

CHALMERS



Exploring the Interaction Possibilities when Combining a Tablet and Smartphones in a Game Context

Master of Science Thesis in the Master Degree Programme, Interaction Design

JOAKIM EKENDAHL
VIKTOR INGEMANSSON

Department of Applied Information Technology
Division of Interaction Design
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2012
Report No. 2012:079
ISSN: 1651-4769

Exploring the Interaction Possibilities when Combining a
Tablet and Smartphones in a Game Context

JOAKIM EKENDAHL

VIKTOR INGEMANSSON

© JOAKIM EKENDAHL & VIKTOR INGEMANSSON, 2012.

Technical report no 2012:079

Department of Applied Information Technology

Chalmers University of Technology

SE-412 96 Göteborg

Sweden

Telephone + 46 (0)31-772 1000

Cover:

A sample of the final version of the game that was implemented during the project. The picture illustrates an ongoing trade. The game is described thoroughly in section 5.

Göteborg, Sweden 2012

Exploring the Interaction Possibilities when Combining a
Tablet and Smartphones in a Game Context

JOAKIM EKENDAHL

VIKTOR INGEMANSSON

Department of Applied Information Technology

Chalmers University of Technology

ABSTRACT

Smartphones and tablets are becoming more common. The aim of this study was to find what interesting interaction possibilities emerge when combining a tablet with a smartphone. The research was performed by implementing a game using this medium. A Settlers of Catan game was created using the tablet as the board and the phones as the private hands of the players. The application was implemented using Bluetooth, and a system based on drag and drop, using portals to send items between the devices was found useful to mediate the board game experience. Another important aspect that can be used in a game like this is the fact that users can use oral communication to plan common decisions, which can simplify the use cases for certain actions. It is also possible to add special rules that only work in a digital setting, but it seems it is difficult to find rules that make the game more interesting.

Keywords: tablet, smartphone, Android, Settlers of Catan, board game.

Table of Contents

| | |
|---|----|
| 1. Introduction..... | 1 |
| 1.1 Purpose..... | 1 |
| 1.2 Scope..... | 1 |
| 1.3 Objective..... | 1 |
| 2. Background..... | 2 |
| 2.1 Settlers of Catan..... | 2 |
| 2.1.1 Original game..... | 2 |
| 2.1.2 Digital versions..... | 2 |
| 2.2 Related work..... | 3 |
| 2.2.1 Joining Bits and Pieces..... | 3 |
| 2.2.2 Settlers of Catan for Microsoft PixelSense..... | 3 |
| 2.2.3 Shared-Screen Social Gaming with Portable Devices..... | 4 |
| 2.2.4 Smartphone Game Controller..... | 4 |
| 2.2.5 Multimodal Multiplayer Tabletop Gaming..... | 4 |
| 2.2.6 Exploring Automation in Digital Tabletop Board Games..... | 4 |
| 2.2.7 Weathergods..... | 4 |
| 2.2.8 Tabula Imaginarium..... | 4 |
| 2.2.9 Tisch..... | 5 |
| 2.2.10 Wizard's Apprentice..... | 5 |
| 2.3 Methods..... | 5 |
| 2.3.1 Incremental/Iterative development..... | 5 |
| 2.3.2 Agile development..... | 6 |
| 2.3.3 Scrum..... | 6 |
| 2.3.4 Extreme programming..... | 6 |
| 2.3.5 Sketching..... | 6 |
| 2.3.6 Wireframing..... | 6 |
| 2.3.7 Paper prototyping..... | 7 |
| 2.3.8 Playtesting..... | 7 |
| 2.4 Technologies..... | 7 |
| 2.4.1 Java..... | 7 |
| 2.4.2 Android..... | 8 |
| 2.4.3 Cocos2D..... | 8 |
| 2.4.4 JSON..... | 8 |
| 2.4.5 Bluetooth..... | 8 |
| 2.4.6 Subversion..... | 8 |
| 2.4.7 Eclipse..... | 8 |
| 3. Planning..... | 9 |
| 3.1 Initial suggestions..... | 9 |
| 3.2 Approach..... | 10 |
| 3.3 Post-decision planning..... | 11 |
| 3.4 Time plan..... | 11 |

| | | |
|-------|--------------------------------------|----|
| 4. | Realization..... | 14 |
| 4.1 | Pre-development phase | 14 |
| 4.1.1 | Mechanics pre-study | 14 |
| 4.1.2 | Game research | 15 |
| 4.1.3 | Game decision and interactions | 18 |
| 4.1.4 | Sketching and further testing | 20 |
| 4.2 | Application development | 22 |
| 4.2.1 | Exploration..... | 23 |
| 4.2.2 | Iteration one | 23 |
| 4.2.3 | Iteration two | 24 |
| 4.2.4 | Iteration three | 26 |
| 4.2.5 | Iteration four..... | 27 |
| 4.3 | Playtesting sessions..... | 29 |
| 4.3.1 | Final tests..... | 30 |
| 5. | Results..... | 31 |
| 5.1 | Gameplay | 31 |
| 5.1.1 | Extensions | 33 |
| 5.2 | The applications..... | 35 |
| 5.2.1 | The board application | 35 |
| 5.2.2 | The hand applications..... | 38 |
| 5.2.3 | Tab version specifics | 40 |
| 5.2.4 | Portal version specifics..... | 41 |
| 5.3 | System architecture..... | 43 |
| 5.3.1 | The common library | 43 |
| 5.3.2 | The phone project (tabs)..... | 45 |
| 5.3.3 | The phone project (portal) | 45 |
| 5.3.4 | The tablet project | 46 |
| 6. | Discussion..... | 48 |
| 6.1 | Project goals | 48 |
| 6.1.1 | Interaction possibilities..... | 48 |
| 6.1.2 | Game experience mediation | 49 |
| 6.1.3 | Utilizing features | 50 |
| 6.2 | Time plan..... | 51 |
| 6.3 | Gameplay | 53 |
| 6.4 | Phone version comparison..... | 54 |
| 6.5 | Future work | 55 |
| 6.5.1 | Other implementations | 55 |
| 6.5.2 | Research in Settlers of Catan | 55 |
| 6.5.3 | Application improvements..... | 56 |
| 7. | Conclusion..... | 58 |
| | References | 59 |
| | Appendices | 61 |

1. Introduction

Smartphones and tablets are becoming more common and most people nowadays own some kind of touch-display device. The Mobile Touch group at Chalmers is interested in investigating the interaction and communication possibilities between phones and tablets. The core part of this is the combination between the devices, as it seems there has been little research previously done in this field. To explore this there is a need for a suitable context to base the content on. Board games are often played in groups with both private and public items, which make them good targets for this.

1.1 Purpose

This project aims to research and explore the use of smartphones in combination with a tablet, to function as interactive devices when playing a game. The tablet would act as a public area and the smartphones as private areas for the participants. Basically this means using the combination to achieve a new experience when playing a game that normally is played on a board with self-owned game elements such as cards, pieces or tokens that are hidden from other players.

1.2 Scope

The interaction between the phones and the tablet is the main focus. It is also important that the graphical user interface supports this interaction, but information visualization has lower priority. There is no need to develop the game application more than necessary in terms of graphical design. The game should be based on an existing game to minimize gameplay design. However some rules should be added to test new interaction possibilities.

The applications will be created for Android phones and tablets, as time can be saved by focusing on just one platform. In the future, a version could perhaps be made for iPhones and iPads too, but not in this project.

1.3 Objective

Following are some questions that are to be answered during the project:

- Which interesting interaction possibilities can be used through combining several smartphones with a tablet?
- How can the board game experience be mediated with only digital game components displayed on a tablet and smartphones?
- Which new exciting features can be added to utilize the digital media for improving the game experience?

2. Background

This section contains some background information about the chosen game, as well as related work and some information of existing technologies and methods that have been used in this project. The intention is that the most used titles, which are referred to later in this thesis, are to be presented and described here.

2.1 Settlers of Catan

There are a number of games that are suitable for the tablet/smartphone-platform. Ultimately, the decision was to make a version of *Settlers of Catan*. In this section, the game rules will be discussed. There are also several digital versions of the game which have been looked at to see how one can solve the challenges to convert a physical game to a digital one. Other suitable games that have been considered are mentioned in section 3.1.

2.1.1 Original game

The aim of the game is to settle the island of Catan by building settlements, cities and roads, using resources harvested on the island. The player to first achieve 10 victory points is the winner. One victory point is given per settlement, and another is added if it is replaced with a city. The game also contains a set of development cards. These cards can be played for special actions at a later stage of the game, with the exception of the victory point cards which award a victory point directly on acquisition. Points are also given out for the longest connected road and the largest army (most knight cards played).

The island is a tile map based on 19 connected hexagons, and each hexagon has a resource and a value in the range 2-12. Resources are given out to all players based on the roll of two dice in the beginning of every turn. Each hexagon with a value matching the combined dice result will award players resources if they have settled around that hexagon. The resource that is given depends on which type of tile that is triggered. One resource is given for each settlement, and two for each city. There is also a special case when the result of the dice equals 7. In this case the robber can be moved, and block the resource of another hexagon. A resource will also be stolen from a nearby player, if such a player exists.

BoardGameGeek (2012a) has much more information about the game on their website. They also mention the success of the game; "Multi-award-winning and one of the most popular games in recent history due to its amazing ability to appeal to non-gamers and gamers alike".

2.1.2 Digital versions

There are a few existing Settlers of Catan-based digital games on the market. Exozet Games GmbH has made a game called *Catan* for a number of platforms. It is available for most mobile devices, as well as Nintendo DS and Amazon Kindle. The Android version is shown in figure 2.1. Exozet Games are also the authors of a web based game that can be played online. With the web game as an exception, all players have to play on the same device, either towards real players or with artificial computer opponents. The web version can be played with other players over the Internet.



Figure 2.1: Catan for Android.

There is also a game developed for Xbox 360 by Big Huge Games and released in 2007. This game can also be played online or with computer controlled opponents. Unlike the previously mentioned game, *Catan* for Xbox features a 3D environment that can be rotated to change view. In both the Xbox version and the version for mobile devices, players that play using the same screen have their resources and other items visible to all players.

2.2 Related work

Some previous similar research has been done in this area, both theoretical and practical. A few relevant studies are introduced in this section.

2.2.1 Joining Bits and Pieces

Lundgren (2002) explored how board games could be combined with computational features using ubiquitous computing. Basically, this meant introducing new mechanics and items to tabletop games that required digital components. Two games were designed in this study, but they were not realized completely. The conclusion of the paper was that there were many opportunities for digital components in board games that could be used to create new and exciting games in the future. These games are, however, more vulnerable and more expensive to create than normal board games.

2.2.2 Settlers of Catan for Microsoft PixelSense

Settlers of Catan has previously been explored by Vectorform in a digital touch-based environment (Lamb, 2010). It was made on a Microsoft Surface (now called Microsoft PixelSense). This was implemented having real dice that were recognized by the built-in cameras inside the device. Since all players would stand together around the PixelSense and see everything on the screen, the privacy of each player's own cards have been solved by providing an x-ray visor to cover the cards. The player then can see their own cards, but also keep them hidden to the other players.

2.2.3 Shared-Screen Social Gaming with Portable Devices

Kauko and Häkkinen (2010) investigated an interaction method based on a shared screen between two devices. The main purpose was to examine portable devices, such as mobile phones, using a face-to-face setup but also a console game setting was tested for comparison with a side-by-side setup. 40 testers were used in this study, and the conclusion was that the oral communication was significantly higher when playing with the mobile phones. Ergonomics, distance, spectators and more seem to have affected the experience too.

2.2.4 Smartphone Game Controller

Katzakis et al. (2011) researched the possibility of using a smartphone as a controller for a three-dimensional game. The idea was to use the orientation of the phone combined with some gestures to enable users to move an object in a three-dimensional environment. A test was conducted with a prototype game where users had to move a UFO through several circular gates in a predefined order. A number of control variants were tested and the conclusion was that a version based on point and click worked best for this particular game.

2.2.5 Multimodal Multiplayer Tabletop Gaming

This study by Tse et al. (2007) was aimed at exploring how digital games could be more social. The aim was to get away from the current multiplayer experience in many games with split screen or online play. Instead, the idea is to let players play on a shared screen using understandable gestures and speech which also would allow other players to easily follow what is happening. The games Warcraft III and The Sims were tested based on these types of interactions.

2.2.6 Exploring Automation in Digital Tabletop Board Games

A study where the automation possibility when putting a board game into a digital context was done by Wallace et al. (2012). They specifically looked at a game called Pandemic and carried out this study by letting several user groups test three different game versions with various levels of automation. The results showed that "while game automation can positively affect gameplay, it can also negatively impact player awareness and enjoyment of the game". The conclusion of the study also states that flexibility of the interface should be sought to facilitate for the user in many aspects.

2.2.7 Weathergods

Another example as a part of a study on tangible interaction, which is treating a particular game is the paper *Weathergods: tangible interaction in a digital tabletop game* by Bakker et al. (2007). This part of the study is about taking the advantages from board game and combine with the benefits that digital games provide. The result was a game where "tangible play pieces have been combined with the dynamic multimedia aspects of digital tabletops", which is described as a "unique hybrid game" with the desired properties.

2.2.8 Tabula Imaginarium

Johansson and Lundberg (2011) created an application for iPads aimed to aid players in tabletop role-playing games. The purpose was to find out if players could engage in role-playing game without physically being present at the same place. The idea was to transfer the game to the iPad medium, while at the same time keeping the application generalized to work with nearly all tabletop role-playing games. An integral part was that everything done on the iPad by one player would be synchronized with the iPads of all other players.

2.2.9 Tisch

Tisch is an application running on Microsoft PixelSense (formerly known as Microsoft Surface) developed by Hartelius, Fröhlander and Björk (2012). Its purpose was to convert the tabletop experience to a digital medium without hindering the players from using their own rules. A key point here was to create a very general platform that worked even with house specific rules. The experience could be enhanced further by adding extra digital support. For example, excise (Cooper, 2003) could be reduced from a game by playing it like this. A few games were tried and tested and the conclusion was that people are interested in playing games like this, because of all the help given by the digital media, while still allowing them to play the game in their own way.

2.2.10 Wizard's Apprentice

Peitz, Björk and Jäppinen (2006) created a prototype of a computer-augmented board game called *Wizard's Apprentice*. The purpose of the game was to fulfill some gameplay design goals concerning the social adaptability of a game. A few main points were to explore changing levels of involvement in the game and how to avoid disturbing the social life outside of the game. A preliminary evaluation pointed out that the less active role as the wizard was seen as not fun even if the intention was that it should require less involvement. The game did, however, allow for users to take small pauses and to socialize. Overall, the experience was concluded as more like a board game than a digital game.

2.3 Methods

Several methods have been used in this process, both actual development methods and methods used to find ideas and try out suggestions. The most important ones are presented in this section.

2.3.1 Incremental/Iterative development

"Incremental development is a staging and scheduling strategy in which various parts of the system are developed at different times or rates and integrated as they are completed", Cockburn (2008) describes the incremental method. In the same way he describes the iterative method: "Iterative development is a rework scheduling strategy in which time is set aside to revise and improve parts of the system". Based on the weaknesses of the waterfall model, where every activity is finished before the next one begins, both these methods exercise a more cyclic approach to the software development process. All functionality does not need to be correct at once, the process may very well be divided into layers where functionality is added as new layers are covered. Essentially modifications always can be done to things that earlier have been completed, things from previous iterations are not set in stone.

2.3.2 Agile development

Agile software development is much based on iterative and incremental development. The core of this actual group of methods can be seen as to allowing changes in a rapid manner, to be flexible about requirements and such. The four prioritization pillars in agile development are as follows (Beck et al., 2001):

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

2.3.3 Scrum

This method uses practices both incremental and iterative agile development. It is more oriented towards project management in the sense of that it focuses on institutions where it is difficult to plan ahead. Scrum incorporates the members taking team roles, the process divided into so called sprints and there are also specific product artifacts and meeting structures that a Scrum-based process should follow. Contained in the meeting structure is the so called *Daily Scrum* which occurs every day during a sprint. Here three questions are to be answered by the team members; "What have you done since yesterday?", "What are you planning to do today?" and "Any impediments/stumbling blocks?". The time for this meeting is not allowed to exceed 15 minutes.

2.3.4 Extreme programming

Extreme programming, also known as XP, is a methodology that builds upon simplicity and frequency. Copeland (2001) means that it is important to write simple and understandable code, as well as reviewing and testing it frequently. Among twelve core practices in XP, a few of them concludes that small code release should be put into production early, using a common system for names and descriptions. One practice states that the process of refactoring should be used, which means revising and editing the code frequently. It is also encouraged to work in pairs to be able to continually see and discuss each other's code.

2.3.5 Sketching

"The intention of sketching is to separate design from the process of making", Lepore (2010) describes sketching. In general this is considered a fast way of visualizing an idea or testing out a layout of a design. It is preferably done with less details and the outcome is often of low material value, meaning that a sketch can be easily thrown away and redone. However the value lies in communicating ideas and see explore alternatives.

2.3.6 Wireframing

According to Lepore (2010), "A wireframe's purpose is to communicate and explore the concepts that come out of sketching—that is, those concepts you actually want to pursue further during user interface design". She describes wireframing as a sort of continuation of, or the next step after sketching where the ideas from sketching are further refined and designed to communicate the concept that was previously produced. The essence is however to have a quick and flexible method to explore and test various ideas at an early stage of development.

2.3.7 Paper prototyping

A definition of paper prototyping was stated by Snyder (2003): "Paper prototyping is a variation of usability testing where representative users perform realistic tasks by interacting with a paper version of the interface that is manipulated by a person "playing computer", who doesn't explain how the interface is intended to work". As explained the intention is to have users interacting with a not yet built interface to still be able to test it and see if it would be worth constructing. This is often done by having a list of scenarios and then ask the tester to perform certain tasks from these scenarios. Then the interface is evaluated from these tasks depending on how easy it was to perform them.

2.3.8 Playtesting

A basic, yet useful method to try out something by acquiring opinions and see how a certain product behaves is to simply test it, preferably with people from outside the development. There are a lot of different test methods with various fields to cover, for example unit testing which tests only a small part of a whole system while system testing covers the whole system. However playtesting is commonly used when developing computer games, board games or role-playing games. They can be performed in various modes such as *open* or *closed*, depending on who are allowed to participate in the testing. The core is anyhow to expose the product to people in a possible target group to help improving the current version.

2.4 Technologies

Technologies used in the project, such as programming languages, structures and programs will be explained in this section.

2.4.1 Java

Java is an object-oriented programming language originally developed by Sun Microsystems, now merged into Oracle Corporation. Typically, Java programs are compiled to class-files that contain byte code. These files require that the device has a Java Virtual Machine, but they can be run on any platform as long as that requirement is fulfilled.

Sun (2012) describes the language with a set of adjectives; "A simple, object-oriented, network-savvy, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, dynamic language". By simple, they mean that many of the unnecessary features of the language C++ that could be considered confusing have not been integrated into Java. Memory management is also not needed, since the Java engine contains an automatic garbage collection. Sun also explains other mentioned properties; Java has an extensive library for network connections, it is strongly typed and many errors can be discovered at compile-time. It has strong support for multithreading and a set of synchronization primitives to facilitate this.

2.4.2 Android

Android is an operating system used on smartphones and tablets. It was originally developed by Android Inc. but was bought by Google in 2005. It is now maintained by the Open Handset Alliance, a consortium of companies in the telecommunication line of business. Most applications for Android are developed using the Android SDK, which has a toolset consisting of a customized version of Java and a framework for creating Android applications. The graphical user interface can be designed in XML (Extensible Markup Language) that can be converted to graphical elements with the Android libraries.

2.4.3 Cocos2D

Cocos2D is a graphics framework originally constructed for 2D games. An iPhone OS version has also been made, which has a port to Android. The framework contains many useful tools for creating game graphics, such as sprites, transformation actions and integrated sound and physics engines.

2.4.4 JSON

JSON is an abbreviation of JavaScript Object Notation. It is a lightweight format commonly used for data interchange. A well-defined protocol called JSON-RPC is normally used to communicate data between server and client when the JSON format is used in this setting. This protocol can be used to call methods remotely from the client to the server, and data can be returned similar to how a normal method works.

2.4.5 Bluetooth

Bluetooth is a technology to connect adjacent devices wirelessly. It works similarly to local area networks but the signals are sent and received directly from a small chip in each device, and does not require an external connection.

2.4.6 Subversion

Subversion (SVN) is a version control system. It can be used over networks, which allows users on different computers to work together. When local changes are committed, the system will merge the changes with the repository on the server. Old versions are also stored and it is possible to revert to a previous version, or branch your project at any time. Since all files are stored on the server, it also works as a backup in case any of the local versions should be lost.

2.4.7 Eclipse

The integrated development environment (IDE) used in this project is a program called *Eclipse*. It is mainly used for Java programming but it can also be used for other purposes. The program has a built-in framework to support plug-ins. In this project, *Subclipse* is used, which is a plug-in to handle the subversion system directly from Eclipse. In order to facilitate Android development, a plug-in that can be used to create Android projects, and run them directly on the phone or an emulator, is also used.

3. Planning

This section goes through how the project was planned and what was the intention with the overall process.

3.1 Initial suggestions

In the beginning of the project several suggestions were brought up as possible game concepts to use. They consisted of three already existing pure games and one more general concept. The first of the game suggestions was *Poker* or *Texas Hold'Em*. This was chosen because of the simplicity of the game contents. With a card deck as the only game component, the focus could be taken from implementing a lot of components and instead work with the interaction part of the game.

Another game suggestion was *Play-by-Play*, a hockey-themed game previously developed by one of the group members. This game was originally developed as a combined board and card game, but is also available in a digital form since January 2012. As it involves both private and public cards, similar to the previously mentioned game, it would also have been suitable for the environment. However the theme and style were narrower and this was not suitable when deciding what game to use.

The last of the three existing games was *Settlers of Catan*, a seemingly popular board game that involves two types of cards, owned game pieces and a dynamic game board. This game requires many things to be transferred between the private and public area since each user is getting new cards all the time and is also required to play them rather quickly. There are already several expansion sets released after the original game, so the concept itself is very extensible which would make it possible to adapt the game even more to the situation and make use of new technology.

There was a general concept idea that was based on having a game engine for card games. This would provide a way for users to freely play any card game with an arbitrary rule set just by interacting with cards in different ways. As a trivial example there could be some interaction that would have sorted the card hand by color or value, where either way is wanted depending on the type of game played. Another example of actions used by some games is to send cards to the table, which in this case would be to send cards to the tablet. This would also be possible with an interaction, for example select several cards and flick them towards the edge of the smartphone. To summarize, the basic idea here is to have all interactions available and leaving it to the users whether to use them in what game they are currently playing.

Apart from already existing games and concepts, one option was to make an entirely new game, or possibly a game based on another game. This alternative would allow for even more possibilities to adapt and tweak the game to fit the chosen medium in the sense of getting the most out of it. The downside is that much resource would have to be spent on developing the game rules instead of the game implementation.

3.2 Approach

To determine which concept to work with, it was decided to test a number of board games and look for interesting content. Perhaps something could be used in making an own game, or give useful ideas for extending an already existing game concept. The plan was therefore to play and analyze games and find features suitable for the platform so that this could be used to its full potential. Hopefully the results of this could lead to a suitable decision of a game concept that would fit the situation and make the most out of it.

After conducting this initial research and having ended up with a choice of game, the thought was to deepen the knowledge about the game and perhaps tweak the rules by using the acquired results from the game research. The aim was after all to find a game that had something special to add to the smartphone-tablet combination, and not come up with a game that just as well could be played on a desktop computer.

To explore and find out what interesting interactions could be found in the smartphone-tablet situation, it was decided to build some sort of a software prototype for testing the interaction possibilities. The initial thought was to try and just send data in the form of plain text between the devices and from that test the possibilities with interactions. It was also expected that much of the interaction would involve just one device, which of course would be useful too.

Before the implementation of the entire game, it was clear that there would be two separate applications; one for the smartphones and one for the tablet. The planning process was intended to involve a requirements specification process along with stating a list of use cases and actions possible to perform in both applications. Also, to provide some structure to the programming, the different components should have been identified to be able to construct a class diagram that could be followed by the time of the actual start of the implementation. In the same way, the user interface would require some effort in the sense of sketching, wireframing and also paper prototyping.

The methods used in the project process were intended to have an agile approach, with the requirements, use cases and designs being subject to change at all times. The process was meant to be iterative with different layers of the game being developed one after another, probably after establishing some sort of stable base version. This would involve a parallel development of the user interface and the general game engine. Preferably the two separate applications would also be developed in parallel, mostly because of the importance of being able to start testing the application more formally as soon as possible. With multiple things in progress at the same time, and probably also working at individual areas, it felt necessary to have some kind of method to share information. One approach that was looked at was the Scrum method that can be used with this kind of iterative processes. Since there were only two members of the development team, it was a bit hard to apply and instead regular discussions and daily briefings, as parts of the Scrum methods, were incorporated to the process.

3.3 Post-decision planning

As soon as the decision fell on Settlers of Catan, it was also clear that this game needed a lot of attention and hours to really get into and deepen the knowledge about. The approach to this then was to try and play the game as much as possible, and also test the other already existing digital versions of the game. The latter of the tasks was mostly done to get ideas of how to represent or implement the various game components needed to carry out the game. In the sense of changing the medium from an analog board game with real physical game pieces, cards and such to a digital platform with entirely new opportunities, it was crucial not only how to model all components in a good way but also what was possible to include other than the original contents and rules.

The idea was to start out with the original version of Settlers of Catan, but also look into the options of tweaking the game so it would fit the chosen platform. The vision was a game with special behaviors that should require this platform to function, and in that way be tailored for the specific situation. To do this the plan was to explore the several already existing versions of the game and investigate if new specific mechanics could be added to refine the uniqueness of the game but still keeping the original environmental board game experience.

3.4 Time plan

A rough time plan was initially constructed containing the expected phases of the project along with some deliverables or milestones that were supposed to be finished by certain times during the project timeline. The entire plan was divided into eight different areas where the anticipated stages at the first weeks of the project were concept discussion, interface design and platform testing. As the titles describe, this was mainly the way to determine and specify what to work with and how it should look roughly. There was also of course the need to learn about the technicalities of frameworks and techniques, but also the hardware that was supposed to be used.

After trying out the various things from the initial areas, a more detailed specification was to expect by the middle of the process where the development should be started for real. Along the implementation phase it was also planned to have the applications continually tested, and of course every significant progress during the entire project should be documented which was conveyed in the diagram by having the documentation phase spanned over all weeks. As a summarizing phase it was also intended to analyze and evaluate the project by the end of the process, which was about the time where the final milestones should be finished. The time plan is included as figure 3.1.

- Concept discussion**
Goals, contents (game), rule changes, game testing
- Interface design**
Sketching, paper prototyping, wireframing etc.
- Platform testing**
Testing devices, communication, software prototyping
- Application specification**
Requirements, use cases, class diagrams etc.

- Implementation**
Coding, unit testing, system testing
- Play testing**
Interaction testing, game experience evaluation
- Project evaluation**
Analyzing results, comparing with goals
- Documentation**
Logging, writing report

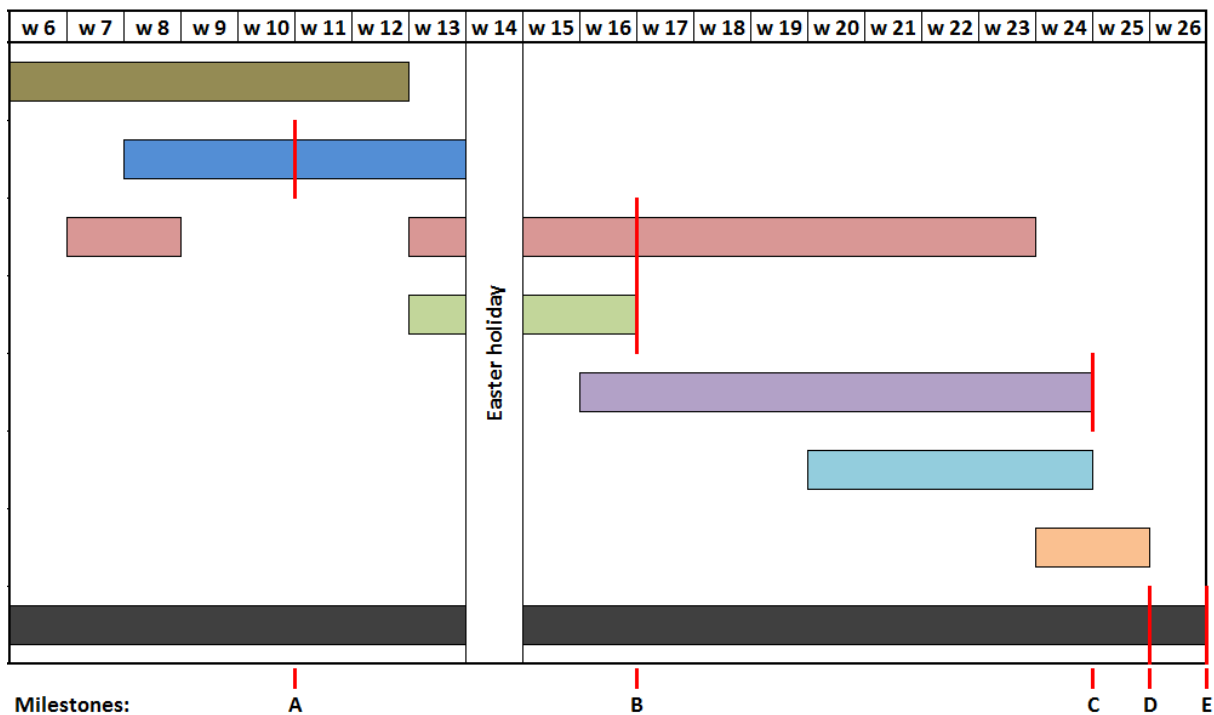


Figure 3.1: Time plan with milestones.

Deliverables and intermediate steps in the project are listed below and also included in the timetable. These sub goals were intended to assure the continuity in work along the entire project process.

| Milestone | Description |
|------------------------------------|--|
| Test application (A) | Software that allows fiddling and exploring possibilities with the medium |
| Specification of modified game (B) | Document that describes the core game with all the intended tweaks and new features |
| Phone application (C) | Software to be used on a phone by a single user, for administering private game components |
| Tablet application (C) | Software to be used on a tablet by all users, for administering public game components |
| Presentation (D) | Oral presentation of the project, including defense |
| Opposition (D) | Opposition of another group's project |
| Report (E) | Written report covering the whole project |

Table 3.1: List of milestones with associated descriptions.

4. Realization

This section contains a chronological description of how the work was carried out, from the research and decision stage to the final development phase.

4.1 Pre-development phase

In this section the preparatory work is described in terms of the different research that was made during the beginning of the project.

4.1.1 Mechanics pre-study

Provided along with the initial task that was the origin for this thesis, was a report written by Lundgren (2002). It contained a list of game mechanics that consisted of mechanisms emerging when putting a game into a digital medium. By going through this list, discussing how these would affect a potential game, the knowledge about the situation grew and thus the idea of making full use of the medium arose. As the game Settlers of Catan seemed most reasonable at the time, the thoughts were mostly focused on how the mechanics would work if they were incorporated into that game. Following is a list of Lundgren's mechanics with their definitions and the conclusions drawn at the time for how they could be used.

Active board

The board by itself generates secret information and/or collects information regarding what is going on. Sometimes it displays this information.

Usage: This could be implemented by having the tile values hidden in Settlers of Catan and then show them only to players with adjacent buildings.

Anonymous trading

Via a special reader and special tokens, players can propose simple deals they would like to make. The outcome is calculated and shown.

Usage: This could be included in the trading part of Catan, for example by putting an anonymous trading proposal that someone could accept without knowing the trading partner.

Active dice

The probability of the outcome of a die roll may be skewed or may depend on the outcome of previous rolls.

Usage: This would work together by replacing the Catan dice rolls with probabilities and in that way calculate the resource distribution result.

Active Tiles

Tiles recognize which other tiles are next to them and react to this.

Complex commodities

The price of a certain commodity relies heavily on prices of other commodities. The relations are complex and prices are calculated by the computer and then displayed.

Usage: They could work well together with resource probabilities since the location and nearby tiles could then affect the probability for a single tile.

Espionage

Players have the opportunity to get more or less accurate information about another player's resources, but may give away information regarding their own resources in the process.

Usage: This could be used in Catan to find out hidden information from other players and also possibly to detect espionage where options like sending false information, counter-espionage, or eliminating the spy would be possible.

Ubiquitous information

The game keeps track of when certain conditions are fulfilled and acts on it. It is very hard or impossible for the players to know if a condition is fulfilled or not.

Usage: This would be used just by showing information, for example how many victory points the players have in Catan.

There were also other mechanics that were not considered useful.

Computerized clues

The computer distributes clues about certain information in the game, known only by itself. At certain stages in the game all information may be revealed.

Secret bidding

Players may bid secretly on an item. Only the player who wins the bid, and the sum that is bid is displayed.

Secret partnerships

Players are divided in teams, without knowing with whom. The combined actions of a team lead to certain consequences in the game (calculated and displayed by the game itself), wherefore it is important to figure out who is on which team.

Active surface

The surface of the board changes its properties during the game.

4.1.2 Game research

After the mechanics study, a number of games were about to be tested for identification of interesting concepts or additional mechanics that could be of essence when determining a game to be working with. Some of the games being tested were chosen based upon advice from personnel at the office, and other games simply looked interesting while investigating the game stash provided by the institution. The following data is a compiled version of the comments about the games during the research test sessions.

Mammut

"Another day, another mammoth hunt. But the spoils of the hunt remains to be divided, and everyone tries to secure the largest share for themselves in this quick and clever family strategy game. Each round the tiles are shuffled in the bag and dropped onto the table. The face up symbols show the spoils of today's hunt.

The tiles are then divided by a unique mechanic: When it is your turn, you may either A) Take any number of tiles from the pool, or B) Claim that another player has been too greedy, taking all of that player's tiles, but returning at least one tile to the pool (you must of course show yourself to be a little less greedy).

The next player without tiles then follows in turn. This way the size of the pool will gradually increase, and the round is over once the last player without tiles decides to take what's left in the pool. Every player will then have a share of tiles, and a scoring phase follows.

Tactical play will help you get the meat, fur, tusks, animals, and tools you want (each tile has a different way of scoring), but you may also play cards for immediate effects or to secretly influence the scoring." (BoardGameGeek, 2012b)

The impression of this game was that it would be easy to implement the game board since it was simple. It would also be beneficial to automate the board, tiles and points since this do not require any interaction from user. However, if it was to be automated, the only thing that would be left to the user is choosing cards and tiles which means that it would not be much of a game since most of the things made in the board version would have been removed. A potential digital implementation would however preferably have the board situated on the tablet, and cards on the smartphone. The user could interact with the board to pick up tiles, and choose from the tablet or the phone depending on whether the tiles are picked from the board or from another player.

Pandemic

"Based on the premise that four diseases have broken out in the world, each threatening to wipe out a region. The game accommodates 2 to 4 players, each playing one of five possible specialists: (dispatcher, medic, scientist, researcher or operations expert). The game is unlike most boardgames as the gameplay is cooperative, rather than competitive. Through the combined effort of all the players, the goal is to discover all four cures before any of several game-losing conditions are reached." (BoardGameGeek, 2012c)

The game did not seem suitable for the environment since you cooperate and in certain modes have all the cards visible and therefore leaving no private information to be displayed on the telephone. You could very well change the availability of the information thus making it more challenging to the acquire info, but then the gameplay would be significantly changed. Possible interactions in a digital environment would be choosing an action on the tablet or the phone depending on whether a card is played or not. Pieces would be moved on the board and cards played from the phone (for travelling, building and researching), which would also be used for transferring cards between phones. The choice of cities to cure would likely be made on the board.

Macao

"At the end of the 17th century, Macao – the mysterious port city on the southern coast of China – is a Portuguese trading post in the Far East. The players take on the role of energetic and daring adventurers. Many exciting tasks and challenges await the players, whether they are a captain, governor, craftsman or scholar. Those who chose the wisest course of action and have the best overall strategy will earn the most prestige at the end.

Macao lasts twelve rounds, and in each round players select one new card from a display specific to that round, two of which were revealed at the start of the game and others that were revealed only at the start of the round. The deck of 96 cards includes all sorts of special abilities, with the more powerful actions costing more resources to put into play.

One player rolls six different-colored dice, then each player selects two of those dice (possibly the same ones chosen by opponents), then places cubes equal to the number and color of the two dice on a personalized "ship's wheel". For example, if a player chooses the blue die that shows a 5, that player places five blue cubes on the ship's wheel position five spots away from the current round. (A player can never claim more cubes than the number of remaining rounds.)

Players rotate their ship's wheels each round, then use the cubes available to them in that round to perform various actions: Activating cards selected in that round or earlier rounds; buying city quarters and collecting the goods located there; moving that player's ship around Europe to deliver those goods; acquiring gold coins; taking special actions with card previously activated; and advancing on a turn order track.

Players score points by delivering goods, paying gold coins, using the powers on their cards, and building in Macao. Whoever has the most points at the end of twelve rounds wins." (BoardGameGeek, 2012d).



Figure 4.1: Macao.

It was very tedious to prepare the game since there was a massive number of game components and many things to keep track of. The setup time was almost an hour, including reading the preparatory rules. This however would make it convenient with digital version, to remove the unnecessary setup of all components. Regarding the visibility of the cards they were publicly visible all the time but rather hard to keep track of. It was difficult even to remember what effects your own cards contributed with. In a digital version, all information would possibly not need to be displayed all the time but instead require an action to actively examine them. As for the interactions, much could be automated thus leaving just the choice of cards, action cubes and the actions themselves to the players' own interactions. A potential problem would then be that a huge amount of visualization during the actions is then needed to make sure the player's decisions are made at the right moment.

There were also some other games that were thought of, and in some cases also tested, but without significant amounts of useful results. **Scrabble**, which is about building words with different letter pieces and placing them on a board to get points and bonuses for certain rare letters, could be implemented with the espionage mechanic since it could be quite important or at least an advantage to know what letters another player possesses. The only private matter contained in the game are the letter pieces, but perhaps also the potential information from spying on other players. For example, if player A spies on player B then A gets information that B does not know that A has. Then this information itself becomes private information to player A, and would then need to be displayed somewhere as additional private data.

A game that already exists in a lot of varieties is **UNO**, a card game with unique numbered cards and special cards like "Draw 2" or "Choose color" where the player who loses all the cards first wins. In this game it could be useful to spy on other players, since that information would most likely affect what card is to be played. Other interesting possibilities was found in regular card games like the Swedish game **Kotten** or **Tecken**, which is played in two teams of two where the most unique action is to smash two piles of cards when one player in the team gets four of a kind. This would require support for hitting the card stacks to get points and also support for the somewhat fast pace in the game.

Finally, one of the two last games tried was **Hearts** at King.com, a digital variant of the regular card game *Hearts*, a game where the players compete for several rounds in not getting any hearts or the queen of spades. This digital version had very nice visualization where you can see exactly how every player interacts with their cards. The detail level is really high in where each card ends up and how the cards behave in relation to each other. The interaction is simple, with only clicks and it works fast and smooth. The last game was also a kind of card game called **Glory to Rome**. It consisted of an own area in the form of a camp that however was publicly visible and was not suitable to placed on a smartphone. Otherwise there was a clear public area where cards were discarded and picked up, but it felt rather inappropriate for the smartphone-tablet combination all-in-all.

4.1.3 Game decision and interactions

After taking the final decision to choose Settlers of Catan as the concept to work with, the research of the game started. There was however a slight problem with the project members being only two since the original board game requires a minimum player count of at least three. Although there were other existing versions that could be tested, and the first one to be examined was the *Catan* game developed by Exozet for the handheld console Nintendo DS. Playing this game gave some impressions of what was possible to achieve when putting the game into a digital environment, and also perhaps some ideas of how to visualize the central trading mechanism in the game. The importance of making sure that the user follows what is happening on the screen was also highlighted and one suggestion to improve this could for example be by adding sound to the game.

In the first play session of the original board game version Settlers of Catan, the game was played as usual but with focus on the interactions and the thoughts of how they would be performed in a digital context. A general list was produced containing the most basic and common interactions with their correspondent potential digital action, shown in table 4.1.

| Action | Interaction description |
|-----------------------|--|
| Place building | drag on touch screen to place and confirm |
| Resources | animate the resource cards OR hide the resource distribution |
| Roll dice | shake the smartphone? |
| Steal cards | drag from other player's pile of cards |
| Build | choose from menu, paid automatically, place on touch screen |
| Trade | initiate trading and choose trading partner, send cards to touch screen and accept |
| Buy development card | choose from menu, paid and moved to the hand automatically |
| Move robber | drag on touch screen and confirm |
| End turn | flick the dice on the touch screen to the next player (moves automatically) |
| Play development card | choose card from the hand, individual events |

Table 4.1: List of interactions.

From the results of the mechanics pre-study (see section 4.1.1) some ideas of how the game could be extended were then discussed. The espionage mechanic which seemed the most interesting at the time could very well be used by adding a spy function to the game. The alternatives would be to for example have it as something you buy with resources, or to have it as a development card. A thought was also to be able to spy by interacting with the robber. A trivial example of this would be to get information each time you move the robber.

As for other mechanics, it was already suggested in the mechanics pre-study to replace the resource values and thus also the dice rolls and instead use probability and let the game itself randomize which tile that would give resources. The resource distribution itself was another discussed topic, whether to keep it open and visualized or to have it hidden from the public and just visualized privately for each player in their own smartphones. The impression was that the distribution contains much information in the original setting since everyone claims their cards by grabbing the different cards from their piles beside the game board. In this way each player can, if they want, keep track of which resources the other players take. Another possibility is that a player can forget to take a resource, which can serve as some kind of penalty for not being sufficiently observant. Thus the alternative to keep the visualization open would require it to convey as much of the information as possible to preserve the feeling from the board game experience.

The other option which involved hiding the distribution would leave the players unaware of what resources are acquired by other players, which could work well with the spying action since this that would be more powerful when not knowing anything about the other players' cards. Another interesting effect would also be the importance of information gotten from the placement of settlements and cities on the board.

Since a player is informed when getting a resource from a certain tile that player also gets the information about other players getting the same resource if there are other buildings adjacent to the same tile. This of course requires the player to know which tile has been distributing resources, and also that there are any other buildings next to that certain tile.

The last of the initial ideas of how to tweak the gameplay was about giving the knight cards more significance. One suggestion was to let the knights protect the player from being robbed. An example would be to place a played knight in an owned settlement or city and by that choose which building to protect. The next player who moves the robber would then have to choose from the currently not protected buildings when deciding who to rob. In relation to the spy function, a similar thought as for the robber was to be able to use a knight in some way to prevent having a spy being played against oneself. This could for example be done by playing a knight when being spied on, thus eliminating the spy. However this would require a card being played by a player who currently does not have the turn, which may not be wanted.

4.1.4 Sketching and further testing

By getting some feeling for what possible actions could be done and also roughly how frequent they were, the illustration of the game interface for the controller started. This was at first mainly done using the wireframing method where ideas were drawn by hand on printed wireframing templates and the various ideas were compared to each other. To be able to test multiple layouts quickly and also have the possibility to switch between several alternatives, the sketches were also made digitally. Both the phone version and the tablet version were sketched and were not just facilitating in trying out different layouts for the graphical user interface, but also leading to new ideas such as modifying the game board structure. Several sketching examples are attached as figure 4.2-4.5. The idea of modifying the game board was to replace the hexagon tiles with more arbitrary generated terrain. The intention with this was to get a nicer, perhaps more realistic and random layout of the board and also make use of the computational ability to randomize this part.

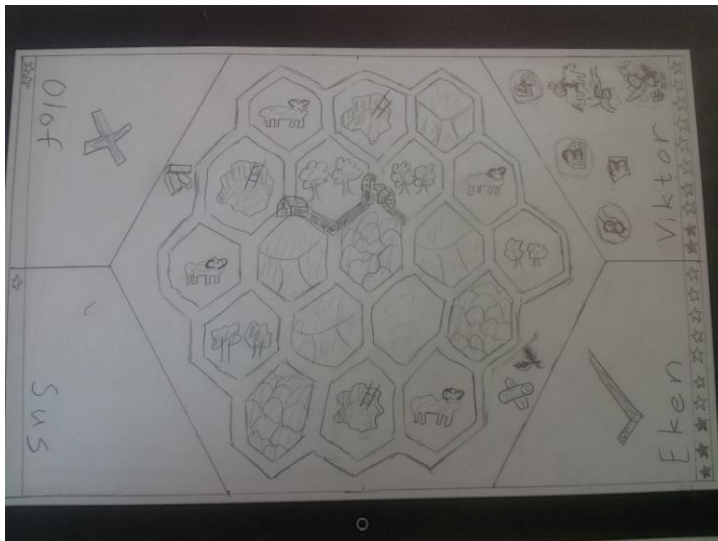


Figure 4.2: Sketch of the board application.

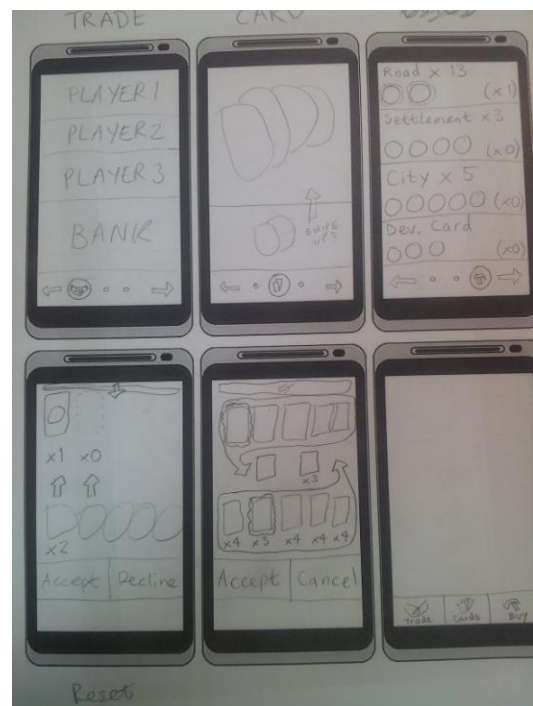


Figure 4.3: Sketch of the controller application.

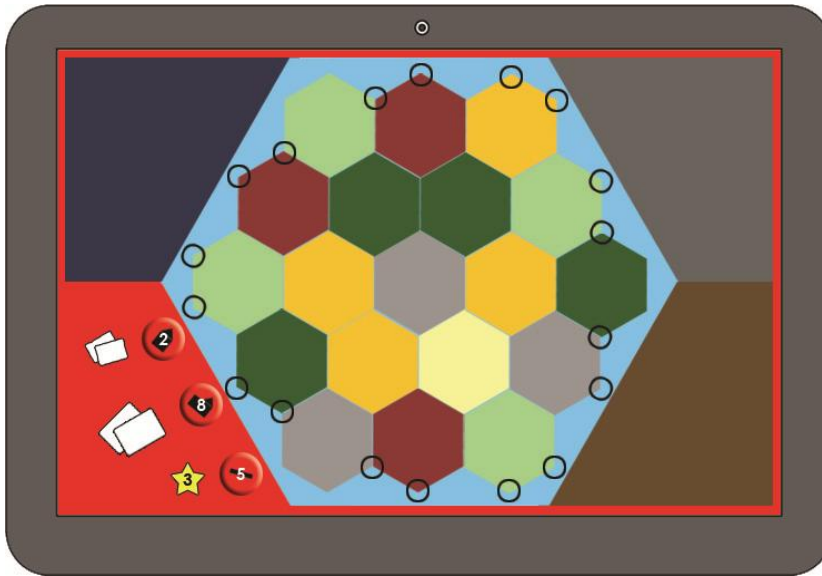


Figure 4.4: Digital sketch of the game board.



Figure 4.5: Digital sketch of the controller.

Beside the sketching there was also a need of determining more precisely what actions the user should be allowed to do, which was why a list of use cases or user actions were compiled. This list was also complemented with what responses to expect from each device after the performed actions (see appendix I and II).

Furthermore some other version of Settlers of Catan were tested, initially the online version available at PlayCatan.com and also Catan for Android. In the latter version some findings were made, for instance that the game had a good visualization of the resource distribution where each player sequentially gets the distributed resources to allow the users to keep track of what everybody gets. The development cards were displayed so that the user could see and read information about each of the different cards contained in the deck, along with some kind of button for buying a card which looked like a card itself. In general the highlighting was rather bad since it was hard to see and notice. An example of this was the visibility of having too many cards on hand, which is quite important to know. However, the trading interface felt better but lacked clearness of what is happening when a trading proposal is accepted by several players.

Later one testing session was carried out, trying to test some ideas that were recently brought up. The main thing in focus during the first custom playtesting session involved breaking the current tiles into six pieces, thus creating what could be seen as a more random terrain (see figure 4.6). With this new board layout some interesting effects were spotted, for instance sometimes a settlement cannot be placed within two road lengths from other settlements because it would then be placed at one road length from another angle.

The first placed settlement around a tile is more crucial here, since it determines how many additional settlements could be placed around the current tile and the surrounding tiles. In some cases it may be up to four, which is one more than the original version. The pointy edges of the tiles also made it possible to place a settlement and from that building get resources from up to six different tiles. In the new layout it was also sometimes required to build more roads in order to get somewhere, which could be explained by that the circumference in the non-hexagonal case was always eight instead of six. As a summary this had too many effects on gameplay, which was partly why this idea was rejected. Also, it would take a long time to implement and that time could be spent on other custom rules instead.



Figure 4.6: A test with triangle-based terrain.

4.2 Application development

The programming work was started after the sketches were done. The first part consisted of exploring Android and libraries that were planned for use. When that was done, some code was reused to start coding a test application for interaction. The testing project later became the real project. Changes that were done in the program were discussed right away and communication was important. Pair programming was also used at times to be sure that most bugs were discovered immediately, and it worked well.

To get feedback from users, a few testers were used in different stages of the program to test playing the game and trying out the applications. They were questioned for feedback and suggestions which were noted down on paper. The list was examined after tests, and some of the points brought up led to changes in the game. In the end of the project, a final test with completely new users were performed and analyzed.

The development process has been divided into four iterations following the iterative development approach. The iterations are not very well defined but they follow a pattern that improves upon an already existing program. The exploration section explains the work that was done before the actual game applications were created, and does therefore not count as an iteration.

4.2.1 Exploration

First of all, some Android tutorials were tried out to get a good understanding of the system and its libraries. In parallel, some research on specific implementations needed for this project was performed. The part that was considered most important about the project was getting the communication between the different devices working. The two valid alternatives for that seemed to be to implement a Wi-Fi or Bluetooth connection. Since Internet is not always available and people are supposed to sit close together, Bluetooth seemed to be the best choice. The two types were also very similar to implement, where the main difference was that the devices had to be paired before they could connect in a Bluetooth-based system.

An important sample project that was looked at was the BluetoothChat from the Android development team (Android Inc., 2012). It was used as a basis to create a server and client communication that suited better in this project. The main difference was that the example project had both a client and a server on both devices while the tablet only uses the server part and the phones only uses the client in this project.

Before starting to make an actual game board with hexagons, a graphical library was also needed. Since Cocos2D can be used on both iPhone OS and Android (Cocos2D for Android, 2010), it seemed to be a good system to learn. A contributing factor was also that there were related people who had used it previously and were satisfied with it. An introductory tutorial was followed in this case as well, which consisted of making a very simple game.

4.2.2 Iteration one

The initial intention was not actually to start on the real project but to try and create a test project where items could be sent between the phone and the tablet. Even so, there had to be an item to send and no reason was found not to send an item already present in the game, such as a house. Shortly the user interface for buildings that was previously sketched was also made. JSON was selected as the format to send messages between the devices. The reasons were that there was prior knowledge working with that format, and that it was lightweight and easy to use. A library called JSON.simple (2012) was imported to the project. However, the JSON-RPC format was not used. Instead, a customized protocol was created because the current situation was slightly different than a regular client-server setup. Unlike the RPC that calls functions on the server from the client and eventually returns something, it was with this system possible to make calls both ways. Data was not returned directly from the function but could be sent back using a reverse call. This system made it easier to send objects directly to the client since there was no need for the client to poll the server for information about the current state.

A board was created with a large hexagon at first. Players could connect to the game but not do anything else. The large hexagon was soon divided into several tiles, with marked positions for roads and buildings. Buildings and roads could be placed by clicking on one of these markers. At this time, the game did not care at all about whose turn it was. Resources could also be sent from the phone to the board by tapping the icons. Soon more functionality was added, such as the robber and the ability to drag resources directly from the board to a player. This feature was implemented since it was very useful when debugging. The robber is shown on the topmost hexagon in figure 4.7.

This image also displays built roads and settlements with different owners. The turn was implemented displayed in the form of a border using a unique color for each player which is also visible in figure 4.7. Player areas soon followed, colored in these player colors. The turn could be changed by tapping a player area. More constraints were also added to the game, for instance a player had to own a building and the resources of its cost in order to build one.



Figure 4.7: Example of iteration one.

When attempting to implement the dice and resource handouts, it became evident that a better structure was needed for finding relations among the tiles, edges and vertices on the board. A custom data structure was therefore implemented which better described the relation between tiles. Shortly thereafter, it was followed by a menu button that triggered a dice roll and resource distribution. When more constraints were added for building placement, a next round button was also added to separate the initiation phase from the rest of the game. Development cards were also added, and some of them could be played while others were only playable using manual input.

4.2.3 Iteration two

After feedback from testers, a second iteration was started with more focus on exploring interesting interactions. First of all, the building placement was changed from tapping to a drag and drop sequence. The idea was that the tabletop game feel would be preserved in a greater sense. A second version of the phone version was also made built on the same idea of exploring more interaction types. In this version, which will be called the portal version, all items were positioned on the same screen and whatever the player wanted to do, an item could just be dragged to the portal to send it to the game board. To better illustrate this, portals were also added on the player areas of the tablet. At this point where the interaction parts were somewhat revised, a table of possible interactions for each necessary action during a game was produced and is attached as Appendix V.

A number of game components were also polished during this iteration. The dice became visible on the screen and all the value labels were scaled and colored in a way that made them more similar to the values of the real board game. The resource icons were changed to more suitable ones, since the old ones were considered unclear by most testers. The trading was also implemented on the first version that fitted better to that user interface. Previously, players had to click on the resources to move them to the board, while they could be sent directly after this change. A bank chest was also added on the board, and the bank trade was implemented similarly to how some of the development cards worked - items could be given to the bank chest, and in return you could drag resources from the board.

There were also some new implementations. Harbors were added to the tablet board, but they did not impact the game at this moment. They are visible in figure 4.8 as arms that come out from the hexagon board. Furthermore, a victory point display was added to the phone apps. Longest road and largest army were also calculated, and soon accounted for on this display. They were also visible on the player areas in the form of a simple text. In order to better illustrate the cost of buildings on the portal version, a cost display was added for buildings that had been picked up but not yet dropped. The bank trading was also improved to take harbors into account, and a label was added to indicate the trade cost when resources were dragged over the bank chest.

A problem discovered during tests were also that multi-touch did not work, which meant several players could not trade resources at the same time. Multi-touch support was therefore added as well, and many things that relied on single-touch had to be modified in order to support this. For instance, the newly added cost labels for buildings and bank trades had to be remade to support that several items were moving simultaneously.



Figure 4.8: Example of iteration two.

A large effort was also given to refactoring the code, since the intention was not to re-implement everything that had already been done. First a list of prioritized requirements was written and a class diagram was made as a basis to use for creating the new structure. A common project used for shared files was added. New classes were also made and some code was moved from one class to another. A good improvement with this new system was that sending items between a phone and the tablet was much more generalized and therefore much easier to reuse later.

4.2.4 Iteration three

The game was nearly working fully now, but many actions had to be performed using cheats. The focus on the third iteration was to make the game fully functional and start working on some custom rule sets. One of the few missing things of the game before this iteration was the ability to steal a random resource from a player using the robber. This was implemented first of all, and in order to choose player a dialog window was added.

The dialog approach was suggested by a tester, and it seemed easy for the player since it is displayed exactly which players can be selected, and at the same time it also stops other players from cheating it. The number of resources and development cards owned by players was also added to the player area since it is important information to know when you are stealing from someone. Along with this, the new labels and old ones for road length and army size were given icons instead of just plain text. The green and yellow players' colors were also exchanged for magenta and orange in order to create a bigger contrast between the pieces and the tiles, and also between the player areas and the resources flying on top of them.

More dialog windows were worked on shortly. It was hard to find a good and logical solution for many of the remaining actions, but nearly all of them required some kind of user input to choose resources. A general choose resource dialog was then made, which could be adopted both to choose resources to get from bank trade or a year of plenty card, or to choose resources to give away when robbed. A monopoly implementation was also made where the player only choose a resource type.

Trading on the tab version was also changed so that the target could be selected directly, as planned originally. A user interface similar to the dialog interface was used for this. The harbors also had to be accounted for, and a small ratio display was added above the resources if the bank was the selected target. Since this feature was missing from the portal version, a ratio was displayed when a resource was hovering over the bank chest on the tablet.

The first custom rule to be implemented was a weather feature. It was based on a combination of *active board* and *active dice*, mentioned in section 4.1. To make it work in our game, the dice are removed altogether but the idea is similar. A weather icon was added in one of the corners as can be seen in figure 4.9, and the probability that one resource would be dealt was changed depending on the weather. Normal dice did not work for this, which meant a secondary probability display would have to be used instead of digits. This new system was based on the direct probabilities and showed a number of icons depending on how likely that tile was to produce resources. One icon could then be replaced with a red frame, indicating that the resource was missing, or another icon could be added if it was extra prosperous. This system was quickly added also to the normal mode, since it looked good and gave additional information.

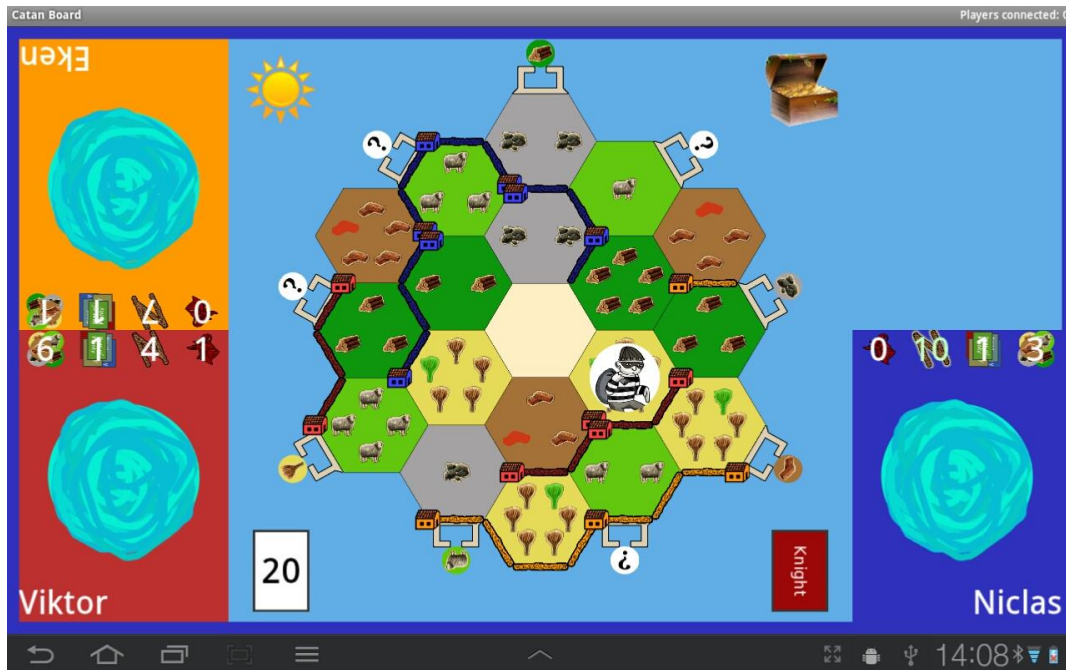


Figure 4.9: Example of iteration three.

Another feature that was added, based on a studying the game rules, was the ability to start with a predefined board, where the first settlements were already placed. This was an option also in the classic game, suggested for games with novice players. As a result of all the different modes, an options menu was created in order to be able to enable or disable them. A hidden resource distribution rule was also added to this menu, where resources handed out would originate from the bank rather than the tiles, and the icons of the resources were replaced by question marks.

4.2.5 Iteration four

The game was fully playable at this stage, but some game logic was still missing. The turn was not changed automatically, and players had to click on the next player's area to hand it over. The game was also started automatically and there was no way to restart the game without restarting the whole application. Many rules were never checked by the game engine. Other problems could occur when players performed actions in an incorrect order, or when the game had not finished animating an action.

The turn was fixed so that it automatically followed a clockwise order, other than during the second part of the initialization state where the rules state that the order is reversed. A button was added to the phones which allowed players to end their turn. The dice were thrown automatically in the beginning of every turn so players did not have to worry about that. As a result of this change, the next round and throw dice button could be removed from the game menu on the board, and the player areas did no longer need to respond to taps from players. The roundness of the board was also fixed by flipping all objects of the board so that they were facing the turn holder. This is illustrated in figure 4.10 where blue on the opposite side is the turn holder, which has turned all icons upside down.



Figure 4.10: Example of iteration four.

Following this, a starting screen was made, where the game board was replaced by a big start button. An end game feature was also added to go back to the start screen. Players that left the game were now also just considered offline if the game was started. A change was also made so that players could only join the game if it was not started or if one or more players were offline. So far, players that joined the game automatically joined on the first free position. A change was also made which enabled players to choose a corner, with its corresponding color, in a dialog that popped up as soon as a connection was established.

Some additional constraints were added to the game, both to check rules and to make sure everything was done in the correct order. A number of new checks were implemented. Examples are only one development card can be played at a time, next turn cannot be started before the robber is moved and the game will not allow any other choices when a player is choosing resources in a dialog or when resources are being animated on the board. To better illustrate this, a mind bubble was added for users that were making choices.

A few additional user interface improvements were also made. Development cards were hard to follow, as they were only showing a message on the board when played or purchased. They were fixed to animate and be visible to all users, like most other items. The robber was changed to flash instead of rotate when it could be moved, since it was more visible that way. The phones were also updated. Some buttons were grayed out on the tab version, when they could not be tapped. Similarly, items that could not be picked up on the portal version were grayed out. A turn indicator was also added on the phone since it was sometimes hard to understand when the turn changed. Finally, a vibration signal was added whenever the turn changed on the new turn holder's phone.

4.3 Playtesting sessions

The game has been tested during the project, mainly as a sort of alpha testing combined with experimental observation study where design flaws were identified and corrected afterwards. The intention has also been to test how the environment and the situation with the devices affect the overall game experience. Thus the first tests were mainly focused on putting the current version of the applications into context.

The first two official test sessions were not held until during week ten and eleven of the project process. This was mainly due to the choice of waiting with the tests until the software had been developed to a stage where most of the gameplay was finished and the game could almost fully be played. The first tests however identified quite several bugs in the program and there were also some difficulties regarding the look and feel of the game as well as the interactions. A general common thing from both tests was the impression of not being able to keep track of things, which was expected due to the lack of visually displayed information on both devices. For example the participants were not able to see the amount of cards possessed by other players, which can have some significance in certain parts of the game.



Figure 4.11: A playtesting session.

About a month later, in week 15 and 16, three additional tests were performed. With new functionality, including interfaces for choosing resources and robber victims among others, the game was once again tested to find a new set of flaws and receive both positive and negative response on new features such as the weather function. Although there was a useful discussion at the end of the three currently mentioned tests, there was no formal evaluation of the tests. This was brought up after the last tests, and the conclusion was to try to formalize and add more structure to the test, to be able to compare the two different phone apps and perhaps get some more concrete results out of the tests to be able to draw some conclusions. For instance, the tests could be validated from the project goals to verify how well they have been achieved.

Although the questions asked during the discussions at the end of the playtests were somewhat based on the goals, the results still lacked some kind of formality or structure as would be achieved by filling in a questionnaire or such. It was not until towards the end of the project that some time was actually decided to be spent on making a questionnaire and try to add some more structure to the tests by for example divide the time to test both phone applications and gather more input of comparing the versions. However the difficulties of getting test subjects at this point made the effort not as useful as was hoped.

4.3.1 Final tests

When the applications were considered final, a more formal test was performed. A questionnaire was made with both closed and open questions. The closed questions were there to make it possible to extract statistics from the data and the open ones were intended to give more concrete feedback about what was good and bad about the game. The initial question was about previous experience with the game, which was seemed relevant since that could impact the experience of this game. After that, users were asked to select if they preferred to play the board game, this version or another digital version with a motivation. Finally, they were asked for feedback concerning the custom rules. All the questions are attached as appendix VII.

The structure of the final tests was intended to be a sufficient base for evaluating and comparing the different phone versions - the tab and portal version. Thus one half of the participants was given the tab version to start with while the other half got the portal version. At about half of the game time they were asked to switch to the other version to try that as well. The idea of not letting all participants start with the same version was to eliminate the reasoning of for example that the first used version would always be the best, when comparing the versions after the test.

To also be able to get input regarding the newly added features, there was planned to be an introduction to the original game version without any added rules. This would hopefully make the testers feel a little more confident in knowing how to interact with the devices and it would also be necessary for testers who have not played the board game version before. When starting the actual play session the first rule set was decided to incorporate the 'random board' option so the players would try the placement of settlements and roads. Also the spy cards were included in this version, mainly since it does not involve so many changes to the game and does not affect gameplay significantly.

However the weather was also included in the first session which was a big change to the introduction setting since it removed the numbers on the fields and also the die rolls. Although it could have been a good thing to play one session for each new rule, the intention was to play as few sessions as possible since players tend to think strategically and does not want to start over and rethink their strategy multiple times. So the aim was to just play one session with spies and weather options, leaving the 'hidden distribution' option to briefly be shown in a short second session.

5. Results

The final results are presented in this section. First a description on how the game is played is provided. Afterwards, a more precise description of all elements of the application is presented. This includes elements that are not part of the actual game. The last part is about the system architecture used to build these applications.



Figure 5.1: The physical game setup.

5.1 Gameplay

In the beginning of the game, players can choose either to use a default setup of the map with included start positions or the little more advanced mode where everyone have to place their two first settlements by themselves. If the players use the advanced mode, a settlement will automatically be taken from the phone of the first player. The system will also mark every hexagon corner with a circle to indicate valid positions for placement. The player can then drag the settlement to any of these places and drop it, or alternatively tap directly on the marker to let the system move the settlement in position.

As soon as the settlement has been placed, a road is taken from the player too. This time, the hexagon edges connected to the settlement are marked with a road outline. Similarly, the player can drag or tap the road into position as shown in figure 5.2. After the road is placed, the turn will automatically change to the next player in clockwise order. That player uses the same procedure to place a settlement and a road, only some spots are no longer valid for placement, as indicated by the markers. When the last player is done placing the first settlement and road, the order will be reversed and the same player can immediately start placing the next two buildings. This time resources are dealt from all adjacent tiles. When all players have placed two houses and roads, the order will go back to clockwise, and the real game will start.

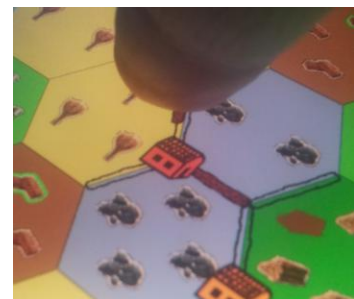


Figure 5.2: Placing a road.

In the beginning of each turn, the dice will be thrown automatically and display their result in the center of the screen. An example of this can be seen in figure 5.3. Triggered tiles will deal resources. If the value is seven, the robber is activated. Firstly, if any players have more than seven resources in total, they will be robbed. This will bring up a dialog box on the robbed players' phones that prompt the users to give away half of their resources. The players can then select resources in the dialog and the chosen resources will be sent away as soon as all choices are made. After this the turn holder will select a new tile for the robber to block and a random resource will be stolen from a player with a settlement or city adjacent to the new robber tile. If more than one player is available for robbery, a dialog to pick player will first be brought up on the robbing player's phone.



Figure 5.3: A die roll.

After the resources have been given out, the turn holder can select an action. There are two phone versions of the game. One with actions divided in tabs based on taps, and another compact version where everything is done by dragging items to a portal in the center of the screen. There are four possible items to buy and development cards can be played if they have been previously purchased. It is also possible to trade resources with other players or the bank.

In the tab version, there is a tab called *build* with a list of buildings and their cost and count. The user can tap on a building to send it to the tablet for placement. In the portal version, the user instead has to drag the building from the building pile to the portal. The placement performed in the same way as in the initiation phase. However, the game rules regarding the placement of houses and roads are now in effect. Cities are built on top of houses, and valid positions are marked with a larger circle. Development cards can also be bought, in the tab version this can be done from the *develop* tab. The portal version has a button in the top right corner. For both buildings and cards, users are unable to send items to the board if they cannot afford it or if any other reason should hinder them from building or buying.

Development cards work much like the buildings. All cards but victory point cards can be played from the development tab or by dragging a card to the portal. In either case, they are immediately played when they reach the tablet. The knight triggers the robber and increases army size count. The monopoly card brings up a dialog to pick a resource to claim. Year of plenty also opens a dialog, but this time to pick two resources. The road building card ignores the cost of the two next roads, and automatically takes one road at a time from the player to place on the board. As soon as a victory point card is acquired, the point will be given to the player.

Trading is quite different between the two versions. In order to trade with the tab version, you'll first have to select your target. The trade tab contains a list of players, and the bank is also included in the end of the list. After the player has selected trade target, the resources to give away are selected. When the player is satisfied with the choice, the send button can be tapped which will send the resources to the target. There is also a cancel button in case the user wants to undo the choice or send to another player. If the bank is the receiver, the user will be prompted to select resources in return for the trade.

Trading from the portal version is done by dragging resources to the portal. At that point they can be moved around on the board. If the player drops it on another player's portal, it will be given to that player. If it is moved over the bank chest, the trading ratio will be displayed and it can be dropped to trade. The player will then be prompted with the same bank trade dialog as used in the tab version.

When the player has finished the turn, there is a need to notify the system. To do that, there is a button on the phone to end the current turn. It is placed in the menu that is opened by the menu button.

5.1.1 Extensions

There are a few optional extensions made to game that can be enabled in the options menu. These are not options that are available in the board game, neither in the base version nor in any of the expansions. Their impacts on the gameplay will be investigated in this section.

Weather

When playing with the weather option enabled, the probabilities for resources to be given out is changed depending on the current weather. The chance that one type of resource will be given out is increased for one resource and decreased for another. Because of the way die values work it is not possible to modify the probabilities on specific resources with normal dice. Instead, one to two hexagons will be randomly selected without the use of dice. The same probabilities are kept but two tiles are never linked the same way as they are when playing with dice.

| Weather | Resource | Lumber | Bricks | Grain | Wool | Ore |
|---------|----------|--------|--------|-------|------|-----|
| Sunny | | | - | + | | |
| Rainy | | - | + | | | |
| Snowy | | | | - | | + |
| Windy | | + | | | - | |
| Cloudy | | | | | + | - |

Table 5.1: Matrix showing which resource is affected by what weather.

Hidden resource distribution

When hidden resources are enabled, players can no longer see directly which tiles are being activated to give out resources. They can, however, see that unknown resources are given out and to which players. The tile values are still visible, though. This means that you can make educated guesses on what resources other players receive. It also means that the dice have to be hidden even if the rolls are done in the normal manner.



Figure 5.4: Hidden resource distribution.

Spy cards

The spy card is a new type of development card. It can be played even in the same round as another card, and it brings up a dialog window that displays all players' resources when played, as displayed by figure 5.5. This can, for instance, be used before playing a monopoly card to see what the best resource to claim is.

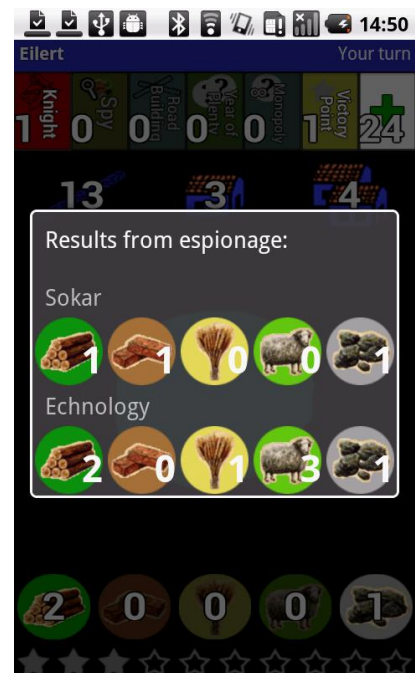


Figure 5.5: Espionage.

5.2 The applications

The game is played using two different Android applications. One is used for the tablet to act as a board and another is used in the phone to act as a personal hand. Two versions have been developed for the phones in order to see what works best. These were mentioned in the last section but more in-depth information will appear here.

5.2.1 The board application

Since the communication is done using Bluetooth, the board application will first prompt the user to turn on Bluetooth if it is not already enabled. If the user does not accept, the application will be closed with a message explaining that it cannot be used without Bluetooth. The tablet acts as server and in order for the phones to connect to the tablet, the phones need to be paired with the tablet. If they are not, they can only find the tablet if it is discoverable. Therefore, there exists a button in the menu to make the tablet discoverable for five minutes.



Figure 5.6: Final game board.

Players can connect to the game as soon as Bluetooth is enabled, provided they are paired. There are four slots for players, and each slot has its own area in a corner of the screen. Each slot is also associated with a color that is used to paint the area. Each area is half transparent until a player connects. That will give the area full opacity and the text will change from "(Empty)" to the name of the player. All states are shown in figure 5.7. The center of the screen is occupied by a large start-button that can be used to start the game. There is also an option menu that can be opened up from the menu button menu. The options menu contains four checkboxes. One box can be checked to start the game with a random map, where the initial settlement placement is done by the users. The other three are used for the extended rules; weather, hidden resource distribution and spy cards.



Figure 5.7: Different states of the player area.

When the game is started, all unused player areas will be removed. Additionally, the existing ones will be equipped with a portal and four information labels. The labels display the total number of resources and the total number of development cards along with the longest connected road and the current army size (amount of knight cards played). The latter two use green text and are highlighted when that player has the longest road or largest army, which is exemplified in figure 5.8. The resource count text turns red if the player has more than seven resources and is eligible for robbery. New players can no longer connect, but if a player goes offline, either by leaving voluntarily or by being disconnected by some other means, any player can reconnect on that slot. Offline players are marked using the same half transparency that was used for unused slots before the game was started, with the label "(Offline)".



Figure 5.8: Highlighted labels on player area.

If no one plan to replace an offline player, it is possible to kick him from the game completely, from a submenu under the menu button. Kicking a player will remove the player area and no items will be given to that player anymore even if that player's buildings are still on the board. The kicked player will never receive the turn. Offline players, on the other hand, are counted as any other player. The only difference is they cannot do anything until a player has reconnected in their place. The game can also be ended before it is finished, using the back button. It is also doable by tapping the end game button accessed using the menu button. This will return the game to the initial state and all slots are once again available.

It is always possible to see whose turn it is by looking at a border inside the screen. It is colored in the current player's color. All buildings owned by that player also use the same color. When a turn is started, the dice will be shown in the center of the screen if they are used. The value labels of the matching tiles will be encircled to indicate that it was triggered to give out resources. The circles are visible even when the weather is used, despite that the labels themselves are not shown. Naturally, they are not visible in the hidden resource distribution mode, as the whole point of it is to make the resource types secret. If any houses or cities are nearby and resources are given out to them, resource icons will fly from the tile to the receiving player's portal in the corresponding player area. There is a short delay between every resource in order to keep them from overlapping.

If the robber is triggered and there are players with more than seven resources, a mind bubble will be shown on those players' areas. When they have selected which resources to give away, the bubble will be removed. When all players are done, or if no player had more than seven resources, the robber will start flashing to indicate that it has to be moved. It can then be dragged on the board and it will snap to the center of a tile it is dropped on. If it is dropped outside all tiles or on the tile it originated from, it will just snap back to the starting position and continue blinking. As soon as it is moved to a valid location, the robber will stop blinking. At this point a resource will be stolen from a player if possible. If there is only one opponent with resources to steal, it will be automatically chosen and a resource will fly from that opponent's portal to the turn holder. Since the resource is secret to other players, the resource will not use its normal icon, but an icon with a question mark in the center. If there are more choices to make, the player selects an opponent on the phone and the mind bubble is brought up on the selecting player's portal meanwhile.

When a player plays a road, settlement or city, it will appear on that player's portal. Markers will also show up to mark available positions for placement. These are removed if the player returns the building. If the user chooses to drag it to position, it will follow the finger around until it is dropped. If it is a road, it will be rotated every time it is moved passed an edge to fit the angle of the edge. When it is dropped in a valid position, it will snap in place. In case of an incorrect placement, the building will be moved back instantly to the portal. If the building is placed by a tap, the building will move quickly towards its position and snap by itself. Houses that are replaced by cities will be moved to towards and through the portal back to the player. If by some reason a player is not eligible to play a building, it will be turned back in a similar manner. Normally, this is prevented already before it is possible to place the building.

Development cards are stacked in a pile in a corner of the hexagon map. When a player buys a development card, it will fly towards and through that player's portal with only the back side visible. When a card is played, it will come back with its front side visible and it will be moved to a discard pile in another corner. Knight cards trigger the robber and cause it to start blinking. The same procedure is done to move the robber and steal a resource as explained previously. Year of plenty cards and monopoly cards trigger a user choice, and the mind bubble is therefore shown on the player's portal. The year of plenty chosen resources will then appear on the bank chest located in a third corner, completely visible to everyone, and moved to the player. All resources claimed by a monopoly card will fly from the other players' portals to the one who played the card. Victory point cards cannot be played, and played spy cards do not display anything on the board.

If the player is using the tab version in the phone, then resources will fly from one portal to another when they are being traded. Alternatively, they can be moved toward the bank chest if the bank is targeted. In that case, the mind bubble will show up on the player's portal. When the selections are done the chosen resources will fly from the chest back to the portal. In the other version, resources are just placed on the board and they can be dragged to another portal or to the chest. If they are dragged above the chest, the ratio will appear as a label following the finger around as long as it stays on top of the chest. Bank trading this way also brings up the mind bubble and the chosen resources will fly to the portal in the same way. If a player wants to cancel a trade, the resource can simply be moved back to the player's own portal.

5.2.2 The hand applications

There are as mentioned two versions of the hand application. One version is more focused on the core Android user interface, and everything is done by simple taps. The other one is more compact and most actions are performed by dragging items to a central portal. They do, however, have some things in common.

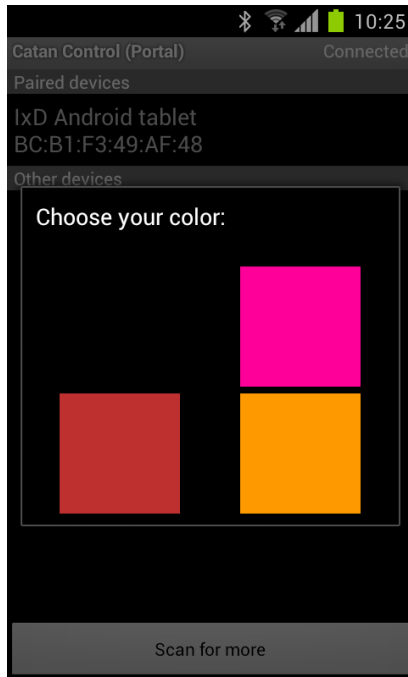


Figure 5.9: Connecting to the game.

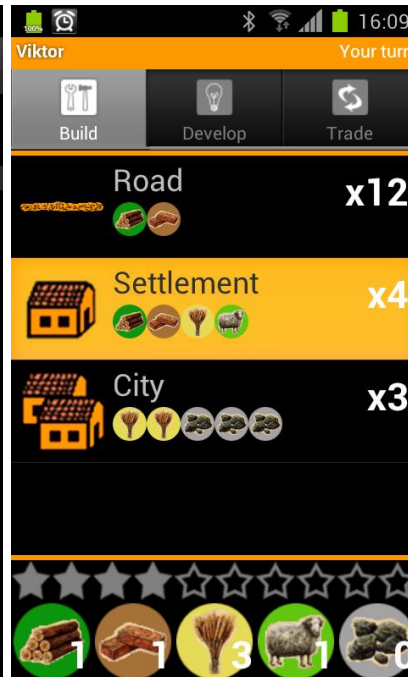


Figure 5.10: Final controller (tab version).

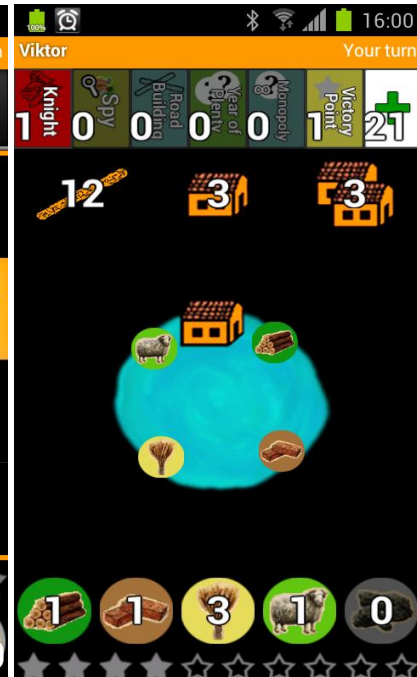


Figure 5.11: Final controller (portal version).

Firstly, connecting to the game works exactly the same for both versions. A list of paired Bluetooth devices will appear on startup. Only devices that are classified as laptops are shown to filter the list as much as possible. Ideally, you would want to show only tablets but they share classification with laptop computers so that is not possible. If a tablet is not already paired, there exists a button in the end of the list to search for new devices. Only devices that are flagged as discoverable can be found so the discoverability needs to be turned on for the tablet first. If a new device is tapped in the list, a pairing request will show up on both the phone and the tablet. Accepting this request will pair the devices and start a connection. A connection will also be started if an already paired device is tapped. If the connection is unsuccessful, for instance because the device is not in Bluetooth range or the game is not started, an error message will be displayed. Otherwise, a dialog will show up with a four button grid. Each button has the player color of the corresponding corner. Corners that are already taken by other players will have their button removed, and the dialog will update whenever a change is made to the available slots. This is exemplified in figure 5.9. If the game is started, only slots with offline players are available. Tapping on a button will connect the phone to the game on that slot and bring up the specific user interface for the used version of the phone application.

When connected to the game, most actions are performed in different ways on the different versions. However, the menu button triggers a menu that is the same on both versions. It contains two buttons; one used to end the turn and one used to synchronize the phone with the tablet. The end turn button is used to notify the system that the player has finished all actions for this turn, and is ready to initiate the turn for the next player. The synchronization button will update the count of resources, buildings, development cards to the values stored on the tablet, which acts server. Since the tablet also keeps track of choices that are to be made by players, it could also open a dialog on the phone when synchronized. Ideally, this button should never be used and it will probably be removed in a final version. However, it is currently a useful button in case something goes wrong.

Dialogs that appear during the game are also the same for both applications. They are shown at any point the user is asked for specific input. When choosing a player to rob, the dialog will display a list of players that are available for robbery. Tapping on a player will trigger the robbery action on the tablet. The dialog for selecting monopoly resource will show one icon for every resource, and a title telling the player to claim a resource. The resource will be claimed just by simply tapping on it, and that also closes the dialog. The dialog to choose resources from bank or year of plenty is nearly identical. The title tells the player to choose resources, but when a resource is tapped a number prefixed with a plus will show up telling how many of that type is selected. When all choices are made, the dialog will close down. Similarly, when choosing resources to give away when robbed, the same view will be shown, but this time the numbers will be prefixed with a minus to indicate that you will lose them. This is illustrated in figure 5.12.

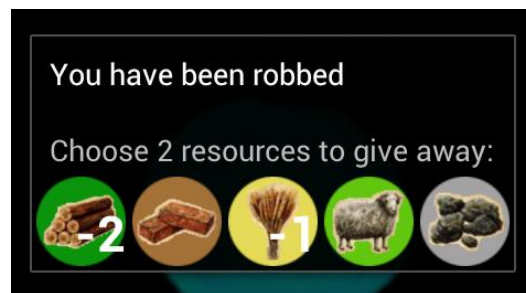


Figure 5.12: Robbing dialog.

There are also a number of error messages that can appear in the phones. All error messages are triggered by the tablet and any items played will be returned in conjunction with the message. Users attempting to play something before the game is started will be informed with a message telling the player to wait for the game to start. If a player tries to build a building or buy a development card without affording it, the game will complain about insufficient resources. This applies even if the resources are taken first when the building is actually placed. If there is no position to place a building, another error will be returned. Some actions are also blocked until players are finished making choices on their phones, and when the robber is triggered to move. Many actions, such as end turn, are also hindered with an error message if the game is busy with a trade or by dealing out resources. Finally, there are also some error messages that should never appear that could be triggered a phone is not synchronized with the tablet.

5.2.3 Tab version specifics

This version of the hand application is based around a tab container in the top of the screen for different types of actions. The first tab is used to construct buildings and roads. The second is used to buy and to play development cards. Lastly, there is a tab dedicated to trading. Since resources are a very important part of the game, they are positioned in the bottom of the screen and are visible from any tab. The victory points are also visible from anywhere. They are shown in the form of ten stars. Unearned victory points only show an outline, while earned points are filled following an order of left to right.

The build tab in figure 5.13 uses a list of buildings to construct, and each list item can be tapped in order to send the building to the tablet for construction. Each building has a small frame with the icon in the player color. There is also a title in every frame telling exactly what kind of building it represents. The cost for each building is also displayed below the title, with a line of resource icons. If more than one resource of the same type is used, several icons of that resource are displayed. Lastly, there is a number on the right side of the frame, indicating how many buildings of that building that the user possesses. The top of the develop tab in figure 5.14 uses the same user interface. There is a button to buy a development card with a development card icon, a title and resource cost, along with number of development cards remaining in the pile. The actual cards are placed in a horizontal line below. It is scrollable and the card count is displayed in the bottom right corner. These cards can be tapped to be played.

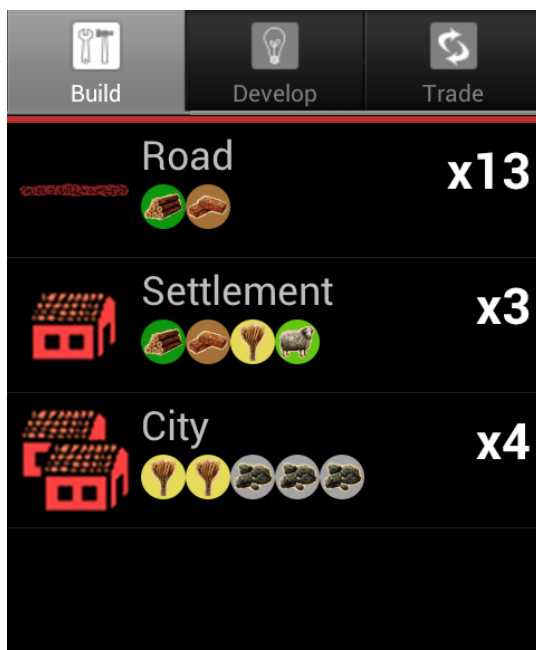


Figure 5.13: Build tab.

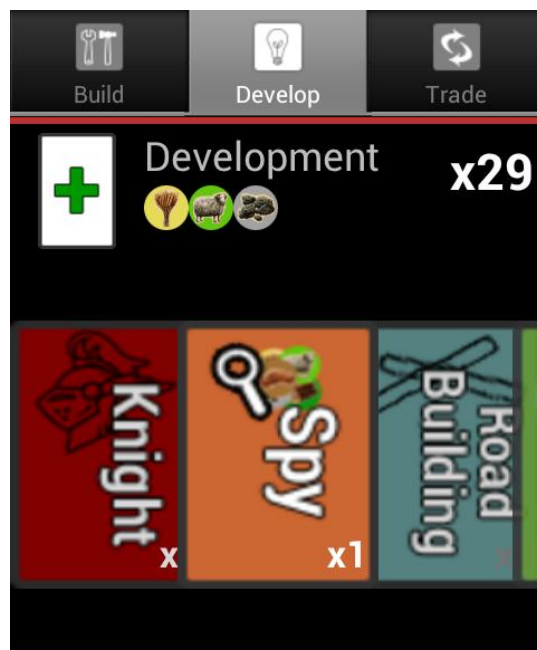


Figure 5.14: Develop tab.

The last tab is the trade tab. The default view here is to see a list of players and the bank. All players are colored in their respective players colors and the bank is black. Tapping on either list item will bring up another view for trade with that target. A copy of the resource list that is used on all tabs is shown in the center here. However, the number with the count is initiated as zero on all resources. Tapping on a resource in the general resource view in the bottom will transfer it to the central resources. In essence, this means that the count is decreased by one in the bottom resources and increased by one in the central view.

Similarly, a resource can be given back to the bottom by tapping one of the central resources. If the bank is the target, each resource also has the bank trade ratio, given by owned harbors, displayed above its icon. The resources are also transferred in groups of size of this ratio. There are two buttons below the central resource icons; a green button labeled "Send", and a red button labeled "Cancel". When tapped, the send button will transfer all selected resources to the targeted player or bank. The cancel button will return all resources to the player. Both buttons will also close the trade menu and return back to the list of trade targets.

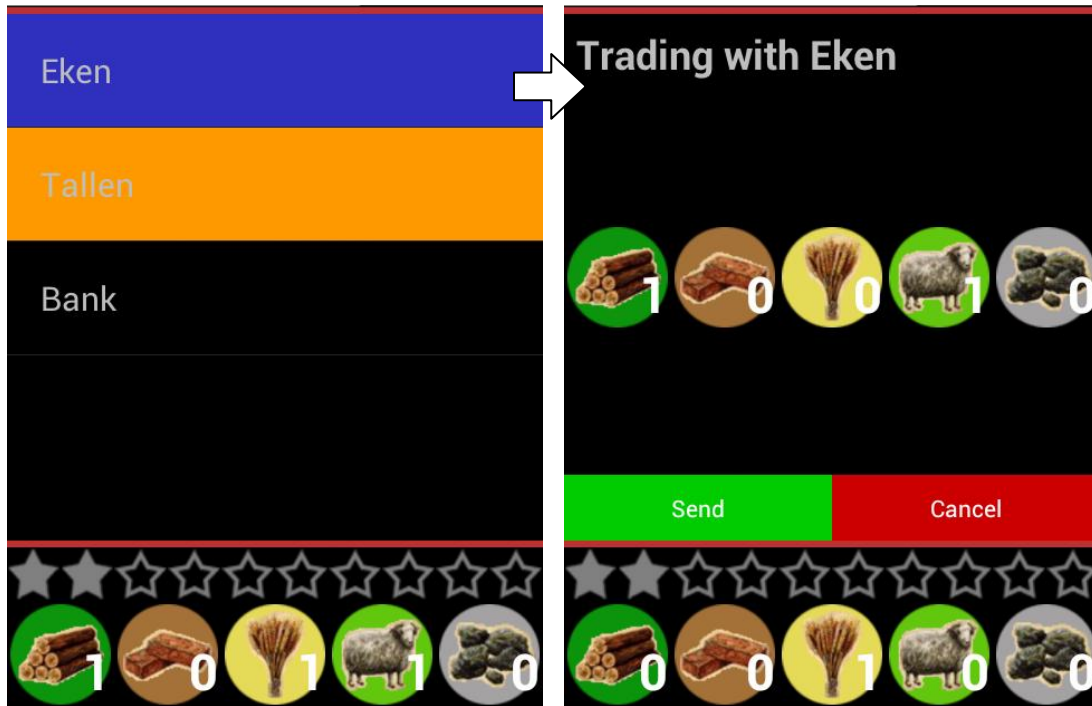


Figure 5.15: Trade tab.

5.2.4 Portal version specifics

The portal version is more compact, but also less informative. Most actions are performed using drag and drop instead of tapping in this version. The center of the screen is occupied with a large portal, using the same image as the portals used in the board application. The upper part of the screen is reserved for buildings and development cards while the lower is taken up by resources and victory points. The points are using the same stars as in the tab version, except they are positioned below the resources this time, as they are one of the few elements that cannot be dragged to the portal. All other items are represented by an icon with the count attached as a label. If the count is less than or equals to zero, the icon will be half transparent which also indicates that it cannot be dragged. A negative count is represented with a minus sign without number, and it can happen in rare occasions that will be explained later. An exception to this rule is the victory point card that is unplayable and therefore it is also not possible to pick it up even the icon has full opacity.

In this version, all items can be played by dragging them to the portal, buildings and development cards alike. However, there is one exception; the development cards are bought just by tapping on a button located as the rightmost element in the development card row. When a building or the development button is touched, the resource cost will be displayed as small resource icons that are rotating around the portal. If several items with cost are touched at the same time, the total cost of them all will be displayed. Trading is done by dragging resources directly to the portal, and then handled by the board application. If the intent is to trade with the bank, only one resource need to be taken, since the rest will follow when the resource is dropped on the bank chest on the tablet.

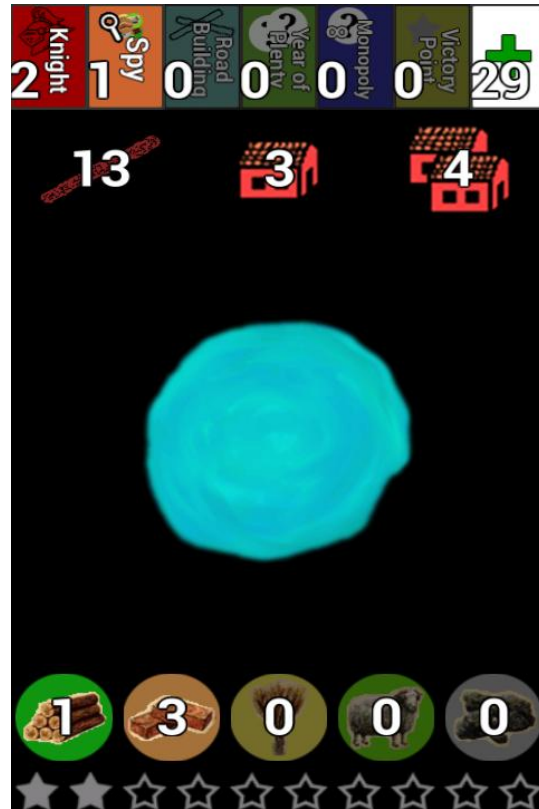


Figure 5.16: Portal screen.

Resources that are taken by force by the tablet are not just immediately removed as they are in the tab version. Instead, they will fly to the portal automatically, much the same way as when resources are dealt out on the tablet. Incoming items are also animated, but they appear on the portal and fly toward their position. This applies also to such items that are returned in conjunction with error messages. If an arriving resource is used to buy something before it has reached its target position, the system will temporarily mark the resource as negative. However, as soon as the received resource has reached its destination, the count will be zero again.

5.3 System architecture

The system is divided into four projects. There is one version for every application, and a shared library project used by all of them, which will be referred to as the common project. The final class diagram is provided in appendix IV.

5.3.1 The common library

Even if the game is divided into several applications, many things are shared between them. That is why there exists a shared project that is used as a library. Even if it is defined as a library, most classes are not generic enough to be usable in other programs. A few classes are there to be used for both phone versions. Others are used to manage data that is used both on the phones and the tablet. There is also a receiver interface that is used by classes that can parse the JSON messages sent between the phones and the tablet.

It also includes common resources that are libraries. Most notably, the Cocos2D library is used on the tablet and the portal version of the phone, and the JSON library is used to write and parse JSON strings.

Item

This class works like an enumeration class. It has a set of public instances that can be used directly for comparisons. There are also a number of subclasses contained as inner classes. It is not possible to extend a traditional enumeration, which is why it is defined as a separate class. The subclasses are called *Resource*, *Tile*, *Building* and *Development* and they have their own sub sets of items. The base class also has a few values on its own. These values are a representation of each subclass. A method can be called on each subclass to find which item type it is part of. For instance, calling `getItemType()` on `Resource.LUMBER` will return `Item.RESOURCE`. This class is used mostly for comparisons, and used to keep track of items that are moving on the screen.

ItemSprite

ItemSprite extends *CCSprite*, which is part of the Cocos2D library. Naturally, it is not used on the tab version of the phone application, since it does not use Cocos2D. It is used to easier keep track of items that are moved around on the screen. It works like a normal sprite and loads an image file that can be transformed and moved around on the screen. Additionally, it has a reference to an *Item* and possibly also an owner. It is, therefore, much easier to find out what it represents when it reappears in another place of the code.

PlayerController

This is a very simple class to define a player. It contains a name, and a MAC address. The address is used as a key to differentiate between players, and it needs to be provided in order to send data over Bluetooth. The name is the name that is written on labels in the game to identify the player.

PlayerData

This is the class that keeps track of how many resources, buildings, and development cards are owned by a player and much more. Other things include length of longest road, army size, and selections that are supposed to be made by the player. An instance of this class exists on both the phone and the tablet. The tablet version is considered the correct one, and it is for instance used to determine if a player has enough resource to buy a building. It is also used to display the resource count visible on the player areas in the game. The phone uses it primarily to update the view correctly.

The reason why this class is not merged with the *PlayerController* is that this class keeps track of items per game slot, while the other class is used to keep track of actual players. If a player goes offline *PlayerData* pays no notice, it still contains all the stuff that is owned by the player. When a player comes back online on that slot, a synchronization will be requested, and the data on the phone will be updated even if it is a different player.

Receiver

This is an interface that is used by classes that should be able to receive messages from the Bluetooth connection. Basically, this means nearly all classes in this project extending *Activity* from the Android library. The *Server* and *Client* classes require a reference to call for updates on when they receive a message from the other device. No inherent activity action is used for this and a method called `receiveMessage()` is therefore defined to take care of this, which takes an object of JSON data as input.

Client

This class is used on the phones to establish a connection with a *Server* class and in that way link the phone with a tablet. It contains methods to connect and disconnect, and it can send messages to the server. Classes with the implemented *Receiver* interface can also register themselves as listeners that should be notified when the client receives a message from the server.

The client contains two subclasses that extend *Thread*. They are running separate threads to avoid the program from locking when it is waiting for messages. The first one is called *ConnectThread* and it is used to establish a new connection. When it receives a signal, the thread will be closed and replaced with an instance of the second class, *ConnectedThread*, which also includes streams that can write and read data to and from the server.

ConnectActivity

This activity class is using a user interface that can be used to find and connect Bluetooth devices. It is extended in both phones, but the only difference between them is that they load a project-specific class.

ControllerActivity

This is the main class for the phones. Since the user interface of the phones are very different from each other, this class focuses mostly on taking care of input from the tablet and then delegating it to the functions implemented in the subclasses. Even so, it does contain code that is shared between both applications, such as how to draw the dialog windows.

5.3.2 The phone project (tabs)

This section will discuss the classes that are used solely by the tab version of the phone application. This project uses the core Android user interface.

ConnectActivity

This is an extension of the *ConnectActivity* in the common project. The only difference is that it is defined to open the *ControllerActivity* of this project when the player enters the game.

ControllerActivity

This class extends the *ControllerActivity* defined in the common project. It is designated to draw the common parts of the graphical user interface, including the tab host and the resource view and victory points located in the bottom part of the screen. Other than that, it stores all data that is used by the different activities that are tied to a specific tab.

BuildActivity

This is the class responsible for the building tab. It contains the *ListView* that is used to draw the building buttons.

DevelopmentActivity

The tab that takes care of development cards is implemented in this class. It contains two lists, one vertical with only one item used to buy development cards, and a horizontal scrollable one with all cards.

TradeActivity

There are two views in this class that are exchanged when the player selects which target to trade with, and when a trade is completed or canceled. The initial one contains a *ListView* with all opponents and the bank, and the second view contains an additional representation of resources. The code to transfer items between the two resource bars is also contained in this class.

5.3.3 The phone project (portal)

The portal version is built upon the Cocos2D library, and nearly everything that is specific to this version is implemented in the *GameHand* class.

ConnectActivity

This class extends the *ConnectActivity* class provided by the common project. It is only used to initialize the right *ControllerActivity*.

ControllerActivity

Its super class is *ControllerActivity* in the common project. It is mostly delegating version specific calls to the *GameHand* class. It also sets up a view for that class to use.

GameHand

This class extends *CCLayer* that is used as a background to add other items on. All items, such as buildings and development cards are drawn on top of this layer. The *GameHand* also takes care of touch events and moving items on the screen. Items dropped on the portal will trigger an event that is handled by the *ControllerActivity*.

5.3.4 The tablet project

This is, much similar to the portal version, built on the *Cocos2D* library. It is also the largest project, both because it has to take care of the game board which is a much larger part of the game than each hand and because it is a server that has to keep track of all the resources of individual players too.

Server

This class is very similar to the *Client* class, but it is designed to receive a connection request rather than initiating a connection itself. To do this, the *ConnectThread* is replaced with an *AcceptThread*. This thread is listening for connections, and as soon as it receives one, it will start a *ConnectedThread*, with input and output streams. All of these threads will be running simultaneously and they are stored in a map, using each connection's MAC address as key.

Options

This is a simple class to store settings. It can be used in the code to check if an option is enabled, and contains methods to read and write options to and from the memory of the phone.

BoardActivity

This is the main class for the tablet. It implements the interface *Receiver* which allows the possibility to receive messages from the server and delegate them to different methods. The whole view is reserved for an *OpenGLView* that displays a game board. Most game data is also stored in this class.

GameBoard

GameBoard is an extension to the Cocos2D class *CCColorLayer*. It is visible in the form of a light blue layer, which is also responsible for drawing the board and everything that is on it. The class is mainly designated to draw stuff and move items around, and it takes care of touch events in order to let users pick up and tap on certain elements. Most of these items are instances of *ItemSprite*. However, there are also several callback methods that are run after an action such as when a movement has been finished.

TileMap

This is an interface that is used to manage a tile map. It exists in order to retrieve a map of all the tiles, edges and vertices that are created using the *Hexagon* class. There are functions to return all tiles, all edges or all vertices. It is also possible to find a tile directly using a coordinate.

Hexagon

This class defines a hexagon tile and its position, but it is much more than that. It contains an inner class called *HexMap* which implements *TileMap*, and is a data structure. Tiles are added using a static method in the *Hexagon* class, and it is possible to either create a new root or add one as a neighbor to another tile. A map is automatically expanded when more items are added and it can then be returned in the form of a *TileMap* using the `getMap()` method.

In addition to the *HexMap*, there are two inner classes called *Edge* and *Vertex* that represent edges and corners on a tile. These can be retrieved using get methods on an instance of *Hexagon*. It is possible either to retrieve a list of edges on a tile or to specify which side is wanted. Each edge also contains two vertices defined as left or right. Since all classes have access to the tile map, it is also possible to find all hexagon tiles that are connected to a specified edge or vertex.

PlayerArea

The player areas that are located in the corners of the screen are drawn using this class. It extends the Cocos2D class *CCNode*, which is a class that represents a position, and it can be scaled and rotated. This is a class purely used as a view, and the text element that is used to show the name of a player has to be updated using a `setLabel(text)` method. Similarly, the values for resource count, development card count, road length and army size has to be updated from outside.

ProsperityMeter

This class also extends *CCNode*. It is only a representation of how prosperous a tile is. It shows a number of resources depending on the probability that a tile will be triggered. It is also used to indicate which resources have been improved or worsened depending on the weather.

6. Discussion

The project will be discussed in this section. Firstly, the objectives that were stated will be compared to the results. Afterwards, the time plan and the gameplay will be analyzed and the phone versions will be compared. Future work that can be made to explore this medium further will also be discussed along with improvements to the implemented game.

6.1 Project goals

The project will in this section be compared to the goals that were stated in the initial plan. Much of this content will be based on questionnaire answers from the final test along with the results of the various game test sessions and group discussions with test subjects during the test sessions.

6.1.1 Interaction possibilities

"Which interesting interaction possibilities can be used through combining several smartphones with a tablet?"

It is not possible for the tablet to locate the position of the phone without external tools, which eliminates a lot of possibilities for a simple game. What did work were the portals used to transfer items between the phones, which seemed to be popular among the testers. None of them had any problems understanding it as soon as they understood that the image represented a portal. The intention with using the portals was mainly to create a visual representation of sending items between a phone and the tablet. Probably this is also enhancing the understandability of what is happening in the game since a lot of items are flying between the devices.

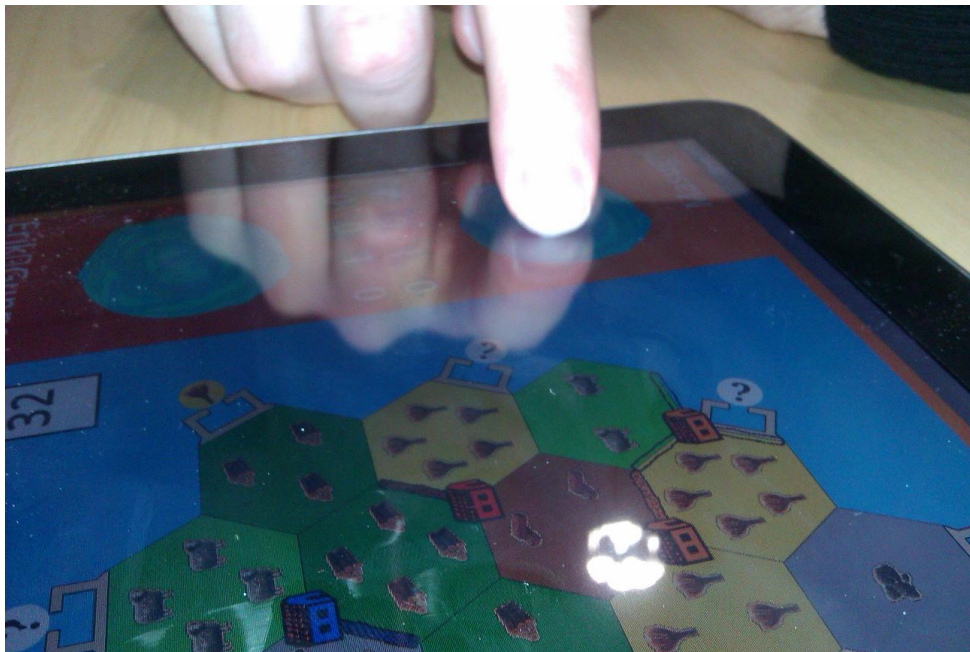


Figure 6.1: Drag and drop interface.

The general impression was also that the dragging actions provided by the portal version of the phone app led to a funnier interaction with the device which can be seen as quite similar to the board game where pieces are grabbed and lifted from a player's own area onto the game board. Since most elements are animated on the tablet, animations on the phones also make the game more consistent. Another feature that was tested was the vibrator of the phone. It was a useful way to notify people that it was their time to make a move, which was a problem in earlier versions of the game.

6.1.2 Game experience mediation

"How can the board game experience be mediated with only digital game components displayed on a tablet and smartphones?"

Since it is hard to measure the experience when playing games, the significance here was much dependent on how the game was perceived by those who got to try the game and compare it to the board variant. Even so, what seemed most logical was the use of drag and drop for placement. As mentioned in the previous section this type of interaction is typically good for mediating the management of game pieces in a board game. Placing a house which is normally done by grabbing the house, lifting it and placing it on the board, is in this way done by pressing and dragging the house to the intended place.

Another thing was to focus on that users should actually talk with each other. The game should not be designed like a game played over the Internet. Instead, the fact that people are sitting together should be utilized. In Settlers of Catan, talking is mainly used for trade purposes. The trading in the other played Catan games required the users to input both what they wanted, and what they were willing to give away, and then all trade participants had to accept or decline and possibly make new offers. This was a complicated procedure that took away part of the game experience. When everyone can talk to each other, the social aspect is kept which is very positive for game experience. Additionally, the trade use case could be simplified since it was no longer necessary to select which resources to give away since the trade partner was always present and could make changes to the trading proposal, much like it works in the tabletop version.

This could also be applied to other games, such as Dragon's Gold (BoardGameGeek.com, 2012e). In that game people have to negotiate about the loot of dragons in a limited time. This would work with normal negotiation and some kind of selection feature in the platform with tablets and mobile phones, while a digital game would more or less be required to change the rules since even a chat is too slow to use for negotiation.

6.1.3 Utilizing features

"Which new exciting features can be added to utilize the digital media for improving the game experience?"

Only a few rules have been tested in this game, but generally there are a few types of additions that utilize the medium well. These are mechanics that uses the computing capabilities of a computer to either calculate complex relations between game elements or to generate more advanced randomness than normal dice. It is also possible to store hidden information and selectively choose players that have access to this information.

The initial plan with exploring the field of tablets and smartphones in combination was to find a way to make use of new possibilities within this medium. This resulted in the digital representation of the game with some additional features showing how the medium was utilized. Three features were implemented and it would of course have been interesting to attempt even more but time constraints prevented that. In general the feedback about custom features was positive. All of the additional rules were liked by a majority which is very positive since they sometimes target a smaller group of players such as expert players only.

Lundgren's computer augmented mechanics was examined and a combination of the *active board* and *active dice* mechanics in her report was implemented in the form of weather. The die rolls were changed depending on the weather and the chance of the weather changing was increased in every turn in a way that would be extremely difficult to solve in the board game. The general impression based on the final tests was that people thought it was a nice addition but it did not affect their gameplay. What they did like was that it added another level of randomness, which in turn meant another excitement factor. One person compared it to how it feels when you have more than seven cards at hand and you are hoping that the robber will not be triggered. Similarly, he was hoping that the weather would not change. The persons who disliked it claimed that it was just another unnecessary and weird random factor that just complicated the game.

The *Computerized clues* mechanic was also implemented in mode where the resource distribution was hidden. The clues were that players could see who got resources but not which resources. Based on that and the fact that resources were coming with short delays on cities, it was possible to find out which resources were given most of the time. Some players were able to calculate what kind of resources were given out nearly all the time, which gave them an advantage over people that cared less or were bad at keeping track of them. It did however not change the game as much as expected. Most likely this is due to that keeping track of resources is not something most users do anyway. Although a majority of the final game testers thought this feature was good as an option, the other players had stronger opinions. The players who were positive to this feature had very different reasons. For example, one player liked to find out which resources were given out, and another player liked that he did not have to care about what others players received and could focus on himself. The dislikers thought it gave them extra work to find out what they otherwise knew, and it removed the aspect of coziness from playing the game. Overall, it is definitely something that could be added as an option in a game for more advanced users to add an extra layer of complexity.

Parts of the *espionage* feature were also added to the game with the spy development card. There is, however, no way to detect a spying attempt and then be able to counter-spy or send false information as initially intended. A more advanced version was tested on the board game but it seemed to be an underpowered and too complex feature. Therefore, the simpler version was added. It would be possible to implement also in the normal game by asking everyone to show their hands but perhaps it was more practical on the phone since the resources were listed in a table that was easier to understand. It would also be possible to hide the spy card when it is played, which would remove the ability to have it in a non-digital game. In the current implementation, the spy card could be very powerful in conjunction with a monopoly card. It could also be used before a knight card to decide who to steal from. Even so it was deemed as very situational and often people saved it for a good moment that never came up. The final test results indicated that the concept was good since people claimed they liked it, but few people had actually used the card. The card was probably too situational for this game, but perhaps a more advanced version of espionage could be fun and interesting in another game, such as a role-playing game where the rules are generally much more complex and players are more accustomed to that.

6.2 Time plan

The time plan was held time-wise but the order and length of the different activities was changed. The concept discussion was finished early, already on the third week. The initial plan was to start game testing with other people to discover features, but it was harder than expected to get people to join. The first session of interface design in the time plan was also ended in the third week, but this time according to the plan. It was of course possible to investigate different alternatives and do more detailed designs but it did not feel like it would have been well spent time, and there was already another session planned at a later time.

The platform testing was perhaps started a bit later than planned, on Friday week three. Since the initial test sessions were excluded, it was not a problem. It was harder than expected to draw the line between testing and implementation, as much of the game was implemented for test purposes as a software prototype. The development continued longer than planned before the applications specification with requirements and a class diagram was started. They were, on the other hand completed in less than one week. Part of the reason why it was made so quickly was that the game rules were quite clear, which meant it was very easy to set up the requirements.

The code was also started already, in the form of the software prototype. It could be used as a base on how to structure the real program. The work on this program was started in working week eleven, in the form of a refactoring of the software prototype. Even before that, the game was working better than expected at the time, and the playtesting sessions begun already in week ten. However, it continued all the way to the last week, in order to perform some final testing sessions that were never planned initially.

The interface design that was supposed to be done in parallel with the coding was to a large extent skipped. Most of the application was already implemented based on the initial designs and there was no need to spend additional time on it unless large changes were made. One such large change was when the portal version of the phone application was added. A few alternatives were sketched based on the idea of a drag and drop interface for the mobile phone. Other, smaller interface changes were just discussed and implemented directly. An example of this was the dialog windows that was explained in words, and only changed slightly after its initial implementation.

During the whole process, a log was made to keep track of the progress and to take notes on discussed topics. The final report was not started until working week seventeen, but much of its planned content was already noted down in the log. The final evaluation was also on time. Even so, it felt more as if it belonged to the documentation, so it could have been removed from the time plan. Table 6.2 shows the roughly calculated time in hours as an estimation of the amount of time spent on each part of the project.

| Part | Time (hours) per member |
|------------------------------|--------------------------------|
| Planning | 25 |
| Game research | 25 |
| Sketching | 35 |
| Playtesting | 45 |
| Learning (tutorials etc.) | 95 |
| Building test application | 110 |
| Implementation | 270 |
| Documentation (reports etc.) | 165 |
| Presentation | 30 |
| Hours in total | 810 |

Table 6.1: Time estimation for each project part.

6.3 Gameplay

As the resulting game shows, it is absolutely playable. Every rule has been implemented in the final version, except for being able to play a knight card before starting your turn. This makes it fully possible to play the digital version on somewhat equal premises as the original board version. Though feedback from testers still says that the board game of course is the "real deal", but surely there are some observed advantages with the tablet and smartphone version. Almost all of the final testers have played the board game before and could compare the different settings with the result of the board game being the best option. However it felt more appealing to play with a smartphone and a tablet than playing another digital version. It is clear that both the social aspect of sitting together and the distinction between public and private elements is necessary for the game experience to be as good as possible.

The game is implemented with very low fault tolerance, which means that the board game still can be played more freely thus allowing to undo actions and even cheat. Of course this has both advantages and disadvantages in the sense of that it is much harder to cheat, since the tablet acting as game board contains all true information and is able to correct a mobile device that contains faulty information which may have been acquired by a bug or cheating in some way. On the other hand the limitation of not being able to correct an undesired action can be quite frustrating, especially if the action was unintentional. For example, when placing a settlement, the building can sometimes unintentionally be dropped at the board and not be possible to move after that. This may be a larger disturbance than to for instance regretting a bought item, which has been intentionally bought. There were plans of supporting an undo action, or at least make it possible to move a building after placement. Here the latter option seemed to be the best since it would be less interruptive for the user, but none of the ideas were sufficiently prioritized to actually be implemented in the game.

An approach taken with the game was to minimize excise (Cooper, 2003) in the game. That is usually a good thing, but as pointed out by the Tisch team (Hartelius, Frölander & Björk, 2012), it is by some people considered part of the fun to throw the dice on your own. Others just think it takes up time that could be better spent elsewhere. Currently, users are not able to throw dice manually, but part of the fun is of course removed when no physical object is used anyway. An idea that was considered was to allow people to shake the phone and make a drop-movement with it on the board, similarly to how dice are thrown. However, it was not possible to detect the relative positions of the devices and just shaking it would not be the same. Even so, it is possible to construct buildings using drag and drop interaction, which resembles the act of moving pieces on a physical game. The lazier users who just want to play the game can just tap their buildings into position. It was seen from the testing sessions that some people preferred to drag their buildings even if they knew that they could tap.

A consequence of the current implementation of the game is that many things are automated. From the study by Wallace et al. (2012) the different aspects of automation was looked at. While it often facilitated for many users, it was also observed that it could confuse users and leave them unaware of occurring events. This was also a significant observation in this project.

As mentioned previously both the die rolling and building placement in some cases, are two things that are more or less automated. In the die example the current user is relieved from performing an active task to roll the dice. In return the user has to be aware that this action is performed by the system in order to keep up with what is happening. Since the user tests showed that this was not the case in most times, the vibration feature was considered as a support for alerting a user at the time of the event.

Another example of automation is the dealing and paying of resources, where a resource will be automatically given to or taken from a player without the player being required to make any effort. It is another illustration of where a user does not need to spend time on an action that may be considered unnecessary as the game is handling all rules. Besides the option of wanting players to be able to miss the chance of taking resources, the most important thing in this event is to notify a player who acquires or pays a resource since that is the point which can cause confusion if the player is not aware of it. The main solution would be the animation of resources primarily on the tablet where resources are flying from the source of the resource to its destination. This was supported differently by the phone versions since one of them supported animations similar to the tablet and the other did not. Probably that was an important factor of the ability to see what was happening in the form of actual movement on the screen.

6.4 Phone version comparison

As previously mentioned, there are two phone versions. Their differences, advantages and disadvantages will be discussed here along with feedback from players. The tab version with the divided user interface allows more space for information, as everything does not need to fit in the same space. For instance the prices for buying objects were visible directly on the build screen. In the portal version, one first had to touch the object to see the price. However everything could be seen on the same screen in that version, so that it was possible to see the development cards and the buildings at the same time.

The final tests revealed that the portal version was most popular. It was favored by 50% of the testers, while 33% preferred the tab version. The remaining persons did not favor one version over another. The main points to why people preferred the portal version was that it was considered more fun with drag and drop, and that it was good to see everything on the same screen. Most people still considered the tab version easier to understand and use, and that was also the main point to why some people favored that one.

There also seemed to be a correlation between previous experience and which version they preferred. Less experienced users seemed to like the tab version to a larger extent, likely because it was easier to use. Even so, the difference was not big enough with this test sample to draw any conclusion about this, especially not considering that the most experienced player also preferred the tab version. Some people also argued about the trading mechanics. Unlike the other version, the tab version did not require players to use the tablet for trading. This has less to do with the actual interaction methods on the devices, but many people disliked having to drag the resources around on the tablet. Most likely the portal version would have been even more popular if people were able to select trade targets directly from the phone.

6.5 Future work

There are many improvements that can be done in this subject. Firstly, it would be interesting to see what one can do with other games and if any other interaction possibilities can be discovered. There are also many improvements that can be made to this project, both of research purposes and convenience improvements for players using the application.

6.5.1 Other implementations

Even if more can be done with the *Settlers of Catan* application, many other implementations of this system with tablets and mobile phones can be explored. A reusable library would be a good start. The classes *Client* and *Server* could likely be used as a base for this, and possibly also the *ConnectActivity*, but most other classes are very specific for this game. A more general approach for sending objects that are identical on the other device would be good to have as well.

With a more generalized library, it would be easier to implement more games. A possible suggestion is the card game engine which was discussed in section 3.1. It is an appropriate choice since it is very different from *Settlers of Catan*, but still a feasible application for this medium. Other board games could also be worked on to see if any undiscovered opportunities exist.

6.5.2 Research in *Settlers of Catan*

There is more research to do also in this game. There are more interaction possibilities to be investigated but also things to discover by testing additional rules. Currently, most interaction is done by dragging and dropping items and tapping buttons. Other movements such as flicking and pinching could be investigated, but it is difficult to find good uses for it without making things more complicated than they have to be. Something that can be implemented in the game that is not possible in a board game is zooming. For this purpose, pinching and spreading is a good choice, and in order to move a zoomed screen, a flick movement could be used. Rotation is also an option to use more advanced interactions. A rotation movement with two fingers could be used for this. These interactions are very standardized and perhaps not very interesting but it is difficult to find other situations where they are suitable. Shaking is also not tested; the initial plan was to use it for throwing dice.

Other things that could be investigated are new game rules. Only a few of the proposed rule modifications are implemented currently. Anonymous trading and a more randomized terrain not based on hexagons could be interesting to investigate. A rule set using complex commodities, where the probability that a tile will give out resources depends on nearby tiles could perhaps also be used. The weather could also make a bigger impact on the game. For instance, it could change the bank trading ratios, so that resources impacted negatively by the weather will be more valuable and favorable resources require larger quantities to be traded with.

6.5.3 Application improvements

The focus of the game was not to make an outstanding game application, and there is also much that can be done to improve it. These points are mainly consisting of things that would make the application more user friendly, and thus more enjoyable for new players. However, things such as sound and graphics will also be discussed. The proposed suggestions from the previous section could of course also be used to improve the game but here are some additional suggestions.

Many of our testers considered it hard to follow what was happening all the time. The movement speed of items could be reduced, but if it is reduced too much the game will feel slow, so it is a hard balance. There are other things that can be done, which probably have a larger impact. For instance, the dice that are displayed in the middle of the screen could shrink instead of being removed, and move to the active player's area until the next turn starts. Similarly, the weather that is currently visible in a corner could be shown with a large icon in the center of the screen whenever it is changed, and then shrink and move to its normal position. Resources are a little trickier as they are supposed to be secret after pickup, but an idea that may work is to show small representations of resources that have recently been given to a player on that player's area for a few seconds. Recent resources could also be marked on the phone to inform the user of the last events.

Another aspect that needs to be improved is the visualization. An idea that has been considered is adding a border to resources that are moved on the board. This border would use the player color so it would be easy to see who owns which resource in a trade. Neutral resources would, in this case, use a black border or no border at all. It can at times be difficult to understand which items can be picked up, and where items can be dropped. Drop areas could be highlighted or marked when they are valid targets. This is already true with building and road positions, but it could be extended also to the portals in each player area and on the phones. An additional improvement would be if the marker was changed when a dragged item is positioned correctly but not yet dropped.

The screen that is shown on the phones before the game is started is also a good candidate for change. Currently, it looks like the game is started except the player owns zero buildings, zero resources and zero development cards. A completely different screen should be shown which makes it clear that the game has not yet started. Depending on how this screen will look, the initial phase where users place out their first two settlements could be included in this.

There is currently no sound in the game. Sound could be added to more or less every action in the game to make it more interesting. The digital *Catan* game from Exozet Games, has different sounds for different resources, and when a wool resource is given out it sounds like a sheep. Perhaps more subtle sounds are better for this game. Nearly all graphics could also be improved. The portal has been mentioned many times and some testers did not even understand that it was a portal. A real graphics artist would be more suitable for creating the graphics, provided all graphics follow the same style.

Other features that are missing include saving the game, so that players are able to continue at a later time. Players should also be able to undo actions. The plan for this was to allow them to continue moving stuff until another action had been made, but perhaps a normal undo button or some other implementation would have worked better. A tutorial or help feature with information on how to play the game would also be very useful. A description available in the game about what the development cards do has also been requested. This could be implemented with a long tap on a development card that opens up a description dialog.

There are also a number of other things about the user interface that can be improved, which could be found with more tests aimed at the user interface. Two issues that have been brought up already are that buying development cards is inconsistent in that they are bought by tapping when everything else is done using drag and drop in the portal version, and that it is illogical that only one resource should be dragged when trading with the bank. The latter could be fixed by allowing the system to take resources that are already present on the board when it takes the extra resources used for bank trades.

7. Conclusion

The idea of combining a tablet with several mobile phones has been successful. Bluetooth has worked well for this purpose but it seems that the Android system for Bluetooth is a bit unstable and can fail to create a connection at times. Hopefully, this will be fixed in future Android phones. The concept works better than initial expectations of an alternative to the board game where people would like it only if the board game was unavailable. About one third of the testers in the final test actually prefer this version out of all the available versions of the game. This is very good considering that the remaining two thirds all prefer the board game.

The advantages mentioned about this version was that much information was displayed that have to be calculated manually in the board game, such as the longest road and the amount of victory points one has. It is also quicker since many things are handled automatically in combination with shorter distances required to grab pieces and cards. The fact that the tablet is small and in one piece rather than with lots of small pieces is practical since it can be carried around and played anywhere. However the small size can also make it hard to see what is happening. Part of the board game feeling with physical pieces is also lost, even if the social aspect is still there. Another disadvantage is that the game requires several devices and electricity for all of them. Players also like being able to tweak the game rules in their own ways, but are unable to when the rules are set. This point could perhaps be avoided if the game was implemented in another way, and does not have to be considered a disadvantage of this medium.

The limitations of working less on information visualization, graphics and sound was a good choice since it saved much time. However, it was apparent that some users initially had problems understanding the user interface, and with greater visual feedback it would have been more apparent what to do. There have been no complaints about the lack of sound effects. This could either have to do with a too small test sample or because people are not used to sound effects when playing board games. This could potentially also be an advantage of playing a digital version of the game.

The interaction possibility that has mainly been explored is the use of dragging and dropping items on a portal, which can be moved further on the other device. Another useful feature is the use of signaling the user with a vibration. Other than that, it was difficult to invent new interactions as the devices were unable to find the relative position of each other. The board game experience can to a great extent be mediated using drag and drop, which is similar to moving pieces in a board game but it is not as good as physical contact with pieces. Also, the fact that people can talk with each other can be utilized to create a better experience. Interesting new rules using computer calculations and hidden information can be added to a game with success, but much work has to be put into balancing and tweaking these additions to fit in a specific game. Likely, it would be more reasonable to make a new game that is originally developed with these rules in mind.

References

- Android Inc. (2012) Samples. *Android Developers*.
<http://developer.android.com/tools/samples/index.html> (28 Jun. 2012).
- Bakker et al. (2007) *Weathergods: tangible interaction in a digital tabletop game*. TEI conference: Conference proceedings. February 15-17, 2007, Baton Rouge, Louisiana, USA.
- Beck et al. (2001) Manifesto for Agile Software Development.
<http://agilemanifesto.org/> (19 Jun. 2012).
- BoardGameGeek (2012a) The Settlers of Catan. *BoardGameGeek.com*.
<http://boardgamegeek.com/boardgame/13/the-settlers-of-catan/> (19 Jun. 2012).
- BoardGameGeek (2012b) Mammut. *BoardGameGeek.com*.
<http://boardgamegeek.com/boardgame/95613/mammut> (19 Jun. 2012)
- BoardGameGeek (2012c) Pandemic. *BoardGameGeek.com*.
<http://boardgamegeek.com/boardgame/30549/pandemic> (19 Jun. 2012)
- BoardGameGeek (2012d) Macao. *BoardGameGeek.com*.
<http://boardgamegeek.com/boardgame/55670/macao> (19 Jun. 2012)
- BoardGameGeek (2012e) Dragon's Gold. *BoardGameGeek.com*.
<http://boardgamegeek.com/boardgame/1042/dragons-gold> (25 Jun. 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).
- Cocos2D for Android. (2010) cocos2d-android-1. *Google Code*.
<http://code.google.com/p/cocos2d-android-1/> (28 Jun. 2012)
- Cooper, A. & Reimann, R. (2003). *About Face 2.0: The Essentials of Interaction Design*, Wiley; 1st edition.
- Copeland, L. (2001) Extreme Programming. *ComputerWorld*. December 2001.
http://www.computerworld.com/s/article/66192/Extreme_Programming?taxonomyId=063 (20 Jun. 2012).
- Hartelius, U., Fröhlander, J. & Björk, S. (2012) *Tisch - Digital Tools Supporting Board Games*. FDG conference: Conference proceedings. May 29-June 1, 2012, Raleigh, NC, USA.
- Johansson, J. & Lundberg, P. (2011) *Tabula Imaginarium - An Ipad application for aiding a spatially separated tabletop role-playing game*. Göteborg: Chalmers University of Technology. (Master's thesis in Department of Computer Science and Engineering).

JSON.simple. (2012) json-simple. *Google Code*.
<http://code.google.com/p/json-simple/> (28 Jun. 2012)

Katzakis et al. (2011) Smartphone Game Controller. *ITE Technical Report*, vol. 35, issue 25, ss. 55-60. Osaka: ITE.

Kauko, J & Häkkinen, J. (2010) *Shared-Screen Social Gaming with Portable Devices*. In ACE conference 2010, Lisbon, Portugal.

Lamb, L. (2010) A New Frontier for "The Settlers of Catan". *Vectorform*. August 2010.
<http://blog.vectorform.com/2010/08/02/a-new-frontier-for-the-settlers-of-catan/> (25 Jun. 2012)

Lepore, T. (2010) Sketches and Wireframes and Prototypes! Oh My! Creating Your Own Magical Wizard Experience. *UXmatters*. May 2010.
<http://www.uxmatters.com/mt/archives/2010/05/sketches-and-wireframes-and-prototypes-oh-my-creating-your-own-magical-wizard-experience.php> (21 Jun. 2010)

Lundgren, S. (2002) *Joining bits and pieces - how to make Entirely New Board Games using Embedded Computer Technology*. Göteborg: Chalmers University of Technology. (Master's thesis in Department of Computing Science).

Peitz, J., Björk, S. & Jäppinen, A. (2006) *Wizard's Apprentice; gameplay-oriented design of a computer-augmented board game*. In ACE conference 2006, Hollywood, CA, USA.

Snyder, C. (2003) *Paper Prototyping: the fast and easy way to design and refine user interfaces*. Morgan Kaufmann Publishers; 1st edition.

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

Tse et al. (2007) Multimodal Multiplayer Tabletop Gaming. *Computers in Entertainment*, vol 5, issue 2. ACM, New York, NY, USA.

Wallace et al. (2012) *Exploring Automation in Digital Tabletop Board Games*. CSCW conference: Conference proceedings. February 11-15, 2012, Seattle, Washington, USA.

Appendix I: User actions

User actions:

- Join game
 - Select game to join (phone)
- Place settlement/city/road
 - Drag from building icon in player area (tablet)
 - Drop at highlighted positions on game board (tablet)
- (Roll dice)
 - Shake phone? (phone)
- Buy building/card
 - Select from buying menu (phone)
- Trade with bank/players
 - Select trading partner (phone)
 - Select resources to give away (phone)
 - Bank: Select resources to take (tablet/phone)
 - See current offer (tablet)
 - (Reset current offer) (phone)
 - Accept/decline trade offer (phone)
- Play development card
 - Select resources to take (*Year of Plenty* card) (tablet/phone)
- Move robber
 - Drag from robber's current position (tablet)
 - Drop at highlighted positions on game board (tablet)
- Draw card from opponent
 - Select cards (tablet/phone)
- Discard cards
 - Select cards (phone)
- (End turn)
 - Flick dice onto game board (tablet)

Appendix II: Device actions

Tablet actions:

- Inform player about turn
- Indicate where to place settlement/city/road
- (Animate dice)
- Pay/give resources (slide)
- Highlight current building
- Give development card
- Show currently chosen cards involved in a trade
- Trade resources (slide)
- Highlight robber
- Indicate where to place robber
- (Show cards to take)
 - Player cards face-down (robbing)
 - Bank cards face-up (*Year of Plenty* card)
- Move dice to next player

Phone actions:

- Show which menu is currently displayed
- Show cards menu
 - Display resource/development cards
- Show buying menu
 - Display which resources are required/possessed
 - Display the amounts of remaining buildings/cards
 - (Display the amounts of buildings/cards currently allowed to be purchased)
- Show trading menu
 - Display available trade partners
 - Display amounts of owned/offered cards
- (Show cards to take)
 - Player cards face-down (robbing)
 - Bank cards face-up (*Year of Plenty* card)

Appendix III: Requirements

Functional Requirements

Priority 1 - should be contained

- Connect to game
- Place building on board
- Change player (end turn)
- Get resources from bank
- Throw dice (begin turn)
- Buy building
- Buy development card
- Play development card
- Trade resources with player
- Trade resources with bank
- Find harbors owned by player
- Move robber
- Choose player to rob
- Give resources to bank
- Choose resources from bank
- Choose resource type
- Give resources to players
- Count victory points for each player
- Charge resources
- Find longest road
- Find largest army
- Restart game
- (+New rules)

Priority 2 - would be contained

- Change player name
- Change player color
- Choose player position
- Zoom board
- Pan board (if zoomed)
- Rotate board

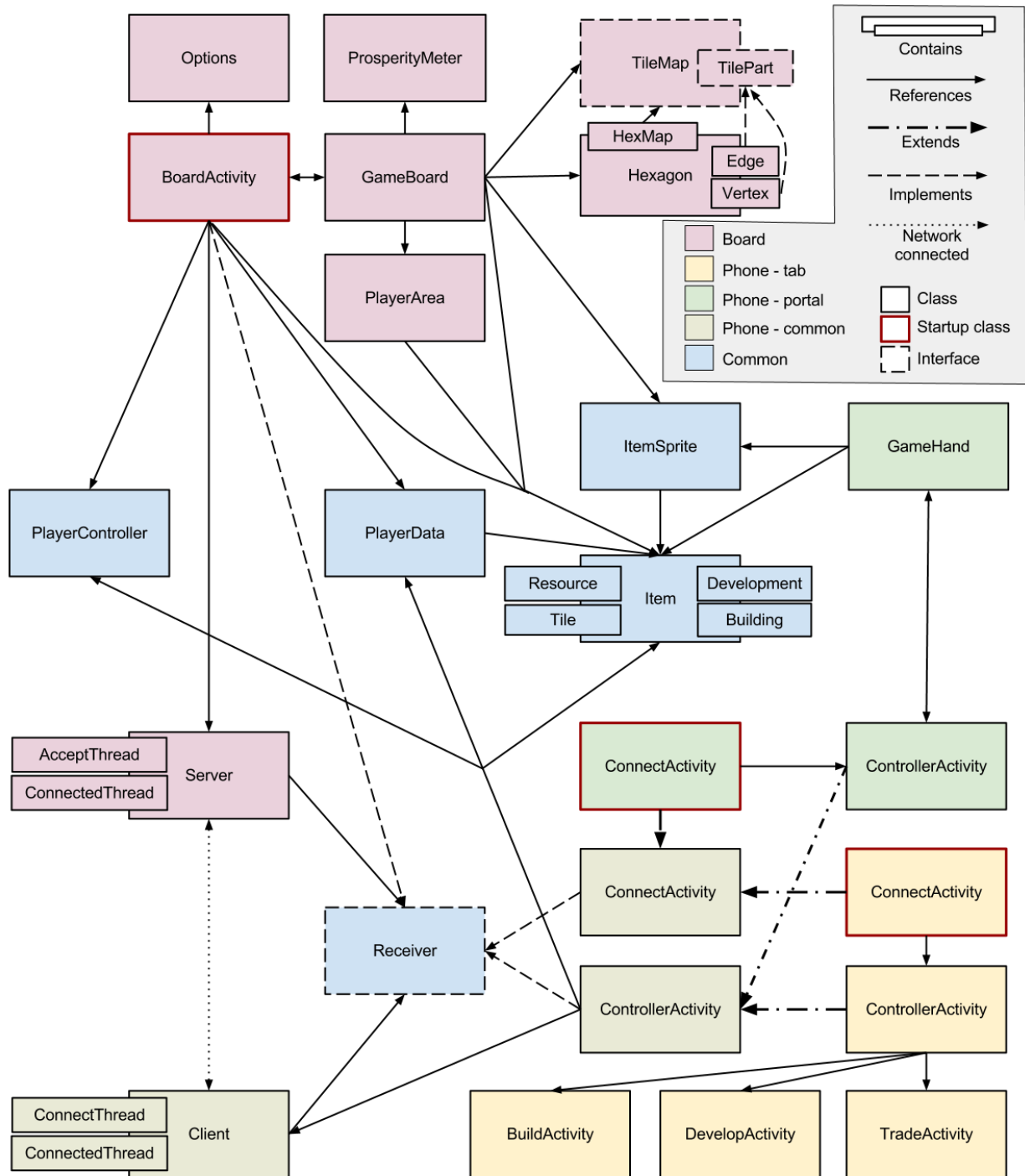
Priority 3 - could be contained

- Options menu
- Game statistics
- Other rule sets to choose from

Non-Functional Requirements

Could be run on android 2.2 (phone) and android 3.0 (tablet)

Appendix IV: Class diagram



Appendix V: Jones table of interactions

■ Used on phone ■ Used on tablet

| User actions | Interaction 1 | Interaction 2 | Interaction 3 | Interaction 4 |
|----------------------------|------------------------------|---|--------------------------|------------------|
| Place building | Click on position | Drag to position | Tilt to position | |
| Change player (end turn) | Click on button | Pinch (close) portal | Drag dice to next player | |
| Get resources from bank | Auto | Drag from bank | Drag from tile | |
| Throw dice (begin turn) | Auto | Click on button | Shake phone | Drag (roll) dice |
| Buy building | Click on button | Drag to portal | | |
| Buy development card | Click on button | Drag contract to portal | Drag card from board | |
| Play development card | Click on button | Drag card to portal | | |
| Play Knight | [Move robber] | | | |
| Play Road building | [Place road] x 2 | | | |
| Play Year of Plenty | [Choose resources from bank] | | | |
| Play Monopoly | [Choose resource type] | | | |
| Trade | Select target | Drag to portal and drag to target | | |
| Trade with player | Click on resources/accept | Accept | | |
| Trade with bank | Click on resources/accept | [Give resources to bank] + [Choose resources from bank] | | |
| Move robber | Click on tile | Drag to tile | Drag to part of tile | |
| Choose player to rob | Click on player area | Click on building | Auto | Click on button |
| Give resources to bank | Click on resources | Drag to bank area | | |
| Choose resources from bank | Click on button | Drag from bank area | Click on spinners/accept | Drag from tile |
| Choose resource type | Click on button | Click on tile | | |

Appendix VI: Playtest notes

Playtest #1 (120417)

Choose side around the board?

"Everything is upside down"

The house is lost, just a number

Numbers are upside down

Not lock the phone screen

Give back resources on failure

The dots on the number pieces are missing, red numbers

Choose robber victim: Press the building or automatically, choose on the telephone

Needs to be clearer whose turn it is!

Summary:

- Think more about the "roundness" of the board
- Error handling ("drop" buildings, give back resources etc.)
- Clarify who's turn it is

Playtest #2 (120426)

Move house after placement!

Help button

Border unclear

Save dice rolls

Press a card to get info

Confirm quit?

Unclear that a card is bought by pressing "+"

See number behind robber? Press for transparency

Implement trade on the phone tab app

Highlight latest event

Highlight marked building

Mark turn on the telephone

Roll dice on the phone

Move robber and choose on the phone

Show how many resources was put in the bank, show popup upon taking resource

Saved stats

Discussion:

Problem with updating on ZTE-BLADE phone?

Helping info - how many points everything is worth

Save on the tablet

End turn button, or close the portal

Enter name on your own telephone

More...

Event history

Playtest #3 (120524)

Enlarge the touch area for roads!

What is happening?! (Things happen too quick...)

Have to be able to trade resources simultaneously!

Sheep look to be disabled when boosted

Who got resources? Leave the marker, or rather lighten up the tile during the whole turn

Is it my turn? Show that the turn is active

Resource number is wrong

Amount of development cards left only updates for the player who bought the last card

Discussion:

More convenient to play digitally

Good animations, good visualization

Portal color related to player color?

Portal unclear graphically

No need to count points

Animated change in weather - what does the weather mean?

Mark clearer when something happens (especially robber!) - Phone vibration?

Exponential velocity of moving things?

Trade sack - move several resources at once

Playtest #4 (120525)

Houses are upside down (resources and numbers too?)

The mystical disappearance of the markers!

4 [sheep] to 1 [?]?

Ending dialog

Tornado - all resources are re-distributed

Weather - can take into consideration when the robber is placed (less useful for players late in the round)

When trading with the bank the resources are first sent on the board, then removed in the telephone (after the choice)

Clear visualization of trading and buying

Discussion:

Easy to miss trading process when it was not required dragging the resources (Tab version)

Good feedback - good highlighting

Not necessary with highlighting of resources

Highlight robber

Weather-development card?

More information about development cards (by clicking or holding) - display at lower part of screen

Playtest #5 (120529)

American English! :O

Players connected: 0?

Strange characters in the name ("\\n")

Name oriented towards you? Don't you know your own name?

Name in the title

House already on the board at start, not aware of this - alert of placing directly

Turn goes anti-clockwise?

Sheep look fewer when boosted, just outline?

Where to press end turn?

What did I get? Plus signs by the resources?

Visualization of new turn!

Place out more resources when trading with the bank

End turn - excise!

Choose resources when being robbed - add/remove?

Message after being robbed?

Highlight current actions on the board?

Distribution twice from the same tile

"The most important resource"!

The 6 looks like a 9 on the labels showing number of resources on hand

Resources move slowly when being robbed (having >7 cards)

How many roads/knights to get points?!

Put out more resources - position in a ring as the resources on each tile?

Don't restart the game if Bluetooth is being turned on

Discussion:

Lock the telephone if you wish to put it down (recap when it is turned on)

Poor track of the weather

Progress when connecting

Knight can be played before rolling the dice

Can only play one development card per turn

Resources on top of the robber cannot be picked up

Red player always begins

"Ooh! Dice!"

"Your turn" instead of "connected"

Robbing dialog acting weird!

It was fun! ... But crashy...

- "WTF?!" - "We're playing Catan!"

One telephone strange - not disconnecting, required to restart Bluetooth

Playtest #6 (120619)

Tab version:

- + Disabled marking
- + Consistent
- + Good overview
- + Trade easy to understand
- Send stuff
- Too big development cards, proposal: two rows of three cards

Portal version:

- + Better with development cards
- + Funnier to drag
- + Funnier when you know the game
- Development cards was bought directly, inconsistent
- Less user friendly, builds on that you know the game
- Trade with the bank
- Dark blue on black background, unable to see the road when disabled (having 0).

The latter version tested was the best...

Game rules are missing. The menu button for new turn was difficult to find but worked once you knew its location. Somewhat strong colors on the game board, while the houses and roads should be clear for larger contrast. Vibrate in other situations, when being robbed for example? Larger chest. Blinking border at the start of a new turn. No one noticed the starts at first, but they were appreciated when noticed. Animation when you get stars. Mode where you place houses before the numbers appear.

Appendix VII – Questionnaire

Settlers of Catan – Final Evaluation

Please help us by giving some thoughts of your game experience when playing the tablet/smartphone version of Settlers of Catan.

Game versions

What versions of Settlers of Catan have you played before?

- This version (tablet/smartphones)
- Board game version
- Online version
- Handheld version (Mobile Phone/Nintendo DS)
- Xbox 360 version

How do you prefer to play Settlers of Catan?

- With smartphones and tablet
- Digitally (online, handheld)
- The classic way (board game)
- In another way

Why do you prefer to play in that way?

Please give advantages and disadvantages of the different versions.

Which controller version did you like most?

- The tab version
- The portal version
- They are equal

Why was one version better than the other?

Please give advantages and disadvantages of the different versions.

Game features

Did you like the weather addition?

- Yes
- No

What did you think about it?

How did it affect your way of playing? Did it improve or worsen anything?

Did you like the hidden resources addition?

- Yes
- No

What did you think about it?

How did it affect your way of playing? Did it improve or worsen anything?

Did you like the spy card addition?

- Yes
- No

What did you think about it?

How did it affect your way of playing? Did it improve or worsen anything?

Appendix VIII - Division of work

The project has been cooperatively worked on and no concrete division of the project parts has been made. However there have been some main responsibilities on several areas. These are listed below. Included here are also the common areas where the work has been divided evenly.

Viktor Ingemansson

- Concept
- Research
- Sketching
- Graphics design
- Bluetooth
- Cocos2D
- Multi-touch interaction
- Game logic
- Playtesting
- Report

Joakim Ekendahl

- Concept
- Research
- Sketching
- Digital sketching
- Graphics design
- Core Android design
- JSON
- Playtesting
- Progress notes
- Report