a mobile browser. Those few test persons who recognized the web application only noticed it because of the URL shown in the address bar. Nobody perceived any performance drawbacks or connection problems during interaction due to the client-server communication. One participant even claimed not to recognize the web application because of the well-perceived response times. However, only 3 test persons would appreciate the interaction with a web-based remote control. As main reasons they mentioned the key benefits "*no download or (de-)installation necessary*".

But most users did not care about the technical implementation and had stated not to know the difference or they would have been satisfied with any realization as long as it just works.(cf. figure 5.7) One user (who noticed the web application) had argued not to care about the implementation but added "*I would put more trust into a Webapp because I can look up the source code.*"

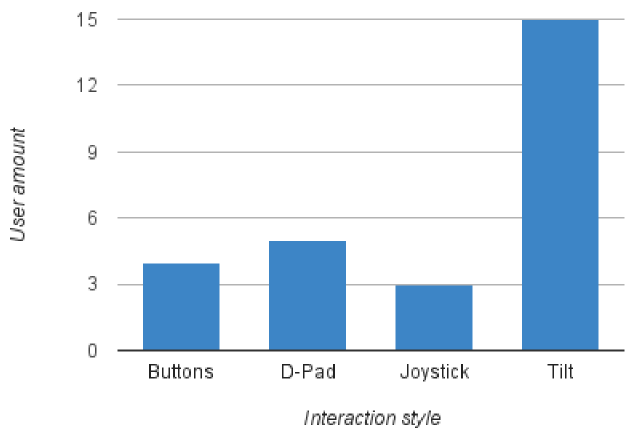

Figure 5.6: User ratings about familiarity of smartphone interactions.

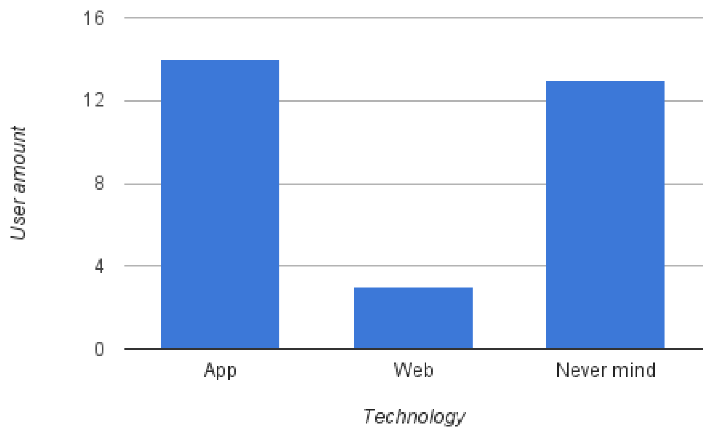

Figure 5.7: User preference of remote control technology.

Almost as many participants would prefer an app installation over the web application. But by analysing the reasons for that preference, it appears that many concerns are made about the requirement of an internet connection e.g. "*with an app no internet connection is needed*".

Only 3 users preferred apps because of higher convenience or just by habit. But most arguments hint that many users may associate costs due to data volume consumption by using browser-based remote controls. For that reason most participants tended to prefer a native app.

5.3 Guidelines for web-based interaction methods

As confirmed by the user study the support of feedback mechanisms is essential for a seamless user experience with web-based remote controls. Especially surface gesture-based interactions like button, d-pad and joystick could benefit from e.g. vibrational signals. The mentioned user requirements also support the findings discussed in 2.1 that feedback must be considered in designing a mobile device interaction with large public displays. Kaviani et.al defines some design strategies for providing feedback and describes the interaction between humans (the spectators) and the interactive display application with the *Norman's Seven Stage Model* [25].

> *"A button widget, for instance, changes it appearance every time a user clicks on it. This is actually not the presentation of an associated state change in the system but a feedback for receiving the input. Such feedback helps users verify the correctness of their inputs while revealing possible mistakes or slips."*[25]

As a consequence the ATREUS framework was extended for implementing a feedback channel with WebSockets as described in 4.

Motion gesture-based interaction techniques such as tilt enable a higher focus on the remote screen and can raise the gaming immersion even without feedback. But the realization of this method must be handled with care. Perceivable inaccuracy or error-prone combinations of interaction styles (e.g pressing button and tilting device) lead to a poor user experience and reduce the satisfaction rapidly.
The following characteristics which are relevant for achieving a seamless user experience with web-based remote controls are summarized:

- Avoiding a blindfold control by providing a feedback mechanism or enhancing the layout.
- High accuracy during interaction; deviations must not be perceived by the user.
- High speed especially for tasks with short reaction times.
- Avoiding unusual interaction styles inappropriate for certain tasks (e.g. dragging left, but jumping right). The interaction should be perceived as natural as possible.
- Combinations of different interaction methods should not affect each other or lead to interaction problems.

- Use perceptions of familiar interaction mechanisms for easy usage and to reduce the learning effort.
- Basically web-based interaction methods are accepted as long as the start procedure is convenient and users need not to take care about the connectivity. If possible provide a local network for communication to reduce the fear of higher costs. In the ideal case they do not miss a native app and cannot realise any difference in performance or convenience.

CHAPTER 6

Use cases for smartphone controllers in public spaces

Moving away from the gaming context this chapter investigates the usage for web-based remote controls in other application areas in general. Based on the studies in the literature chapter 2.1 four representative use cases in public spaces were picked out and further discussed with a focus group during a workshop. The aim was to identify suitable use cases for using the own mobile device as web-based remote control and to analyse the user's needs and expectations from such a system. Above all, the people's preferences and concerns regarding web applications are worked out for improving further development and to evaluate how users can benefit from a web-based approach.

6.1 Method and workshop procedure

For collecting many different ideas and allowing a free creativity process two User-centered design techniques *Co-constructing stories* [11] and *Instant Card Technique* [8] were combined to get qualitative feedback from the focus group. With the storytelling concept of Buskermolen and Terken users were motivated to recall memories, experiences and dreams that was very useful for envisioning the usage of web-based remote controls. The *Instant Card Technique* helped the participants as inspiring material and triggered ideas with focus on use cases only for public spaces. Overall the discussion with the focus group of 5 participants took about two hours. All conversations were audio recorded and the corresponding transcript in German and the introduction questionnaire are attached in the appendix D. Principally the focus group workshop were organized in the following two interview phases:

Phase 1 - Identifying possible use cases:

After a short introduction and a warm-up questionnaire for gaining information about their background and smartphone experience, the participants had to evaluate four use case scenarios with the assistance of so called “idea cards”. These cards are based on the instant card concept of Beck et. al. but extended with more pictures that the people can keep an open mind and without classifying them into defined context groups. The topics presented with the cards were examples from our daily life where we are already accompanied with public displays: “public places”, “public events or exhibtions”, “shopping” and “public transport”. Every idea card contained four sample pictures to illustrate one of these scenarios as shown in figure 6.1.

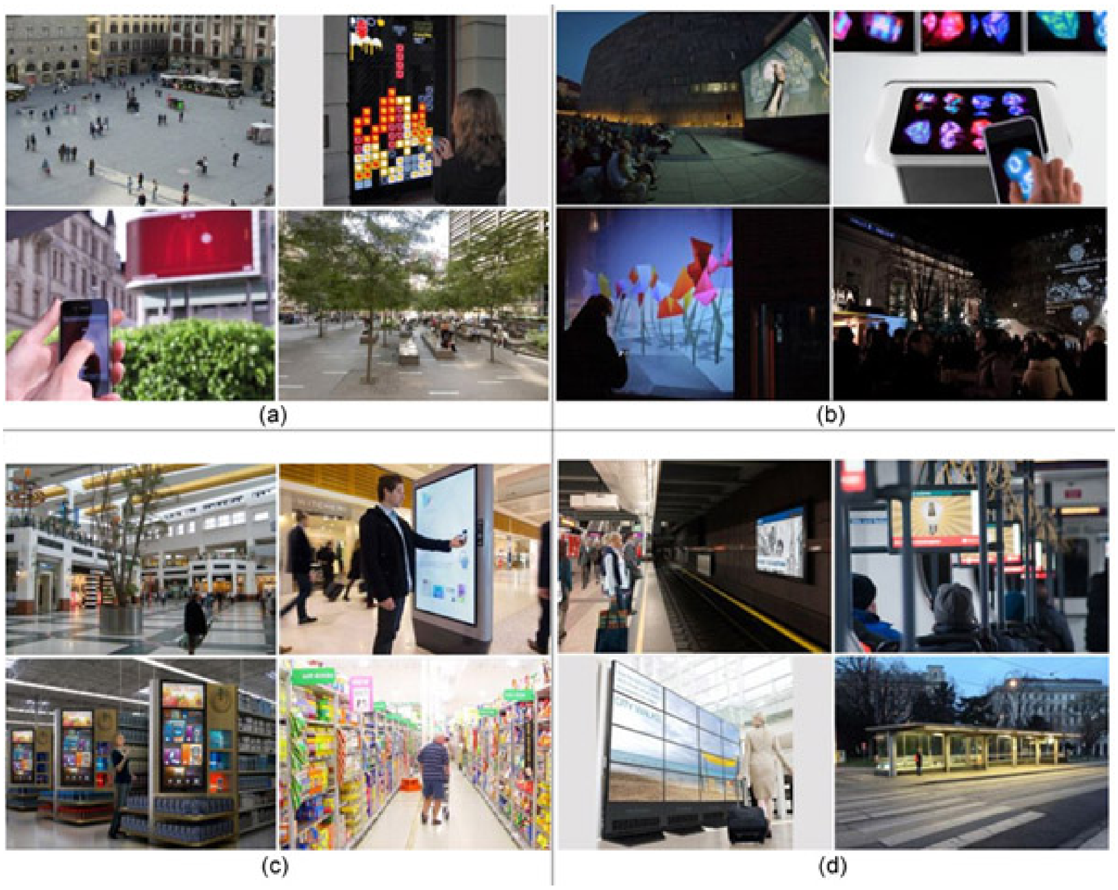

Figure 6.1: All idea cards: each containing four sample pictures with the scenarios: (a): public places, (b): public events or exhibitions, (c): shopping, (d): public transport.

Then every participant got these four idea cards and one empty card which could be optionally used for additionally sketching a missing idea. Next every person was asked to rate each scenario with 1-4 points (4 for the most appealing use case) on its own which would be most interesting for using the smartphone as a remote control for public displays (cf. figure 6.2). After a preparation time of about 5 minutes the assigned points per public scenario were quickly evaluated and the results were noted on a flip-chart. The shown ratings gave a first discussion impulse for answering why they would (would not) tend to use their smartphone as remote control in a specific context. Each of the four idea card topics were discussed within a conducted interview with the focus group and relevant points were took down onto the flip-chart. In the end the use case which emerged to be the favourite by the focus group is chosen for a more detailed discussion in the second workshop part.

Phase 2 - Discussing one preferred use case:

The aim of this interview phase was to analyse different interaction issues and user's needs or concerns by reference to a specific context. Therefore, one use case as result from the first discussion part was considered in more detail. The focus group was asked to imagine this certain use case similar to the storytelling method described in [11] and interviewed about different issues regarding the usage of their smartphone and the public display. The pictures and some notes or sketches on the flip-chart were used to support the conducted discussion. Figure 6.3 and 6.4 show the workshop setting and the discussion materials. For not losing the thread it was always pointed out to focus on the main usage of controlling the public display with the smartphone and that the mobile application is purely web-based.

Figure 6.2: Use case rating:
e.g. 2 points for public places, 3 points for public transport and 1 point for public events or exhibitions.

Figure 6.3: Workshop setting for the focus group.

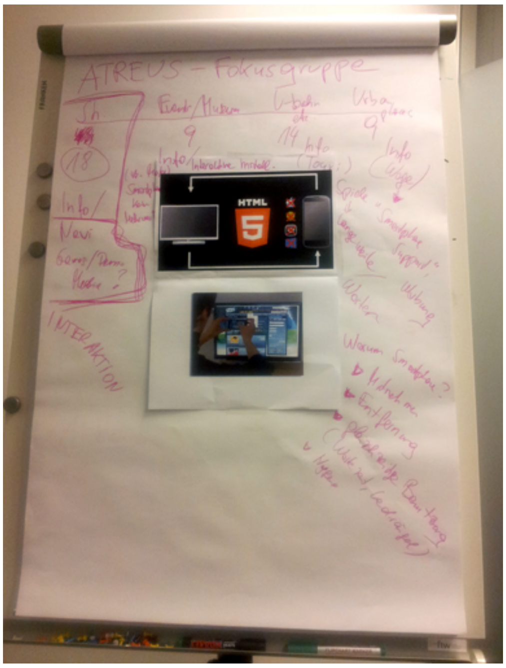

Figure 6.4: Illustrating notes on the flipchart with rating points per use case.

6.2 Analysis of focus group discussion

For evaluating the workshop different qualitative data were collected with the warm-up questionnaire, the rating of the use cases, the notes from the flip-chart and the audio record of the interview. The complete transcription and the questionnaire of the focus group discussion is attached in the appendix D. All data was prepared by using the thematic analysis method [10] to figure out different aspects for using a web-based remote control in the most preferred use case. The results of the use case rating of the first interview phase and the associated flip-chart notes helped to find a consensus between different opinions of the focus group and to structure the evaluation topics. The answers of the introduction questionnaire helped to assess the personal experience and motivation for better understanding the arguments. For instance, the topic "information take-away" from the first interview phase or "finding products" from the second interview phase were important scenarios for the participants and thus were documented on the flip-chart (cf. figure 6.4). Then for further analysis all statements from the transcript referring this topic were grouped to summarize the key points of the discussion. Repetitive arguments by several participants or opinions most persons agreed were joined into a core statement for developing the key aspects for the user acceptance.

The focus group consisted of 5 participants aged between 23 and 33 years who are very well experienced with the usage of smartphones or other mobile devices. They claimed to use their smartphone on an average of 4 hours a day but only 1 participant had experience in using the smartphone as a remote control with *VLC Remote*[1]. 3 participants have already used the smartphone with a public screen via QR Code or Bluetooth technologies.

According to the results of the use case ratings as demonstrated in figure 6.5 the scenario "*Shopping*" gained the highest score followed by "Public transport". The following discussion clarified the main reasons for this ranking. Almost all participants could imagine a meaningful application for different shopping issues. In comparison to the other scenarios the focus group missed the practical benefit in using their smartphone with a public display except for entertaining purposes (e.g. public gaming) or as a pastime. Only one participant would prefer a system for public events or exhibitions over the shopping use case but mainly for gaining personalized information during a concert or a museum trip.

> *"I can imagine being in an open air concert and controlling the screen in such a way that I can get some information autonomously - meaning personalized in that case. Or for example in a museum for quickly gaining some information."*

Another person suggested a special application as learning tool for children as a meaningful use case for a museum exhibition.

[1] `https://wiki.videolan.org/Control_VLC_from_an_Android_Phone/`, Accessed: 2015-18-03

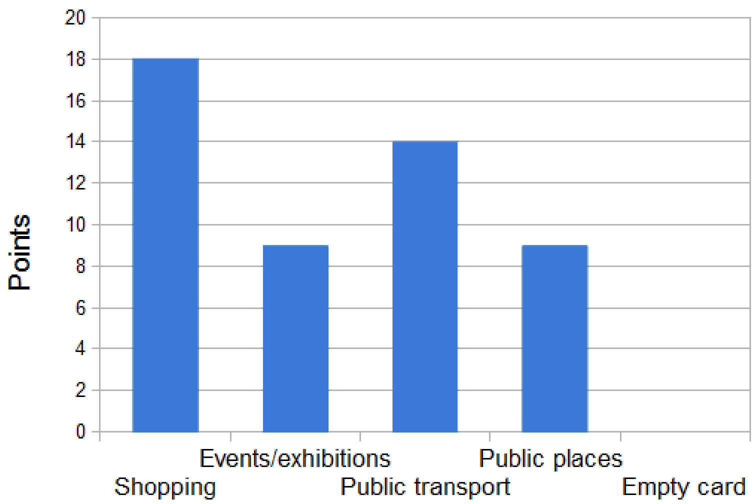

Figure 6.5: Rating result of the four use cases. Most preferred use case "shopping" with 18 points, no usage of optional empty cards.

The most appealing use cases for scenarios regarding public transport or public spaces were applications for entertaining and marketing purposes which was mainly supported by opinions like "*at the subway there is enough time and I am bored anyway*" or "*while waiting everybody is looking at the own smartphone anyway, maybe playing something together would be more fun*". Regarding the smartphone controller as a navigational tool the wish came up for a location-based orientation guide combined with a public display especially at unknown places or abroad. They found it very helpful to retrieve or take away information at any time desirable without accessing the internet over the own mobile network. They also suggested to use the smartphone with the public display as an interactive navigation help. Similar arguments relating information, navigation, orientation guidance, gaming and advertising purposes were mentioned as main reasons for shopping:

> *"For me it is important for comparing products online with the smartphone. This [shopping] comes into my mind because it would be an extension of something what I already do."*

> *"I am partly overwhelmed with the offerings while buying groceries. I want to orient myself much better. Where can I get something? How can I optimize my shopping route?"*

But most use cases are not provided or poorly conceived for this application area, although the discussion revealed people would appreciate it more to enhance the shopping experience. Therefore, the shopping scenario was regarded as most useful by the focus group. The following examples were discussed in the second interview phase to analyse the usage and requirements of web-based remote controls in this specific context:

- **Finding products and navigation**:
  The participants stated that they would prefer using the smartphone controller as orientation guide in public places for finding a shop or product as well as within a big shopping centre. Assuming a large screen at the shopping centre entrance the focus group suggested to use the smartphone for choosing or searching a product. Then a map of the shop with the product's location should be shown on the public display. This orientation plan can be further explored by using the smartphone as a web-based remote control. During the interview they also mentioned concerns about long waiting times due to a big crowd in front of the display. That's why they affirmed the advantages of interacting with the own mobile device even at long distances. Additionally they desired to download the information about the route from the map and use it for navigating with the smartphone.

- **Product information and demonstration:**
  Another benefit which was discussed by the focus group was getting more information about a specific product through a public display and a smartphone controller. With the big screen size the properties of a product can be represented in more detail and the smartphone allows new possibilities for exploration. As an example for buying furniture that is not stored, a 3D representation could be shown on a large screen and rotated with the accelerometers of the smartphone to view it from all different angles. Another sample mentioned in the context of buying a car was to start a virtual test drive on the display and control the car with the own mobile device.

- **Getting help or advice:**
  The focus group also expected support from the shop employees or other customers during shopping which can be requested with the smartphone and the public screen. Some examples mentioned were: a collaborative voting or experience reports about a product, recommendations by other customers or personalized offers and advertising. But in this case some participants had concerns about the privacy issues in publishing the own shopping behaviour on a public display.

  With respect to using the smartphone as a web-based remote control an idea was described for buying clothes. Assuming that every changing room has a screen which is synchronized with several screens for the shop employees, the customers could control the information on the employee's screen with the smartphone. If the clothes do not fit, the customer can declare all wishes by controlling the remote screens and call the employee without leaving the changing room. The benefits of using the own mobile device in this case are additionally features like e.g. scanning tags on the clothes or saving personal shopping preferences on the device.

- **Special offers and promotions:**
  Gaming applications on a public display controlled by the smartphone were often mentioned in conjunction with special offers or innovative product promotions. But all participants agreed they would expect a voucher or a similar offer as reward for playing with other customers, otherwise they would not be motivated to participate. The main reason was that in contrast to the situation at public transport places people usually do not have much time when doing the shopping.

This interview with the focus group showed that using smartphones as web-based remote controls varies depending on the different tasks in a certain use case and where, when and in which context the interaction takes place. Nevertheless, all participants mentioned several benefits in using such a system and also mainly appreciated the web-based approach. To analyse the probable user behaviour in the shopping context, the research methods used in the workshop were too limited because it was difficult for the participants to imagine real and detailed interaction steps. For gaining more insights in this topic a user study with a prototype or simple mock-ups would be a more promising method. All user expectations and preferences are summarized in the next section.

6.3 Key aspects for user acceptance

Although the workshop focused on using smartphones as remote controls, the wish of gaining and downloading information from the display kept coming up for all four use cases. The main reasons mentioned were: avoiding a long waiting period in case of too many users, taking away personal information or saving complicated information (e.g. directions) for later and for extreme distances reading information on the smartphone screen.

> *"On a screen, where it is readable - that is ok for me. But if there is a long way to the display, I would prefer to download the information and carry it with me."*

According to such frequently statements getting information is seen as an essential factor why users prefer smartphones in combination with public displays. Further advantages which correspond to results from the literature review were stated by the focus group:

- Overcoming large distances between user and display.
- Support of multi-user interaction for avoiding long waiting periods and scramble.
- Avoidance of touching dirty displays or other devices at public spaces.

The display size, a quick connection and guaranteed privacy in some situations (e.g. ratings) were main aspects which were also relevant for the general user acceptance.

One basic requirement for the focus group and also mentioned by participants from the aforementioned user study was the internet connection. *"Do I need an internet connection for that?"* – this consideration was always adduced for further evaluation of the use cases. This was caused because many people associate a web-based application with higher costs. The participants of both studies argued that they did not want to waste their free data volume just for interaction purposes and therefore tend to avoid a mobile network connection. But a system with an easy pairing method and working over a local Wi-Fi connection, as realized for both studies, was largely accepted and could overcome this challenge. Furthermore, all participants appreciate the key advantages of the web-based approach that no app installation is necessary and the independence of any mobile platform.

> Question: Why would you prefer a web application?
> *"Because I am sure for Windows phone there is no app for that..."*

To enhance the motivation for pairing the own smartphone with a public display – also referred as *serendipity* [7] – the following aspects were relevant for all participants:

- The connection establishment must be easy and quick e.g. avoiding entering an IP address manually.
- The pairing process must not be automatically, users do not want to be forced to connect to a public display.
- Accessing the display by several users must be well-regulated to avoid an interaction mess. A time-regulated access management for exclusively one user could be useful for some kind of tasks.
- Connecting with the public display should also be possible over large distances e.g. using Wi-Fi or Bluetooth technologies instead of touch-based methods like NFC.

All these requirements can be realized with latest web technologies as already discussed in the previous chapters.

CHAPTER 7

Discussion

In this chapter the results of the framework development, the user study and the focus group are further evaluated and compared with insights from previous related work in chapter 2. The goal is to gain more detailed knowledge about: realizing present interaction technologies for mobile devices with current web technologies, the user experience and other challenges in coupling the smartphone with a public display. Therefore the following research questions from 1.3 were elaborated:

1. *RQ1: What existing interaction techniques can be realized with state-of-the-art web technologies on smartphones?*
2. *RQ2: How do people use their smartphone as a web-based remote control for interactive applications?*
3. *RQ3: What use cases can be identified for using smartphones as web-based remote controls with public displays?*

RQ1:
The development of the framework and the experiments with the web-based remote control demonstrators proved that *surface gesture-based*, *motion gesture-based* and *camera-based* interaction techniques, similar to existing approaches as discussed in section 2.2, can be realized with today's web technologies only. Moreover, web features such as vibration or sound can be used to support different feedback mechanisms. While the HTML5 specifications and the mobile browser support are improving, several aspects for implementing a web-based remote control still have to be considered to achieve a reliable functionality for a seamless user experience. In particular the support of *motion gesture-based* and *camera-based* methods needs further optimizations.

To analyse all these strengths and limits of latest web technologies, essential requirements were deducted from related work in 2 for developing the *ATREUS framework*. In contrast to prior work about framework development for mobile phones as input devices it does not depend on proprietary systems or device characteristics.[46, 29, 22] The main benefit was providing a system which allows the development of completely platform-independent remote controls for mobile devices. Furthermore, user-centred features like enabling feedback mechanisms or ensuring a reliable and easy connection process – all relevant for a seamless user experience – were considered in the system design. Because of the purely browser-based interaction techniques, no app installation is necessary which encourages a low threshold of use. Up to now only early attempts with Bluetooth technologies were made to avoid the installation process but proved a failure due to the low user acceptance of enabling Bluetooth on the mobile phone.[26, 15]
Based on these results the *ATREUS framework* met all basic conditions for realizing smartphone controllers for interactive applications on public displays. The following experiments with different devices indicated good results and a major support of HTML5 features by common browsers. Although some web features are still in draft stage by the W3C. The access of all hardware features relevant for interaction was widely assured which further approved the usage of a browser-based approach contrary to mobile technologies studies referred in [21, 47]. An overview about current HTML5 features and their further development were discussed in chapter 2.4 in more detail. However, the different browser behaviour has to be taken into account already in the implementation. As evaluated in section 4.4 web-based remote controls were primarily affected by the used browser version. But there exists some implementation possibilities to overcome these problems e.g. using meta-descriptions for suppressing default browser behaviour. While surface gesture-based interaction methods worked without any problems in common mobile browsers, motion gesture-based and camera-based remote controls had to be handled with care. For example, several browsers returned different angle values for the device orientation and Apple devices did not support the live media feature for accessing the camera. The widely used mobile browsers *Google Chrome* and *Opera mobile* provided the best results for all demonstrators tested. In summary the further development and the increasing support of today's web technologies are still in progress but result in a highly promising approach for web-based remote controls for interacting with public displays.

RQ2:
In general the usage and the preference of a certain interaction style depend on the specific task. Factors like speed, accuracy and naturalness of the interaction style have shown an affect on the user behaviour in completing a task and, therefore, should be considered in developing smartphone controllers. The web-based interaction approach is widely accepted and do not have any influence on the user experience as long as the connection process is convenient and there are no additional costs for the users.

Considering this user perspective part of the master thesis an user study was conducted in cooperation with *FTW* for analysing how people would use their smartphone as a web-based remote control. Since gaming applications are very popular for most people and present several challenges regarding speed and accuracy, the gaming context 2.3 was used for further evaluation. Therefore, we compared four different gamepads which were implemented with the aforementioned framework. With observations, logging data and interviews the interaction with the web-based remote controls was investigated regarding user behaviour, user experience and user acceptance. More details about the user study was given in chapter 5.

The results revealed that the chosen interaction method had a notable influence on the user behaviour in solving a certain task. Regarding the frequency of gazes on the mobile screen, surface gesture-based techniques (e.g. button and d-pad) required more gazes than motion gesture-based methods (e.g tilt). This was caused by a lack of tactile or audio feedback what made it difficult for users to fully concentrate on the remote screen. In comparison to a more tangible classic gamepad the physical properties of a mobile device were not sufficient. Thus the support of feedback is very important for smartphone controllers in particular during tasks that needed quick reactions when switching the button or doing some corrections. Kaviani et al.[25] considered a feedback process already in the design phase of public display applications for mobile device interaction. Based on all these insights the *ATREUS framework* was extended to allow the development of web-based remote controls with feedback features.

Despite the higher requirement of control gazes surface gesture-based gamepads performed better in accuracy, speed and number of errors and were higher preferred by the participants especially for precise tasks. Similar findings were made in the explorations of Cairn et.al [12] affirming that the design of the touchscreen interface had an affect on the interaction performance. But arguments in [42] assessed the acceptability of motion gesture-based interaction fairly high which did not hold true for the user study. The tilt control was rated the lowest in most relevant user preference and performance categories throughout all tasks. It turned out that this controller is not suitable for all kind of tasks and needed tailored applications to achieve a natural user experience. Previous researches also showed that for appropriate games a more natural mapping with motion-based interactions were possible. [12, 22] Combinations of different interaction methods (e.g. tilting and pressing button) in one gamepad should be taken carefully because for some tasks the interaction appeared unusual or led to many unintentional errors.

The familiarity with common interaction styles on smartphones or classic gamepads did not indicate a significant affect on the user preference but can be helpful in reducing the learning effort and simplifies the interaction. Further relevant factors resulted from the user study to ensure a seamless user experience were summarized in the guidance of section 5.3.

Concluding from all results the user experience with web-based remote controls was well-perceived, no complains about delays in reaction times or other performance issues were made.

The user preference in comparison to native apps was only impaired because many participants associated web applications with higher costs based on the idea of requiring a permanent internet connection. But this could be challenged with an implementation without needing an internet connection by providing the communication over a local Wi-Fi network. All in all most users did not recognize the difference to a native app and are satisfied as long as it works fluently and without additional effort.

RQ3:
Some examples of identified use cases for using smartphones as input devices are given in table 7.1. Whereas the user study focused on gaming applications, the results of the focus group indicates that people could benefit more from use cases in a shopping scenario. Basically the discussion revealed that users appreciate the advantages of using the own mobile device as a web-based remote control in public spaces. But to achieve a wide user acceptance, appropriate pairing methods and interaction styles have to be provided to fulfil people's need and expectations in a specific use case.

As detailed in chapter 6 a workshop with 5 smartphone-affine participants in two interview phases was done to get more insights into the users' requirements for a certain use case. The creativity process and the flow of the discussion were supported by idea cards which represented the use cases "public transport", "public places", "shopping" and "public events or exhibitions". As a result the use case *"shopping"* was preferred by the participants because new developments in this application area were missed and thus argued to be most useful in comparison to the others. Based on the classification of Sarah Clinch [13] similar use case scenarios were mentioned by the focus group in a shopping context as summarized in table 7.1.
The evaluation showed that although the focus of the discussion was interaction with smartphones, the focus group regarded the functionality of *Information Take-away* as an essential part of each use case. Even so all participants appreciated the benefits of smartphones as input devices like overcoming large distances, support of multi-user applications and improved hygiene as also described in the work of Ballagas et al.[7]
To encourage people getting engaged in the interaction with public displays, the opinions in the focus group agreed with the findings of Kaviani et.al [25] like e.g. privacy for specific tasks. One main request was a quick and easy pairing method with the public display to reduce the threshold of use. Several aspects that were important for the participants in establishing a connection were summed up in chapter 6.3. All participants preferred a connection which helps overcoming large distances between user and display to avoid unnecessary walking. According to the study of Baldauf et al. [4] and derived from the discussion *display pointing* would be a suitable pairing method that meets all the mentioned requirements.

USE CASE	Interaction	Information Take-Away	Co-Displays & Cyberforaging	Personalization
Finding products & shop navigation	Navigation	Keeping shop map	Exploring shop plan	—
Product info & demo	Tilting 3D product model	Saving product information	Exploring product	—
Getting help or advice	Controlling employee's screen	Saving personal shopping preferences	—	Votings or personalized advertising
Special offers & promo	Gaming for marketing purposes	Downloading voucher	—	—

Table 7.1: Shopping use cases for possible interoperation between smartphones and public displays.

As already known from the user study also the focus group had concerns about the need of an internet connection because the web-based approach is associated with higher mobile network costs. But after a short demonstration that the web-based remote controls also worked without any internet connection by communicating via a local Wi-Fi network, all doubts were eliminated. This caused a significant positive affect on the user acceptance for the browser-based interaction approach. Furthermore, the focus group appreciated the main benefit of platform-independence because some of them have often experienced that some native apps are not available for their smartphone.

CHAPTER 8

Conclusion and future work

The topic of this master thesis was investigating browser-based remote controls for interactive applications on public displays. With the increasing smartphone usage, the growing processing power of mobile phones and the spreading popularity of large screens interactive applications have become indispensable in our modern information age. Various scenarios primarily at urban spaces present many sophisticated challenges for which new technologies have to be explored. This research elaborated one promising approach for future applications in this area. The aim of this study was not only to explore latest web technologies for new interaction methods but also taking the user perspective and possible application areas into account. For this purpose three main aspects were focused on: the development of web-based interaction technologies, the investigation of the user experiences with web-based remote controls and the analysis of the user acceptance by identifying different use cases with public displays. Therefore, the following three methods were used for this research: the framework development, the user study and the focus group. All elaborations were based on the previous literature review.

Summarizing the insights of all research methods the usage of browser-based remote controls turned out to be a very useful and widely accepted interaction technology for public displays. Although some HTML5 features are still in draft stage or can evoke different browser behaviour, using latest web technologies proved to be suitable for common interaction techniques. Appropriate interaction styles depending on a certain task and different challenges of the use case have to be developed. The guidelines derived from the results of the user study and the focus group provided some helpful indicators for developing efficient web-based remote controls. The supported features of latest web and hardware technologies are further improving, so a wide range on new advanced interaction methods can be realized to enhance a seamless user experience.

This work mainly focused on the investigation of surface gesture-based and motion gesture-based interaction styles but for future work camera-based and optimizations for the motion gesture approach should be further explored. Additionally the support of feedback with mobile devices should be more taken into account and examined in a pursued user study. For a more detailed and future research it would be interesting to perform all experiments and user studies with browser-based remote controls in a real urban setting. The specific challenges and various use cases of public spaces could reveal further requirements which would verify the capabilities of web-based remote controls.

APPENDIX A

Technical documentation

A.1 Source code snippets

```
# format:
# (id of html element)=(keycode to be
    simulated)
# keycode can be either specified as int or as
     keyevent constant
cross_up=VK_UP
cross_down=VK_DOWN
cross_left=VK_LEFT
cross_right=VK_RIGHT
buttonSelect=VK_SPACE
buttonStart=VK_ENTER
buttonA=VK_A
buttonB=VK_B
```

Listing A.1: gamepad.map

```
public abstract class FeedbackMessage {

    protected Map<String, Object> jsonData =
    new HashMap<String, Object>();

    private static ObjectMapper mapper = new
    ObjectMapper();

    public void setBroadcast(boolean broadcast
    ) {
        jsonData.put(AtreusConstants.
    FEEDBACK_BROADCAST, broadcast);
    }

    public String getMessageString() {
        ByteArrayOutputStream baos = new
    ByteArrayOutputStream();
        try {
            mapper.writeValue(baos, jsonData);
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } finally {
            try {
                baos.close();
            } catch (IOException e) {
                // TODO Auto-generated catch
block
                e.printStackTrace();
            }
        }
        return baos.toString();
    }

}
```

Listing A.2: FeedbackMessage.java

```
    public static void main(String[] args) {

        log.info("ATREUS start....");

        final AtreusServer as = AtreusServer.
getInstance();

        // enable feedback function
        SettingsManager.setFeedback(true);
        SettingsManager.save();

        as.setEventListener(new
IMobileEventListener() {

            @Override
```

```java
            public void mobileClientDisconnected(String id) {
            }

            @Override
            public void mobileClientConnected(String id, Session session) {
                // send three feedback messages when new client connects:
                // broadcast
                as.sendFeedback(new VibrationFeedback(
                        new long[]{500, 300, 1000}));
                // broadcast
                as.sendFeedback(new SoundFeedback(new String[]{
                        "/ATREUS_mobile/sounds/success.mp3", "0.5"}));
                // targeted
                as.sendFeedback(id, new TextFeedback(
                        "this is a targeted textual feedback for id " + id
                                + "!"));
            }
        });

        as.start(SettingsManager.getIP(), SettingsManager.getPort(),
                AtreusConstants.HTML_PATH, AtreusConstants.FB_DEMO);
```

Listing A.3: FeedbackTest.java

```java
public class AtreusServer {

    private static final Logger log = LogManager.getLogger(AtreusServer.class);
    private static AtreusServer serverInstance = null;
    private Server server;
    private IMobileActionListener mobileActionListener = null;
    private IMobileEventListener mobileEventListener = null;
    private HashMap<String, Session> allSessions = new HashMap<String, Session>();
    private ScreenStreamer screenCaptureThread = null;

    public static AtreusServer getInstance() {
        if (serverInstance == null) {
            serverInstance = new AtreusServer();
            log.info("ATREUS http server started...");
        }

        return serverInstance;
    }

    public void start(String ip, int port, String resourceBase,
            String welcomePage) {

        server = new Server(port);

        AtreusUtils.initWebsocketSettings(ip, port, AtreusConstants.JS_FILE);

        // create/register Handler for WebsocketServer
        WebSocketHandler wsHandler = new
```

```
WebSocketHandler() {

            @Override
            public void configure(
WebSocketServletFactory wsServlet) {
                // TODO Auto-generated method
stub
                wsServlet.register(
AtreusServerSocket.class);

            }

        };

        // create handler for html files
        ResourceHandler resource_handler = new
 ResourceHandler();
        resource_handler.setWelcomeFiles(new
String[]{welcomePage});
        resource_handler.setResourceBase(
resourceBase);

        HandlerList allHandlers = new
HandlerList();
        allHandlers.setHandlers(new Handler[]{
wsHandler, resource_handler});

        server.setHandler(allHandlers);

        try {
            server.start();
        } catch (BindException e) {

            log.error("Problem binding to port
 " + port + " : "
                    + e.getMessage());
            JOptionPane.showMessageDialog(null
, "Problem binding to port "
                    + port + ". Please try
another port!", "Server error",
                    JOptionPane.ERROR_MESSAGE)
;

            if (server != null) {
                try {
                    this.stop();
                } catch (Exception ex) {

                }
            }

        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        try {
            server.join();
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            // e.printStackTrace();
        }

}
```

Listing A.4: AtreusServer.java

```
@OnWebSocketConnect
public void onConnect(Session session) {
    // store reference to session
    this.session = session;

    // create unique client identifier
    this.clientID = session.
getRemoteAddress().getHostString() + ":"
            + session.getRemoteAddress().
getPort();
```

```java
        log.info(this.clientID + " tries to connect...");

        // send screen resolution
        if (session != null) {

            String reply = null;

            if (SettingsManager.getMaxUser() > 0
                    && AtreusServer.getInstance().getAllSessions().size() > SettingsManager
                            .getMaxUser()) {

                reply = "{\"responsecode\": 1, \"message\": \"Connection failed, max user reached\"}";

            } else {

                reply = "{\"responsecode\": 0, "
                        + "\"message\": \"Connection successful\", "
                        + "\"screenx\": " + SettingsManager.getScreenX() + ", "
                        + "\"screeny\": " + SettingsManager.getScreenY() + ", "
                        + "\"screenwidth\": "
                        + SettingsManager.getScreenWidth() + ", "
                        + "\"screenheight\": "
                        + SettingsManager.getScreenHeight() + ", "
                        + "\"feedback\": " + SettingsManager.isFeedback() + "}";

                // add to list of connected clients
                AtreusServer.getInstance().getAllSessions()
                        .put(this.clientID, session);

                // call mobileEventListener
                if (AtreusServer.getInstance().getEventListener() != null)
                    AtreusServer.getInstance().getEventListener()
                            .mobileClientConnected(this.clientID, this.session);

            }

            try {
                Future<Void> fut;
                fut = session.getRemote().sendStringByFuture(reply);
                fut.get(1, TimeUnit.SECONDS);

                log.info("Replied to " + this.clientID + ": " + reply);
            } catch (Throwable t) {

                t.printStackTrace();
            }
            // close connection if maxuser reached
            if (SettingsManager.getMaxUser() > 0
                    && AtreusServer.getInstance().getAllSessions().size() > SettingsManager
                            .getMaxUser())
```

```
                session.close();

        }

    }

    @OnWebSocketMessage
    public void onMessage(String msg) {

        if (msg.startsWith("{")) {

            MobileData data = null;
            ObjectMapper mobileMapper = new ObjectMapper();
            try {
                data = mobileMapper.readValue(msg, MobileData.class);
                data.setClientId(this.clientID);

            } catch (JsonParseException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (JsonMappingException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

            log.info("Received command from client " + data.getClientId()
                    + ": " + data.toString());

            IMobileActionListener actionListener = AtreusServer.getInstance()
                    .getActionListener();
            if (actionListener != null)
                actionListener.triggerAction(data);
        }

    }
```

Listing A.5: AtreusServerSocket.java: onConnect and onMessage

```
    public static void main(String[] args) {

        log.info("ATREUS start....");

        AtreusServer as = AtreusServer.getInstance();

        // read server address from config
        if (SettingsManager.isShowqr()) {
            JFrame qrwindow = QRCodeCreator.getQRCodeWindow(SettingsManager
                    .getURL());
            qrwindow.setVisible(true);

        }

        // set actionlistener
        if (SettingsManager.getRemoteControl().equals(
                AtreusConstants.DRIVING_WHEEL)) {
            as.setActionListener(new DrivingWheelListener());

```

```
        } else if (SettingsManager.getRemoteControl().equals(
                AtreusConstants.FB_DEMO)) {
            log.error("Only for feedback demo, start FeedbackTest.java");
        } else if (SettingsManager.getRemoteControl().equals(
                AtreusConstants.GAMEPAD)) {
            as.setActionListener(new GamepadListener());

        } else if (SettingsManager.getRemoteControl().equals(
                AtreusConstants.GAMEPAD_BTN)) {
            as.setActionListener(new DemoGamepadListener());

        } else if (SettingsManager.getRemoteControl().equals(
                AtreusConstants.GAMEPAD_DPAD)) {
            as.setActionListener(new DemoGamepadListener());

        } else if (SettingsManager.getRemoteControl().equals(
                AtreusConstants.GAMEPAD_JOYSTICK)) {
            as.setActionListener(new DemoGamepadListener());

        } else if (SettingsManager.getRemoteControl().equals(
                AtreusConstants.GAMEPAD_TILT)) {
            as.setActionListener(new DemoGamepadListener());

        } else if (SettingsManager.getRemoteControl().equals(
                AtreusConstants.MINIVIDEO)) {
            ScreenStreamer.start(SettingsManager.getPort() + 1,
                    SettingsManager.getScreenRect());
            as.setActionListener(new MinivideoListener());

        } else if (SettingsManager.getRemoteControl().equals(
                AtreusConstants.POINTER)) {
            as.setActionListener(new PointerListener());

        } else if (SettingsManager.getRemoteControl().equals(
                AtreusConstants.SMARTLENS)) {
            PhotoReceiver.start(SettingsManager.getPort() + 2,
                    SettingsManager.getScreenRect());

        } else if (SettingsManager.getRemoteControl().equals(
                AtreusConstants.TOUCHPAD)) {
            as.setActionListener(new TouchpadListener());
        } else {
            log.error("No corresponding listener found");
        }

        as.start(SettingsManager.getIP(), SettingsManager.getPort(),
                AtreusConstants.HTML_PATH,
```

```
    SettingsManager.getRemoteControl());

    }
```

Listing A.6: StartAtreus.java

```
<!DOCTYPE html>
<html lang="de">
<head>
<meta charset="UTF-8" />
<!-- prevent zooming in -->
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no" />
<meta name="application-name" content="ATREUS mobile library" />
<meta name="description" content="Classic gamepad interface for ATREUS platform" />
<meta name="author" content="http://atreus.ftw.at" />
<title>ATREUS classic gamepad</title>
<script type="text/javascript" charset="utf-8" src="ATREUS_mobile/js/jquery-1.11.1.js"></script>
<script type="text/javascript" charset="utf-8" src="ATREUS_mobile/js/json2.js"></script> <!-- json fallback for old browsers-->
<script type="text/javascript" charset="utf-8" src="ATREUS_mobile/js/atreus-connection.js"></script>
<script type="text/javascript" charset="utf-8" src="ATREUS_mobile/js/atreus-controls.js"></script>


<!-- add interaction eventlisteners and other functions for your mobile application -->
<script>
$(document).ready(

    function() {

      //Object Events
      var buttonObjects = [ document.getElementById("cross_up"),
          document.getElementById("cross_down"),
          document.getElementById("cross_left"),
          document.getElementById("cross_right"),
          document.getElementById("buttonSelect"),
          document.getElementById("buttonStart"),
          document.getElementById("buttonA"),
          document.getElementById("buttonB") ];

      addControlEvent(buttonObjects, "touchstart", buttonPressed);
      addControlEvent(buttonObjects, "touchend", buttonReleased);

      addControlEvent(buttonObjects, "touchenter", buttonPressed);
      addControlEvent(buttonObjects, "touchleave", buttonReleased);

    });
$(document).on('touchstart', function() {
  launchFullscreen(document.documentElement);

});
```

```html
</script>
<link type="text/css" rel="stylesheet" href="ATREUS_mobile/css/atreus-style.css" />
</head>
<body>
  <div id="main_box">
    <div class="remoteControl">
      <div class="cross_area">
        <div id="cross" role="img" aria-label="atreus cross">
          <div id="cross_up"></div>
          <div id="cross_down"></div>
          <div id="cross_left"></div>
          <div id="cross_right"></div>
        </div>
      </div>
      <div class="button_area">
        <div id="buttonSelect">
          <img src="ATREUS_mobile/images/buttonSelect.png" />
        </div>
        <div id="buttonB">
          <img src="ATREUS_mobile/images/buttonB.png" />
        </div>
        <div id="buttonA">
          <img src="ATREUS_mobile/images/buttonA.png" />
        </div>
        <div id="buttonStart">
          <img src="ATREUS_mobile/images/buttonStart.png" />
        </div>
      </div>
    </div>
    <div class="logo_area">
      <div id="logo" role="img" aria-label="atreus">
        <div class="innerLogo"></div>
      </div>
    </div>
  </div>
</body>
</html>
```

Listing A.7: gamepad.html

A.2 Communication protocols

API	Inter-action type*	Action (HTML Event)	JSON Data {interaction type, action, ID, x, y, z, sensorData}
Touch events	BUTTON = 0	DOWN = 1 (touchstart)	{0, 1,currentTargetID, x,y,null,[target-attribute]}
		UP = 0	{0, 0,currentTargetID, x,y,null,[target-attribute]}
		LONGDOWN = 3 (touchend)	{0, 3,currentTargetID, x,y,null,[target-attribute]}
	TOUCH = 1	MOVE = 2 (touchmove)	{1, 2,currentTargetID, endX,endY,null, [multiTouchAmount, windowWidth,windowHeight]}
		SWIPE_RIGHT = 4 SWIPE_LEFT = 5 (touchend)	{1, 4,currentTargetID, endX,endY,null,[xDiff, yDiff]} {1, 5,currentTargetID, endX,endY,null,[xDiff, yDiff]}
		SCROLL_UP = 6 SCROLL_DOWN = 7 (touchend)	{1, 6,currentTargetID, endX,endY,null,[xDiff, yDiff]} {1, 7,currentTargetID, endX,endY,null,[xDiff, yDiff]}
		ZOOM_IN = 8 ZOOM_OUT = 9 (touchmove)	{1, 8,currentTargetID,null,null,null,[scaling,x0, y0,x1,y1]} {1, 9,currentTargetID,null,null,null,[scaling,x0, y0,x1,y1]}
Key-board events	KEY = 2	keyboardPressed	{2, null,null, null,null,null,[keycode,keyCharacter]}
Motion event (orien-tation)	ORIENT = 3	deviceOrientation	{3, null,null, alpha,beta,gamma,null}
Motion event (acceler-ation)	ACC = 4	ACC = 0 (devicemotion)	{4, 0,null, x,y,z,[interval]}
		ACC_GRAVITY = 1 (devicemotion)	{4, 1,null, x,y,z,[interval]}
		ROTATION = 2 (devicemotion)	{4, 2,null, alpha,beta,gamma,[interval]}
Geo-location	GEO = 7	sendGeoLocation	{7, null,null,latitude,longitude,altitude,null}

* protocol parameters 5, 6 reserved for video, audio.

Table A.1: Overview ATREUS JSON interaction protocol

API	**Feedback type**	**Action**	**JSON Data {feedback type, feedback parameter}**
Vibrate	VIBRATION = 0	vibrationFkt	{0, [duration, delay, duration]}
Audio	AUDIO = 1	soundFkt	{1, [AudioSource, volume]}
Text	TEXT = 2	textFkt	{2, "text"}

Table A.2: Overview ATREUS JSON feedback protocol

APPENDIX B

Test results

ATREUS framework tests

HTML files & features	Safari iOS	Android Browser	Google Chrome		Opera Mobile		Firefox mobile	
Browser version: **Platform:** **Device:**	600.1.4 **iOS 8.0.2** iPhone 6	1.6.28 **Android 4.4.4** SM-G850F (Samsung Galaxy Alpha)	38.0.2125.102 **Android 4.4.4** SM-G850F (Samsung Galaxy Alpha)	34.0.1847.114 **Android 4.0.3** GT-I9100 (Samsung Galaxy SII)	24.0.1565.82529 **Android 4.4.4** SM-G850F (Samsung Galaxy Alpha)	20.0.1396.73172 **Android 4.0.3** GT-I9100 (Samsung Galaxy SII)	32.0.3 **Android 4.4.4** SM-G850F (Samsung Galaxy Alpha)	32.0.3 **Android 4.0.3** GT-I9100 (Samsung Galaxy SII)
drivingwheel.html	+	+	+	+	+	+	+	+
Problems with...								
• websockets							~ *	~ *
• deviceorientation								
• touchstart								
• touchend								
• touchmove							x	x
• fullscreen	x					x		
http://www.gamesbasis.com/mario-kart.html								
* different sensor values								
feedbackdemo.html	~	~	+	+	+	+	+	+
Problems with...								
• websockets								
• vibration	x	x						
• sound	x	x	x	x	x			
• text								
FeedbackTest.java								
gamepad.html	~	~	+	+	+	~	~	~
Problems with...								
• websockets								
• touchstart								
• touchend								
• touchenter	x	x	x	x	x	x	x	x
• touchleave	x	x	x	x	x	x	x	x
• fullscreen	x	x				x	x	x

http://www.gamesbasis.com/super-mario-flash.html								
gamepad_btn.html	~	~	+	+	+	+	--	--
Problems with...								
• websockets								
• vibration	x	x						
• sound	x	x	x	x	x	x		
• text								
• touchstart								
• touchend								
• touchmove								
• fullscreen	x	x					x	x
• canvas**							x	x
StartDemo.java								
** API should work on FF without using canvas								
gamepad_dpad.html	+	+	+	+	+	+	--	--
Problems with...								
• websockets								
• touchstart								
• touchend								
• touchmove								
• fullscreen	x	x				x	x	x
• canvas**							x	x
http://www.gamesbasis.com/super-mario-flash.html								
** API should work on FF without using canvas								
gamepad_joystick.html	+	+	+	+	+	+	--	--
Problems with...								
• websockets								
• touchstart								
• touchend								
• touchmove								
• fullscreen	x	x				x	x	x
• canvas**							x	x

http://www.gamesbasis.com/super-mario-flash.html ** API should work on FF without using canvas								
gamepad_tilt.html Problems with... • websockets • touchstart • touchend • deviceorientation • fullscreen • canvas** http://www.gamesbasis.com/mario-kart.html * irrelevant for tilting function ** API should work on FF without using canvas	+ x	+ x	+	+	+	+ x	+ x x*	-- x x
minivideo.html Problems with... • websockets • touchstart • touchend • touchmove • fullscreen • layout * screen flashing	+ x	+ x	+	+	+	+ x	~ x ~ *	~ x ~ *
pointer.html Problems with... • websockets • devicemotion/ rotation/rotationRate ~ ... different calibration values optimized for Chrome/Opera	~ ~	-- x	+	+	+	+	~ ~	~ ~
smartlens.html	--	--	+	+	+	+	--	--

Problems with... • websockets • MediaStreamTrack (getSources) • Media Capture Stream (getUserMedia) • Image Matching • Canvas • touchstart • fullscreen * only front camera ** Matching error	x x x	x* x** x					x x x	x x x
touchpad.html Problems with... • websockets • touchstart • touchend • touchmove (swipe/scroll/zoom) • geolocation • keypress • fullscreen	+ x	~ x x	+ x	~ x	+ x	~ x	+ x	+ x

+ … remote control works completely or main parts works completely
~ … remote control works partly
-- … main remote control functions fail

x … API not supported or does not work
~ … does not work correctly or unoptimized

green … works in > 80% of all modern browsers
orange … works in < 70% of all modern browsers
red … works in <= 50% of all modern browsers

Reference: http://mobilehtml5.org/ (Oct. 2014)

APPENDIX C

User study

Gamepad 1: ____________________

Auf dem Bildschirm werden blaue Symbole verschiedener Größe an zufällig gewählten Positionen erscheinen. Versuchen Sie, den weißen Kreis **so schnell und präzise wie möglich in die Mitte** dieser Symbole zu steuern und mit dem rechten Knopf zu bestätigen. Der Test beginnt nach dem „Klick“ auf das „Start“-Symbol in der Mitte. Sobald Sie das aktuelle Ziel korrekt angeklickt haben, erscheint wieder der „Start“-Button, mit dem Sie den nächsten Test starten können.

1. Training

2. Test

Block 1 ☐   Block 2 ☐

Observation hints:
Zeichen von Engagement, Interesse:

Zeichen der Irritation:

Anzahl Blicke auf Smartphone während Test:

Block 1:

Block 2:

Spezielle Handhabung:

Fragen:

Wie gut konnten Sie mit Hilfe dieses Gamepads die Aufgabe lösen?

Sehr gut	Gut	Mittel	Schlecht	Sehr schlecht
☐	☐	☐	☐	☐

Wie einfach war es, das Gamepad zu verwenden ohne daraufzublicken?

Sehr einfach	Einfach	Mittel	Schwierig	Sehr schwierig
☐	☐	☐	☐	☐

Was ist Ihnen speziell aufgefallen? Warum aufs Gamepad geschaut?

Zusammenfassung Gamepad 1: ____________________

		Mentaler Aufwand		
Sehr niedrig				Sehr hoch
☐	☐	☐	☐	☐
		Physischer Aufwand		
Sehr niedrig				Sehr hoch
☐	☐	☐	☐	☐
		Genauigkeit		
Sehr niedrig				Sehr hoch
☐	☐	☐	☐	☐
		Geschwindigkeit		
Sehr niedrig				Sehr hoch
☐	☐	☐	☐	☐
		Komfort generell		
Sehr niedrig				Sehr hoch
☐	☐	☐	☐	☐
		Schwierig zu benutzen		
Sehr schwierig				Sehr einfach
☐	☐	☐	☐	☐
		Macht Spaß zu benutzen		
Kein Spaß				Sehr viel
☐	☐	☐	☐	☐
		Lernaufwand		
Sehr schwierig				Sehr einfach
☐	☐	☐	☐	☐

Kommentare bzgl. der Technik generell bzw. Verbesserungsvorschläge:

Positiv:

Negativ:

Gamepad 1: ___________________

Pacman

Bei Pacman müssen Sie die gelbe Spielfigur durch ein Labyrinth bewegen, um sämtliche Punkte darin aufzufressen. Dabei muss den bunten “Geistern” ausgewichen werden, um kein Leben zu verlieren. Die blinkenden Punkte machen den Pacman für einige Sekunden unverwundbar.

1. Training

Probieren Sie das Spiel aus, um mit der Steuerung vertraut zu werden.

2. Test

Beim Test ist nun das Ziel, alle Punkte **möglichst schnell und mit so wenigen Leben wie möglich** einzusammeln.

Benötigte Leben: ___________________ Level completion time: __________

Wie gut konnten Sie mit Hilfe dieses Gamepads die Aufgabe lösen?

Sehr gut	Gut	Mittel	Schlecht	Sehr schlecht
☐	☐	☐	☐	☐

Sonstige Kommentare, Zitate:

Beobachtungen:

Super Mario Bros.

Bei diesem Spiel müssen Sie die Figur von links nach rechts durch ein Level steuern ohne Feinde zu berühren oder in ein Loch zu fallen. Die Figur kann dabei nach links und recht bewegt werden und mit Hilfe des rechten Buttons springen.

1. Training

Probieren Sie das Spiel aus, um mit der Steuerung vertraut zu werden.

2. Test

Beim Test ist das Ziel, das erste Level **möglichst schnell und mit so wenigen Leben** wie möglich durchzuspielen. Ich bitte Sie, dass Sie im Spiel keine Abkürzungen verwenden wie z.B. die grünen Rohre, sondern die Figur nur “oberirdisch” bewegen.

Benötigte Leben: ___________________ Task completion time: __________

Wie gut konnten Sie mit Hilfe dieses Gamepads die Aufgabe lösen?

Sehr gut	Gut	Mittel	Schlecht	Sehr schlecht
☐	☐	☐	☐	☐

Sonstige Kommentare, Zitate:

Beobachtungen:

Final Interview

Welches Gamepad würden Sie am ehesten für welches Spiel verwenden? Bitte erstellen Sie eine Reihenfolge:

	Richtungstasten	Steuerkreuz	Joystick	Neigung
Pacman				
Super Mario				
Spiele allgemein				

(Bestgereiht = 4, schlechtgereiht = 1)

Begründung/Kommentare:

Wie beurteilen Sie die Gamepads, die Sie kennengelernt haben? Bitte erstellen Sie eine Reihenfolge:

	Richtungstasten	Steuerkreuz	Joystick	Neigung
Mentaler Aufwand				
Physischer Aufwand				
Genauigkeit				
Geschwindigkeit				
Komfort				
Schwierig zu benutzen				
Macht Spaß zu benutzen				
Lernaufwand				
Allgemeiner Favorit				

(Bestgereiht = 4, schlechtgereiht = 1)

Begründung/Kommentare:

Waren Sie mit einem der Spiele vertraut bzw. haben es bereits oft gespielt?

☐ Pacman ☐ Super Mario Bros.

Waren Sie mit einem der Gamepads auf einem Smartphone vertraut?

☐ Richtungstasten ☐ Steuerkreuz ☐ Joystick ☐ Neigungssteuerung

Die Gamepads, die Sie verwendet haben, waren keine Apps, sondern Web-Anwendungen in einem Browser. Ist Ihnen dabei etwas aufgefallen?

☐ Ja ☐ Nein

Was?

Würden Sie für ein Smartphone-Gamepad lieber eine Web-Anwendung oder eine App verwenden oder spielt die technische Realisierung keine Rolle für Sie?

☐ App ☐ Web-Anwendung ☐ egal

Warum?

APPENDIX D

Focus group questionnaire and transcription

V. Welche mobilen Geräte besitzen und benutzen Sie?

Handy	Smartphone	Tablet	Sonstiges: ___________
❐	❐	❐	❐

VI. Wie oft verwenden Sie ungefähr Ihr Smartphone pro Tag?

weniger als 1h pro Tag	1h - 3h pro Tag	4h - 7h pro Tag	7h - 10h pro Tag	mehr als 10h pro Tag
❐	❐	❐	❐	❐

VII. Für was verwenden Sie Ihr Smartphone am häufigsten? Bitte erstellen Sie eine Reihenfolge:

Kommunikation (Telefonieren, SMS, Messaging etc)	Surfen	Spielen	Navigation	Sonstiges: ___________

(Bestgereiht = 5, schlechtgereiht = 1)

VIII. Wie gut können Sie nach eigenem Ermessen mit Smartphones umgehen?

Sehr gut	Gut	Mittel	Schlecht	Sehr schlecht
❐	❐	❐	❐	❐

IX. Haben Sie Ihr Smartphone jemals als eine Art Fernsteuerung verwendet? Wenn ja, bitte beschreiben Sie kurz für was und in welchem Zusammenhang:

Ja	Nein
❐	❐

____

____

X. Haben Sie Ihr Smartphone jemals mit einem Screen oder ähnlichen Vorrichtungen (inkl. Printplaketen) im öffentlichen Bereich verwendet? Wenn ja, bitte beschreiben Sie kurz die Interaktion:

Ja	Nein
❐	❐

____

____

XI. Ich habe Interesse daran mein Smartphone als Fernsteuerung für Screens im öffentlichen Bereich zu verwenden:

Stimme überhaupt nicht zu	Stimme eher nicht zu	Stimme teilweise zu	Stimme eher zu	Stimme völlig zu
❐	❐	❐	❐	❐

XII. Ich habe Interesse daran mit anderen Leuten an öffentlichen Plätzen an einer Interaktion teilzunehmen und gemeinsam einen Screen mit meinem Smartphone zu steuern:

Stimme überhaupt nicht zu	Stimme eher nicht zu	Stimme teilweise zu	Stimme eher zu	Stimme völlig zu
❐	❐	❐	❐	❐

Focus group Transcription

1.September 2014, 17-19h
gesamte Aufnahmedauer: 87:51 min, Transkription [06:19- 81:37]

***(kursiv** → meine Fragen / Interview / Kommentare zur Thematik)*
(grau → Anmerkung, Ironie)

Interview Phase 1 [06:19 – 42:15]

Videovorstellung (06:19) ATREUS: **https://vimeo.com/69836416**
Eigenständige Bewertungsphase (09:30 – 13:40, ~ 5 Minuten)

Diskussionsstart: (~ab 15:00)

Würde es euch interessieren mit mehreren gemeinsam etwas anzusteuern?
(Bezug auf vorhergesehenes Video)

BE: Nein, wo ist der Sinn?. *(mit mehreren gemeinsam)* Ich weiß nicht, ob so etwas interessant ist. Meist ist das nur ein Gag wie z.B: auf der Landstraße Wien-Mitte, da gab es mal eine telering - Kampagne. Da konnte man mit dem Inder tanzen, indem dich die Kamera erkannt hat und man konnte dann etwas auf der Leinwand machen. Klingt auch immer sehr lustig, aber ich denke, dass haben auch nicht sehr viele Leute gemacht. Mir fehlt der Mehrwert, außer dass es nur ein Gag ist.

BT: Vielleicht nur gelangweilte Studenten beim Schottentor.

A: Nur als Zeitvertreib, wenn der Zug Verspätung hat. Dann mit 50 anderen Zocken, wäre vielleicht eh recht nett.
BA: Das wäre doch das totale Chaos bei 50 Leuten.
A: Naja, es kommt auf das Spiel an. Es geht vielleicht nicht mit allem.
BT: So etwas zahlt sich nur aus, wo man lange Wartezeiten hat. Wenn viele Leute sind, ist es schwer sinnvolle Sachen zu machen. Ich kann mir keine Anwendung dafür vorstellen. Wenn ich mir das so anschaue, wäre es sinnvoller so etwas zu verwenden, um Informationen zu beschaffen: z.B.: in einem Museum: Ausstellungsstück chinesische Vase → Abspielen von einem Video mit dem Handy.

BE: Für Ansteuerung fällt mir eine Shopping – Anwendung ein: 30 Leute voten gemeinsam für ein Produkt, dann gibt es das um 20% billiger – dort macht es Sinn, viele Leute zu motivieren.
E: Mir kommt vor, ich habe so etwas schon gesehen. (Kopenhagen oder so?) Da waren bei Produkten auf einem Plakat in der U-Bahn QR-Codes oben. Diese konnte man mit dem Smartphone einscannen und zu Hause abholen. Das ist voll praktisch nach der Arbeit – so etwas würde ich sofort machen.

Zum Thema Informationen in der U-Bahn: Bei einem Ausfall möchte ich wissen, wie komme ich am schnellsten zum Bahnhof. Dann kann ich selbst etwas ansteuern und rumnavigieren und meinen Weg selbst finden. z.B.: mit der Straßenbahn 33B komme ich am schnellsten hin.

BT(19:48): Brauche ich dafür eigentlich eine Internetverbindung?
(AWT: abhängig von Anwendung und Anbieter, aber in diesem Fall nicht notwendig. Hinweis auf Beispiel im Versuchslabor verwendet lokales Netzwerk ohne Internet)

Evaluierung der Punktevergabe → Shopping = Gewinner, Warum interessiert euch Shopping am meisten? (21:42)

BE (22:09): Für mich ist diese Anwendung jetzt schon wichtig. Ich mache viele Produktvergleiche online z.B.: Geizhals, da vergleiche ich jetzt schon Angebote am Handy. Hier wäre es eine Erweiterung dessen, was ich schon tue. Daher ist das der Use case, der mir am ehesten einfällt.

BT (22:30): Beim Lebensmitteleinkauf: bin ich teilweise jetzt schon von den Angeboten überfordert. Deswegen mache ich zuerst eine Internetrecherche. Vielleicht wäre es aber einfacher sich gleich so mehr Informationen zu beschaffen. In einem Shopping-Center will ich mich besser zu recht finden: wo gibt's was? Wie kann ich mich besser orientieren? Wie kann ich meinen Weg besser optimieren.
Ich möchte das dann schneller und einfacher. Jetzt muss ich mich extrem anstrengen, um das Internet nach neuen Produkten zu durchsuchen.

E (23:16): Auch ich recherchiere viel beim Einkaufen übers Internet. Ich mach viel über den Onlineeinkauf, aber in den Geschäften ist noch nichts implementiert. Vielleicht gibt es mehr in den Elektronikgeschäften, aber bei Lebensmittel gibt es da eher noch nichts.

A (23:24): Bei einem großen Baumarkt: man sucht so ein spezielles Teil (z.B.: OBI Hütteldorf hat viele Stockwerke). Ich würde es verwenden als Navigation im Gebäude, damit man was schneller findet. *(Da fragst einen Typen in einem Stock und der sagt, er sei nicht dafür zuständig... dann gehst in den nächsten Stock, dann ist der Typ vom Bad nicht dafür zuständig...)* Es wäre fein, man gibt die Teile in den Screen ein und der sagt, dann wo man es findet.
E (24:29): Oder man hat beim Angebot wieder so einen QR-Code, den scannt man ein und der sagt, wo man es findet.

Ok, wir nehmen an im Geschäft wäre ein Screen und ihr könntest darauf etwas ansteuern? Was würdet ihr ansteuern? (z.B.: Werkzeuge bewegen?)

BT (24:45): Was würde es mir bringen, wenn da Werkzeuge rumfliegen?
A: Gibt's das nicht schon wo? Hab das wo gesehen, dass man einen Roboter ansteuern kann. Dem sagt man: „bring mir das“ u. er bringt das dann.

Also ihr würdet es dann eher als Infomationsbeschaffung verwenden? (25:00) Und was ist mit der Display-Ansteuerung?

BT: Ja, wenn dann nur für mich alleine zur Information oder wie man dorthin kommt.

BA:(25:29): Vielleicht eher noch als Spiel? *(Bezug Ansteuerung)*
BT: Ja, eher für ein nerviges Kind, dem drückt man dann sein Handy in die Hand...

BA (26:08): Für mich eher noch interessant in so Märkten wie z.B: Media Markt, da gibt's so Spielekonsolen und da macht man das halt dann mit dem Handy.

Museum – Warum für dich am interessantesten? (26:20)

BA: Ich würde das so nutzen, dass man im Open Air Konzert sitzt und den Bildschirm so ansteuern kann, dass man sich Informationen holen kann. Sodass jeder es selbst machen kann – also personalisiert in dem Fall oder im Museum z.B.: dass man sich schnell Informationen holen kann.
Also eher zur Informationsbeschaffung?
BA: Ja.

Und was wäre mit z.B.: so Installationen?

A (27:08) Beim TMW *(Technisches Museum Wien)* gibt es schon so interaktive Sachen, wo man z.B.: ein Radl drehen kann, das könnte man ins Digitale umwandeln. Da gibt es schon so „Infoscreens" – z.B.: man will wissen, wie was geht → das am Handy auswählen und es kommt am Screen. Oder gleich ein Flugzeug ansteuern. Dort würde es viele Anwendungsfälle geben.

BE(28:07): Projektion vorm Museumsquartier und diese ansteuern *(Bezug auf Bild von Idea card, Schneeflocken)* Etwas Steuern oder Voten usw. ist zwar nett, aber ob das ein Mehrwert für die Menschen ist, weiß ich nicht... Oder so Kunstinstallationen, wo sich so ein Ding verändert, je nachdem was die Leute am Handy machen.

E(28:45): Es gibt schon ur viele Museen, wo man manuell was ansteuern kann, ob das jetzt digital einen großen Boost auslöst, weiß ich nicht... *(zweifelt es eher an)*
Es wäre kein Mehrwert für dich mit Smartphone dort *(Bezug Museum)* ***etwas anzusteuern?***
E: Für mich nicht so spannend, ehrlich gesagt.

A(29:29): Für die Kinder hat das sicher einen Mehrwert. Der Generation 30+ ist das egal, ob man das liest oder eine Animation am Handy ansieht.
BT: Ist im Prinzip ja auch eine Art Informationsbeschaffung. Statt Lesen der Tafel, schaut man sich einen Film an.

A(29:50): Ja, wenn dann müsste das schon mehr als nur ein Film sein. Es müsste was Dynamisches sein, was sich wirklich ändert, aufgrund der Interaktion. Weil sonst ist es nur eine Fernbedienung wie bei einem Videorekorder.

BA(30:01): Ich kann mir ehrlich gesagt noch gar nicht vorstellen, wie das Ansteuern mit dem Smartphone einen Mehrwert bringt?
BT: Wenn man z.B.: zu zweit am Bahnsteig steht und auf einer richtigen Leinwand Tetris spielt.

***Generell fehlt euch der Mehrwert?* (30:30)**

BE: Mir fehlt der Mehrwert in der großen Menge. Alleine oder zu zweit kann ich mir schon noch was vorstellen, aber was machen die dann – was Sinn ergibt – zu 50?
BA: Ja, wenn 's ein Spiel ist, was für eines?
A: Spiele machen ja generell keinen Sinn. Der Sinn ist, dass man Spaß hat.
BT: Zu viert geht ja noch, aber wenn dann alle am Bahnsteig stehen und das machen. Aber wenn dann alle dort stehen und spielen wollen...
A: Aber machen das wirklich so viele?
BT: Ich nehme an, wenn sich das durchsetzt.
BE(31:06): Cool wäre so ein Autorennen, und du hast das Handy als Controller. Mit einem Split-Screen würde es aber auch nur zu viert oder zu acht gehen.
E(31:29): Das was zu 50 gehen würde, wäre z.B.: so ein Millionenshow-Spiel. Du hast verschiedene Antwortmöglichkeiten, so wie bei „Quizduell." *(aktuelle Spiele-App zum Lösen von Quizaufgaben)*
Alle: Ja, das könnte gehen.

Thema U-Bahn / Public Transport (32:19)

BA: Gerade in der U-Bahn eben, wo es eh schon so Screens gibt, da könnte man was gemeinsam spielen.
Warum gerade in der U-Bahn?
BE: In der U-Bahn hat man eh Zeit, da ist mir eh fad.
BA: Da schaut sowieso jeder auf sein Handy und macht irgendwas anderes, das kann man vielleicht gleich gemeinsam machen, ist vielleicht lustiger.

E(32:50): Besonders im Ausland, da hab ich nie Internet. Auch wegen Roaming und so, da hab ich immer einen Netzplan auf Papier. Aber wenn ich kein Internet brauche und ich kann so per WLAN mir den runterladen, wäre das ur super eigentlich. Gerade im Ausland wär super.

BE(32:59): Das wäre aber mehr der Case, du ladest dir das auf dein Handy.
E: Ja, genau, das würde ich so verwenden. Die Einheimischen wahrscheinlich nicht so, weil die wissen eh wohin.

BA(33:41): Interaktion kann ich mir hier am besten vorstellen, weil andere Leute warten müssen...
Einwurf aller: … und die haben eh nichts besseres zu tun.

Public places: (33:53:)

BT: In erster Linie vielleicht wieder eher als Informationsbeschaffung. Ich bin irgendwo und möchte Informationen, was ich da in der Umgebung tun könnte.
Im In- oder eher im Ausland ?
BT: In beidem – z.B.: in Floridsdorf. An einem Ort, wo ich mich nicht auskenne, könnte ich vielleicht etwas posten wie „Hilfe kenn mich nicht aus" und alle „liken" das.
A: Oder jemand kommt und hilft dir dann.
BT: Ja, genau!

BA(34:48): Da geht es doch um einen öffentlichen Raum, der draußen ist, oder?
AWT: *Ja – draußen: Parks, Alleen, Plätze usw.*
BA: Ok... also draußen bin ich froh, wenn irgendwie einmal kein Screen da ist. *(Zustimmung von E)*
A: Das ist ja nicht wirklich draußen, sondern dass ist eher in der Stadt draußen.
(→ nicht überall Screens akzeptabel z.B.: nicht im Wald (Natur), daher Annahme für weitere Diskussion: Stephansplatz)

BT(35:50): Eher noch so bei Geschäfter, wenn man vorbeigeht und die haben so einen Screen im Schaufenster.

E(36:06): Habe das wegen dem Bild zuerst eher nur mit Entertainment verbunden, aber ich finde die Idee mit der Navigation doch nicht so schlecht. *(Bezug auf BEs vorherigen Kommentar).* Angenommen ich wäre neu in der Stadt und will mir Sachen anschauen. Und ich bekomme dann die Info, wenn du die Strecke so und so gehst, kannst du dir das und das anschauen. Das wäre wiederum im Tourismusbereich praktisch – die Idee finde ich nicht schlecht.

BA(36:31): Beispiel am Times Square: überall sind Bildschirme, mit denen man interagieren kann – die Idee könnte schon ganz lustig sein.

A (36:55): Es gibt immer so Tourismusinformationen. Angenommen man hat dort so einen Screen und man hat eine Frage, dann könnte man einen Art Skype-Videocall starten. Manche Sachen kannst einfach nicht „googlen" – z.B.: wo gibt's das beste Sushi hier?

BE(37:33): Mir fallen da wieder so Voting-Geschichten oder Angebote ein. z.B.: Sommerkino: der Film mit den meisten Stimmen wird gezeigt. Oder eine Werbefläche in der Öffentlichkeit: man kann sich einen Gutschein aufs Handy runterladen für die Werbung, die gerade gezeigt wird.

BT(38:06): *(Bezug generell Info runterladen)* Auf einer Leinwand, wenn's wo einsehbar ist, ist es ok. Aber wenn das ein Weg ist, wo ich weiter hingehen muss würde ich es lieber gleich auf meinem Smartphone mittragen.
A: Wenn der Weg zu lang ist – dreimal rechts, dreimal links ...– und dann nicht mehr weißt wohin musst.

BE(38:35): Wenn es nur einen Screen gibt ist es schwierig, aber so können 50 Leute gleichzeitig am Smartphone was machen. Es gibt keine Wartezeiten.
BA: Das ist ein Grundpunkt. Weil sonst stehen bei so Screens immer gleich 10 Leute in einer Reihe.
BE: ... da scheiß ich dann auch drauf.

BT: ***(macht angewiderten Gesichtsausdruck und Handbewegung beim Berühren eines Touchscreen als Reaktion auf Hygienebedenken bei so vielen Leuten gleichzeitig)***
A(39:23): *(Bezug BTs Verhalten)* Deswegen hast ja das Smartphone als Fernbedienung, da brauchst den Screen nicht angreifen.
BE: So wie die iPads beim Mecki... die sind „ur-grindig" *(Anspielung auf schmutzige Touchdisplays beim McDonalds)*
A: Aber wahrscheinlich ist das Smartphone eh „grindiger"... *(Ironie)*
BE: Ja, aber das ist wenigstens mein eigener Dreck.

Die Möglichkeit Information mitnehmen zu können – ist euch das bei allen Use cases wichtig, oder gibt es da Ausnahmen? (39:33)

A: Also das mit den Informationen mitnehmen gilt für mich bei allen.
BA: *(Bezug Idea Card Infoscreen im Bus)* So wie da brauche ich das aber nicht unbedingt.
BE: Ja, wenn jeder seinen eigenen Screen hat, brauche ich auch nicht unbedingt mein eigenes Handy z.B.: im Flieger. *(Bezug Bildschirme pro Platz zum Video schauen bei Langstreckenflüge)*
BT: *(Einwurf)…* Außer vielleicht doch als Fernsteuerung, weil da *(Flugzeug)* sind die Fernbedienungen wirklich grausig.
Alle: Ja.
A: Bei der U-Bahn kannst den Screen gar nicht angreifen über den Gleisen.

Mehrheit für „Shopping" (41:31) – Nachfrage: warum nicht für dich?
BA: Ich bin jetzt zwar überzeugt. Aber am Anfang war für mich noch unklar, was man da mit einem interaktiven Display machen soll? Es ist für mich ungewohnt, ich kann mir das hier nicht so vorstellen, *(Bezug Shopping)* bei anderen Sachen ist mir das klarer. Es hängt davon ab, was für Ideen man als erstes kriegt. Was hat für mich einen Mehrwert? Was bringt mir das?

Interview Phase 2 [43:30 – 81:37] – „Shopping“ Kontext

Zusammenfassung beschriebener Hauptaktionen (neben Informationsbeschaffung) am Flipchart: Navigation, Spiele, Aktionen
→ Was würde euch am meisten zu sagen?

BE(44:18): Navigation und Aktionen finde ich beides ziemlich wichtig. „Games“ sehe ich beim Shopping nicht so, weil ich da eher zielgerichtet bin und weiß was ich will.

A (44:35): „Games“ eher beim Media Markt. Sehe ich eher bei einer jüngeren Zielgruppe halt.
BE: Spiele wäre dann der Ersatz für eine X-Box, die du dort rumstehen hast.

BT(44:49): Spiele sind für mich nur interessant, wenn man jemanden mit hat, den ich beschäftigen möchte und mir während dem Einkaufen auf die Nerven geht.

Wären für euch Spiele mit einer Aktion gekoppelt eventuell interessanter?(45:10)

A(45:10): Wenn man lang genug spielt und dann was kriegt...
BT: Aber ich hab ja nicht unendlich viel Zeit im Geschäft
BA: Ja, aber wenn dann was gratis kriegst?
BT: Da geh ich lieber mit Hilfe der Navigation schneller einkaufen und gehe dafür 1h länger arbeiten, was besser entlohnt ist...
BA: ... aber wenn's ein lustiges Spiel ist?
BT: Aber was ich da kriege, da kann ich mir 5 Spiele mehr kaufen.

E(45:47): Beispiel Kleidung einkaufen: Ich habe ein Shirt Größe M, aber brauche es in L und ich möchte wissen: Gibt es das überhaupt hier? Wenn ja – wo hier? Und wenn nein – in welcher Filiale dann? Die Info brauche ich die ganze Zeit...Echt mühselig immer zur Kasse gehen und fragen.
BE: Bsp.: QR-Code am Etikett: scannst es mit dem Smartphone ein, schaust wo es das online gibt – fertig und bestellen.

A(46:50): Gibt es das nicht schon wo? Wo man das Gewand quasi virtuell anprobieren kann?. Ich geh zum Screen und sage ich möchte das anziehen und es wird simuliert am Screen. Im Winter muss ich dann nicht aus meinen 20 Schalen raus. Das wäre für mich eine Interaktion.
BT: Aber kann man das nicht einfach so vor dem Spiegel vorhalten?
A: Es kommt drauf an, das sieht man oft nicht so gut.

Bezug auf Spiele im Geschäft:(48:08)
BA: Ich würde nichts machen, wenn man dafür nichts bekommt.

BT: Oder was Lustiges z.B.: von Screen zu Screen hanteln, wie bei einer Schnitzeljagd als eine Art „Promo-Aktion“.

Annahme: Use case = Navigation → Wo genau möchtet ihr das verwenden? z.B.: im Geschäft / zu einem Geschäft?(49:00)

BT: Eher im Geschäft. Ich weiß, was ich brauche... zum Geschäft finde ich selbst hin.
E: Wenn's in einer anderen Filiale ist, will ich schon wissen wo diese ist.

Annahme: Ihr seid im Geschäft und wollt zu einem Produkt navigieren →
Was würdet ihr euch erwarten? Wie kann man euch dazu motivieren?(49:21)

BE: Was ich erwarten würde... Werbung für irgendwelche Angebote.

Einwurf andere Idee für Shopping von BE:

BE(50:00): Zum Beispiel man ist in einer Kabine bei der Anprobe und es passt nicht: Übers Smartphone den QR-Code einscannen und an den Screen an der Kassa schicken. Die Dame bringt das dann in der richtigen Größe.
Wie würdest du sowas für eine Anwendung zum Steuern sehen?
BE: Ja, bei z.B.: mehreren Screens an der Kassa 1-3 für die Mitarbeiter. Da möchte ich diese bei einem Mitarbeiter ansteuern und ich schicke mit dem Smartphone die Info an den jeweiligen Mitarbeiter-Screen.

BT(51:31): Bei Schildern, wo Angebote draufstehen stehen aber oft nicht alle Angebote. Es gibt oft viel mehr preisgesenkt als draufsteht. Da würde ich gerne eine Produktkategorie am Smartphone auswählen und Angebote suchen z.B.: “Getränke” → „xyz“ ist gerade im Angebot. Das möchte ich dann nicht nur klein am Smartphone anschauen, sondern gleich groß am Schirm. Und vielleicht findet es auch jemand, der gerade vorbeigeht, auch interessant.
Wäre das für euch interessant, wenn BT das macht?
BT: Ja, wenn ich nicht gerade die Damenhygiene-Artikel durchstöbere...
(Ironie, aber Privatsphärebedenken)
BA(53:11): Ich finde, das ist eher praktisch für dich. Ich denke der Haupteffekt von dem ganzen ist, dass man was kleines auf einen großen Bildschirm bringen kann.
BT: Aber ist es nicht auch was für die Verkäufer – die Anregung von Impulseinkäufen der Kunden?
Bsp.: Jemand ist gerade in der Gemüseabteilung und sieht das Angebot, was die Person auch gerade braucht. Sonst wäre mir auch egal, was andere kaufen.
Was wäre für euch der Mehrwert, dass es am öffentlichen Bildschirm ist und jeder mitschauen kann? (im Kontrast zu Angebot kommt gleich direkt auf das Smartphone)
BT: Ich bin total „schaß-augert“ *(schlecht sehend) –* Ich habe zwar schon ein Smartphone mit einem großen Display, aber ich sehe noch immer nichts.

BA(55:59): Es kommt drauf an, ob man manches privat machen will. Wenn ich jetzt z.B.: „Bier" schaue.
BT: Bei manchen Sachen ist es mir egal, bei manchen nicht... Es könnte mir unangenehm sein, wenn da jemand mitschaut. Vielleicht ist es eher aus Verkäufersicht interessant, nicht immer aus Kundensicht.

Zusatzfrage zurück zum Beispiel „Umkleide" → Warum brauchst du hier ein Smartphone, wenn eh ein Display in der Kabine wäre?
BE: Mit dem Smartphone könnte ich einen „Tag" am Kleidungsstück scannen. Es wäre zwar auch eine Kamera am Display möglich und so scannen, aber wahrscheinlich wäre das für das Geschäft einfacher und billiger, weil eh jeder ein Smartphone mit hat.
BT: Vielleicht kann man damit auch persönliche Daten mitnehmen: z.B.: das hat gut oder schlecht gepasst.
BE: Mit dem Smartphone könnte man auch eine „Wishlist" machen: z.B. hab ich das probiert, aber es passt doch nicht ganz, vielleicht kaufe ich es doch wo anders.
BA: Dann kriegst auch gleich personalisierte Werbung.
BE: Genau.

Beispiel Navigation → wollt ihr das dann auf dem Handy oder am Display sehen?

BE(58:01): Am Display in kleinen Geschäften und bei wenig Information z.B.: Produkt ist dort im 3.Regal. Beim großen Shoppingcenter z.B.: SCS, wäre es besser am Handy, anstatt zu merken ich muss dort und da hingehen – also teils, teils.

BA(58:40): Ich denke mir ständig: wie kann man das Produkt einscannen, wenn man das Produkt erst braucht und noch nicht hat?
E: Man hat vorher ein Prospekt mit Code oder ich tippe es selber ein.
BA(59:38): Man hat am Smartphone einen Plan von Produkten. Man wählt eines aus und es blinkt dann am Screen, wo es ist oder der Weg dorthin blinkt auf am Bildschirm. Man sieht wo man ist und wo das Produkt ist.

Fallbeispiel (Skizze Flipchart): Ein Display in der SCS (Shopping City Süd) mit einem Plan und ihr habt euer Smartphone bei der Hand → Wie würdet ihr das angehen? Was braucht ihr, damit es für euch nicht zu umständlich wäre? (60:00)

BT: Beispiel-Annahme: Ich brauche schnell eine Milch, weil der Kaffee ist schon heiß in der Kanne und ich bin in einem Geschäft, wo ich noch nie war... dann möchte ich rasch wissen: wo ist das Kühlregal? Ich würde mein Smartphone nehmen, mich dort ins Netz einbuchen, und z.B.: im Browser die Datenbank durchsuchen mit "Milch". Dann möchte ich eine Auswahl mit welche Produkte gibt es in dieser Kategorie und an welchem Standort. Dann bekomme sehe ich einen Lageplan, wo das Produkt ist.

Wo würdest du dir das Display erwarten?
BT: Beim Eingang.
Was wäre der Vorteil für dich das Produkt direkt am Display beim Eingang zu sehen und nicht gleich am Smartphone? (61:16)
BE: Bei einem Lageplan ist sicher die Größe von Vorteil, weil am Smartphone kann man mal reinzoomen bis man überhaupt sieht welches Geschäft das ist.
BT: Wenn ich auf dem Weg bin - wo gibt es Milch? → z.B.: dann sehe ich mehrere Geschäfte z.B.: Merkur. Dort gehe ich dann hin und sehe dort wieder den nächsten Bildschirm usw.

Würdet ihr das auch so verwenden oder habt ihr andere Vorschläge oder Erwartungen? (62:29)
BA: Dass, wenn ich mit dem Handy hingehe, es sich gleich mit dem Screen verbindet oder ein Menü auf das Handy ladet, sodass man gleich auswählen kann.

Also ihr würdet gern wieder Information am Handy mitnehmen? → Ja
BT: Manche tun sich mit dem Smartphone leichter, weil sie sich mit dem eigenem besser auskennen. Es sollte auch möglich sein nur das Handy zu verwenden, falls wieder mehrere drauf zugreifen wollen.

Würdet ihr es auch gern gemeinsam mit anderen verwenden wollen? (63:23)
BE: Bei einem interaktivem Lageplan müsste dann aber jeder eine eigene Farben haben, damit man weiß welcher Weg ist seiner.
BT: Leute, die Milch suchten, suchten auch Erdbeeren... *(Anspielung Online-Empfehlungen)*
BE: Interessant wäre es dann diese Menschen dann zu verkuppeln quasi.
BT: "Kuppel-App" – Bernd bei den Artischocken suchte auch nach...
(Gelächter, ironische Kommentare → gemeinsame Interaktion wird aber in diesem Fall eher abgelehnt)

E(64:44): Kann man den Plan jetzt eigentlich am Smartphone mitnehmen?
Wäre das für euch wichtig?
BE: Ja, wäre schon praktisch, dass ich weiß wohin ich gehen muss – eine Art Navi wie im Auto.
BT: Eine Art "Supermarkt-GPS"

Würdet ihr mit anderen interagieren wollen, wenn es mit einer Aktion verknüpft wäre? Oder um anderen zu helfen? (66:02):
BT: Produkte bewerten.
BA: Oder wieder so Votings.
A: So Bewertungen wären sicher interessant. z.B.: kauft sich einer eine Induktionsplatte und merkt die ist „ur scheiße", dann wäre es fein, wenn das gleich dort wo auftaucht.
BT: Ist dann aber wieder die Frage, wie das mit Daten sammeln ausschaut? Weil ich will z.B.: meine Daten nicht hergeben.
A: Oder man kann das gleich mit einer Aktion verbinden. z.B.: Man bewertet 10 Produkte und bekommt dafür -10% auf den nächsten Einkauf.

Möchtet ihr diese Bewertungen dann am großen Display abrufen?
BT: Ja, wenn gerade sonst nichts läuft und ihn gerade keiner bedient.
A: Wenn z.B.: jemand ein Video davon gemacht hat wie die Herdplatte gerade einen „schaß“ macht, kann man es sich dort anschauen. Weil das will man sich ja nicht runterladen *(aufs Handy)*, ist ja ur-viel Traffic.
E: Das Problem mit solchen “Bewertungsgeschichten” ist, dass es nach einiger Zeit missbraucht wird. Da mischen sich dann die Firmen ein. Bei Amazon muss man da jetzt auch schon genauer schauen, was jetzt „fake“ ist oder echt ist, oder bezahlt ist.

Fallbeispiel weitere Annahme: Ihr habt schon das Produkt ausgewählt. Wie könnte euch das Display beim Weg zum Produkt weiterhelfen? (69:10)

A: z.B.: einen virtuellen Durchgang.
E: Ja, bei einem großen Shoppingcenter schon, wenn's mehrere Ebenen gibt.
BA: So „Google Maps“ Sachen halt, das ist jeder schon gewohnt, dass man so 2D Karten sieht.
BE: Wobei das sicher schwierig ist, wenn das mehrere Leute sind, die sowas gleichzeitig machen wollen, sieht man das ja nicht. *(Zustimmung von E: Genau!)* z.B.: Schaut sich einer 10 Minuten lang den Weg an und dann stehen da noch 5 andere Leute, die auch was suchen. Da weiß ich dann nicht, ob das so cool ist, wenn man sich das am großen Display anschaut.
E: Deswegen habe ich vorher gefragt, ob das auch am Smartphone geht.

Was für Art von Aktionen würdet ihr euch am Public display vorstellen?(70:46)

BE: Zum Beispiel Grillkohle: ein paar Leute stehen vor dem Display, wenn sich noch 5 andere finden, die die Grillkohle billiger wollen, dann gibt's diese auch billiger. Jeder wählt das aus, dann kommt eine Notification am Smartphone und am Screen sieht man dann die Grillkohle gibt's jetzt tatsächlich billiger.
E: Also bin da wieder skeptisch... vielleicht könnte da ja wieder die Firma dahinter stecken. Aktionen lösen eine Art Kaufrausch aus.
BE: Ja, so wie bei den Reiseportalen. Beim Buchen steht immer :10 Leute schauen sich das auch gerade an.
E(72:12): Bei so öffentlichen Angeboten bin ich eher skeptisch. Mich würde eher bei einem Angebot der Vorrat interessieren: gibt es das dort noch? Und wenn nicht, gibt es das noch wo anders?

Wo seht ihr einen Vorteil darin die Aktionen am Smartphone in Kombination mit einem Public display zu erhalten? (72:42)
E: So wie bei Zeitschriften im Briefkasten – das am Display und mehr Infos dann am Smartphone. Und dann zum Navigieren wieder wo das ist , so wie vorher.

A(73:00): Vielleicht das Produkt in 3D mit den Neigungssensoren anschauen, dass man es auch umdrehen kann.
E: Oder damit man das Siegel nicht aufreißen muss.
BT: So drüber scannen und man sieht das Produkt von innen.

BA: Sowas macht vielleicht eher Sinn, wenn es das Produkt dort nicht gibt. Man möchte das bestellen, schaut sich das vor Ort in 3D an und nimmt es dann.
A: z.B.: ein Möbelstük, was gerade nicht in der Ausstellung ist.
BA: Das kann man dann in 3D mit dem Handy bewegen. *(zeigt Neigung mit eigenem Handy)*

A(74:28): Bsp. Autokauf: Man könnte sich virtuell ins Display „reinsetzen“, ohne dass man das Leder schmutzig macht. Und das Auto mit einem Joystick wie bei der Wii lenken.
BA: Ich kaufe mir das Auto auch, wenn ich das mit dem Handy steuern kann,... *(Ironie)*

Würdet ihr dabei auch mit anderen interagieren wollen?(74:50)
A: Ja, wäre ein cooler Gag für einen Autohändler. Wäre ein guter Use case, wenn man was zu zweit machen könnte, was auch nicht nerven würde. z.B.: ein Rennspiel und man kann damit mehr interagieren, als wie wenn's nur ein Foto ist.
Es ist auch vom Produkt abhängig, man kann das nicht gut mit allen machen z.B.: Bohrmaschine, Milchpackerl von allen Seiten anschauen?? *(angezweifelt)* – macht nur bei bestimmten Produkten Sinn in 3D anzuschauen.

Vorher wart ihr mehrheitlich eher gegen eine gemeinsame Interaktion beim Einkauf, aber warum würde euch das beim Autokauf interessieren?
A: Weil es dort die Möglichkeit gibt, das von Innen anzuschauen, aber wenn jemand sich z.B.: eine Bohrmaschine anschaut, während wer anderes was anderes anschaut, dann gibt es da keine Synergie.
BT: Beim Autokauf kann jeder sein Spielzeug herzeigen. Es gibt einen gegenseitigen Austausch und man kann ins Gespräch kommen.

Was seht ihr für Vorteile, wenn diese Anwendung (im Kontext Shopping) web-basiert ist? (77:15)

A: Weil's für Windows Phone sicher wieder keine App dazu gibt...
BE: Weil man wieder zu viel Speicherplatz braucht.
BE: Braucht man, um das ganze nutzen zu können, Internet? Eine App kann ich nur runterladen, wenn ich Internet habe.
BT: Das ist für mich auch interessant. Am Ende des Monats ist meist mein Datenvolumen verbraucht. Ohne Internet wär schon praktisch.

A: Wie funktioniert die Verbindung eigentlich? Wenn ich keine App drauf habe, die die Umgebung scannt?
E: Man geht einfach vorbei und es geht die Startseite auf und schon ist man eingeloggt, wenn man eh im gleichen Netz ist.
A: Ja, aber wenn man das automatisch macht, nervt das dann nicht?

Wie müsste für euch der Ablauf für den Verbindungsaufbau sein, dass ihr es ohne Probleme ausprobieren würdet?

BT: Es soll nicht automatisch passieren, ich möchte das selbst ansteuern. Aber z.B.: dazu halten oder die URL in den Browser eingeben, wäre ok.
A: Weiß nicht.. das würde mich schon nerven, wenn ich eine URL eingeben müsste.
BT: Es sollte auch schnell gehen.

A:(80:01): Vielleicht sollte man sich auch aussuchen können, ob man einen exklusiven Zugriff darauf haben will. Ich möchte einstellen, ob sich noch ein anderer verbinden darf. Oder einer bekommt für 5 Minuten alleine das Recht und wird dann automatisch „rausgehaut". Damit nicht einer alle anderen blockiert.
BA: Aber es ist sicher schwieriger, wenn man etwas mit „Ranhalten" verwendet, wenn jetzt mehrere sind oder auch nur zu zweit
BT: Ja deswegen, wenn dann eher was mit Reichweite *(zum Verbinden)*.
BA: Oder auch z.B: am Times square, da wäre das *(„Ranhalten")* schwierig bei so einer Entfernung.

Bibliography

[1] M. Baldauf. ATREUS - advanced web technologies for remotely controlling ubiquitous screens. `http://matthiasbaldauf.com/atreus/`, 2014. Accessed: 2014-18-03.

[2] M. Baldauf, P. Fröhlich, F. Adegeye, and S. Suette. Game over? comparing on-screen gamepad designs for smartphone-controlled video games. *ACM Transactions on Multimedia Computing, Communications and Applications (to be published), TOMM*, 2015.

[3] M. Baldauf, P. Fröhlich, J. Buchta, and T. Stürmer. From touchpad to smart lens: A comparative study on smartphone interaction with public displays. *International Journal of Mobile Human Computer Interaction, IJMHCI*, 5(2):1–20, 2013.

[4] M. Baldauf, M. Salo, S. Suette, and P. Fröhlich. The screen is yours – comparing handheld pairing techniques for public displays. In *Ambient Intelligence*, pages 32–47. Springer, 2013.

[5] R. Ballagas, J. Borchers, M. Rohs, and J. Sheridan. The smart phone: a ubiquitous input device. *IEEE Pervasive Computing*, 5(1):70–77, Jan 2006.

[6] R. Ballagas, M. Rohs, and J. G. Sheridan. Sweep and point and shoot: phonecam-based interactions for large public displays. In *CHI'05 extended abstracts on Human factors in computing systems*, CHI EA '05, pages 1200–1203, New York, NY, USA, 2005. ACM, ACM.

[7] R. Ballagas, M. Rohs, J. G. Sheridan, and J. Borchers. BYOD: Bring your own device. In *Proceedings of the Workshop on Ubiquitous Display Environments, Ubicomp*, volume 2004, 2004.

[8] E. Beck, M. Obrist, R. Bernhaupt, and M. Tscheligi. Instant card technique: how and why to apply in user-centered design. In *Proceedings of the Tenth Anniversary Conference on Participatory Design 2008*, pages 162–165. Indiana University, 2008.

[9] S. Boring and D. Baur. Making public displays interactive everywhere. *IEEE Computer Graphics and Applications*, 33(2):28–36, 2013.

[10] V. Braun and V. Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, Jan 2006.

[11] D. O. Buskermolen and J. Terken. Co-constructing stories: a participatory design technique to elicit in-depth user feedback and suggestions about design concepts. In *Proceedings of the 12th Participatory Design Conference: Exploratory Papers, Workshop Descriptions, Industry Cases-Volume 2*, pages 33–36. ACM, 2012.

[12] P. Cairns, J. Li, W. Wang, and A. I. Nordin. The influence of controllers on immersion in mobile games. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 371–380. ACM, 2014.

[13] S. Clinch. Smartphones and pervasive public displays. *IEEE Pervasive Computing*, 12(1):92–95, 2013.

[14] N. Davies, A. Friday, P. Newman, S. Rutlidge, and O. Storz. Using bluetooth device names to support interaction in smart environments. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 151–164. ACM, 2009.

[15] D. Dearman and K. N. Truong. Bluetone: a framework for interacting with public displays using dual-tone multi-frequency through bluetooth. In *Proceedings of the 11th international conference on Ubiquitous computing*, Ubicomp '09, pages 97–100, New York, NY, USA, 2009. ACM, ACM.

[16] U. M. Gewista. Gewista urban media – infoscreen. `http://www.gewista.at/DE/Produkte/Infoscreen/Infoscreen.aspx`, November 2014. Accessed: 2015-24-01.

[17] R. Hardy and E. Rukzio. Touch & interact: touch-based interaction of mobile phones with displays. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, MobileHCI '08, pages 245–254, New York, NY, USA, 2008. ACM, ACM.

[18] R. Hardy, E. Rukzio, M. Wagner, and M. Paolucci. Exploring expressive NFC – based mobile phone interaction with large dynamic displays. In *First International Workshop on Near Field Communication, NFC '09.*, pages 36–41, Feb 2009.

[19] G. Holm. Mcdonald's - interactive pick 'n' play billboard. `http://cargocollective.com/gustavholm/McDonald-s-Pick-n-Play-1`, 2011. Accessed: 2014-19-08.

[20] W. Hu, S. Lian, X. Song, and T. Li. Mobile camera based cross-screen interaction by object matching and tracking. *IEEE Transactions on Consumer Electronics*, 59(3):452–459, August 2013.

[21] N. P. Huy and D. V. Thanh. Developing apps for mobile phones. In *7th International Conference on Computing and Convergence Technology (ICCCT)*, pages 907–912, 2012.

[22] M. Joselli, J. R. da Silva, M. Zamith, E. Clua, M. Pelegrino, E. Mendonca, and E. Soluri. An architecture for game interaction using mobile. In *Games Innovation Conference (IGIC), 2012 IEEE International*, pages 1–5, 2012.

[23] C. Kaazing. About HTML5 websockets. http://www.websocket.org/aboutwebsocket.html, 2013. Accessed: 2015-24-02.

[24] N. Katzakis, M. Hori, K. Kiyokawa, and H. Takemura. Smartphone game controller. In *Proceedings of the 74th HIS SigVR Workshop*, 2011.

[25] N. Kaviani, M. Finke, S. Fels, R. Lea, and H. Wang. What goes where?: designing interactive large public display applications for mobile device interaction. In *Proceedings of the First international Conference on Internet Multimedia Computing and Service*, ICIMCS '09, pages 129–138, New York, NY, USA, 2009. ACM, ACM.

[26] H. Kukka, F. Kruger, V. Kostakos, T. Ojala, and M. Jurmu. Information to go: Exploring in-situ information pick-up "in the wild". In P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler, editors, *Human-Computer Interaction–INTERACT 2011*, volume 6947 of *Lecture Notes in Computer Science*, pages 487–504. Springer Berlin Heidelberg, 2011.

[27] P. Lubbers and F. Greco. HTML5 web sockets: A quantum leap in scalability for the web (kaazing corporation). https://www.websocket.org/quantum.html, 2013. Accessed: 2014-10-02.

[28] P. Luojus, J. Koskela, K. Ollila, S.-M. Mäki, R. Kulpa-Bogossia, T. Heikkinen, and T. Ojala. Wordster: collaborative versus competitive gaming using interactive public displays and mobile phones. In *Proceedings of the 2nd ACM International Symposium on Pervasive Displays*, pages 109–114. ACM, 2013.

[29] S. Malfatti, F. Ferreira dos Santos, and S. Rodrigues dos Santos. Using mobile phones to control desktop multiplayer games. In *Brazilian Symposium on Games and Digital Entertainment (SBGAMES)*, pages 230–238, 2010.

[30] D. C. McCallum and P. Irani. ARC-pad: absolute + relative cursor positioning for large displays with a mobile touchscreen. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 153–156. ACM, 2009.

[31] D. N. Mozilla. Orientation and motion data explained. https://developer.mozilla.org/en-US/docs/Web/Guide/Events/Orientation_and_motion_data_explained, March 2014. Accessed: 2015-24-02.

[32] D. N. Mozilla. HTML5 developer guide. https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5, January 2015. Accessed: 2015-22-02.

[33] Nintendo. Spielstile Wii U. https://www.nintendo.at/Wii-U/System-Eigenschaften/Spielstile/Spielstile-663026.html, 2015. Accessed: 2015-19-02.

[34] A. Olwal, S. Feiner, and S. Heyman. Rubbing and tapping for precise and rapid selection on touch-screen displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 295–304. ACM, 2008.

[35] M. Oshita and H. Ishikawa. Gamepad vs. touchscreen: a comparison of action selection interfaces in computer games. In *Proceedings of the Workshop at SIGGRAPH Asia*, pages 27–31. ACM, 2012.

[36] N. Pears, D. Jackson, and P. Olivier. Smart phone interaction with registered displays. *IEEE Pervasive Computing*, 8(2):14–21, Apr. 2009.

[37] T. Pering, R. Ballagas, and R. Want. Spontaneous marriages of mobile devices and interactive spaces. *Communications of the ACM*, 48(9):53–59, Sept. 2005.

[38] A. Research. Mobile apps beware – rapidly accelerating mobile touch web is perfect example of mobile cloud computing. `https://www.abiresearch.com/blogs/mobile-apps-beware-rapidly-accelerating-mobile-touch-web-is-perfect-example-of-mobile-cloud-computing/`, February 2010. Accessed: 2015-25-01.

[39] A. Research. 2.1 billion HTML5 browsers on mobile devices by 2016 says ABI research. `https://www.abiresearch.com/press/21-billion-html5-browsers-on-mobile-devices-by-201`, July 2011. Accessed: 2015-24-01.

[40] A. Research. 1.4 billion HTML5-capable mobile devices in 2013, but developer uptake requires stronger OS, chipset integration. `https://www.abiresearch.com/press/14-billion-html5-capable-mobile-devices-in-2013-bu`, April 2013. Accessed: 2015-24-01.

[41] A. Research. More than half of mobile devices able to connect to external displays in 2018. `https://www.abiresearch.com/press/more-than-half-of-mobile-devices-able-to-connect-t`, September 2013. Accessed: 2015-24-01.

[42] J. Ruiz, Y. Li, and E. Lank. User-defined motion gestures for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 197–206, New York, NY, USA, 2011. ACM, ACM.

[43] E. Rukzio, K. Leichtenstern, V. Callaghan, P. Holleis, A. Schmidt, and J. Chin. An experimental comparison of physical mobile interaction techniques: Touching, pointing and scanning. In P. Dourish and A. Friday, editors, *UbiComp 2006: Ubiquitous Computing*, volume 4206 of *Lecture Notes in Computer Science*, pages 87–104. Springer, 2006.

[44] D. Sin, E. Lawson, and K. Kannoorpatti. Mobile web apps - the non-programmer's alternative to native applications. In *5th International Conference on Human System Interactions, HSI 2012*, pages 8–15, 2012.

[45] N. Swamy and G. Mahadevan. Event driven architecture using HTML5 web sockets for wireless sensor networks. *White Papers, Planetary Scientific Research Center*, 2011.

[46] T. Vajk, P. Coulton, W. Bamford, and R. Edwards. Using a mobile phone as a "wii-like" controller for playing games on a large public display. *International Journal of Computer Games Technology*, 2008:1–6, 2008.

[47] S. J. Vaughan-Nichols. Will HTML 5 restandardize the web? *Computer*, 43(4):13–15, 2010.

[48] W3C. I've seen the future, it's in my browser. `http://www.w3.org/html/logo/downloads/HTML5_sticker.png`, 2014. Accessed: 2015-19-02.

[49] W3C. Standards for web applications on mobile: current state and roadmap. `http://www.w3.org/2015/01/mobile-web-app-state/`, January 2015. Accessed: 2015-22-02.

[50] V. Wang, F. Salim, and P. Moskovits. *The Definitive Guide to HTML5 WebSocket*. Apress, February 2013.

Acronyms

ARC-Pad Absolute and Relative Cursor - Pad. 19

ATREUS Advanced Web technologies for Remotely Controlling Ubiquitous Screens. 5, 43, 63

CD gain Control-Display gain. 19

CSS3 Cascading Style Sheets Level 3. 24, 28, 38

DTMF Dual-tone multi-frequency. 23

FTW Forschungszentrum Telekommunikation Wien. 5, 40, 41, 43, 63, 82

GUI Graphical User Interface. 17

HTML5 HyperText Markup Language 5. 2, 24, 28–30, 37, 38, 40, 43, 44, 53, 54, 58–61, 81, 82, 87

HTTP HyperText Transfer Protocol. 34, 37

NFC Near Field Communication. 2, 3, 10, 22, 23, 30, 64, 80

QR Quick Response. 3, 11, 12, 25, 44, 53, 76

RFID Radio-frequency identification. 9, 10, 12, 22

W3C Worldwide Web Consortium. 2, 28, 29, 82

WHATWG Web Hypertext Application Technology Working Group. 28