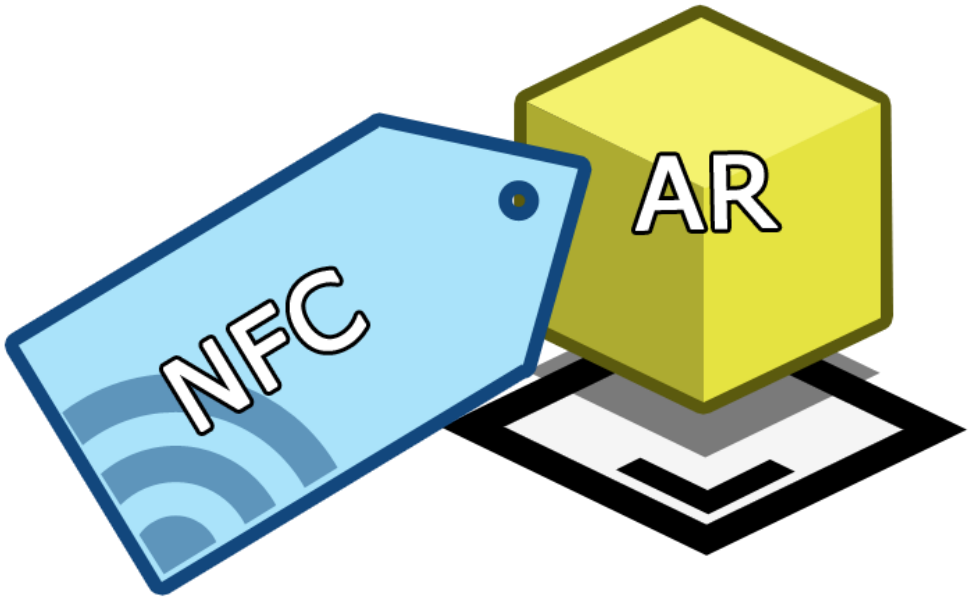
CHALMERS



NFC and AR Interaction in Mobile Gaming

Master of Science Thesis in the Master Degree Programme Interaction Design

ERIK EINEBRANT
TOBIAS JOHANSSON

Department of Applied Information Technology
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2012
Report No. 2012:078
ISSN: 1651-4769

NFC and AR Interaction in Mobile Gaming

ERIK EINEBRANT
TOBIAS JOHANSSON

Examiner: STAFFAN BJÖRK

Chalmers University of Technology
Department of Applied Information Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover:
Graphical icons representing augmented reality and near field communication, technologies that will be further presented in section 2.

Department of Applied Information Technology
Göteborg, Sweden September 2012

Acknowledgements

We would like to thank our supervisors Peter Ljungstrand and Staffan Björk, who let us do our thesis at Interactive Institute, and have helped us a lot during the process. A big THANK YOU to Interactive Institute who have purchased a lot of cool and useful stuff for us to use (mobile phones and Android tablet). Finally, thank you Mr. Gimmick for always being there for us, even in the darkest moments.

Abstract

This thesis explores how augmented reality and near field communication can be used as means of interaction in mobile gaming. In specific three prototype games have been developed and are analysed to find new design patterns which are supported by augmented reality and near field communication. The methods and the process of the development are described and analysed.

Four interaction design patterns were discovered during the course of the project which helped to show that augmented reality and near field communication can be used for new forms of interaction in mobile games.

Keywords: Near field communication, augmented reality, smartphones, games, design patterns, Android, arbject

Table of Contents

1. Introduction

This section introduces the project which this thesis is based on. The background and motive for the project are presented as well as which stakeholders were present during its execution. Furthermore the main question is presented and there is also a brief presentation of the approach to answering it. Preset limitations of the approach are also listed.

1.1. Background

There are strong indications that the mobile game market will grow in the near future (Business Insider, 2011), (The Guardian, 2011). The mobile platforms are also becoming increasingly powerful according to Moore's law (Moore, 1965). This means greater possibilities in use of new and demanding technologies when developing games for a mobile platform. This project aspires to explore the use of such new technologies on mobile platforms. Specifically this project focuses on smartphones as the mobile platform and two technologies, namely Near Field Communication (NFC) and Augmented Reality (AR). The reason is that several modern smartphones already have support for both of these technologies. Furthermore both NFC and AR have gotten noticed, although none of them is used to any great extent.

The Interactive Institute (II) is a research institute based in Sweden. They conduct research in the fields of design and IT and develop concepts, products and services. II has a group in Gothenburg called the Game Studio, focusing on games. The studio joined an EU project in late 2011 called Experimedia (Experimedia, 2012). This is an FP 7 EU project that explores forms of live social interaction to be able to develop a facility for Future Media Internet experiments. One form of social interaction is games, both analog and digital. This is one thing that the Interactive Institute Game Studio is interested in. Experimedia is also performing research on augmented reality, and one of its testbed technologies is an augmented reality platform.

The Future Internet Assembly (FIA) consists of projects that strive towards strengthening European involvement on the future Internet (Future Internet Assembly, 2012). Experimedia is one of these projects, and therefore attends the conventions and exhibitions arranged by FIA.

1.2. Question Formulation

Computer and video games have been around much longer than mobile games. Therefore these platforms have defined a dominant paradigm for how to design a digital game. Mobile games therefore work pretty much the same as computer and video games. Most games require the user's full attention and does not account for what happens outside of the screen. There are also mobile games that use networking, but they rarely consider that people in the nearby area play the same game. For this reason it may be interesting to explore new technologies that can be used in developing these kinds of games leading to the question this project aspires to answer.

How can AR and NFC be used for new forms of mobile games,
which go beyond the dominant paradigm?

1.3. Approach to Problem

The intention of this project is to create a number of game prototypes that together can provide examples of how AR and NFC can enable new types of mobile gaming. User studies will be performed iteratively to investigate how these experiences are perceived by players. Specifically, these games will be applications for mobile phones that support both AR and NFC and the games will be suited for one to four players.

1.4. Limitations

To carry out this project, some limitations have to be considered.

- The project will only focus on mobile games, so the first limitation is that mobile devices will be the only devices used for the games.
- The project will only focus on games using both AR and NFC. This means that only devices that have support for AR and NFC can be used in the project. The devices must also have relatively high-end processing units to be able to render the AR with acceptable frame rate.
- The prototypes of the project will not be full games, but only concepts to try out different ideas. If the games were to be complete much more time would have to be dedicated on such aspects instead of exploration. This would get in the way of finding new ways to interact using AR and NFC.

2. Theory

In this section, relevant topics for the project are explained. Firstly the target technologies of the project, AR and NFC, are introduced. Thereafter the idea of ubiquitous computing will be presented to give the reader insight into the context in which the technologies will be used.

2.1. Augmented Reality

The term "augmented reality" was coined by Boeing researcher Caudell, 1992. The technology uses sensors to determine the orientation of the user's view. A sensor in this case can for instance be a camera or simply an accelerometer. The data gathered from the sensors are analysed by a computer unit. The analysis is then used to determine where AR content such as a 3D model or text is to be projected onto a transparent display. The content is then reposition in relation to the sensors input such that from the user's perspective the content seem to stay in the same physical space when seen through the display.

The display can be a head-mounted display set (Caudell, 1992, Starner, 1995) and allow for AR to be projected where the user looks. The display can also be the display of a mobile device such as a smartphone and then use the camera of the device to show how the AR content is projected into the real world.

There are various techniques of projecting AR onto a real space. Three examples are *location-based AR*, *marker-based AR* and *markerless AR*.

- Location-based AR determines the user's current position and orientation (e.g. via GPS and compass) to show information about the surroundings.
- Marker-based AR makes use of special markers to project information on. The marker can be anything from a simple pattern to a high resolution photography. If this marker is found within a device's camera image the orientation can be determined by comparing the found marker with a pre-stored picture of that marker. Once the orientation is determined content can be projected in relation to this orientation.
- Markerless AR avoids the use of markers by searching for existing patterns using for example optical flow. In Optical flow the motion of objects, surfaces and edges in a scene are analysed in real-time. The relative motion between the camera and the scene can thus be used to project 3D content as if it the content stayed in the same physical position.

2.2. Near Field Communication

In 2004 Nokia, Philips and Sony established the Near Field Communication Forum (Vanderkay, 2004). In 2006 the forum formally defined the architecture for NFC technology thus allowing various standards to be formed (NFC Forum, 2012). NFC is a collection of radio communication standards for mobile devices such as smartphones. The proximity between communicants is the main feature of this technology. It requires the devices to touch each other or be very close together, usually a couple of centimeters. The short range makes it harder for other people to intercept the data transfer. It is simpler than Bluetooth because it does not require that devices pair prior to communication. On the other hand, NFC operates at a slower bit rate than Bluetooth and Wifi, making it unsuitable for transferring larger portions of data.

NFC is based on several existing Radio Frequency Identification (RFID) standards. RFID is a way of transferring data from a chip, often referred to as a tag, to a device with a reader. This is done using radio frequency electromagnetic fields. RFID tags can be powered by the electromagnetic field from the reader and thereby avoid the use of a battery. This means that tags can be used very flexibly. For instance they can be put on a posters, or inside an object. Because NFC is based on RFID standards, NFC readers can also read several types of RFID tags.

2.3. Ubiquitous Computing

The original concept of ubiquitous computing was coined by Mark Weiser in 1988 (PARC, 2010). It can also be called pervasive computing. Weiser thought of the upcoming computing era as the ubiquitous computing era, where each person owns and uses several computers in their everyday life without realising that they are actually using computers. Instead computers would function as extensions of the human mind (Weiser, 1996a).

Ubiquitous computing is a way for humans and computers to interact through everyday objects, rather than just via a regular computer. A user might not even be aware that a computer is being used. Examples of ubiquitous computing can be a refrigerator which monitors its content and can therefrom suggest dishes, order new articles from a store and warn the user of spoiled food.

Following the ideas of ubiquitous computing are those of calm technology. In a ubiquitous computing era it is important that computers can work without demanding the users immidiate attention. Calm technology should provide enhanced peripheral reach while not increase the more direct flow of information (Weiser, 1996b).

3. Previous Work

This section presents products and services which utilize AR and NFC and were available at the start of the project. The chapter aspires to give the reader insight into the technological context which existed prior to project execution. The chosen products does by no means cover all earlier work in the field but give examples of many applications.

3.1. Work in the Field of AR

While AR has been around for about 20 years (Caudell, 1992), and much research has gone into the subject, it is not very common on the commercial market. In earlier applications the user was often required to carry around heavy and uncomfortable equipment or be stationary. As computer systems have become more powerful and smaller later examples of AR can for instance be applications on smartphones.

3.1.1. AR Games

AR Pacman is a game where the player wears a backpack containing a computer which is connected to a special pair of glasses and a camera. The game projects three-dimensional dots on the street in front of the user who is supposed to walk around and pick up these dots. One player plays as Pacman and the other players play as the ghosts, hunting Pacman. To win a game as a ghost, the player must physically touch the player who plays Pacman (AR Pacman, 2004).

Figure 1: AR Pacman

AndAR Pong is a smartphone game in which the player places a marker on a table. This marker then acts as a game board. The player then controls a paddle either by touching the screen of the phone, or using an additional marker which then acts as the paddle (AndAR Pong, 2010).

Kid Icarus: Uprising is a game for for the handheld game console Nintendo 3DS. It features an augmented reality mode where players can use cards with AR markers to battle each other. The idea is that players should collect cards to get as many game characters as possible. A player puts the cards on the table and look at them through the camera of the 3DS. Different characters will then be projected onto the cards (Nintendo, 2012).

Figure 2: Kid Icarus: Uprising

3.1.2. AR Glasses

AR glasses are glasses through which AR can be seen. There are a number of such glasses which are or are getting available to the public. Vuzix is a company that specialises in video and augmented reality glasses. They have made several different models. Another example of AR glasses is Google Glass which is one of Google's new projects. This project aspires to take the functionality of a smartphone and put it in a pair of wearable AR glasses (CNN Money, 2012).

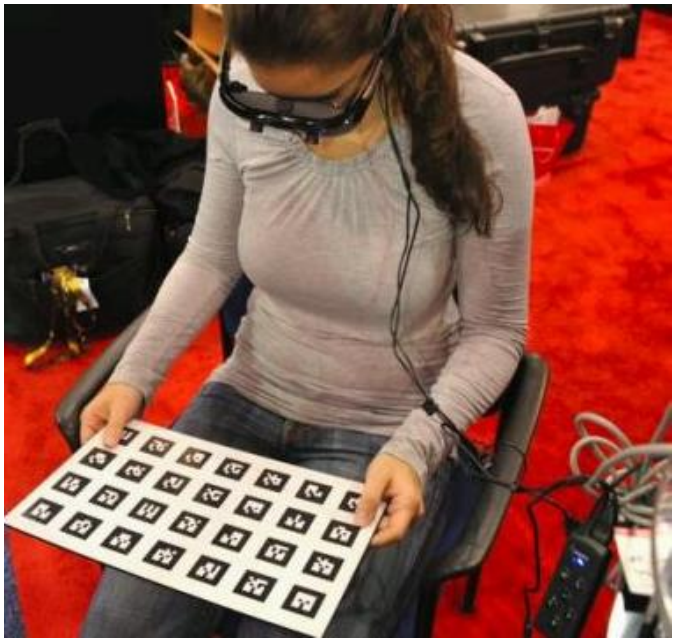


Figure 3: Vuzix Glasses

3.1.3. Smartphone Applications

Layar is a famous augmented reality browser with which the user can interact with real physical objects. A user can also use the platform to create a personal layer. For example a restaurant owner can use Layar to create an image that, when scanned by a visitor, asks for a review of the visit to the restaurant. Layar features an augmented reality interface which can be used to recommend or warn

other users from visiting a specific restaurant or café. Layar can also use location-based AR to show tweets from people who have been at the user's current location before (Layar, 2012).

Google Sky Map is an application for smartphones which allows the user to explore the night sky. It uses location-based AR to show the stars and planets that the phones camera is currently aimed at (Google Sky Map, 2011).

Wikitude is an augmented reality browser that shows the user information about the surroundings when using a camera view. Wikitude uses location-based AR (Wikitude, 2012).

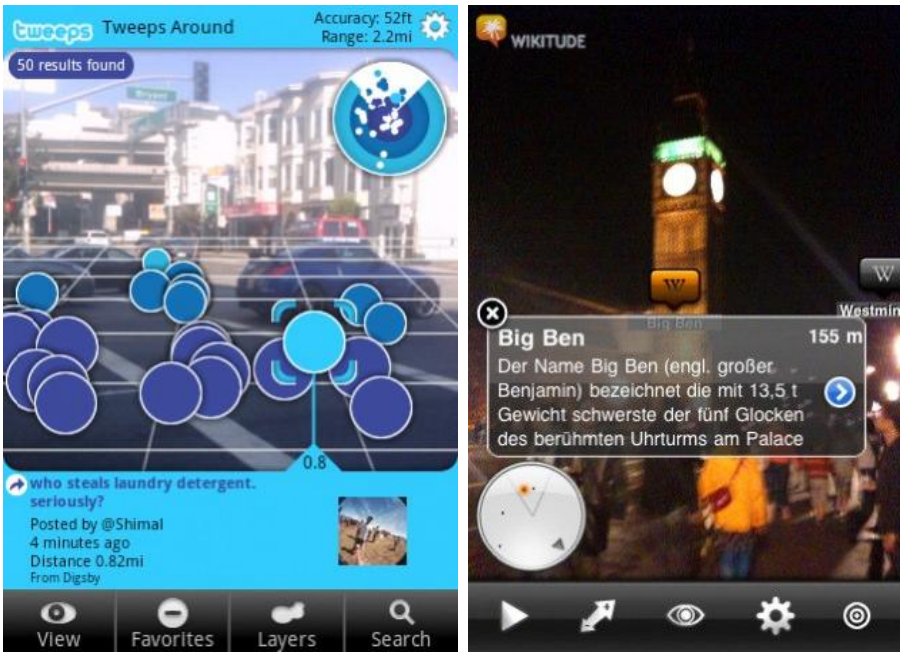


Figure 4: Left: Layar, right: Wikitude

3.2. Work in the Fields of NFC and RFID

In the six years that have passed since the NFC forum formally outlined the architecture for NFC (NFC Forum, 2012) a number of products using NFC have been released. Nokia were fairly early out in using this technology however it is not until recently that other mobile phone companies have started to work with the technology.

3.2.1. Mobile Phones

Nokia were early to implement NFC enabled mobile phones and were also part of the original team which formed the NFC forum (Vanderkay, 2004). The Nokia 6131 NFC, released in 2006, was also the first mobile phone ever with NFC capabilities (NFC World, 2012a). Google entered the market of NFC enabled mobile phones in 2010 with their smartphone Google Nexus S (Hildenbrand, 2010). Since then they have continued to popularise NFC as many other mobile phone developers are using their Android OS and therefore NFC API.

3.2.2. NFC Games

Nokia have developed a few early NFC games. An example of such a game is Nokia Shakespeare Shuffle (Nokia Beta Labs, 2012). In this game players put a number of cards with RFID tags on a surface. An NFC enabled phone is then touched against one of the cards. This causes a part of a famous Shakespeare quote to be played. The cards can then be rearranged and the player must tap the phone against the cards in such an order that a full quote is read out within a certain time limit.

Figure 5: Nokia NFC Game

Nintendo's next console, the Wii U, is to be released in late 2012. The Wii U has a special hand controller that will utilise NFC. It is not revealed as to what it will be used, but presumably players will be able to purchase products from Nintendo's online store using NFC (IDG, 2012).

3.2.3. Payment Systems

Octopus is a system used for paying by smart cards. It is used in Hong Kong and was introduced to the public in 1997 for use in public transit. It was expanded in 2000 for use with other products and services as well. Today Octopus is widely used in Hong Kong for paying in stores and for public transportation. The card uses RFID technology and is compatible with NFC enabled phones.

Google Wallet allow a user to register a MasterCard or a Google prepaid card to a Google Wallet account. The account can then be used for payment in stores. The user touches an NFC enabled phone against a reader to perform the payment (Google Wallet, 2012).

Figure 6: Google Wallet

3.2.4. Access Cards

Several modern apartment buildings uses RFID tags as keys for the front door. The user tap the tag against a reader next to the door. If the user have the right permissions the door will open (Aptus, 2012).

Figure 7: Left: Aptus RFID tag, right: RFID door lock

4. Methodologies and Methods

This section presents the methodologies and methods used within the project. The choice or development of methodologies was not a given focus in the project. While the work could have been carried out using different methods, these alternatives are being mentioned in the next chapter. The chapter also describes relevant design domain patterns as well as specific materials which were relevant to the project.

4.1. Methodologies

The methodologies described below are the ones that were later used in the project. Methodologies used for exploration and implementation as well as evaluation are present.

4.1.1. Participatory Design

Participatory design is a design approach where people, who will be affected by the product, are involved in the design process. These people can be for example employees at the company, end users, designers or customers. Stakeholders are invited in several stages of development to work together with the designers, researchers and developers. The idea of participatory design is to let the users, which will use the product later on, participate in designing the product.

This approach can yield good results when developing a product, as the final user can give input that the designers did not think or know about (Faulkner, 2000). Another advantage is that the users might feel that the product is actually developed for them, and therefore feeling more involved in the process. Hence, it is more likely that the end result of the product is important to them (Preece, 2007). By using participatory design, the users' expectations will also be more realistic as they are part of the development process from an early stage.

4.1.2. Agile Development

An agile approach is characterised by iterative and incremental development. Agile development differentiates from for example the waterfall model in that the agile approach allows for several tests of the product during the project's life cycle. In the waterfall model the product is instead tested at the end of the development.

Incremental development is part of agile development. The word *increment* means *add onto*. In incremental development the project is divided up into smaller pieces, which are then developed independently and at different times. They are then integrated into the main project as they are completed. An alternative method to incremental development is to develop the whole system and have a so-called *big-bang* integration at the end (Cockburn, 2008).

Iterative development is another ingredient in agile development. The word *iterate* means *re-do*. Every now and then the product needs to be revised and redesigned. A project using an iterative development process consists of several iterations, where in the end of each iteration the product is tested and revised. When enough feedback is collected, the product goes back into development to fix the issues that came up during the testing. Not only user interfaces are changed during this process but also system architecture and algorithms (Cockburn, 2008).

4.1.3. Pair Programming

Pair programming is a programming style described in Extreme Programming (Copeland, 2001). The idea is that two programmers work next to each other on a common task. According to Copeland programming on an assignment in pairs, you get higher-quality code, and you need to spend less time testing and debugging.

4.1.4. User Testing

User testing is a collection of methods used to determine a user's interaction with a product. Typically it used to determine how well users do in completing certain tasks and what problems they encounter during this interaction (Cooper, 2007). This methodology requires that the prototype product is so far developed as to allow a coherent experience. This means that usability typically occurs in the later stages of the development. According to Cooper, 2007, it is important to see that the results of user testing is quantifiable, well administrated, can be used to correct design issues and that resources for such corrections are available.

4.2. Methods

The following specific design methods were used when collecting information from the test users. While the specific methods are explained here there will be justification for their use in the following chapter.

4.2.1. Workshops

Workshops are an information gathering technique which focuses on aspects present in a group collaboration rather than a person to person relation as in Interviews (Preece, 2007). Workshops are useful for identifying disagreement within a group as stakeholders may be unaware of their different opinions. Christoph Hienerth et al. proposes that so called lead users are likely to provide potentially profitable new ideas in a study analysis from 2007 (Hienerth, 2007). A lead user is in this case users that may experience need for products that most users will experience first in a future. Identifying such users and inviting them to workshop sessions can potentially result in a good environment for product innovation. The choice of participants for a workshop must generally be considered as participants might fall into the shadow of other more driven participants.

4.2.2. Brainstorming

The primary goal of brainstorming is to rid of as much preconception as possible and allow a more open-minded design. This is typically an applicable technique early in a project once the fundamental questions regarding what the product is and what its purpose is have been answered (Cooper, 2007). The idea generation that is done during a brainstorming session should be unconstrained and avoid criticism even though many of the ideas are likely to be strange. The generated data can then be saved and eventually used later on in the development.

4.2.3. Interviews

Interviews can be used in many variations to generate both quantitative and qualitative feedback from current or potential users. It allows for the developers to understand the effects of the product's behaviour on the users and how the users behave in response. Among many things interviews can be used to gather knowledge regarding tasks and activities (that the product is required to accomplish or doesn't support) as well as the mental model of users. According to Cooper, 2007, users of a product should be the main focus of the design effort.

4.2.4. Questionnaires

A questionnaire is a common way of gathering data regarding users' opinions. The phrasings of questions require thought as it is important that the tester understands the question. The content of a questionnaire is similar to that of a structured interview, but questionnaires can easily be deployed to a large group of testers. This method can often be used in conjunction with other methods such as a user test (Sharp, 2007).

4.3. Design Patterns

Design patterns are an approach to problem solving introduced by Christopher Alexander in his book *A Pattern Language* in 1977 (Alexander, 1977). According to Jan Borchers patterns are solutions that are proven efficient against recurring problems (Borchers, 2000). In other definitions design patterns also include activity patterns which are not necessarily proven but rather observed approaches to

tackle recurring problems (Bayle, 1998). This section further presents three design domains of patterns relevant to the project.

4.3.1. Software Design Patterns

As to define design patterns suitable for software design Erich Gamma et al. wrote the book Design Patterns in 1994. Here design patterns are defined as descriptions of communicating objects and classes. These descriptions are created to solve general design problems in certain contexts (Gamma, 1994). In software design, design patterns allows for an approach which lies on an abstraction level between data structure and full application architecture. Thus one application may contain several software design patterns used to confront various issues within the full architecture.

4.3.2. Interaction Design Patterns

Interaction design patterns refer to patterns regarding design for interaction between a user and a system. They can be used for several purposes such as establishing good standards for interface design as well as allow for comparison between alternatives (Tidwell, 2010). Furthermore a design pattern reminds the designer that the product in question will eventually be used by someone else (Borchers, 2000).

4.3.3. Gameplay Design Patterns

Gameplay design patterns are an established terminology among experienced gamers and game designers to express gameplay patterns found in games. The goal is to make it easier for them to share their knowledge while avoiding misinterpretation. In this report the meaning is heavily based on the definitions and content on the Game design pattern wiki (Björk, 2012a).

4.4. Tools and Materials

Materials, in this session, refer to relevant hardware and software for the project. In specific such materials that were available and used during the project will be presented. Various relevant development tools are also briefly described. The purpose is to give the reader insight into the specific technical context in which the project was carried out.

4.4.1. Smart Devices

A smartphone is sometimes described as a computer-like mobile phone, featuring a computer-like OS. Some smartphones are NFC enabled, meaning that they are equipped with an NFC reader which also making them compatible with several RFID tags. Moreover there exist touch pads which are smart devices on a scale between a smartphone and a laptop computer. During this project two smartphones were considered:

- Google Nexus S running Android.
- Google Galaxy Nexus (Google Galaxy Nexus, 2011) running Android.

The touch pad Eee Pad Transformer Prime was also a target of interest for the project.

4.4.2. RFID tags

RFID tags come in many different shapes and operate on various radio frequencies following various standards. An example of an NFC compatible standard is FeliCa, where the tag is predefined to be read, written or read-only, and the theoretical memory limit per service is one megabyte (NFC Forum, 2012b). The project had access to a large variety of RFID tags whereof several were NFC compatible. Among others there were tags in shapes of stickers, adhesive tape and plastic cards.

4.4.3. AR Markers

For marker-based AR certain markers are used to determine the orientation of AR objects. Such markers can be something between a very simple pixelated symbol to a high resolution picture. The marker is image analysed by a camera in order to determine its orientation. Therefore the marker should be asymmetric, so that there are no two orientations of the marker which looks similar.

4.4.4. Development Environment

Android is an OS which is supported by a large range of smart devices. Android applications are written in Java allowing for object oriented programming and several standard libraries (Sun, 2012). In order to develop Android applications several tools are available for Java.

Google have developed an Android API for Java as to allow development of Android applications. All basic functionalities of devices using Android are made available via this API. It also allows for reading of the various sensors that might be available on an Android device such as the NFC reader (Android Developers, 2012).

In order to program efficiently an Integrated Development Environment (IDE) is generally used. Eclipse is the IDE officially supported by Google for Android and is therefore well equipped for such development (Eclipse, 2012).

A version control system is a tool which allows for several developers to work on the same files stored on an external server. Apache Subversion (abbreviated SVN) revision control system is such a tool and there is a plug-in developed for Eclipse (Apache Subversion, 2012). Google supplies developers with a free service to host code on their servers. The code of this project is hosted at Google Code (Google Code, 2012).

4.4.5. AR Framework

Augmented reality relies on several technical aspects such as image analysis and 3D calculations. The development of an AR framework is therefore a complicated task and unsuitable for a project which aspires not to supply an AR solution but to explore the possibilities of AR. Instead an already made AR solution can be used, such as AndAR. AndAR is an Android framework for augmented reality written in Java. It offers the functionality of the underlying ARToolKit, which is a software library written in C++ for creating augmented reality applications (AndAR, 2010).

4.4.6. Interface Framework

In order to simplify debugging as well as supply flexibility in application interaction some sort of interface solution can be used. The Android API features various solutions as to build standard applications interfaces for Android devices. These solutions however focus on development of tool type applications rather than games. For development of custom game interfaces there instead exist various frameworks such as Cocos2D (cocos2d-android, 2010). Cocos2D is and open source project which allows for flexibility in development as well as interface design.

4.4.7. Client-Server Solution

In order for several users to be able to play simultaneously in one shared game session but at their own devices some sort of communications is needed between their devices. While NFC supports communication between devices that are very close together and AR allows for a shared vision none of these technologies supports efficient and flexible data exchange between devices. A standard solution for such communication is instead to use the Internet. A typical approach is that the applications have a client-server based architecture, meaning that several devices can synchronise with each other via a server. The server can then maintain shared values that are sent to the clients when the values are updated.

5. Planning

This section presents how the project was planned before implementation started. Some early choices regarding methods, materials and tools will be motivated and some alternate approaches will be presented. Finally the initial time plan for the project will be presented.

5.1. Method Planning

The focus of the project was to explore how AR and NFC can be used for interaction in mobile games. Because of the available competence and good access to software and hardware the project would furthermore put emphasis on a practical approach. A practical approach would be more intuitive for the available competence and thus limit the time spent on such tasks as getting accustomed to alternate approaches. It would also yield more practical results which could be used to test the interaction hands-on. These fundamental choices served as a base for the future method choices.

5.1.1. Participatory Design

In order to truly explore the target issue of the project it was of interest to limit the impact of internal opinions on the project's implementation and results. Since mobile devices in the format of phones are common among users it was expected that recruiting potential users for external feedback would be fairly easy. For this reason it was planned that participatory design would be utilised due to the availability of users.

The average smartphone user was not expected to be very familiar with neither AR nor NFC because of the technologies relatively limited propagation. Finding field experts was however not expected a time efficient approach in relation to how long it would take new users to familiarise with the technologies. Therefore the participants would simply be users which could be recruited on a short notice and participate on several occasions during the project.

5.1.2. Agile Development

Because of the projects experimental nature it was important to quickly be able to test new directions in order to determine the direction of the project. Therefore it became relevant to implement according to a light-weight agile method. Several such methods are described and are fitting for various circumstances. No particular method was strictly followed as involved parts formed a very small group which also benefitted from good communication. A better described approach could have been chosen such as Scrum (Schwaber, 2004) or Crystal Clear (Rusk, 2006). Due to the project's small size it was however considered that the project would need no strict format of execution as the time spent on administration would at best be similar to efficiency gained from such an approach.

5.1.3. Workshops

As mentioned previously participatory design with a focus on users would be used to allow for a more flexible direction of the project, therefore it became relevant to choose methods to interact with the participants.

The project included users in the design mainly for two purposes; firstly for the development to be more creative and secondly to test the prototypes. For the creative development participants were to take part in workshops where they would collaboratively work on tasks that reflected the state between iterations. For the test sessions users would be invited to participate in fictional scenarios during which they would test working prototypes.

Instead of a workshop participants could have been involved in less hands-on focus groups. This would allow for a more interview-like format of participatory design (Preece, 2007). While this might have allowed for more efficient data gathering the experimental nature of the project did not truly

provide questions to be asked. Instead it was believed that workshops would allow for more creativity.

5.2. Choices of Tools and Materials

Following the reasoning in section 5.1 tools and materials were not in themselves relevant for the project. Instead they were chosen mainly for their availability, flexibility and most of all by how much time would have to be spent to get started with them. This effectively meant that most tools and materials were chosen as a result of the competence within the project.

5.2.1. Android

Android was chosen as the platform to use since there existed previous knowledge about Android within the project. Another reason was that Google had released two phones with NFC readers, so no external readers would be necessary. If an iPhone would have been used instead, an external NFC reader would have been needed since the iPhone does not include NFC compatibility. In addition to this, one Galaxy Nexus was already available for testing so to get started fast it was determined that Android was the platform to use.

5.2.2. AndAR

Since Android was chosen as the platform an AR framework that can be run on Android was needed. One alternative that was inspected early on was Vuforia (Vuforia, 2012), which is Qualcomm's augmented reality platform. This product seemed to have good end results, but was a little too complex for this project's needs. To build an Android application using Vuforia the first step would be to build the native libraries manually, and then build your Android application upon these. This seemed like an extra unnecessary step that was not needed in the project.

The next solution that was tried was AndAR. It is written in Java, and uses ARToolKit as a foundation. It also uses OpenGL to draw graphics, which is a good graphics engine. AndAR is also completely open source, so if anything needed to be changed it was easily fixed. Finally it was easy to start working with AndAR, so it was decided that this framework should be used.

5.2.3. Cocos2D

Cocos2D was used primarily because prior knowledge existed about that library. The version that was chosen is an Android port, building on the iOS version. The iOS version has a rather big community with extensive information. Cocos2D is also completely open source, so any modifications to it would be easily applied, and it uses OpenGL for drawing graphics so it would be able to interact with AndAR in a good way.

While Cocos2D was believed to supply more features than would be necessary to complete the project an alternative solution would have been to use the standard interfaces of Android. This would have allowed for more standard looking applications as well as portability from one Android device to another. On the other hand portability was not a goal in itself for the project and the standard interface also is not very flexible. Flexibility and a custom look were desired in this case as the prototypes would be games and not necessarily have any standard uses.

5.3. Time plan

The time plan was constructed to reflect the choices of methods and priorities. The first version of it can be seen in figure 8. Early implementation would handle participatory user idea generation workshops and implementation of a technical framework. As the technology was implemented user tests sessions were planned later in the project. The technical work would continue throughout the whole implementation process and the end of the project time would be spent on the report.

ID	Activity	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17	v18	v19	v20	v21	v22	v23	v24	v25	v26
0	Initial Planning																				
1	Workshop 1 - Idea																				
2	Workshop 2 - Concepts																				
3	Workshop 3 - Paper prototyping																				
4	Workshop 4 -																				
5	Test session 1																				
6	Test session 2																				
7	Test session 3																				
8	Test session 4																				
9	Technical base framework																				
10	T.F. finetune																				
11	Planning report																				
12	Report																				

Figure 8: Time Plan

6. Implementation

This section presents the implementation process of the project. The first session explains the pre study carried out before the first iteration. Following the pre study session each of the four iterations of the project will be described.

6.1. Pre Study

In the first phase of the project some time was spent preparing and designing for what the project was going to be about. This section describes the various moments of this pre-study.

6.1.1. Stakeholders

The initial idea for the project was rapidly developed on an early stage in collaboration with the stakeholder II. While II allowed for many forms of exploration within the field of interaction design it was clarified that it would be easier to invest resources in the project should it be in a field related to the work within Experimedia. During an early supervision meeting with II a list of sub-projects currently being conducted within Experimedia were presented. One of these sub-projects involved AR. Since AR was also a personal technique of interest within the team it was decided that some early work should go into investigating the possibilities of this field. The initial idea was to work with AR as a tool for gaming in an everyday context. While AR quickly became a favourite within the team II recommended that the project should find its own niche as AR is a fairly well explored subject. II also recommended NFC as a “hot” technique to investigate into. From this context the initial idea to combine AR and NFC was spawned as to add something extra to the project.

6.1.2. Search for Previous Work

In order to quickly orientate within the field of AR and NFC a brief scan of the Internet for relevant projects was made. Browsing results from Google and YouTube indicated that there existed several projects looking AR and NFC separately (see chapter 3), but rarely the combination. This result led to the belief that the project idea was at least not too exploited by earlier work. According both to II’s preferred way of conducting research and the project members own preference it was decided that the project should be based mainly around design, construction and testing of prototypes. Therefore search on the Internet proceeded to determine the availability of existing engines and frameworks for development of AR and NFC applications.

6.1.3. Available Platforms

In the first supervision meeting it was mentioned that II already had a couple of NFC equipped Nokia phones which could be used for studying this technique. These were however quickly disregarded as the phones hardware was a bit outdated and early results from the investigation of AR indicated that a device would need a strong CPU to support AR. The idea to use phones however stuck as this was a device most testers were likely to have a certain familiarity with. It would potentially also allow for much flexibility since modern phones comes with an array of various sensors and ways of interaction. The natural choice was to develop for Android since the project members had previous experience with this platform and also personally owned NFC compatible Android phones.

6.1.4. Available Augmented Reality Tools

With the choice of platform slimmed down to an Android phone, early investigation went into which Android phones would be suitable for using both AR and NFC. The investigation was conducted such that the Internet was browsed for open source AR frameworks which could quickly be tested on the personal phones already available for the team members. The first framework to be tested was Vuforia (Vuforia, 2012). The framework however seemed complicated to adjust for personal use and so another, AndAR (AndAR, 2010), was tested. This framework was easy to grip and understand how to use; therefore it became the initial choice of the AR solution.

Later on in the project another part within Experimedia shared their investigation into available AR frameworks. In this investigation they had decided upon using a framework called Metaio (Metaio, 2012). When investigating the framework it looked promising. However Targine had already reached such a state as to allow for many AR possibilities using AndAR. While not neglecting Metaio it was decided that an eventual change into this framework would happen first in the future. It was believed that the time used to change the engine could currently be better invested into other aspects of the project.

6.1.5. Available Near Field Communication Tools

The Android API has built-in support for NFC, which makes it possible to create applications for Android phones utilising NFC (where an NFC reader is present). The API makes it possible to both read a tag, and to interact with another phone that has NFC support. Interacting with another phone is called *beaming*. To beam from one phone to another you take the two phones and touch them back to back. There will now appear a special interface on the screens. If you touch the screen while in this mode, you will *beam* to the other phone, meaning sending data. This data can be anything from text to music and video.

II provided the project with a wide variety of RFID tags that had been collected from earlier projects. This collection included a large amount of credit card sized plastic cards. Other types of tags available were small stickers, tags attached onto thin plastic strips and loose circuits. The phones that were used were able to read the credit cards, the stickers and the plastic strips, but not the loose circuits. The credit card type was the type with the most tags, so this type was chosen for the project.

6.1.6. Which Methods to Use

During a second supervision meeting with II some methods to get the project started were recommended. During this session the idea to involve many user workshops evolved. The idea was to involve many testers which were already socially connected to the project members. This would make it easier to recruit participants for the workshops while it would also be easy to estimate the participants' suitability as users as well as involve the same testers over a longer period of time. At this point the project goal was still vague and it was decided that the first workshop should take place soon and be used to generate ideas for the upcoming work.

6.2. First iteration

The first iteration consisted of building the technical foundation for the games, as well as two workshops to try to generate some ideas for games. Most of the time was spent on building the technical foundation.

6.2.1. First Workshop

The goal of this first workshop was to generate ideas. A number of participants were recruited to perform collaborative brainstorm sessions in groups. The participants were recruited via phone over the period of a week. The participants were then scheduled for one of two workshop instances carried out in one afternoon each. Coffee and cookies were offered to encourage participants to partake.

Brainstorming is usually used in order to ignore old ideas and explore potentially new ones. In this case there existed a relatively well defined idea of how the project was going to proceed. Therefore it was decided that the materials should be used during the brainstorming session in order to guide the participants towards the existing idea while still allow them to explore the subject. As material a number of cards were created and three categories of words were generated. Two of the categories were generated using a web tool called Random Word Generator (Plus) (Random Word Generator (Plus), 2007). The first category contained verbs and the second contained prepositions. The third category was created without the use of a word generation tool and instead contained words that were generated during a smaller brainstorm session within the team. During this session the goal

was to generate words that reflected interesting issues regarding mobile devices, privacy, AR and NFC as well as games. The three categories of words were stuck to the cards and formed three decks of cards each representing a word category.

During the workshop, the decks were shuffled and the participants picked one card from each deck. They were then told to discuss what kind of application came to their minds when reading the words. During the sessions the participants' ideas were recorded on audio and written down. Afterwards the results were written in a digital form to simplify analysis. Some selected results can be found in Appendix A.

6.2.2. Targine

An early step in the development process was to create a foundation to stand on when building the games. This work was carried out early in combination with looking for suitable AR and NFC solutions as well as mobile devices. This allowed for orientation in the development options while making implementation progress at the same time. Since it was unclear how much time developing a sufficient prototype would take this would also help estimate what could and could not be done within the project. The technical foundation would later be called Targine from the combination of AR, tag (from NFC tag) and engine (as it would be used as engine for the project's prototypes).

One important thing was to find a decent framework for AR to build on. Some research was made to decide what framework to use for building AR applications, and it seemed that AndAR had a good trade-off between functionality and complexity.

The user interface was created using Cocos2D, a software library for creating mobile games. The merging of the AR part and the user interface resulted in a prototype where you could watch augmented reality in the phone at the same time as you could interact with 2D UI components. For the NFC part, all that was needed was a mobile device with an NFC reader.

6.2.3. Mobile Phones

During the time of the first workshop one type of mobile phone was tested to see if it would fit the needs of the technical framework. It was the Google Galaxy Nexus, which has a rather fast processor and an NFC reader. Another option would have been to use another mobile phone without a built-in NFC reader, and to use an external reader. A third option would have been to wait for Samsung's next mobile phone, the Galaxy S3, but the choice fell on the Google Galaxy Nexus. One Galaxy Nexus was acquired for testing purposes. It seemed to work fine, so two more were ordered.

6.2.4. The Arbject

At this point in the project it was still very unclear how AR and NFC would be combined to test their potential for interaction in mobile gaming. However as Targine started to take shape the marker-based AR solution became a very available possibility. In addition to this II had provided for a wide variety of RFID tags including a large amount of credit card sized plastic cards, as was mentioned in section 6.1.5. As Targine had been enabled both to project AR onto simple markers and read RFID tags the simple combination of sticking an AR marker onto a plastic card RFID tag was tested. This yielded very satisfying results and was decided to work as a base for initial testing. The combination needed a name so that it could be referred to during the development and was named Arbject. Arbject comes from the merge of "AR" and "object" and referred to any object which was enabled both for AR and NFC.

6.2.5. Server Solution

At this early stage of development it was prioritised to allow for many development possibilities as it was yet not decided what would be developed. Therefore it became important to allow for some sort of communication between devices so that multiplayer games would be possible. While NFC is a communication technology it only allows for communication between devices and tags that are

physically close to each other. Therefore Targine should support a client-server based architecture for communication between devices. This would allow for content to be shared via an internet connection. Therefore it became relevant to setup a server. The first version of the server solution used polling to check if a new value had been set for every arbject. Every time polling occurred, a new Android asynchronous task was run to send a request to the server for new values. Every time an asynchronous task is run, it creates a new connection to the server. The client polled the server ten times per second and each of these asynchronous tasks was run in its own thread, so every second ten new threads were spawned. This made the application rather slow in the update phase.

6.2.6. First User Tests

During the first user test sessions the participants tried two simple game concepts utilising AR and NFC. They were then asked a number of questions regarding what they thought about the concepts in the context of using AR and NFC. The participants of the tests were recruited over the span of about a week by calling potentially available users. The participants were divided into pairs and scheduled to take part in 30 minute sessions. A pair would arrive, carry out the testing in two scenarios and then individually answer questions in a semi-structured interview. After the interview a new session with the next pair would begin. These sessions were carried out in one day. Users were offered coffee and cookies during the sessions to encourage participation.

The tests involved several design patterns and in particular tested three. Firstly they were multiplayer games meaning that they were played by more than one person. Such games often add a social aspect, disregarding the participants' level of presence to each other (Björk, 2012b). The idea was that this pattern better utilised AR as two people would have a shared AR vision meaning that the AR artefacts were not personal experience but instead had a position which both players could relate to. Furthermore these games utilised a player-player proximity pattern. This means that the game relied on the participants to be physically close to each other (Björk, 2012c). This pattern was chosen for similar reasons as the multiplayer pattern but also for the NFC touch interaction to be tested in a space where both players could relate to each other's touch actions. This would both help the players to learn from each other as well as potentially give further meaning to their physical actions. The test prototypes were not specifically designed for AR and NFC interaction but instead just simple games which could be played using Targine.

The first simple game prototype utilised the rock, paper, scissors pattern. The pattern means that a choice of strategy where there is always at least one other strategy that can beat that choice (Chelaru, 2007). In this case the prototype was simply an implementation of the game "Rock, Paper, Scissors". In this game a player secretly choose one of rock, paper or scissors and then play these choices against another player's choice. Rock beats scissors, which beats paper, which beats rock. The testers in this session had three RFID cards each, and secretly chose one of rock, paper or scissors for each. They did so by tapping the card to the phone and choosing rock, paper or scissors from a user interface shown on the screen. After that, they counted to three and picked one card each. The AR showed what was on the cards, and therefore who had won.

The second game concept was a scavenger hunt. Six cards were hidden in the rooms where the session was held, and then the two persons got three markings each. They were then supposed to search the place for their three cards, matching the markings on their piece of paper. When they found a card, they should touch it to the mobile phone and change the model that was on the card. The player who then finished first won.

The results from the semi-structured interview questions can be found in Appendix B.

6.3. Second iteration

At this point the idea for the project had been further concretised and a working prototype for the technical foundation had been implemented and tested. The feedback from these tests was used as

base for the next iteration. During this second iteration implementation of the first concrete game prototype began. To generate more ideas for future prototypes a third workshop was held. Finally the first prototype was presented during a visit to the Future Internet Assembly in Aalborg, Denmark, where expert feedback and new perspectives were received.

6.3.1. SkidroidXtreme

SkidroidXtreme was the first real game prototype that was developed during the project. The development begun as II offered the possibility to present part of the project at FIA. In order to fit into the theme of Experimedia it was decided that the game would be designed for one of Experimedia's venues, Schladming Skiing Resort. To fit the theme the prototype would be a winter sports game. Building upon the features already available in Targine, SkidroidXtreme needed a few more game design patterns to stand on its own. Therefore the design included the patterns of Manoeuvring, Player Created Game Elements (PCGE) and Collectibles.

Manoeuvring (Björk, 2011c) is used in real-time games where moving a game object is necessary. It usually requires different skills than moving an object in a turn-based game, as the player has to pay constant attention to what is happening, and take the appropriate action where it is needed.

PCGE is defined such that the player sometimes can create parts of the game by himself/herself. It can for example be a character for a role playing game or a level that you are going to traverse later in the game (Björk, 2011b). In SkidroidXtreme the PCGE would consist of levels built from various collectible slope pieces.

Collectibles are, in this case, artefacts in games that can be collected by the player. They can for example be virtual resources that can be spent in-game, keys which unlock new features or simply pieces of a collection which form a gallery. In SkidroidXtreme the collectibles were different slope parts that could be collected by visiting different geographical areas of the Schladming resort as well as win games against other players. In order to give each slope part its own characteristics and potentially involve various local stakeholders at the resort there would typically be a unique part for each slope and restaurant at the resort. Restaurants and other local stakeholders would be allowed to attach extra data onto the parts, such as fun facts or offers.

In order to develop SkidroidXtreme Targine had to be extended with many new features. Among others the engine had to be able to attach multiple 3D models onto the same arbject and to allow for a real-time update of player avatars.

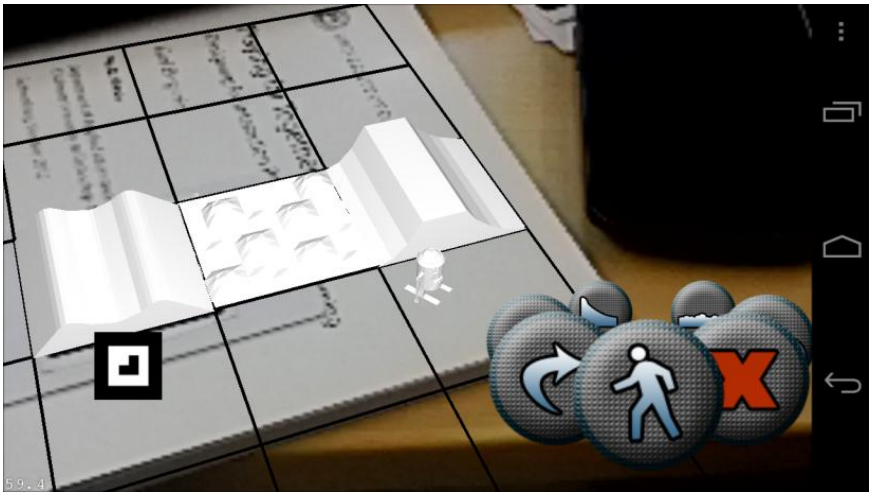


Figure 9: SkidroidXtreme as seen from a smartphone. A non-textured user-made level, consisting of three level parts, and a player's avatar can be seen in 3D.

While developing SkidroidXtreme, two new interaction design patterns were discovered. They were named **Static Arbject** and **Mobile Arbject**. Static arbjects are supposed to be integrated with objects in certain places and should contain meta data with information about that location. Mobile arbjects are arbjects that users are supposed to carry with themselves and play with at any given location. The patterns are further described in section 7.2.1 and 7.2.2.

6.3.2. Second Workshop

The second workshop focused on idea-generation regarding AR and NFC interaction with mobile device games. In this workshop the participants got two game concepts and were asked to explore how they would like to interact with the concepts via smartphones. The goal was to get some user ideas for the future game interaction with NFC and AR. Without external feedback it is otherwise easy to fixate on one or two specific ideas that already exist within the development team.

The participants were recruited via phone over the period of a week and the recruited group included several users which had already participated in the earlier workshop and test session. A total of 12 users were recruited and all users participated in the same session which lasted a little over one hour.

Before the session a short presentation of the project and the workshop was held. During the workshop participants were divided into groups of four or five people and each group was given material to sketch and prototype their ideas. The game concepts were modified versions of the ones used in the first testing session, i.e. a Rock, Paper, Scissors game and a scavenger hunt game. Participants were encouraged to focus on the AR and NFC interaction but could also work on other ways of interaction. To make recruitment easier and to make sure that users were refreshed for the session lighter food was served before the session and coffee and cookies during it.

During the workshop user feedback and discussions were documented on paper. The user-made physical prototypes with descriptions were collected for analysis.

6.3.3. Future Internet Assembly, Aalborg

It was decided that SkidroidXtreme would be showcased at FIA and this lead to many new tasks. For instance it became relevant that the project was explained to other parts of Experimedia prior to the assembly. It also became relevant to develop a graphical style for the game as well as some graphical components for a poster to be displayed at Experimedia's booth. Since other parts of Experimedia were geographically unavailable for each other meetings were done via voice chat conferences. To explain the concept of SkidroidXtreme a small presentation video was compiled and shared via the Internet. A screenshot from this video is shown in Figure 10. The poster content was iteratively developed with another part as well. The preparation work for FIA was carried out simultaneously with the implementation of the prototype itself.

Figure 10: SkidroidXtreme explanation graphics which was used as base for a video tutorial.

At the Future Internet Assembly SkidroidXtreme was officially presented for the first time. Several European Future Internet projects were attending the exhibition. A few people tested the application and gave their opinions. Furthermore there were further discussions regarding the project with other parts of Experimedia.

Figure 11: The Experimedia booth at FIA, SkidroidXtreme promotional graphics can be seen on the top of the poster to the right.

6.4. Third iteration

After FIA, it was clear that something had to be done about the stability of Targine. There were issues especially concerning the server solution. Therefore the next task became to upgrade this part of Targine. Once Targine was upgraded it would be time to develop a new prototype as the project aspired to develop and test AR and NFC in a few games of very different nature. For the new prototype the results from the second workshop would be analysed and potentially used to better understand NFC and AR interaction with mobile devices.

6.4.1. New Server Solution

The first task which had to be done in the third iteration was to improve the server solution. The new server solution used an event handler. When a prototype application received data from the server, it would fire an event which sent the value to an event listener. The event would then handle the

data in an appropriate way depending on the prototype's functionality. This solution allowed for data to be sent and received only when it had to. This also allowed for a single constant connection to the server instead of creating a new connection for every requested update. The use of this approach instead of polling resulted in considerably less network traffic and Targine became much more efficient and robust. The development also became easier, allowing for faster implementation of the prototypes.

6.4.2. Second User Tests

After some internal testing of the new server solution the SkidroidXtreme prototype seemed stable enough for some new user tests. These were carried out as part of a competition during a social gathering that the project members were attending. During this session the guests were given a short introduction to the prototype and then got to challenge each other in a short competitive scenario of SkidroidXtreme. Because of the unstructured nature of the gathering no specific data was recorded but the users still provided useful feedback. Also the context of the test showed how well Targine and SkidroidXtreme performed outside the office.

6.4.3. Game of Stones

As Targine had been confirmed as more stable since the second iteration, work began on the second game prototype. The basic idea was that this prototype was going to have the rock, paper and scissors gameplay pattern which had been investigated in previous iterations. Furthermore, during the first user tests participants had suggested that arbjects could be used as a medium for figure games such as Warhammer (Games Workshop, 2012). These aspects hinted of a strategic game which would therefore be more complex to balance. To avoid spending too much time on game development, since this was not the main focus of the project, it was decided that an existing game could be used as a game base. The new prototype should have features different than those of SkidroidXtreme and thus such a game would be used as base. At this point a card game called A Game of Thrones: The Card Game (AGoT) came into the picture.

AGoT is a complex strategy card game. Not only did it include the sought after features but it awoke the question whether AR could be used to simplify the often crammed information on the game's cards. Cards, in this case, are game assets used to play a game. On the card there can for example be information about a character or a location (Björk, 2011a). AGoT was chosen rather than similar games, such as Magic: The Gathering (Magic: The Gathering, 1993) or Pokémon Trading Card Game (Pokémon Trading Card Game, 1997), for multiple reasons. Firstly it is not as well-known which is potentially useful for testing the prototype as the testers are more likely to experience the game as something new. Secondly there existed local knowledge in how the game worked while the mechanics were still rich and complex as to allow much exploration with AR and NFC.

Figure 12: A typical mid-game setup of A Game of Thrones: Trading Card Game

This new prototype put many new demands on Targine which had so far only been designed as a basis for SkidroidXtreme. Since 3D models are time expensive both to find and create an alternative solution was needed. The models created for SkidroidXtreme could not be reused in this prototype as they came from a very different context and would probably have confused the user. The solution was to implement the possibility to project billboards onto arbjects. Billboards, in this case, are flat surfaces with a texture drawn on them that are projected in a 3D space. Billboards are sometimes used in games to simulate otherwise advanced shapes that could slow down a device should they be implemented as a full 3D model, e.g. leaves or clouds. In this case several billboards were used to represent game cards and various data for game cards such as text and game icons.

Much work went into tweaking the support for billboard graphics as many techniques were tested and each came with both advantages and disadvantages. When a final version had been picked the implementation of game rules began. Making full use of the new server solution system a full game loop could finally be completed. Furthermore some work went into building a system for support of various card types as well as a database of some test cards which featured these types. In order to give the prototype more of a game feeling as well as inspire the creation of cards a medieval fantasy theme was picked.

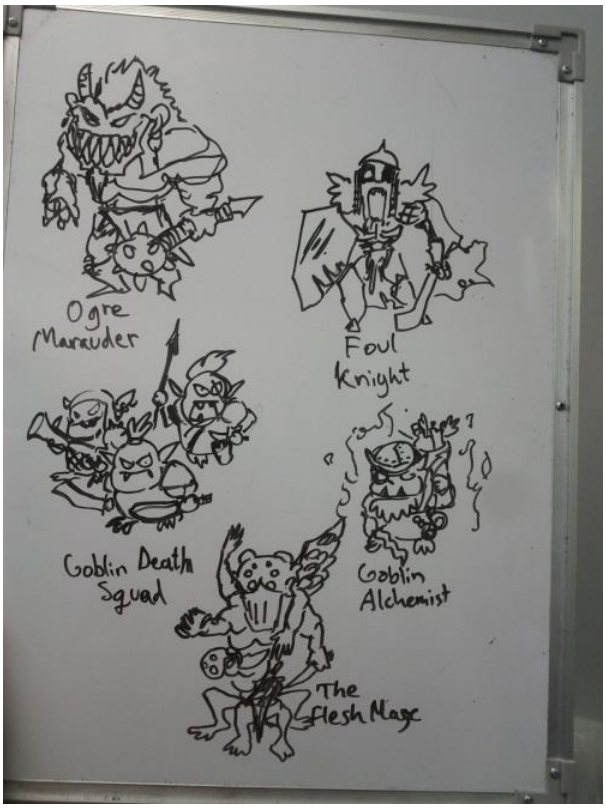


Figure 13: Game of Stones game content concept

While developing Game of Stones a certain aspect of the difference between a digital and physical card game became apparent. While card games often features some hidden information, for instance to make the experience unpredictable or reward user strategies, the digital media makes visibility more flexible. Eventually these observations lead to an interaction design pattern which was named **Individual View**. It describes how a visible game artefact, in this case an arbject which holds a virtual game card, can show AR content differently for two users. The pattern is further described in section 7.2.3.

6.4.4. Experimedia Conference

During the development of Game of Stones II held a conference for Experimedia in which, among other things, various parts got to present their projects. This project was given the opportunity to give a brief presentation by II. Rather than preparing a presentation of the full project it was decided that the presentation would focus on SkidroidXtreme. The reason was that focusing on a certain part would allow for more depth while representatives from Schladming were present, giving SkidroidXtreme extra meaning. The short presentation briefly gave an overview of the project's scope and then quickly explained the concept of SkidroidXtreme. It ended with a short demonstration of the SkidroidXtreme prototype. The project got positive feedback from the Schladming representatives who stated that they would have liked to see this implemented at the resort.

6.5. Fourth Iteration

In the fourth iteration the third and final game prototype was developed. The third and last testing session was also carried through, testing all of the game prototypes.

6.5.1. Scavengers

The third game prototype was named Scavengers. It gives examples of how AR and NFC can be used in a scavenger hunt game, a game type which was also explored during the first user tests and second workshop. At the point where development of Scavengers begun Targine was well developed

and therefore the implementation of the simple prototype took no more than one afternoon. The basic idea would be a treasure hunting game where arbjects are placed in different locations, and each arbject holds a treasure of various types. Players attempt to find and collect treasures which have a type that is valuable for that player. When a treasure is collected the arbject becomes empty.

In order to add extra depth to the game an additional feature was designed. This was to allow player to put traps on arbjects. A player who would then pick up a trapped treasure would gain a disadvantage. In order to make use of AR these traps were designed so that they could be detected if the user investigated the treasure thoroughly. This design was inspired from how a detective might investigate some artefact to find evidence in the details. The design would eventually be identified as an interaction pattern and be called **Detective Data**. The pattern is further described in section 7.2.4.

6.5.2. Third User Tests

In the third, and final, user tests several participants were recruited to test the three prototypes and then give feedback in the format of a questionnaire. The test results varied from day to day, because of instabilities in the applications. Some of the problems that occurred were because of the wireless internet connection.

Participants were recruited over phone and many of the users who had previously participated in the project were invited together with some new users. Roughly one and a half week was spent recruiting and scheduling participants for the tests. Participants were put in pairs, as in the first user tests. The sessions were divided into an introduction, three scenarios were the prototypes were tested and finally a section where they users filled in a questionnaire.

In the introduction the agenda for the session was described briefly. Following this explanation the users were introduced to AR, NFC and the arbject. The users would then train on the interaction with AR and NFC as they would need to use it during the prototype scenarios.

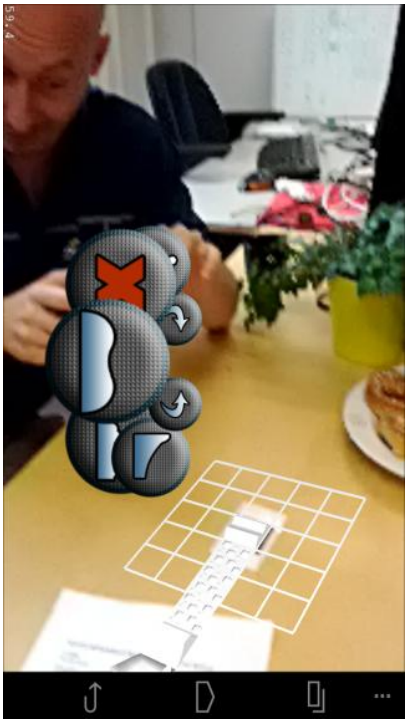


Figure 14: User tests of SkidroidXtreme

In the first scenario the users would interact with the SkidroidXtreme prototype. The participants were told to imagine that they were in Schladming, Austria. They got one mobile arbject each, and

were told that these were their ski passes that they always carried with them. In the next step the users were to build one skiing slope each. When the participants were finished building their slopes, it was decided upon which they would compete. They then added one skier each to that arbject, and started racing against each other. In this specific version of SkidroidXtreme there was no implemented goal, so after some time the race was manually interrupted. The person who won the race was then told that a slope piece unique to the other person had been collected. Finally, the participants were instructed to take the lift up to the top of a specific ski slope. At the top of this slope there was static arbject attached somewhere. It could for example be a monument, a restaurant or a cabin. This static arbject contained a ready-made ski slope that they would race on. The participants were told to add one skier each to the static arbject and start racing. The one that won (again, the race would be interrupted manually) was told that the prize was a specific slope piece that was only accessible through this static arbject.

The second scenario centred on the Game of Stones prototype. Because the game had not been thoroughly tested, the participants were only given limited freedom of their actions during the scenario. The first step in this scenario was to explain all the rules to the participants. Since the game is rather complicated, and contains quite a few rules, this scenario was more of a walkthrough of the game than a real play through. After the rules had been explained, the participants got to play one or two rounds of the game, depending on how stable the game was and how long time it took.

Next up was the Scavengers scenario. It was the last scenario of the session. The two players were instructed how to play the game. Since this game was the simplest and apparently most intuitive in the test, the instruction part did not take very long. When the players had gotten the instructions a number of arbjects were placed in different places in a room. The players then walked around the room looking for the arbjects. When they found an arbject with a treasure they would either collect that treasure or place a trap on it. The scenario took about two minutes and the player with the most points would then be the winner.

After all the scenarios, the session was ended with the participants filling in a questionnaire. The questionnaire was divided into five sections:

- Questions regarding the participant
- Opinions about the SkidroidXtreme scenario
- Opinions about the Game of Stones scenario
- Opinions about the Scavengers scenario
- Opinions about the questionnaire

7. Results

In this section results of the project are presented. The results have been divided into various categories. Firstly there is the arbject which has been the main concept for the projects combination of NFC and AR. Four interaction design patterns which were identified in the prototypes of the project will be presented. The technical foundation Targine on which the prototypes are based will be presented. Finally the three game prototypes will be presented.

7.1. Arbject

An arbject is the combination of an AR marker and an NFC or RFID tag. In this project, the AR marker was printed out and attached to an RFID tag in the size of a credit card. The paper used for the AR marker was ordinary printer paper. Once developed, the arbject could then be used in the games, both as a game board and as an interactive object, such as a playing card.

Figure 15: An arbject

7.2. Interaction Design Patterns

Four patterns were discovered during the course of the project while developing the prototypes. This section describes these patterns.

7.2.1. Static Arbjects

Static arbjects are arbjects that are placed in a location and never moved. They can for example contain information about that specific location, or a game board. In the SkidroidXtreme scenario, the static arbjects contained ready-made skiing slopes that participants could race on. They also contained a specific slope piece that contained information about the location in the form of meta data. Static arbjects also appears in Scavengers where each treasure arbject is static, forcing users to search for them and then investigate them on location.

7.2.2. Mobile Arbjects

Mobile arbjects are arbjects that a user can carry around. These arbjects might contain personal game information, such as personal skiing slope in SkidroidXtreme or cards in Game of Stones. When a user wants to play something with someone else, the arbject can just be put down on a suitable surface and allow for all participants to interact with the arbject. They might be used as portable game boards or containers of tokens in a game.

7.2.3. Individual View
The individual view pattern is implemented by games that will show both private and public information to the players. This information can be for instance certain aspects or states of cards in a card game like Game of Stones. The central idea is that while looking at the same virtual object, two players may see some aspects of this object differently and others in the same way.

7.2.4. Detective Data
This pattern promotes further investigation of an object as a detective would carefully look at an artefact in search of a trace of evidence. The idea is to use the feature of AR to be able to watch an interface of virtual objects from various angles in an intuitive way. This is perhaps a more obvious pattern to be used in games, as most interfaces are not designed to take more than necessary of the user's attention and time. However, should it be important that the user explore an interface in order to further understand it, this pattern can be used. The pattern is supported directly by AR.

7.3. Targine
This is an explanation of the class structure and hierarchy for the applications common engine, Targine. Targine uses a client-server based architecture and builds upon several other frameworks which have been further presented in section 4.4. Specific names in this section are written in italic and represent classes of Targine's architecture. The class diagram can be found in Appendix D.

7.3.1. Client Side
The client side contains most of the code. It includes the main game logic and the drawing and updating of all the graphics.

The *ARToolKit* class offers all the AR functionality from *ARToolKit*. It is a Java wrapper that communicates with the native libraries.

AndARActivity is AndAR's base class. It also holds a reference to *ARToolKit*. *ARManager* is the class that puts everything together. It extends this class to get access to all the AR functionality and creates the *GameLayer* (which handles the GUI). Furthermore it is the base class for *GameCore* (which holds the game logic for each game). It also handles all communication with the server. *ARManager* also declares a number of abstract methods that a potential sub class must implement. These include the handlers for NFC events, touch events and accelerometer events.

GameLayer is the class that handles the GUI. It holds all the interface components and handles the appropriate logic for those components. It uses the software library Cocos2D for creating the GUI.

Every game implements its own *GameCore* class, which in turn extends *ARManager*. This class contains all the logic for the main game loop. By extending *ARManager*, you get access to all underlying technology such as the AR and the server connection and can then build on this. By extending *ARManager*, *GameCore* is also obliged to implement the abstract methods declared in *ARManager*.

Arbject is a base class for the arbjects. Every game implements its own subclass of *Arbject* which is its specific arbject. The *Arbject* class is responsible for drawing and updating the models that are projected onto the arbject in question. Every game uses arbjects in a different way, therefore it is more natural to have a general base class called *Arbject* and let the games extend that class and make smaller more specific tweaks to it.

ServerConnection is the class that handles all the communications with the server. The applications are client-server based, which means that all data is stored on and synced with a server. No data is actually stored on the RFID cards, but the cards' IDs are instead used for setting the correct data for a card through the server. *ServerConnection* uses an event handler. When the application receives

something from the server, it fires an event that sends the value to an event listener. By doing this you don't have to use polling, and only send or receive data when you need to. You also have a constant connection to the server instead of creating new ones all the time.

7.3.2. Server side

The server side of the software architecture only consists of one class. *ARNFCServer* listens for incoming connections from clients, and handles all the incoming messages. One of its member variables is a hash table with the arbject IDs as keys and a string containing values as values. The values typically say what model to show, what text to show etc.

The server holds a list of all the connected clients. When a value is changed in the table, the server contacts all the clients in the list and sends them the new value. Using this approach, we send as many messages over the network as we have to, making the application more efficient.

7.4. Prototypes

Following is a presentation of all the game prototypes which were created during the project. There are figures showing sample scenarios in the games to make it easier for the reader to understand them.

7.4.1. SkidroidXtreme

SkidroidXtreme is a multiplayer 3D racing game with a winter theme in which players can collect level parts and build their own levels. The prototype does not implement the feature to collect level parts but instead allows for levels to be built from a few different parts, simulating that these have been collected by the player. The level parts would also represent different areas of the Schladming skiing resort, this it now implemented neither. The included parts however do hold their own visual characteristics.

The game is played such that the user will touch an arbject, symbolising a skiing pass or some sort of static arbject on the resort, to establish a connection to that arbject. In the prototype there is no difference between one arbject and the other. Conceptually the skiing passes should allow for users to build their own levels on them while the static arbjects should offer specific challenges for the players to complete. Once connection has been established with an arbject an interface is displayed on the device screen. The interface allows the player to put level parts on that arbject.

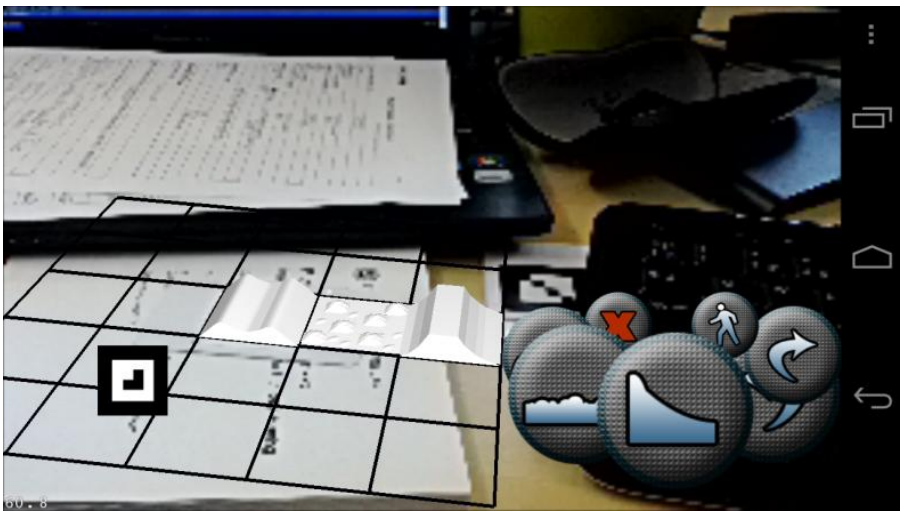


Figure 16: The user interface and a sample level in SkidroidXtreme.

In figure 16 the interface of SkidroidXtreme can be seen. The black grid in the middle is displayed around the connected arbject. The marker of the arbject can be seen to the bottom left, showing the

user which arbject is currently connected. To the bottom right an interface for placing level parts can be seen. On the grid three white level parts of different types have been placed. To place a level part, the user touches a button, representing a level part type, in the interface to the bottom right. The level part will be placed automatically in the next available grid cell. If the player chooses a right or left turn the direction of the track will change accordingly.

Figure 17: SkidroidXtreme icons indicating various level parts and actions

Figure 17 shows all the different options a user can choose from in the interface. The first five are different level part types. The first three are various slope types. The fourth and fifth are left and right turns. The sixth option adds a skier avatar for the player on the level as can be seen in figure 18. The red X is used for removing everything from the scene of the arbject.

The skier in figure 18 is controlled by the player by tilting the device like a steering wheel. If two players are connected to the same arbject, they can add one player each and race each other. Features supporting an actual race such as a start and finishing line are not implemented in the prototype. However the AR allows for players to easily put real physical objects on the level and decide between them where to start and finish.

Figure 18: A skier avatar have been added a level consisting of three level parts.

7.4.2. Game of Stones

Game of Stones is a multiplayer card strategy game, based on A Game of Thrones: The Card Game. It is the most complicated prototype as it contains the most rules and game components. The basic idea is that users gather virtual money, spend it to activate location, attachment and unit cards and then use these activated cards to challenge the other players. A successful challenge will result in the player gaining victory points and the player who first gets a certain amount of victory points wins.

The game is split up into several phases. During each phase all players perform certain tasks either simultaneously or in a certain order. The first phase is a setup phase in which players touch each other's devices together, so that the game is aware of all participants, and chooses an arbject each as their resource card. A resource card is owned by a player and shows that player's amount of money and victory points. The next phase is the plot phase and it is played simultaneously by the players.

During the phase each players pick one plot card from their virtual plot deck determining how much money the player will get that turn and how high the player's initiative will be. The plot phase interface and general model for card interface can be seen in figure 19. Once all players have chosen a plot card the marshalling phase starts. This phase is played by one player at a time while the others wait for their turn. The player with the highest initiative value from the plot phase begins. The current player will draw two cards from a virtual card deck onto a virtual hand. These cards can be of the types units, attachments and locations and events. The event cards were not implemented for use in the prototype. In order to activate a card so that it can be used in the following phase the player must pay a certain cost given by that card with that player's money. Once a player cannot afford to activate any more cards from the hand or simply doesn't want to the player passes the turn to the next player.

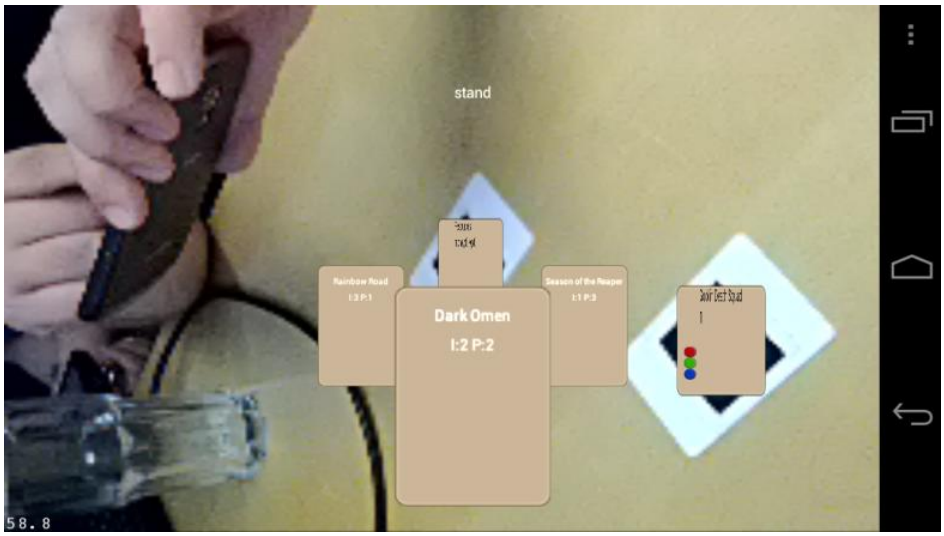


Figure 14: The plot phase of Game of Stones. Three plot cards can be seen with white text on them in the centre of the figure. Behind the plot cards are two arbjects with a card on each of them. The centre one holds a resource card and the one to the right holds a unit card.

Once all players are finished with their marshalling phases the challenges phase begins. During the challenge phase each player can challenge the other players. The phase is player similarly to the marshalling phase in that the player with the highest initiative value starts and then passes the turn to the next. There are three types of challenges; red, green and blue challenge. A player can chose to perform one of each of these challenges during the player's turn. After a challenge type has been picked the player chooses which of the player's activated unit cards which are to participate in that challenge. Unit cards can only be used in the challenge types which are indicated on the card and they have a certain power value. When a user touches a unit card that card is kneeled, meaning that it cannot participate in any other challenge this round, and its power is added to an attack value. Once the current player has chosen its unit cards the challenge is passed to another player by touching that player's device. That player must now defend against the challenge and this is done in a similar way as the current player attacked. In order to successfully defend the defender must have a defence value which is higher than the attacking units summed attack value. A successful defence yields no further results while a successful attack have an effect depending on the challenge type. A red challenge means that the losing player must remove one of his/her activated unit cards. A green loss causes the defender to drop a random card from his/her hand. A blue challenge, should the defender have at least one victory points, causes the winner to steal one victory point from the defeated player. Furthermore, if the defender has a defence value of zero, typically because no defending unit cards were chosen, the challenger gains one extra victory point. When all players are finished with their challenge phases all kneeled cards are stood up, all money is removed and a new plot phase begins.

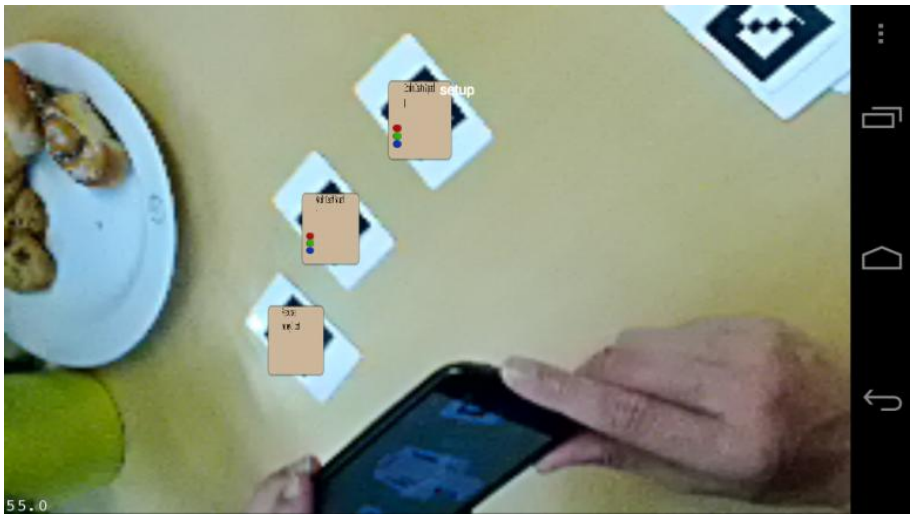


Figure 15: A user looking at his resource card and two unit cards in Game of Stones

Arbjects in this prototype are either empty or contains a virtual card. Most cards may at some point be removed and replaced, i.e. an arbject is emptied during a lost red challenge and then activated with a new card during the following marshalling phase. The resource card, chosen during the setup phase, however works as an AR interface as to allow all players to view certain game properties as in a physical board game. In the prototype all card data on all arbjects was pubic so that all users could see them except for the resource card. Due to a miss in implementation resource data is only updated for the owner of a resource card showing that data.

7.4.3. Scavengers

The final game prototype, Scavengers, is also the smallest and simplest game. It is a multiplayer treasure hunting game, where each arbject contains a treasure. The game features three different treasure types and each participating player gains an individual amount of points for gathering certain treasure types.

Figure 16: Six treasure arbjects in Scavengers. The three treasure types are shown with two of each type.

For each player there is one type of treasure which gives the player two points, one that gives one point and one which subtracts one point from the player. Players are not aware of what treasure types other players are looking for and avoiding. Figure 21 shows the main game interface of Scavengers. To the top left is an icon which shows what type of treasure is good for the player (worth two points). To the bottom left is an icon that shows what treasure type is bad for the player (worth minus one point). Treasure types that are not represented by these icons are worth one point. To the right of the top left icon a number which shows how many points the player currently have, zero in this case. To the bottom right is the button which is used to lay out traps. To the left of that button is a number showing how many traps the player has left.

If a player finds a treasure of the type which is non-beneficial for that player a trap can be placed on that treasure. If a trapped treasure is picked up the points of the original treasure are ignored and the player instead loses two points. It is possible for a player to discover a trap by closely observing a trapped treasure as the trap is represented by a small dot on the treasure.

Figure 17: Trapped treasure, the trap is indicated by a small dot near the centre of the treasure

8. Discussion

In this section the methods and results which were used and obtained during the project will be analysed. At the end of the section some suggestions for future work that can be done to improve the results will be presented.

8.1. Methods

The methods used in the project are fairly well tested, and proven to be good ways to carry out a project. The use of them in this project sometimes gave varying results.

8.1.1. Participatory design

While the involvement of potential end users gave ideas and good feedback on the project it did not provide as much information as had been expected. The idea was to give the project much space to spread out into new interesting branches, a task that in the end was not suitable for the testers. This method was mostly used during the second workshop. The amount of design freedom that was given the users seemed to confuse them rather than allow them think freely. Thus there were few results that had not already been considered. Instead the users could have been given a more concrete scenario and instead of giving them the task to design on their own they could have been modifying a basic design.

8.1.2. Agile development

The agile development process proved to be a good approach as it allowed for great flexibility while allowing for a version of the prototypes to be partially tested when opportunity was given. By using an agile process with several iterations new functionality could rather easily be added in each iteration. It also proved to be good for dividing the project into several parts, with the maximum number of one prototype per iteration. By doing this the development of one game did not interfere too much with the development of another.

8.1.3. User Testing

During all the testing sessions the tests were made in such a manner that they would be quantifiable. This followed Cooper's ideas of how to approach user testing. The test data can be seen in Appendix B and C and the results were used as means to find patterns in uses' opinions. However, while the tests data was collected in a well administrated manner, there was not as much attention given to the recruitment of the testers. Instead testers were recruited after how available and easily accessed they were. While most testers would still fit into the broad scope of this project the amount of testers needed to be able to draw general conclusions of the user group would have had to be much bigger. The gathered data however still sufficed as to give a good view on users' general opinions regarding the project, which was helpful for deciding the next task of implementation.

8.2. Arbject

The arbject concept seems to work rather well. It is intuitive to use and practical if you want to take it with you. Results from the workshops showed that the users understood almost immediately how to use and interact with the arbjects.

One drawback with the RFID cards was that it sometimes was hard to get the phone to read the tag. The users occasionally had to pick up the card from the table or touch the card with the phone in a different angle to get it to react.

One drawback with the AR marker is the risk of glares. When the light is reflected onto the marker it is hard for the application to recognise the pattern. This leads to that nothing is shown on the arbject and that you have to either change your position, or that you have to find a different space with less direct light. Playing outside may be difficult for this reason.

No tests were conducted with other types of arbjects than cards, since they were the most suitable tags available and other tags were not considered to add anything that the cards couldn't do. Hence we cannot say whether or not other types of tags would work as well as cards. It also depends on what type of application you are developing. If you are going to create an application that will utilise a static arbject, a card might not be the optimal solution. Instead you might want to try to embed a small tag into a bigger object, and then attach an AR marker onto that object.

8.3. Prototypes

This section analyses the three prototypes separately specifically looking at how they were perceived during the final user tests and what design patterns they implemented. In general it can be said that all prototypes suffered from an underdeveloped interface. This results from the main focus of the project not being to develop user friendly applications but to explore how AR and NFC can be used in mobile games. Nevertheless the users thought that a better interface would have added to their experience, especially in the cases of SkidroidXtreme and Game of Stones.

8.3.1. SkidroidXtreme

The SkidroidXtreme prototype got the most even feedback. More than one tester stated that the game "had potential". This was also the only prototype to be presented for the stakeholders as it was both shown in an early state during the Aalborg conference and later during a small presentation during the II conference. During the latter presentation it also received positive feedback from representatives from Schladming's ski resort. They showed particular interest in the use of their existing skiing passes as arbjects. They also noticed other features of the prototype such as collectible game artefacts and user generated content.

Overall SkidroidXtreme was probably the game which suffered most from its prototype state. Rules had to be made outside the framework of the prototype in order to give it meaning as a game. The game also suffered implementation wise from being the first to be developed and therefore had to be reworked since it was built on the first version of the underlying Targine engine.

This game showed us clearly the difference of a static and mobile arbject in that it utilised both design patterns for its gameplay. In the final workshop these patterns were tested mostly in theory. To test the patterns in practice the session should have been carried out at a skiing resort or at least in a larger space. This would have given the testers a more realistic context for where the different designs would be encountered. The theoretical test was instead initially chosen as it would require less time and thus made it easier to recruit more testers. During the later sessions of the final user test it was discovered that there was an issue with the wireless network in the testing space. Wireless network is required for the prototypes to work, therefore this issue made it impractical for testers to walk around too much with their devices.

8.3.2. Game of Stones

Game of Stones is different from the other prototypes in that it includes the by far most complex set of rules. While this complexity was designed as a means of testing how AR can be used to overview massive amounts of game data it is only partially tested. This is because text projected on the arbjects easily got too far away from the device's camera. Similarly to what happens when the eye is too far away from a text this meant that the tester could not read it without getting closer. Whether this format still made it easier than reading from a real world game card was never truly tested. Users were however overall feeling that the AR added to the game and that is was easy to interact with the cards projected on the arbjects.

In order to counter this issue some implementation solutions could also have been made. For instance the user could have been allowed to touch the virtual card on the display of the device to focus on that card and allow for detailed information. This approach was however avoided since it was considered too close to the dominant paradigm in which the devices own interface would

typically be used for all interaction. Another solution could have been to adjust the text size as it got further away and therefore harder to read. For instance it could have had a minimum size. This solution was briefly tested but yielded poor results since the text of faraway arbjects tended to occlude each other. Instead of putting more emphasis into this issue the more general solution of forcing the user to adjust to focus on a card was chosen.

In this prototype the design pattern Individual View became very apparent. It was not implicitly implemented in the prototype as the development never got that far, but it was developed as a concept. Testers also spontaneously mentioned it in different forms as they reflected over their experience while playing the game. The concept was also unintentionally included in the workshop. Some data regarding one player's game assets was not updated for the other player and vice versa, meaning that this information became private. The tester could see that the information was present on the other player's arbject, but not read the data on their own device. During the user tests the testers were asked to avoid this occlusion by being required to give out their hidden information if the other player should ask about it. The concept would however have fitted well into the prototype looking at the game on which Game of Stones was based. In this game personal hidden data is a present game mechanic.

8.3.3. Scavengers

Scavengers was by far the game that most testers perceived as being the easiest to grasp of the three prototypes. This was probably because it was the simplest prototype, with very basic rules. In this prototype the Detective Data pattern was developed and tested as a variation of the Individual View pattern. According to the testers the semi-obscured data was at first hard to detect but later easy to discover. This suggests that some modification to the nature of the obscurity of the data could have been beneficial. For instance the obscurity could have been dynamic so that a user had to consider its variable nature in order to find the data. Possibly it could also have been combined with Individual View.

This prototype suffered much from issues in the later sessions of the final user tests either caused by unreliable network connection or unsolved bugs. This resulted in a more positive feedback in questionnaire during the tests performed before the issue became very present. Even so the simple nature of the game however appealed to the testers even when the prototype crashed or was buggy.

8.4. Interaction Design Patterns

This section discusses the four design patterns which were discovered during the project. Their impression on the testers during the final test and their relation to AR and NFC are analysed.

8.4.1. Mobile Arbject

The Mobile Arbjects were present in the SkidroidXtreme and Game of Stones prototypes. In both prototypes they were seemingly easy to understand for the testers. All the participants of the final user test stated that they would be comfortable in using their ski pass as an arbject. A majority of the participants also stated that they thought that it was easy to use arbjects as playing cards. As the only tested arbjects were cards in themselves it is hard to determine whether or not this was a good choice. However since the testers seemed to understand very quickly how to use them it was at least a sufficient format.

Mobile Arbjects provide more freedom in NFC interaction since either the NFC enabled device can be moved adjacent to the arbject or vice versa. However either way of interaction might interfere with the AR view of the arbject depending on how the camera and NFC reader are attached to the mobile device. When the NFC reader gets close to the arbject it is also likely that the camera will get close to it which means that the AR marker is likely to go out of focus or view of the camera. This issue is countered by the fact that the camera vision relates directly to what is seen by the user normally. Most users were able to switch directly between camera vision and just looking at the arbjects

without much training. This might have to do with what level of experience a user previously had with mobile cameras.

8.4.2. Static Arbject

Static Arbjects were present in both the SkidroidXtreme and Scavengers prototypes. Arbjects designed this way were basically as easy to interact with as they were physically identical during the final user test. This might have made it unclear what was the difference between what type of pattern an arbject was following. This issue was probably more relevant in the tests of SkidroidXtreme as there were only one type of arbject available in the Scavengers prototype. In the future the static arbjects should be integrated with some static object so that it is clearer that they cannot be moved. In the Scavengers prototype it was clearer to the participants that the arbjects were static since those were the only types of arbjects used in the game.

8.4.3. Individual View

The Individual View pattern was not tested properly in this project, since it was not fully implemented in any of the prototypes. The pattern was however considered and to some extent explored theoretically for the Game of Stones prototype, where it also occurred unintentionally.

The pattern makes sense in that it is a small modification of how vision usually works supported by the digital nature of an AR implementation. It would by no means be hard to develop game elements using this pattern for both SkidroidXtreme and Scavengers. For instance the vision of certain slope blocks in SkidroidXtreme could be individual or the pattern could be used in combination with the Detective Data design in Scavengers.

8.4.4. Detective Data

The Detective Data pattern was specifically designed to further test the use of AR in Scavengers. The testers during the final user test found the mechanism to place and find traps, which were the semi-hidden data in this case, to work very well. The results however also showed that the testers overall were most pleased with this prototype as they found it fun and intuitive. These aspects may have caused the opinion towards the design of the game to overall be positive as the other prototypes appeared complex for many testers.

Detective Data is relying on neither AR nor NFC. It is however greatly supported by AR where a game artefact can be observed from many perspectives in an intuitive way while the digital media allows for the searched data to be dynamic in its nature.

8.5. Targine

Targine is a general purpose engine for games using arbjects. Its design allows for extensions on the engine to create new applications. Targine is not only designed for games and can be used to create other applications as well. Due to the flexible nature of arbjects they can be placed almost anywhere. This supports development of applications which make use of the geographical location of an arbject.

Targine is in itself not a portable engine. It is written in Java and therefore relies on the Java environment. Furthermore it relies on the Android API for usage of various sensors an Android device might have. While this means that Targine is supported to some degree by any Android device with the appropriate sensors, it cannot easily for example be used on an iOS device. This is however not tested. Targine does however rely on the on several other tools which in themselves have their own portability features. AndAR is written in Java meaning that if another language is required it would have to be re-written. An alternative could be to exchange it for another wrapper for ARToolKit. Cocos2D is available for C++ and Objective-C as well, so it can be used for iOS at least.

There are no specific limitations to the types of games that can be created with Targine. In fact the engine does in itself not feature any specific tools for any game play design patterns. It simply

provides some basic components for a game base. This base can however, as mentioned above, also be used to create other types of applications than games. The rapid development for which is allowed during the implementation of the Scavengers prototype however proves that it can indeed work well as a tool for game development.

Adding custom features to Targine should not be a problem, since it is made rather abstract. The easiest way is to use the existing engine as a base for a new application, and then create custom and more specific game logic classes. This approach allows for easy access to the AR and NFC features of the engine without limiting the new game.

8.6. Future work

This section presents some future work that could be carried out within the project should it proceed. This includes further investigation into the Individual Vision pattern, exploration of how to reduce glares on the markers as well as investigating some other concepts than the arbjects. Future work on the Targine engine is also presented.

8.6.1. Further Development of Individual View

One thing that could be developed further is the possibility of showing individual information to the users. One example is card games, where the users only should be able to see their own cards. If Targine is extended with this feature you could for example be able to create whatever card based game you want.

8.6.2. Markers with Inverted Contrast

One problem with the arbjects used in this project is that they sometimes are hard for the application to recognise the patterns due to bad contrast or glares. One solution to this problem can be to explore the possibilities with markers with inverted contrast. Additionally, you can also examine different kinds of paper to print the AR marker on. By using less glossy paper you lower the risk of glares.

8.6.3. The Eiffel Tower Issue

During the FIA conference another part of Experimedia presented an issue with a smartphone app developed for using location-based AR to look at points of interest from the top of the Eiffel Tower. The issue is that the compass rotations behaves strangely, making the application misinterpret where the user is pointing the smartphones camera. Possibly this is due to the magnetic disturbance from the massive metal structure that is the tower. A solution to this issue could be to instead use the accelerometer of the smartphone and recalibrate its reference position using NFC tags. This idea led to the concept of using a number of tags to determine a location-based AR view rather than the markers of an arbject. This concept could have been interesting to explore should there had been more time.

8.6.4. The Drawing Board

The arbjects of this project utilise very simple AR markers. So simple, in fact, that a user can draw a marker by hand with enough precision to be used by the smartphone for AR. A drawing board with NFC interaction could therefore have become a sort of variation to am arbject which allowed for several markers and touch interaction points. The markers could also have been designed more as a sort of code than an id, meaning that instead of knowing which marker the user is looking at the various dots and lines forming a marker might present some sort of AR object in themselves. Parallel horizontal lines might for instance mean that an AR tower is to be projected and the tower will be higher for each drawn line.

8.6.5. Targine

While Targine proved to be useful, especially when developing Scavengers without adding anything to the engine, much more can be done with it. Targine currently does not support animation in itself which could be very effect full in AR especially when developing games. Neither does it support

sound or use of any other smartphone sensors than the camera, touch and the accelerometer. Also it would have been interesting to support other forms of AR projection than marker-based AR as well as be able to add new markers in-game. In order to support the drawing board idea presented in section 8.6.4 some modifications would also have had to be done to how Targine and its AndAR framework reads AR markers. Targine has neither been thoroughly tested on various Android devices which could have been interesting to do in order to determine its portability.

8.6.6. Other Mobile Devices

In this project only smartphones were used as mobile devices. While it would have been interesting to test with other similar mobile devices it would have been especially interesting to exchange the phone for devices more designed for AR and NFC interaction. In particular it would have made sense to try the three prototypes using AR glasses such as the ones presented in section 3.1.2 and perhaps a sort of NFC enabled glove. This would have meant that the AR and NFC interaction was moved to the user's eyes and hands, making the technology work in a more intuitive way.

9. Conclusion

The Interactive Institute Game Studio is interested in the exploration of games and new technologies. This project was carried out as a part of Experimedia, an EU project which aspires to explore new forms of live social interaction.

During the course of the project a game engine supporting AR and NFC was created. This engine was named Targine. Using Targine three game prototypes were developed and tested. The prototypes were named SkidroidXtreme, Game of Stones and Scavengers. They were developed for a mobile platform utilising AR and NFC for communication between device and arbject. The prototypes were intentionally different in gameplay and content in order to provide a basis for analysis with some breadth.

SkidroidXtreme is a racing game where multiple players can race each other in a player generated 3D obstacle course. Game of Stones is a card based game inspired by A Game of Thrones: The Card Game. It's a strategic game where the players buy units using their resources and attempt use a strategy which beats the opponent's. Scavengers is a treasure hunting game where the participants attempts to collect treasures and place traps for the opponents.

The prototypes were developed using methods from participatory design and agile development and then user tested with walkthrough sessions complemented by interviews and questionnaires. Using the workshop results, the prototypes and results from the user tests a number of potential, new interaction design patterns were discovered and identified. These new patterns were named Static Arbjects, Mobile Arbjects, Individual View and Detective Data.

We believe that by looking at these patterns the following conclusions can be drawn regarding how AR and NFC can be used in new forms of mobile games, which go beyond the dominant paradigm:

- AR can be successfully used as interface for a wide variety of games and allows for visual interaction to be more easily shared among users who are geographically close to each other.
- NFC can be successfully used to interact with content which have an AR representation and therefore make interaction more intuitive in the non-digital space.
- The combination of AR and NFC can be used to move the interaction outside the mobile device and into a virtual real space. This means that not only can mobile games be as available as a physical board game but new requirements for what defines a mobile device might take form supporting the ideas of ubiquitous computing.

References

Literature

Alexander, C., Ishikawa, S., Silverstein, M. et. al. (1977) *A Pattern Language: Towns, Buildings, Construction*. New York: Oxford University Press.
Android Developers (2012) http://developer.android.com (28 Jun. 2012).

Bayle, E., Bellamy, R. & Casaday G. et. al. (1998) *Putting It All Together Towards a Pattern Language for Interaction Design: A CHI 97 Workshop*. Atlanta: SIGCHI Bulletin.

Björk, S. (2011a) Cards. *Game Design Patterns 2.0*. http://gdp2.tii.se/index.php/Cards
(3 Jul. 2012).

Björk, S. (2011b) Player Created Game Elements. *Game Design Patterns 2.0*.
http://gdp2.tii.se/index.php/Player_Created_Game_Elements
(3 Jul. 2012).

Björk, S. (2011c) Maneuvering. *Game Design Patterns 2.0*.
http://gdp2.tii.se/index.php/Maneuvering (19 Aug. 2012).

Björk, S. (2012a) Patterns. *Game Design Patterns 2.0*.
http://gdp2.tii.se/index.php?title=Category:Patterns&until=Landmarks (3 Jul. 2012).

Björk, S. (2012b) Multiplayer Games. *Game Design Patterns 2.0*.
http://gdp2.tii.se/index.php/Multiplayer_Games (3 Jul. 2012).

Björk, S. (2012c) Player-Player Proximity. *Game Design Patterns 2.0*.
http://gdp2.tii.se/index.php/Player-Player_Proximity
(3 Jul. 2012).

Borchers, J. (2000) *A Pattern Approach to Interaction Design.* New York: Darmstadt University of Technology.

Business Insider (2011)
http://articles.businessinsider.com/2011-11-22/tech/30427958_1_mobile-gaming-pcs-and-consoles-games-industry (17 Aug. 2012).

Caudell, T. & Mizell D. (1992) *Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes*. Seattle: Boeing Computer Services, Research and Technology.

Chelaru, V. (2007) *Rock Paper Scissors - A Method for Competitive Game Play Design*. *Gamasutra*.
http://www.gamasutra.com/view/feature/1733/rock_paper_scissors__a_method_for_.php
(3 Jul. 2012).

Cockburn, A. (2008) Using Both Incremental and Iterative Development. *CrossTalk - The Journal of Defense Software Engineering.* May 2008.
http://www.crosstalkonline.org/storage/issue-archives/2008/200805/200805-Cockburn.pdf
(19 Jun. 2012).

Cooper, A., Reimann R. & Cronin, D. (2007) *About Face 3: The Essentials of Interaction Design*. Indianapolis, Indiana: Wiley Publishing, Inc.

Copeland, L. (2001) Extreme Programming. *ComputerWorld.* December 2001. http://www.computerworld.com/s/article/66192/Extreme_Programming?taxonomyId=063 (20 Jun. 2012).

Experimedia (2012) http://www.experimedia.eu/ (18 Aug. 2012)

Faulkner, X. (2000) Usability Engineering. Basingstoke, Hampshire & New York, New York: Palgrave.

Future Internet Assembly (2012) http://www.fi-aalborg.eu/ (18 Aug. 2012)

The Guardian (2011) http://www.guardian.co.uk/technology/gamesblog/2011/nov/17/game-consoles-future-tablets (17 Aug. 2012)

Hildenbrand, J. (21 Dec. 2010) Gingerbread feature: Near Field Communication. Android Central. http://www.androidcentral.com/gingerbread-feature-near-field-communication (17 Aug. 2012)

Hienerth, C., Pötz, M. & Hippel, E. (2007) *Appropriability, Proximity, Routines and Innovation*. Copenhagen: Copenhagen Business School.
CNN Money (2012) http://money.cnn.com/2012/04/04/technology/google-project-glass/?source=cnn_bin (2 Jul. 2012).

IDG (2012) http://www.idg.se/2.1085/1.429210/nfc-i-wii-us-handkontroll (2 Jul. 2012).

Moore, G. (1965) http://download.intel.com/museum/Moores_Law/Articles-Press_Releases/Gordon_Moore_1965_Article.pdf (1 Aug. 2012).

NFC Forum (2012a) http://www.nfc-forum.org/aboutus/ (17 Aug. 2012)

NFC Forum (2012b) http://www.nfc-forum.org/specs/spec_list/ (19 Aug. 2012)

NFC World (2012a) http://www.nfcworld.com/nfc-devices/nokia-6131-nfc/ (2 Jul. 2012).

NFC World (2012b) http://www.nfcworld.com/nfc-devices/nokia-701/ (2 Jul. 2012).

Nokia Beta Labs (2012) Nokia NFC Games. http://betalabs.nokia.com/trials/nfcgames (17 Aug. 2012)

PARC (2010) A Tribute to Mark Weiser Father of Ubiquitous Computing and the Rise of the Mobile Mashup. http://www.parc.com/event/1210/tribute-to-mark-weiser-father-of-ubiquitous-computing-and-the-rise-of-the-mobile-mashup.html (28 Jun. 2012).

Preece, J., Sharp, H. & Rogers, Y. (2007) Interaction Design: Beyond human computer interaction, 2nd Edition. Chichester, West Sussex: John Wiley & Sons, Ltd.

Rusk, J. (2006) Crystal Clear Methodology. *AgileKiwi*. http://www.agilekiwi.com/other/agile/crystal-clear-methodology/ (19 Aug. 2012).

Schwaber, K. (2004) *Agile Project Management with Scrum*. Redmond, Washington: Microsoft Press.

Stack Overflow (2012) http://stackoverflow.com/ (28 Jun. 2012).

Starner T., Mann S. & Bradley R. et. al. (1995) *Wearable Computing and Augmented Reality*. Cambridge: Massachusetts Institute of Technology.

Sun (2012) The Java™ Language: An Overview. *Sun.com*. http://java.sun.com/docs/overviews/java/java-overview-1.html (19 Jun. 2012).

Tidwell, J. (2010) *Designing Interfaces: Second Edition*. Canada: O' Reilly Media, Inc.

Vanderkay, J. (2004) Nokia, Philips And Sony Establish The Near Field Communication (NFC) Forum. *NFC Forum*. http://www.nfc-forum.org/news/pr/view?item_key=d8968a33b4812e2509e5b74247d1366dc8ef91d8 (3 Jul. 2012).

Weiser, M. (1996a) *Open House*, Review, the Web Magazine of the Interactive Telecommunications Program of New York University, ITP Review 2.0, March 1996.

Weiser, M. & Brown, J. (1996b) The Coming Age of Calm Technology[1]. Xerox PARC, October 1996.

Applications and Products

AndAR (2010) http://code.google.com/p/andar/ (28 Jun. 2012).

AndAR Pong (2010) http://www.youtube.com/watch?v=DK5v4MPwjmU (12 Aug. 2012)
Apache Subversion (2012) http://subversion.apache.org/ (28 Jun. 2012).

Aptus (2012) http://www.aptus.se/sv/site/aptusse/Produkter/?groupId=744586&productId=778986 (2 Jul. 2012).

AR Pacman (2004) http://www.gizmag.com/go/3512/ (12 Aug. 2012).

cocos2d-android (2010) http://code.google.com/p/cocos2d-android/ (28 Jun. 2012).

Eclipse (2012) http://www.eclipse.org/home/ (28 Jun. 2012).

Games Workshop (2012) www.games-workshop.com (16 Aug. 2012).

Google Code (2012) http://code.google.com/intl/sv-SE/ (28 Jun. 2012).

Google Galaxy Nexus (2011) http://www.google.com/nexus/#/galaxy (16 Aug. 2012).

Google Sky Map (2011) https://play.google.com/store/apps/details?id=com.google.android.stardroid&hl=en (2 Jul. 2012).

Google Wallet (2012) http://www.google.com/wallet/ (2 Jul. 2012).

Layar (2012) http://www.layar.com/ (2 Jul. 2012).

Magic: The Gathering (1993) http://www.wizards.com/Magic/TCG/Article.aspx?x=magic/rules (16 Aug. 2012).

Metaio (2012) http://www.metaio.com/ (19 Aug. 2012).

Nintendo (2012) http://kidicarus.nintendo.com/uprising/explore-the-game/#ar-cards (2 Jul. 2012).

Octopus (1994) http://www.octopus.com.hk/about-us/corporate-profile/our-history/en/index.html (17 Aug. 2012).

Pokémon Trading Card Game (1997) http://boardgamegeek.com/boardgame/2165/pokemon-trading-card-game (16 Aug. 2012).

Random Word Generator (Plus) (2007) http://watchout4snakes.com/creativitytools/randomword/randomwordplus.aspx (15 Aug. 2012).

Vuforia (2012) http://www.qualcomm.com/solutions/augmented-reality (16 Aug. 2012).

Wikitude (2012) http://www.wikitude.com/ (2 Jul. 2012).

Appendix A: First workshop results

These are samples of ideas that came up during the first workshop.

Fry unlike beer
steka saker man inte gillar
ladda ner bilder på saker man inte gillar, steker

Strike towards storage
Man får någon belöning om man slår sig igenom något

Absorb upon travel
man åker runt och suger upp allt
GPS/app som suger in info från omgivningen
sparar var du har gått, spelar upp historia som var du gått

Curse during safekeeping
Nåt som man använder och litar på men som ibland jävlas med en.
Lägga förbannelse på en mottagares kista, så att de blir skadade om de öppnar kistan
inför varje bana får man en curse på sig. kan vara t.ex. att man bara har 1 i hp

Coagulate throughout outside
man är en stor slemmig boll som stelnar utifrån, till slut kan man inte röra sig.

Detach but protect
Jenka. Ta ut träklossar utan att tornet rasar. Intressant appkoncept.

Hack up 3D
Hackar upp sitt knark i linor. AR-spel.

Hack round thumbs
App where you get a shape that you hack away with your thumbs.

Upon via yes, this is dog
Create a chain of messages among your friends

Appendix B: Second workshop results

1. CG exp	2. SH exp	3. AR contr. to CG*	4. NFC contr. to CG	5. AR contr. to SH**	6. Interface		
2	5	5	5	2	4		
5	3	4	1	3	2		
4,5	3	4	4	1	5		
5	3	2	3	1	2		
3	3	2	3	2	5		
4	3	1,5	1	1	4		
3	1	2	3	4	4		
4	2	3	1	2	3		
3,8125	2,875	2,9375	2,625	2	3,625		
4	3	2,5	3	2	4		
		"If it had worked better it would have been awesome"	"More confusing with more steps"	"would have won without it"	"Lacks feedback"		*CG: Card Game **SH: Scavenger Hunt
		"Mostly there for fun"	"Blipping cards felt redundant"	"Would have been more fun if it was impossible wo see the which card it was witouth AR"	"Easy once it had been explained"		
		"Delayed the surprise, different experience"	"Worked a little so-so, not completely 'snappy'"	"Could potentially see if someone had been there before, fun"	"Didn't understand that about pushing it and then the card picture disappeared, I tried to 'toktrycka'"		
		"Would not have been as fun with just the cards"	"Adds strategy elements but awkward gestures"		"Awkward with the physical phone buttons"		
			"Cool, easy to see the potential"				

Appendix C: Third workshop results

1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	1.10	1.11	2.1	2.2	2.3	2.4	2.5	2.6a	2.6b	2.7a	2.7b	2.8a	2.8b	2.9a	2.9b
y	4	4	3	y	2	1	2	y	y	5	5	3	2	5	y	1	3	3	5	n	1	y	
n	3	3	3	y	4	3	3	y	n	5	4	3	2	3	y	4	4	3	3	y	4	n	2
n	4	4	1	n	3	2	4	y	n	3	5	5	3	4	y	2	1	3	2	y	4	y	5
y	1	1	1	n	1		3	y	n	5	3	1	1	5	y	1	2	1	1	n	1	n	1
n	2	3	2	n	2	1	3	y	y	3	4	3	3	4	y	2	1	4	4	n	1	y	3
y	4	4	4	n	2	3	3	y	y	4	4	3	2	3	y	3	4	3	3	n	1	y	3
n	4	4	3	y	5	2	4	y	n	5	3	2	1	3	y	5	5	3	2	y	2	y	3
n	3	5	5	y	5	1	5	y	n	5	5	4	2	5	y	1	1	3	3	y	5	n	1
y	4	4	2	y	1	2	2	y	n	5	5	3	3	2	y	3	5	5	4	y	3	y	4
y	4	4	3	n	4	2	4	y	n	5	4	2	3	4	y	3	4	3	4	y	3	n	3
n	3	3	1	n	2	3	4	n	n	5	5	5	3	4	y	4	5	3	4	y	4	y	5
y	5	5	5	n	4	3	5	y	n	5	4	2	3	5	y	4	5	2	3	n	3	n	2
y	1	1	2	n	5	5	4	y	n	3	5	5	3	3	y	3	5	4	5	n	5		5
y	5	5	4	y	4	2	2	y	y	5	2	3	3	4	y	3	3	3	3	n	3	y	4
y	2	2	3	n	4	2	5	y	y	5	3	2	1	4	y	3	3	3	3	y	3	n	3
y	4	4	4	n	4	1	3	y	n	5	4	2	3	5	y	4	4	4	4	n	1	n	1
y	4	3	3	n	5	2	3	y	n	5	4	2	3	4	y	4	5	4	4	n	1	n	3
y	4	4	4	n	5	3	3	y	y	5	5	4	4	3	y	1	1	3	3	y	4	y	4
y:11	3	4	3	y:6	3	2	3	y:16	y:5	5	4	3	3	4	y:17	3	3	3	3	y:8	3	y:8	3
	4	4	3		4	2	3			5	4	3	3	4		3	4	3	3		3		3

2.10	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	4.1	4.2	4.3	4.4	4.5a	4.5b	4.6	4.7	4.8	4.9	5.1	5.2	5.3
	3	3	1	4	3	1	2	y		5	5	5	2	4	4	5				y	y	y
	3	3	3	4	2	2	4	y		3	3	5	2	5	5	4	3	4		y	y	y
x	2	2	3	5	5	4	4	n		5	5	4	2	5	5	3	4	5		y	n	y
	1	3	2	4	3	2	3	n		5	4	3	4	5	4	4	3	4		y	y	y
	4	4	3	5	4	4	5	y		4	4	5	2	5	4	5	5	5		y	y	y
x	2	2	5	2	4	2	3	n		4	3	5	2	4	4	3	4	5		y	y	y
x	3	1	2	4	2	2	5	y	x	5	3	4	3	5	5	5	3	2	x	y	y	n
x	3	3	2	5	5	3	5	y	x	5	5	3	5	2	1	5	5	5		y	y	y
	2	5	4	4	5	4	4	y		5	5	5	3	4	3	4	3	4		y	y	y
x	3	4	4	4	5	2	4	y		5	5	4	3	5	5	5	4	4	x	y	y	y
x	5	5	4	3	4	5	4	y	x	5	5	5	2	5	5	4	2	5	x	y	y	y
	1	2	5	3	5	1	5	y		3	5	5	3	3	4	5	3	3		y	y	n
	5	5	3	3	5	3	5	y								5	5	5		n	y	y
	4	2	2	5	4	2	4	y		2	4	5	2	3	4	3	4	5		y	y	y
x	3	3	3	5	3	3	4	y	x	3	3	4	3	4	3	5	2	4		y	y	y
x	4	2	4	5	4	4	4	y	x	5	2	5	3	3	4	5	3	4		y	y	y
	5	4	3	4	5	2	5	y	x	4	3	5	2	3	5	5	4	5	x	y	y	n
x	5	4	5	3	5	5	5	y	x	5	5	5	3	4	5	5	4	5	x	y	y	y
	3	3	3	4	4	3	4	y:14		4	4	5	3	4	4	4	4	4		y:16	y:16	y:14
	3	3	3	4	4	3	4			5	4	5	3	4	4	5	4	5				

Appendix D: Software architecture UML diagram