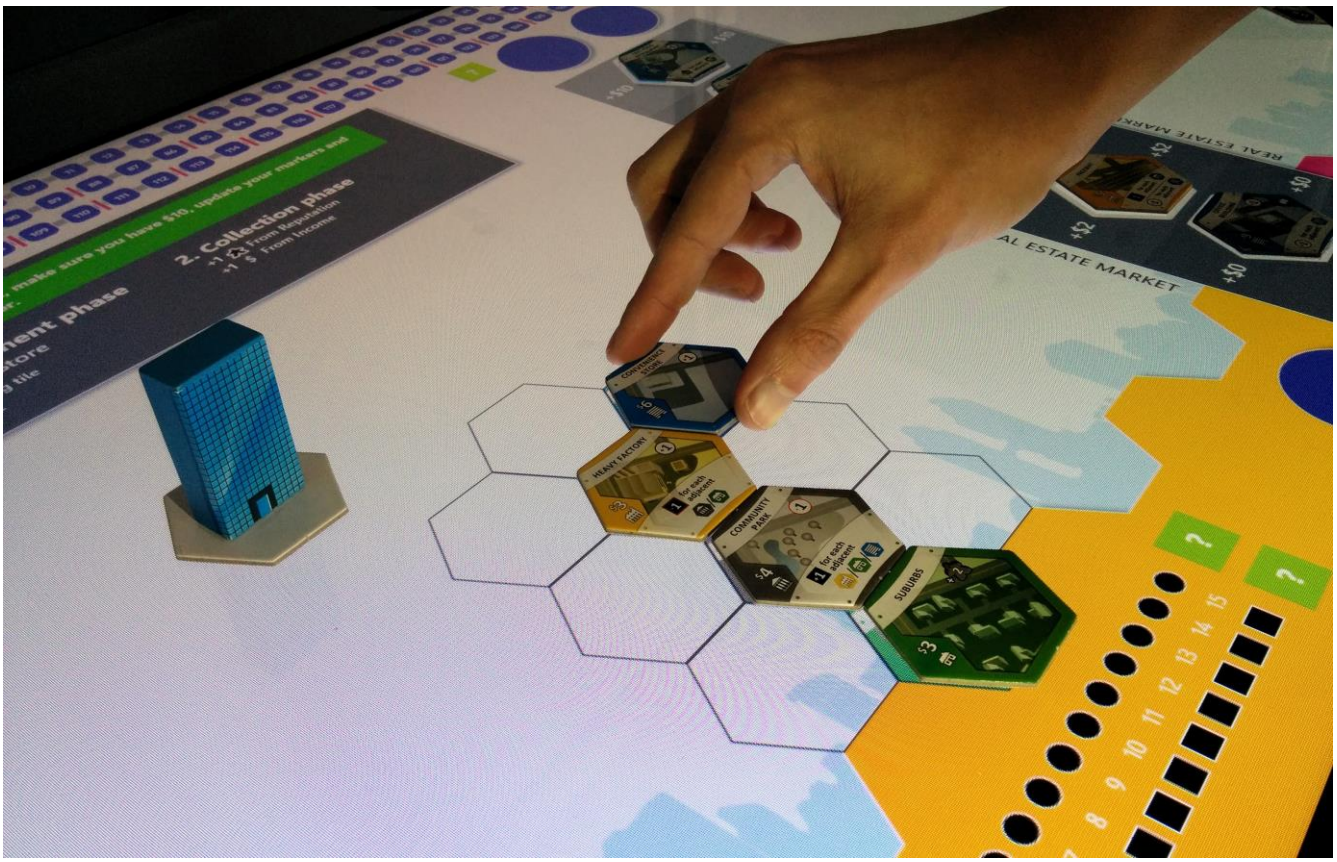
CHALMERS
UNIVERSITY OF TECHNOLOGY



Augmenting Suburbia

Guidelines for making computer-augmented tabletop games easier to learn

Master's thesis in Interaction Design & Technologies

SEBASTIAN LJUNGGREN
ANDREAS LUNDQUIST

Department of Applied IT
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2015

REPORT NO. 2015:111

Augmenting Suburbia

Guidelines for making computer-augmented tabletop games easier to learn

Sebastian Ljunggren
Andreas Lundquist

Department of Applied IT
Chalmers University of Technology
Gothenburg, Sweden 2015

Augmenting Suburbia
Guidelines for making computer-augmented tabletop games easier to learn
SEBASTIAN LJUNGGREN
ANDREAS LUNDQUIST

Technical report no 2015:111
Department of Applied IT
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover:
Placing a tile in *Augmented Suburbia*. *Augmented Suburbia* is the prototype used to explore games and learning in this project. Read more about it in section 7.2 (page 62).

Gothenburg, Sweden 2015

The goal of this thesis is to investigate how computer-augmented tabletop games and games in general can become easier to learn. This has been done by designing, implementing and user testing a prototype of a CATG, *Augmented Suburbia*. This prototype is a CATG-implementation of the tabletop game *Suburbia*. Based on the experience of creating and user testing *Augmented Suburbia*, twelve guidelines for how to design CATGs that are easy to learn have been developed. In addition, concepts that show how the guidelines can be used have been designed. In the future, these guidelines need to be validated through actual usage. While the guidelines were created with CATGs in mind, another future direction is to investigate to what extent they are applicable to game design in general.

Contents

Acknowledgements

A lot of people have helped us in our work with this thesis. Our supervisor, Staffan Björk, has provided invaluable feedback and advice from the start. Morten Fjeld and Pawel Wozniak have helped us with getting started with the *Microsoft Pixelsense*. Most of all we want to thank all the participants in our user tests, both the early tests during the design process and those with the more final version of *Augmented Suburbia*. Their feedback is the foundation for our results.

Terminology

CATG Computer-augmented tabletop games are games that use electronics such as screens and microprocessors in combination with elements from traditional tabletop games such as dies and pawns. They may be viewed as a hybrid between computer games and tabletop games.

Computer game In this thesis, computer game refers to all digital games, including desktop computer games, console games and mobile games.

Tabletop game A game that is played on a table. This includes card games, board games and miniature games.

AI Artificial Intelligence. Often used in computer games when talking about a player that is controlled by the computer.

1
Introduction

Recently, several tabletop games which use electronics have been commercially released (Casey 2014). In this thesis, this type of games will be referred to as computer-augmented tabletop games (CATGs). Combining these two areas makes it possible to do things that previously only were possible in computer games. For example, rules may be enforced, scores may be automatically computed, or it may be possible to play against an artificial intelligence opponent. Since the field is relatively unexplored, there are many possibilities and few best practices have been established.

Computer games are often designed to assist new players when they start to play. Gee (2007) gives some examples of how this may be implemented. Players are sometimes coached by the game through a first level, teaching about controls and mechanics of the game. Another design solution is to only teach a subset of the rules in the beginning, to avoid overwhelming players with details. Also, some computer games give suggestions of actions or general advise about what actions would be beneficial to the players.

However, in tabletop games, players instead need to refer to the rule booklet to learn the game before they can start to play (Prensky 2005). During a play session, rules may also be forgotten or misinterpreted and there may be downtime when rules need to be looked up. Therefore, tabletop games can be hard to learn without an experienced player. Some tabletop games even recommend playing with a limited set of rules the first time. A typical way of learning a board game is to start playing it and introduce rules gradually. However, this requires an experienced player or many interruptions when reading the rule booklet.

Computer games have made players used to being able to pick up games and learn them while playing. On the other hand, players of tabletop games still have to read the rule booklet before playing. This puts them at a disadvantage and may lead to them being perceived as harder and less approachable than computer games. However, CATGs make it possible for tabletop games to have a learning experience that is similar to computer games.

Computer-augmented games have previously been explored in the academia, for example by Benford et al. (2005), Bakker et al. (2007), Bergström et al. (2014) and Mandryk et al. (2002). However, the combination of CATGs and learning has not been explored. This includes how to make CATGs easier to learn as well as make them more approachable for new players. Since game designers do not want players to put aside their games out of frustration from

not understanding, the learning experience is important. While the area is important, game designers have also written very little about it.

We are two students with an interest in both computer games and tabletop games from the Interaction Design & Technologies master's programme at Chalmers University of Technology. We believe that if tabletop games were easier to learn, they would be able to reach a larger audience. Therefore, we find the topic of CATGs interesting and relevant for the future of tabletop games. This is the reason we have chosen to work with CATGs and learning in this thesis.

1.1 Purpose

The purpose of this thesis was to investigate how the experience of learning a computer-augmented tabletop game can be improved. The research problem was to find ways to make it easier and more enjoyable for new players to learn a computer-augmented tabletop game. The goal was to give insights to game designers on how to design CATGs that are easy to learn.

More specifically, the research question of this project was:

How can computer-augmented tabletop games be designed to make them easier to learn?

Since this question is broad and largely unexplored, this thesis' aim was not to provide a full answer to the question. Instead, it was answered by creating guidelines for how to design CATGs that are easy to learn. The guidelines were based on the experience of designing, implementing, and user testing a prototype of an augmented version of the existing tabletop game *Suburbia* with the *Microsoft PixelSense*. The project also included investigating examples of already existing computer-augmented tabletop games, comparing with computer games and tabletop games, and creating new CATG concepts that show how the guidelines can be applied in design.

1.2 Delimitations

While the CATG-prototype of *Suburbia* is important for the process of this project, the focus is to create guidelines. The prototype should be viewed as a tool for reaching this goal. Therefore, the guidelines are a more important result than the prototype. This also means that the goal of the user tests is not to evaluate or validate the success of the prototype, but to come up with ideas for guidelines.

Since the area is relatively unexplored, a comprehensive mapping of all aspects of learning in CATGs or games in general is outside the scope of a Master's thesis. Instead, the focus will be on giving an overview, and then exploring the details of some aspects.

The prototype is not meant to be a digital or interactive rule booklet. Instead, it will show how to play the game by using knowledge of the game state, similar to how many computer games work. The prototype will teach the game during play rather than before the game starts. We do not compare the learning experience of the prototype with an expert player that teaches the game to others, but instead in the situation that players read the rules themselves.

The whole prototype is only meant to be used the first times the game is played. Therefore, it needs to stay close to the original game. One of the goals is to preserve as much of the game as possible.

Since the prototype is only meant to be used while learning the game, the needs of players that already know the rules and enjoy the game will not be our main focus. This means that some of their strategies may be rendered useless and that they feel that the game has been simplified or even ruined. Also, the prototype will focus on learning the basic rules and not teach the players advanced strategies and higher forms of play.

There has been research on how games can be used to teach knowledge outside of a game, such as mathematics or reading. This thesis does not focus on this type of learning, it focuses on how the games themselves are learnt. However, this research may still be relevant to the project.

Since we have no experience of pedagogy, we will explore learning from the point of view of interaction designers. While pedagogical research certainly could be useful we will not use it since it is outside our area of expertise.

2

Theory

Academic research into games and learning is mainly focused on how games can teach for example mathematics or other useful skills but not how to make them easier to learn. However, game designers take this seriously since they want their games to be played. In this chapter, the areas of games and game design and how they relate to learning are presented. Both views from researchers and game designers will be given.

2.1 Games

There are many definitions of what a game is but most definitions seem to agree that a game is something you do outside of "ordinary life" (Huizinga 1955) and that is separated in time and space (Caillois et al. 1961). According to the game designer Costikyan (2002), a game involves players pursuing a goal by making decisions. One additional definition from Fullerton et al. (2004) points out the fact that the outcome should be unequal.

Games exist in many different forms but this master thesis will focus on tabletop games and, in particular, computer-augmented tabletop games. Computer games are also relevant, since electronic components allow a lot of what is possible in computer games to also be possible in tabletop games.

Tabletop games are games played on a table. These games are categorized into several types mostly based on the type of components that are prominent in the game. Tabletop game genres include dice games, tile games, card games and board games. Tabletop games are typically flexible, meaning that they allows for house rules (Benford et al. 2005). Since tabletop games are typically turn-based, there are often downtime when the players wait for the active player to finish his or her turn.

There is no generally agreed upon definition of computer games. Smed et al. (2003) defines a computer game as "...a game that is carried out with the help of a computer program." Furthermore, the computer can have three roles in the game: mediating the game process, visualizing the game state and acting as a participant in the game (Smed et al. 2003). In this thesis, a broad definition that includes computer games, smartphone games and video games will be used.

2.1.1 Computer-augmented tabletop games

Tabletop games combined with electronics are referred to in many different ways. Benford et al. (2005) call them computer-augmented tabletop games, Bakker et al. (2007) talk about tangible interaction in digital tabletop games, while Mandryk et al. (2002) use the term hybrid board/video games. Computer-augmented tabletop games, as we will be referring to them in this thesis, have the potential to maintain the direct interaction between people that tabletop games are good at, while still use the power of computers to calculate and keep track of the game state (Benford et al. 2005).

Bergström et al. (2014) gives an overview of computer-augmented games. Augmenting games can be done in many ways for many different reasons. For example, electronics can be used to enable new mechanics. Electronic components in games also allows for rules and state updates that are complicated and impractical for humans to manage. It can also be used as an AI opponent or to automate excise such as keeping track of the score.

Bergström et al. (2014) also presents their research of several design dimensions related to computer-augmented games. These dimensions can be used to understand the possibilities that technology brings into tabletop games. However, they should not be seen as guidelines when designing. Instead they offer a way to explore this type of games further. Examples of these design dimensions are as follows:

- *"Player-agreed" vs "Artefact-encased" game logic*
  Whether the rules of the game is agreed by the players or entirely encased in the artefact.
- *"Manual" vs "Automatized" excise*
  Whether the work to update and maintain the game state is controlled manually or if it is automatized.
- *"Limited" vs "Rich" audiovisual content*
  A rich amount of audiovisual content can help to visualize the game state or enhance the game experience.
- *"Low-effort" vs "High-effort" modification of game state*
  A "high-effort" modification of game state makes it possible to undo a previous move and to set the game to a specific state to show an example when teaching the game.
- *"Low" vs "High" tangibility*
  The amount of tangible objects in the game.

Eriksson et al. (2005) have been investigating how electronics can be used to enhance different tabletop game tasks. They analyzed several tabletop games and made a list of game state manipulating tasks that were common when playing these games. These tasks were then linked together with available electronics, describing how each task can be enhanced or supported by electronics.

While it may seem like a good idea to move every task to the virtual domain, it may be worth to remember and maintain the social dimension of tabletop games. For example, the act of

throwing dice is something that can easily be implemented by technology, but then you lose the social interaction between the players which is highly valuable (Magerkurth et al. 2004). Xu et al. (2011) explored how these tasks, also called chores, of updating the game state in tabletop games are actually needed to maintain the focus of attention between the players. These tasks are also important to the social interaction between players because they increase the awareness of the other players' turns (Xu et al. 2011).

2.2 Combining games and learning

The game designer Koster (2004) argues that all games can be considered "... very fundamental and powerful learning tools." When a game is fun to play, it means the player is learning something while playing. In older games, this is more obvious, since they often teach basic survival strategies. Today, what games teach can be more abstract, with odds and understanding of some mathematical concept being common in many games.

Also, Koster (2004) claims that games are only fun when they present the right level of challenge. If a game is too easy, we feel it does not teach us anything and therefore we put it away. If it is too challenging, we cannot grasp what it teaches us and we stop playing. When we have mastered a game and it cannot teach us anything new, we grow tired of it.

While Koster (2004) argues that all games are about learning at their core there are some types of games where the connection is more obvious, such as serious games. A serious game is a game in which being entertaining or fun is not the primary purpose (Michael et al. 2005). This does not mean that serious games can not be fun or entertaining, however. Typically, the purpose of a serious game is to teach something that is useful outside of the game. This can be educational (for example numbers or reading), a political or philosophical idea or to get people to change their habits. Educational games, often referred to as edutainment, is an example of serious games.

Gee (2007) has explored computer games and their usage as learning tools. He has created a list of 36 learning principles that games with good learning experiences have in common. One example of these is the *"Psychosocial Moratorium" Principle*. This principle states that when players can take risks without any real-life consequences, they learn from failure without any consequence. A related principle is the *Practice Principle* that states that you need to practice in order to become successful and if this is done in a fun environment it is easier to learn.

Another principle of learning is the *Subset Principle* (Gee 2007). According to this principle, starting with learning in a subset of the full domain can be helpful. The *Explicit Information On-Demand and Just-in-Time Principle* describes how the learner is given information when it is relevant, because it is easier to understand information in the right context. The principle states that this information can be either on demand or just in time.

The *Probing Principle* discusses that learning is a process of probing the game world with an action and then reflecting on the results (Gee 2007). This process of reprobing and then analyzing is the core of many games. Another principle is the *Discovery Principle*. According to this

principle, overt telling should be kept to a minimum. Instead, the player should be allowed to explore on their own and learn from their mistakes.

The *Multimodal Principle* focuses on how different modalities contribute to learning together (Gee 2007). Examples of modalities are text, images and interactions. By combining many modalities it becomes easier to learn. A related principle is the *Material Intelligence Principle*. This principle states that we store knowledge, problem solving and thinking in the environment and material objects in the game world. This reduces the effort needed to memorize.

Linderoth (2012b) disagrees with some of the points made by Gee (2007). His main argument is that just because the player makes progress in a game does not mean that the player has learnt something. Examples include adapting the difficulty when a player has problems, giving hints of what to do next and making it possible to pay for better equipment or to skip levels. Another example is how players often receive more powerful equipment when the game gets harder, which in some cases even leads to the game becoming easier.

Linderoth (2012b) concludes with saying "...observations of someone being able to play and progress in a game cannot be taken for granted as constituting the outcome of advanced learning processes." Instead, what is observed may be progression built into the game system that requires little effort and skill from the player.

Linderoth (2012b) also discusses the difference between *exploratory* and *performatory* actions. *Exploratory* actions are actions where the player learn something new, while *performatory* actions relies on previous knowledge and the player expects certain results.

Prensky (2005) describes his theory of five different learning levels in games. He describes the first level called *How* as the most explicit knowledge, players learn how things are done in the games they play. The second level, *What*, is about players learning what they are supposed to do in the game. For example, they learn the rules, the controls, as well as who is the enemy and what the goal of the game is. The third level of learning, *Why*, deals with learning the reason behind why players do something. This is mostly related to strategies and requires more of the players than the two first levels.

The two last levels of learning are called *Where* and *When/whether* (Prensky 2005). The fourth level is about the context in games. Prensky (2005) talks about that players learn of where they are and what they can do there. Some things can only be done in specific places. Lastly, the fifth level deals with moral problems, whether it is okay to do some things, but not others.

3

Background

In this chapter, an overview of tabletop games (both computer augmented and not) is presented. Examples of both commercial and academic games relevant to the project are also given. *Suburbia* is explained in greater detail than the other games because it is the most important game for this thesis. Furthermore, the field of tabletop computing and the *Microsoft PixelSense* are introduced.

3.1 Tabletop games

Tabletop games have a long history and the popularity for these games is increasing. For example, in 2009 the tabletop game *Settlers of Catan* had sold 18 million copies worldwide. Additionally, the newspaper ICv2 (2014) reports that the retail market for tabletop games in USA and Canada has reached $700 million in 2013.

Some of the classic tabletop games are Chess, Go, and Checkers. These are very old but still very popular. One of the most widespread games, with more than 275 million sold copies, is *Monopoly* (Hasbro 2015). It was based on *The Landlord's Game* that was published in 1906 (Orbanes 2007, p. 22), and several differently themed variants of this game are still released every year.

The popularity of tabletop games can also be seen through the increasing number of Youtube series which are discussing and playing these games. For example, *TableTop*[1] from the Youtube channel *Geek and Sundry* is a web series where Internet celebrities meet up and play games together. These kind of series typically include a walkthrough of the featured game's gameplay. *Geek and Sundry* has over 1 million subscribers on Youtube and more than 12 million social media fans around the world (Graser 2014).

These examples show that tabletop gaming is a thriving genre. There are a lot of commercial examples of tabletop games and the following short descriptions are just a few of these that are relevant to this master thesis.

[1] `http://geekandsundry.com/shows/tabletop/`

Figure 3.1: The tiles and other components of *Istanbul*. Image from *AEG* (`http://www.alderac.com/istanbul/`).

In the tabletop game *Istanbul* you compete with other players while walking around in the city of Istanbul. Each player has a leader along with a number of assistants that can be controlled. The city of Istanbul is built around 16 different buildings, each representing unique actions. The players take turn in moving their stack of assistants to one of these buildings to perform the corresponding action. Rubies are handed out to players who achieve specific goals and the first player to get a specified amount of rubies, depending on the number of players, is the winner. The game is shown in Figure 3.1.

Carcassonne is a tabletop game with tiles where you build a landscape together, consisting of cities, monasteries, roads and fields. The players take turn in adding one tile to the landscape and they have the opportunity to add one of their available workers (called meeples in the game) to that tile. The meeples give points to the owner according to specific rules for each of the four different types of landscape features. When every available tile has been added to the landscape, the winner is the player with most points. In Figure 3.2 meeples and some of the tiles from the game are pictured.

In the tabletop game *Ricochet Robots*, an unlimited number of players try to solve a puzzle in the least number of moves. The puzzle to be solved involves a rectangular grid on which robots can be moved in straight lines with the limitation that they can only stop when reaching an edge. The robots in different colors are placed randomly on the board at the beginning of the game. At the start of each turn, one marker is flipped face-up, displaying a symbol that matches one of the symbols printed on the board. The color of the symbol indicates which robot that should be moved to that symbol. When a timer has reached zero, the player with the best solution will get the marker. After all the markers have been resolved, the player with most markers is

Figure 3.2: Meeples placed on tiles from *Carcassonne*.

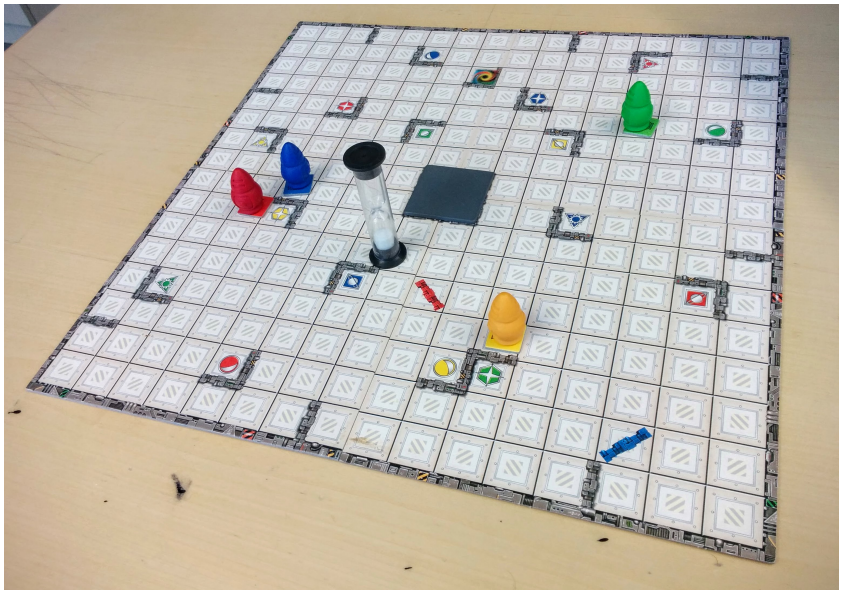


Figure 3.3: The board of *Ricochet Robots*.

Figure 3.4: A suburb from the tabletop game *Suburbia*. To the left a goal and some in-game currency can be seen. At the bottom are the markers used for tracking income and reputation.

the winner. The board of the game is shown in Figure 3.3.

3.1.1 Suburbia

Suburbia is a tabletop game for 1-4 players where you construct your own suburb. By taking hexagonal tiles representing different buildings from a real estate market, you increase the size of your suburb. The goal is to have the highest population at the end of the game. An example of a suburb is shown in Figure 3.4.

The population is both affected by your reputation in your suburb and by the effects of the tiles. Each tile has two types of effects, one that is calculated once when the tile is placed and one that is triggered by other tiles in play. The effects can also be triggered when new tiles are placed.

New tiles are bought from the real estate market. The market consists of seven positions with different costs where the tiles are placed. When a tile is bought, the tiles on the positions above will be moved down to decrease the cost. A new tile is then placed at the top of the market. Tiles can also be bought from the supply of standard tiles or by creating a lake to increase the money temporarily.

Goals are a way of getting more points at the end of the game. When playing with two players, there are two public goals. Additionally, each player gets to choose one private goal. Examples of goals are “Get 20 population if you have the lowest reputation.” and “Get 10 population if you have the most green tiles”.

Each player has, in addition to population, a reputation and an income. At the end of the

player's turn, money is collected based on income, while population is gained or lost according to reputation. The effects of the tiles increase or decrease these parameters. When a player's population increases, gates will be passed that will decrease the reputation and income by one. When the piles of tiles are empty, the game is over and the winner is the player with the highest population.

3.2 Computer-augmented tabletop games

As already mentioned, CATGs are tabletop games that have been augmented by making use of electronics in certain ways. A wide range of technologies can be used for this. These include virtual reality headsets, sensors like RFID and NFC, smartphone applications with image analysis and tabletop computers. This topic is investigated in depth by several researchers who implement their own system to demonstrate different technologies. This section contains descriptions of some of the examples from academia, but also some of the commercial games that were published in 2014 (Casey 2014).

TARBoard is an augmented reality tabletop system with tangible components (Lee et al. 2005). The system uses one camera to track markers that are attached to the bottom of the tangible components, and another camera for the augmentation. This enables the system to avoid occlusion when moving components and updating the game state on the board. A card game was implemented to demonstrate *TARBoard* in which monsters were augmented in the middle of the board when the corresponding card was flipped in the player's card area.

A similar system to *TARBoard* is the *STARS* platform (Magerkurth et al. 2004). *STARS* uses a camera to track the positions of physical objects on a game table. The camera is also able to detect the players' hands. This is different from the previous system because occlusion can occur. The *STARS* platform also uses different technologies to separate public and private information. A large vertical display is used to display everything that should be visible to everyone, while small PDAs are used to display each player's private information.

Song et al. (2007) introduced a projector-camera system which can project a game display on a paper board. This paper board can be rotated and tilted to fit the needs of the player. The system is able to compensate for the viewing angle and display the surface correctly. This is accomplished by using markers in each of the four corners of the paper board. A racing ball game was implemented to demonstrate this system in which the player controls a ball through a maze by tilting the paper board.

Additional examples of augmented tabletop games within academia are Weathergods (Bakker et al. 2007), False prophets (Mandryk et al. 2002) and Battleboard 3D (Andersen et al. 2004).

Recently, several commercial CATGs have been released. One example is *Golem Arcana*. It is a miniature game where the players compete against each other on a battlefield. The game features an infrared sensitive pen (called the *TDI Stylus*) with which you direct your miniatures on the battlefield. A smartphone application is used to display information about units, cards

Figure 3.5: A miniature from *Golem Arcana*. To the left the *TDI Stylus* can be seen.

and parts of the map that have been scanned with the stylus. A miniature and the *TDI Stylus* is shown in Figure 3.5.

Everything you do in the physical copy of the game will need to be updated in the digital version via the stylus. The stylus communicates with the smartphone with Bluetooth technology. The application makes makes calculations of line of sight, the outcome of battles and other things that would be complicated for the players.

Another example of a commercial CATG is *Alchemists*, shown in Figure 3.6. In this game, players take on the roles as alchemists trying to find the hidden effects of eight ingredients. The effects will be randomly distributed between the ingredients at the start of each game. A mobile application will keep track of the distribution and give results to ingredients tests that will occur during the game. In Figure 3.6b the smartphone view of scanning the two ingredients in Figure 3.6a with the application is shown. After six turns of collecting ingredients, mixing ingredients into potions, selling potions and publishing theses, the player with the most reputation as an alchemist will be the winner.

XCOM: The Board Game is a cooperative CATG that uses a smartphone application to direct the players through an alien invasion. The game supports up to four players that cooperate to survive the invasion. The players are assigned to different roles with specific abilities. One of the roles, the *Central Officer*, is in charge of controlling the application and directing the other players. The application goes through different phases in which the players must do things like collecting money, distribute UFOs to different continents and sending out interceptors to fight them. One component of the game (the base) is shown in Figure 3.7a. Each turn the application will ask the players whether the base wase destroyed as shown in Figure 3.7b. If this is the case, the players have lost the game.

(a) The laboratory of one player with two ingredient cards ready to be scanned.

(b) The result of scanning two ingredients with the application.

Figure 3.6: How ingredients are scanned i *Alchemists*.

(a) The representation of the base on the board. At the top the status of the base is shown. To the left aliens attacking the base can be seen. The defenders of the base are on the right hand side.

(b) The application asking for information about the game state.

Figure 3.7: The base in *XCOM: The Board Game*.

Figure 3.8: The *Samsung SUR40*, also known as the *Microsoft PixelSense*.

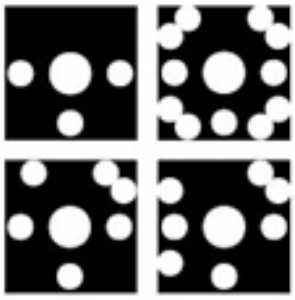


Figure 3.9: Tags consisting of different patterns of dots can be detected by the *Microsoft PixelSense*.

3.3 Tabletop computing and the PixelSense

Like desktop- and laptop computers, tabletop is a form factor for computers. An overview of the form factor and its uses is given by Müller-Tomfelde et al. (2010). Like the name suggests, the inspiration for the form factor is the table. The most distinguishing feature of a tabletop computer is its horizontal screen. Typically, it also has touch input and in many cases it has a mechanism for detecting physical objects on its surface. Being shaped like a table, it is suitable for groupware, where many users co-operate around one screen (Müller-Tomfelde et al. 2010).

Microsoft PixelSense[2] is a platform for tabletop computing that includes APIs and documentation. The *Samsung SUR40* is the only product currently sold with *Microsoft PixelSense*. It is shown in Figure 3.8. It features capacitive multitouch, a 40 inch 1080p display and the abil-

[2]Previously called *Microsoft Surface 2.0*, until Microsoft chose to use the surface branding for its line of tablets.

ity to detect tagged objects on the screen. The tagged objects are detected using tags put on the objects consisting of specific patterns of dots, see Figure 3.9. We will refer to the *Samsung SUR40* as the *Microsoft PixelSense* to avoid using different names for the specification and implementation of the technology.

4

Methodology

In this chapter, design methods that are relevant to this project are presented. They are categorized into three groups (ideation, prototyping and evaluation), which correspond roughly to when in a project they are relevant.

A process, whether it is a software development or design process, can be planned and performed in different ways. In a waterfall process, everything is first planned and specified (Cusumano et al. 1995). Next it is implemented without any changes to the original plan. At the end of the project it is tested and integrated. This process has several weaknesses, such as lack of flexibility when requirements change and and when the problem domain is not stable (Cusumano et al. 1995). Agile or iterative processes attempt to address these shortcomings by focusing on frequent deliveries, adjusting quickly to change and encouraging collaboration between different roles in a project (Fowler et al. 2001). While these processes originated in software development, they have been adopted in other fields such as interaction design (Fox et al. 2008; Patton 2002; Sy 2007).

4.1 Ideation

This part of a design process corresponds to what Jones (1992) calls divergence. In this phase wild ideas are created and the project branches out in many directions.

Brainstorming is the process of generating ideas in a group. Three principles to adhere to when brainstorming are (Wilson 2013):

- Aim for sheer quantity.
- Defer judgement about the quality of ideas.
- Encourage new and wild ideas.

While the principles sound straightforward, adhering to them requires skill and experience from a moderator (Wilson 2013). Otherwise, some individuals may take charge and involuntarily suppress ideas of some types or discouraging less outspoken participants from speaking.

Furthermore, internal politics and hierarchies may limit the productivity of the group, without skilled moderation.

During a brainstorming session, there are many methods that can be used to increase productivity. For example *Six thinking hats* may be used to help participants step outside of their usual roles and switch between different ways of thinking (de Bono 1999). With *Extreme Characters*, unexpected ideas can arise by thinking of the needs of an extreme persona, such as a super hero (Djajadiningrat et al. 2000). *Brainwriting* reduces loud discussion by letting participants write their ideas down and then passing them around (Wilson 2013).

After a brainstorming session, a lot of related and sometimes conflicting ideas may have been produced. To gain insights, the ideas may need to be grouped and ranked. One method for doing this is the KJ-technique, described by Spool (2004). With the KJ-technique, participants are first asked to group ideas silently together. Afterwards, suggestions for names to the categories are given by the participants. Lastly, the participants are asked to identify and rank the top three categories that they think are the most important. After a successful session, these rankings can be put together to create an aggregate list of priorities.

4.2 Prototyping

During the prototyping stage of a design process the viability of ideas from the divergence phase is evaluated. This is similar to what Jones (1992) calls transformation. In this stage several options can be explored. Eventually some will be dropped, narrowing the focus of the process.

Prototyping is the process of creating a prototype which is a partial implementation of a final design (Benyon 2010). Floyd (1984) mentions that the word *prototype* literally means "first of a type", and is used to describe the first version of any new type of goods in mass-production. However, in software engineering and interaction design, this literal translation can not be used directly because there is often only one product to design and the details of it is not always clear from the beginning. According to Floyd (1984), a prototype should be used as a way to find more details about the final version of the product.

A high fidelity (hi-fi) prototype is visually similar to the final product but does not necessarily contain all the required functionality (Benyon 2010). Hi-fi prototypes are typically used in detailed evaluation later in the design process, but the fact that they look so believable can be a problem. A simple mistake somewhere can get the focus of the tester away from what is actually important: the overall design of the product. Hi-fi prototypes are implemented in software and typically takes a lot of time and effort, thus they are not easily discarded (Benyon 2010).

On the other hand, a low fidelity (lo-fi) prototype is a prototype that should be used early in the process to aid idea generation and evaluation (Benyon 2010). As they are typically made of paper they can easily be created and discarded without the creator getting too attached to his or her creation. Lo-fi prototypes can be used for different purposes: ideas can be sketched

quickly to get a common understanding of the design in the group, or the prototype can be of higher quality and be used in early user tests by making it dynamic (Benyon 2010). However, Benyon (2010) thinks that if too much time and effort is put on creating a lo-fi prototype, it is probably better to create a hi-fi prototype instead.

The *Wizard of Oz-technique* can be used to provide similar functionality in a lo-fi prototype as in a hi-fi prototype (Buxton 2007). The idea is that a human responds to the user interaction with a system and acts like it was done by a computer. The user should not be able to understand that the system is not functional without manual involvement. Buxton (2007) mentions that it is the fidelity of the experience rather than the fidelity of the prototype that is important when testing ideas and that is why the *Wizard of Oz-technique* can be good to use.

4.3 Evaluation

After exploring ideas in the transformation phase, new decisions need to be made. Jones (1992) refers to this as convergence. In this phase, the remaining alternatives are compared and evaluated. New ideas and improvements can be found when evaluating the alternatives. At the end of this phase, a unified design has been created.

Interviews are a good way to get new ideas and suggestions (Courage et al. 2005). However, they have several problems, such as taking much time, being subjective, providing few data points and the risk of bias from the interviewer (Courage et al. 2005).

There are several ways to conduct an interview (Courage et al. 2005). In an unstructured interview the interviewer lets the conversation flow freely, allowing detours and posing open question. On the opposite side of the spectrum are structured interviews, where closed questions and a high degree of moderation dominates the discussion. A compromise between the two are semi-structured interviews, where both types of questions are used. During interviews, the interviewers may inadvertently affect answers and interviewees may try to give the answers that they believe are desired (Courage et al. 2005).

An alternative method to interviews can be observation. This means observing the user while they go about their everyday tasks or while they use a certain product (Blomberg et al. 1993). Blomberg et al. (1993) describes how observation can be done in a design project. By observing, a lot can be learnt that the user can not or will not express in an interview. One approach to observation is try to become the “fly on the wall”, minimizing the interactions between observer to the highest extent possible (Blomberg et al. 1993). This can be hard to do, and leads to the loss of opportunities to ask questions when interesting interactions occur. If the observer chooses to interfere more, the observation becomes more similar to an interview.

Another way to learn about the the users’ opinions is surveys. They make it possible to reach many users at once without much effort (Courage et al. 2005). A big disadvantage with surveys is their low response rate, which often is as low as 20% (Courage et al. 2005). Compared to the other methods, it is not possible to pose follow-up questions and elaborate as easily (Courage et al. 2005).

An evaluation can be formative or summative. Formative evaluation is done during the design process, to ensure that a design is on track with its the users' needs during the design process, allowing improvements between tests (Rogers et al. 2011). Summative evaluation is done with a finished design to assess whether it meets its requirements, possibly comparing it against some standard (Rogers et al. 2011).

One way to use formative evaluation on a prototype and improve it continuously is to use the RITE Method, introduced by Medlock et al. (2002). The RITE Method stands for *Rapid Iterative Testing and Evaluation* and is very similar to many other iterative evaluation techniques. However, the main difference is that in this method, changes are made to the prototype as soon as problems are identified. This makes sure that the prototype is improved rapidly and that newer testers will not experience the same bugs and features that previously were decided to be bad. On the other hand, it is not always good to change features early because different people have different views on things.

5
Planning

In this chapter, our initial plan for the project is presented.

In Table 5.1, an overview of our weekly planning at the start the project is shown. The project was divided into four phases with different main focuses. This does not mean we planned to only do one thing within each phase. For example, we planned to continuously write on the report and were prepared to change and update the prototype during the user study.

Weeks	Description
W 1-4	Initial planning and literature study, choose game and technology
W 5-10	Prototype development
W 11-13	User study
W 14-20	Analysis of results and report writing

Table 5.1: An overview of how the weeks will be spent. Note that this table only shows the main activities.

Table 5.2 shows the deadlines we had set for some important milestones. These deadlines were not strict deadlines, they were goals set to motivate the work. If needed the deadlines would be moved later on.

Date	Description
6/2	Choose game and technology
16/2	Planning report
27/3	Finish prototype
29/5	Presentation
5/6	Hand in master thesis report

Table 5.2: Important deadlines for the project.

The goal with our study was to be formative, not summative, since we were exploring a new area. This would affect our choices of methods throughout the whole project. The focus was not to validate our result, but to use it to explore how electronics can affect the learning process of a tabletop game.

Even though the project was split into four phases with different goals the process was never planned to be waterfall. Within the phases the work would be iterative. If needed, choices of methods and tools would changed. The phases would be split into smaller subtasks and worked on iteratively. Also, while the phases have main tasks, they would overlap. For example, some paper prototyping would be needed already during the planning phase. Also, the prototype would be changed between the user tests if needed.

The project was split in four phases (as shown in Table 5.1). The first phase was planned to be an exploration of the research area. During this phase existing research on CATGs and games in general was to be read, existing CATGs were to be evaluated and the project would be planned in greater detail. Another important part of this phase was ideation. The goal of the ideation was to come up with ideas for how electronics can be combined with games to facilitate learning. Another goal with the ideation was to get a clearer idea of what type of result the project would produce. At the end of this phase the technology and game used for the prototype would be chosen. Also, a plan for what methodology to use in the later phases would be produced.

During the prototyping phase, a CATG version of the tabletop game we have chosen was to be implemented. Since it is hard to test a prototype of a game without implementing it, the goal would be to produce a hi-fi prototype. The prototype would be created in small iterations with the goal of implementing one of a few features in each iteration (Fowler et al. 2001). To ensure the project stays on track, the prototype would be continuously tested during its implementation.

After finishing the prototype, it was planned to be user tested. The motivation for testing it was to come up with ideas for how it could be improved and how electronics could be used to make tabletop games in general easier to use. The choice of user testing methodology would need to match this goal.

The last phase was planned to consist of analyzing the findings from the earlier phases and compiling it into the results. Other than the prototype, the exact nature of the result was not decided this early in the project. Depending on what was found during the earlier phases, it could be some sort of framework for analyzing how well an augmented tabletop game facilitates learning, some sort of guidelines or recommendations for how to make CATGs easier to learn or a description of what possibilities CATGs create for learning.

Since the type of result was not decided at this stage, we knew the process needed to be flexible. The division into phases was meant to give some structure and subgoals to the project while still making it possible to adapt the process to a goal that is not yet fully decided and may change during the process.

6

Process

As mentioned in chapter 5, the project was split into broad phases with different goals. During the ideation and planning phase, the scope of the project was decided and the subject was investigated. Also, a brainstorming session generating ideas to use to support learning in games was made during this phase. In the prototyping phase, the prototype of *Suburbia* on the *Microsoft PixelSense* was iteratively designed and developed. Next the prototype was user tested in a formative study to find suggestions and improvements to the prototype and for learning in games in general. Finally, in the analysis phase, new concepts and guidelines were designed based on the experience of developing and testing the prototype.

While it may seem like a waterfall process has been used this is not the case. Instead the phases described should be viewed as descriptions of the main activity at various stages in the project. For example, a lot of changes were made to the code during the user tests and the planning phase included some lo-fi prototyping. Therefore the design process was more similar to an agile process, with smaller iterations within the main phases (Fowler et al. 2001).

6.1 Ideation and planning

At the start of the project, several decisions had to be made. These included what games to work with, how and if to prototype and whether user testing was needed. Furthermore, relevant work by others and a rough idea about what type of result the project would have needed to be identified. To ensure good decisions, we decided to have a long and broad ideation phase.

6.1.1 Pre-study

The first four weeks of this project was dedicated to a pre-study. The result from this was to be written down in a planning report.

In the pre-study we read a lot of articles. A list of articles to read was made to keep track of all of them. Articles were found by searching on Google Scholar and the Chalmers library. Even more articles were found when looking at the reference list of other articles. Both the point-of-view of game designers and game researchers were investigated, with focus on how

Figure 6.1: The post-its from the first brainstorming session.

games are designed to be easier to learn. After an article was read, we summarized it to be able to use it easier later when it was needed.

An important part of the pre-study was to plan the prototype. We needed to choose a tabletop game and a technology to use in our prototype. The study involved playing some tabletop games, but also some new computer-augmented tabletop games. For details about this, see subsection 6.1.3.

At the end of the pre-study we had a planning report with most of the theory we had researched up to that point and a plan (see chapter 5) how the project was supposed to be. At first we said that the research question would be broad and that we would narrow it down after we chose a game and a technology to use. However, we kept it broad even in the planning report because the results we expected to find from the prototype and the project were believed to be able to be used in a more general sense than only for the chosen game and technology.

6.1.2 Brainstorming

To get ideas for guidelines, we had brainstorming sessions early on. The goal of the sessions was to find ideas for how games (both tabletop and digital) are learnt. Mainly, the focus was to look at what is already implemented in existing games. Since there were only two participants, who are used to brainstorming, the need for moderation and planning was small. However, other important rules of brainstorming such as withholding criticism were still respected (Wilson 2013).

Figure 6.2: The post-its with ideas after being categorized using the *KJ method*.

Ideas were written down on post-its and put in the middle of a table. After two sessions of one hour each, 63 ideas had been produced. Since the ideas were generated without too much effort, no support methods were used during the sessions. The ideas from the first brainstorming session are pictured in Figure 6.1.

To sort and categorize the ideas, a variant of the *KJ method* (Spool 2004) was used. The goal with the categorization was to find broad patterns that could be used as a basis for the guidelines. The ideas were sorted into piles in silence. Then, with some discussion, the piles were labeled and a priority was discussed and given to each. When needed, piles were restructured by creating new piles, merging piles and moving post-its between them. Figure 6.2 shows the categories created after this session. The brainstorming sessions helped us to start reflect upon the area of research.

Another categorization of the ideas was made with *Trello*, a web application where cards can be sorted in different categories. First a card was created for each idea. Then the ideas were sorted, depending on whether they were applicable in the context of computer games, tabletop games or both. Some ideas were also categorized as outside the scope of our project. The result is pictured in Figure 6.3. The purpose of categorizing the ideas in this way was to see which aspects of learning that are unique to different form factors and whether some can be used in computer-augmented tabletop games.

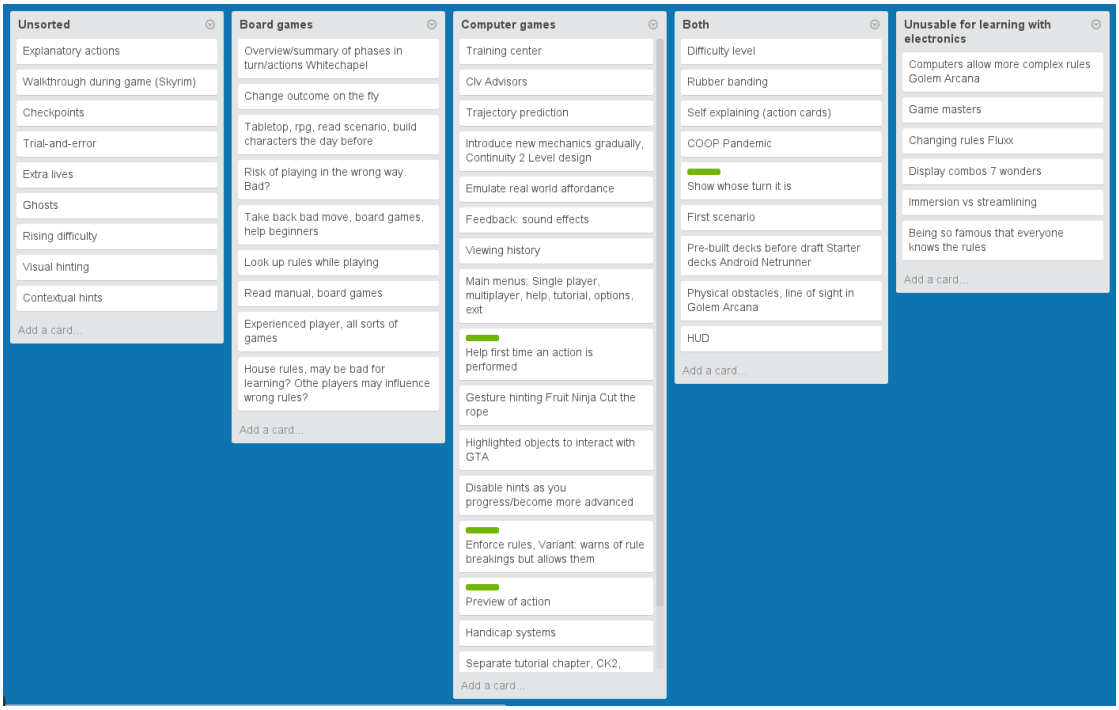


Figure 6.3: The ideas from the brainstorming sessions, categorized in *Trello*.

6.1.3 Planning the prototype development

An important decision made in this phase was to produce a hi-fi prototype. The main motivation for this is that we wanted to playtest the game with the prototype. With a lo-fi prototype using the *Wizard of Oz-technique* (Buxton 2007), playing would take a long time since someone would have to make calculations and update the game state when players make decisions. Furthermore, we wanted the testers to use a hi-fi prototype that was as close as possible to how the real product would look like, making them take it more seriously. Note that this did not mean lo-fi prototyping was not used in the design process, just what the final prototype should look like.

To be able to start prototyping, some key decisions needed to be made: what tabletop game to base the prototype on and what technology to use for the prototype. Since the decisions affected each other they were made together. In order to make the decision, several table top games and CATGs were played. The CATGs were played for inspiration while the regular tabletop games were potential candidates for the prototype. Our previous experience of tabletop games were considered when choosing what games to play. Examples of tabletop games that were played are *Carcassonne*, *Istanbul* and *Suburbia*. Additionally, the CATGs that were played are *Alchemists*, *Golem Arcana* and *XCOM: The Board Game*.

Some of the technologies that were evaluated were smart phones, sensors such as RFID combined with a microprocessor and the *Microsoft PixelSense*. The PixelSense quickly emerged as

a strong candidate. One reason is that its SDK would make it possible to focus on higher level design of the game instead of low level image processing and sensors. Another reason is the novelty of the PixelSense, it is a product that few consumers have seen. Lastly, since there already was a PixelSense already available at the department, convenience and economics played a part in the choice.

After realizing the benefits of the PixelSense, we started evaluating how suitable the games were for implementation on the PixelSense. Here, *Suburbia* was a clear winner. *Suburbia* has a flat design where tiles are placed directly on the table and almost no components are stacked. Since the PixelSense can only sense components placed directly on it, this meant that *Suburbia*, unlike many other games, would need few changes to be playable. Both *Carcassonne* and *Istanbul* use stacked components even though the board is flat. In *Carcassonne* the tiles are flat but the meeples are placed on the tiles. In *Istanbul* the players' components are stacked together and placed on the "place tiles".

Another important reason for choosing *Suburbia* is its complexity. It is a fairly complicated game to play, it requires keeping track of the tiles of all players and performing calculations when placing a tile. When playing the game for the first time, we spent at least two hours reading the rules and setting up the game. While a lot is going on when playing, a turn is relatively simple in *Suburbia*. It consists of choosing a tile, possibly discarding a tile and then updating your markers and money. Since the purpose of this project was not to challenge us with a technical challenge, it turned out that *Suburbia* was a good candidate as the game would be relatively easy to implement. Results from the evaluation of the prototype was considered much higher than the prototype itself.

When making the decision, another strong reason for choosing *Suburbia* was that with a few exceptions, the game can be scored just by looking at the board without knowing the order the tiles were placed in. This would have made it easy to allow players to set up example scenarios and taking back moves later in the game. While this was never implemented in the prototype, it was still one of the motivation that was discussed the most and was one of the inspirations for the guideline *Allow rules to be broken*. The reason why this was never implemented was a technical problem where the system sometimes lost track of the tiles that was already played.

6.2 Prototyping

After completing the ideation phase, work started on creating the prototype. Work was typically divided into iterations of about ten days, with each iteration having the goal to implement some pieces of functionality. Important to note is that the goal of the prototype was to get the players to learn about the original tabletop game. This is a priority that we thought about during the whole design and implementation phase. For example, we tried to use similar components in the prototype as in the original game just to get the players familiar with them.

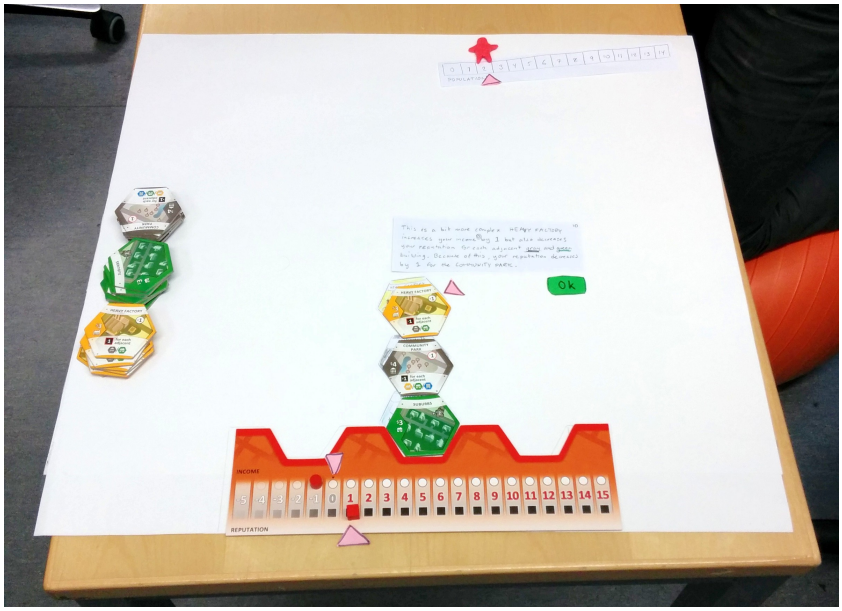


Figure 6.4: A lo-fi prototype that was used to test how the setup of the player's tiles and markers would work.

Typically, an iteration started with some sketches and a lo-fi prototype. After testing the prototype, some decisions of how to implement the new features were made. Then the features were prioritized and assigned to each of us. Next, implementation of the functionality started. When stable enough, the new functionality was tested again, to ensure it worked as intended.

Before anything else, the screen space was determined and decisions about what components to include on the screen and where to put them was made. Then the first iteration started, dealing with how to setup one player's markers and starting tiles. A tutorial was introduced giving explanations of the steps. First a lo-fi prototype, shown in Figure 6.4, was developed, tested and improved based on feedback. Next, implementation started on the PixelSense. When the code started to stabilise, we asked the testers of the lo-fi prototype to test again.

Following the first iteration, setup of the market and goals, logic for the market, showing a tile's effects and playing a whole turn was implemented in the same fashion. The main difference was that in later iterations, more focus was on testing on the PixelSense instead of building lo-fi prototypes like in the earlier iterations. While we describe these iterations separately, we sometimes implemented the features in parallel. In the following sections these iterations will be described in more detail.

6.2.1 Test of screen space

Before starting the implementation, we needed to see how well *Suburbia* would fit on a *Microsoft PixelSense*. To do this, the game was played on a paper with the same size as the the screen

Figure 6.5: A test of how to fit the game on a *PixelSense*. Two ways of placing the player board were tested. The players were positioned opposite to each other where one had the player board at the top and the other at the bottom.

of the PixelSense. Figure 6.5 shows one of the setups we tried out. The first conclusion we could draw was that the screen was not big enough for more than two players even though the game usually support up to four. Therefore all other decisions were made with that in mind.

Components that we did not want to sense with the *PixelSense* like money, the tile piles and the goal pile was put aside as for now. Firstly, we could easily see that the players' tile areas seemed to fit nicely. Also we decided to let the players have their own money and private goals where they seemed to fit. After a play with all three piles of the game, we could still fit every tile on the screen. However, at this early test we decided to put the population tracker outside of the screen because we were still discussing where and how to put it.

In addition to learning what would fit on the screen, this test also allowed us to quickly try different placements of the components. For example, Figure 6.5 shows two different ways of placing the board with the players' markers on, also known as the "home". Either this board was put at the bottom or at the top of each player's half of the screen. We decided to put it at the bottom because it is closer to the player. Private components are usually placed near the users as this area is seen as the personal territory (Scott et al. 2010; Smith et al. 2008).

We decided early that we wanted to have the market in the middle of the screen so that it could be reached by both players. Another alternative that we discussed was to put it along one of the long sides, but that would make it harder to reach some tiles. The orientation of the tiles was discussed a lot. If they were rotated so that they face the players only one player can read the text normally, the other one need to read up-side-down. Therefore it seemed more fair to rotate them towards the long-side of the table, but that was not perfect either as the text was rotated wrong for both players.

After the early conclusions of screen space, a decision was made to only focus on learning of the components on the screen rather than the global setup dealing with shuffling components outside of the screen. Piles with game tiles and goals should be shuffled before the prototype started.

6.2.2 Initial setup and the tutorial

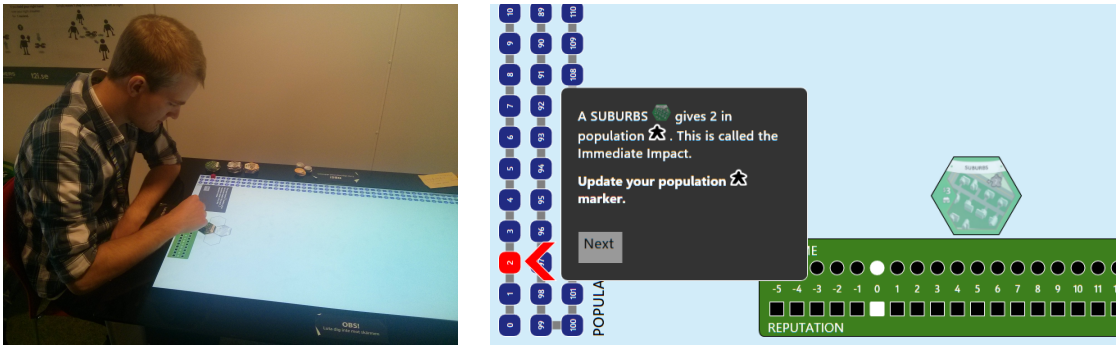
This iteration started by introducing a tutorial where one player was asked to put markers on his or her trackers, to place the first three tiles included in the setup and to update the markers according to the effects of these tiles. The lo-fi prototype can be seen in Figure 6.4. We chose to start simple and only implement half of the screen first to make sure that everything was good before implementing the whole screen. Another reason why we started with only half of the screen was that it is easier and faster to test the lo-fi prototype with only one player instead of two.

This prototype used some existing components of the original game while the graphical user interface was drawn on paper. Only the three standard tiles were used in this prototype. The icons on the tiles corresponds to the form of the markers in the game. We preserved these with one exception, we replaced the original population marker with a meeple to match the population icon. Because we did not have any meeple at the time we created our own from clay.

The tutorial dialogs contained different texts and could be positioned on the screen area. Arrows pointed out important areas where the player should look. With a OK button the player could move to the next step of the tutorial. Alternatively, when a tile was placed at the correct position, the game would recognize that and go to the next step automatically.

The prototype of the first iteration was tested by four different persons whereof one of them had played the game before. The reason why we asked one person that had played the game before was to check that everything was correct according to the rules. The first conclusion from the tests was that they found what were causing the effects was hard to understand. We thought that if the information on the actual tiles was read, the understanding would be higher. This could be done by pointing with arrows at the specific information or using images with the relevant parts highlighted. Secondly, a good thing we noticed was that the testers did not mind the amount of text in the tutorial. One additional thought from the testers was that they would really want the markers on the different trackers to be recognized by the system, similar to the tiles. This was because they thought that the OK button was a bit annoying. However, due to the small size of the markers we understood early that we could not use tags to recognize them.

After the lo-fi prototype was created and tested, the implementation of the hi-fi prototype started. As we had little experience with working with the *PixelSense* before, we started to play around with some examples. Then we began with implementing the different components that were needed in order to finish the introduction tutorial. We knew early on that a hexagonal grid was going to be needed and that tiles were needed to be recognized by using tags.



(a) During user testing. **(b)** One step of the introduction tutorial.

Figure 6.6: Hi-fi prototype of the first iteration.

During the implementation we were having some technical troubles with the recognition of tags. Since we used regular computers when running the program we could not use physical tags directly. Instead, we tried to use the Input Simulator from the *PixelSense* SDK, but it turned out that an error in this program caused it to crash our prototype. A web search showed one person that also had this problem, but we could not find any solution so instead we decided to test the tags on the real hardware.

Simultaneously, the dialogs and the tutorial began to take shape. With the dialogs we added arrows and the ability to move them around on the screen following a tutorial. Each tutorial step could, at this stage of the prototype, contain a description text, an action text, an OK button, an arrow and icons, everything optional. The action text was bold to mark it as more important to read than the description text. See Figure 6.6 for an example of how one step of the tutorial could look like.

Another component that was implemented was the population tracker. We decided to put it along one of the long sides of the screen. The tracker runs back and forth three times and lines indicate the connection between each step. With the motivation that the action to move a tangible marker manually could make the players start to reflect more, a digital tracker was decided to not be implemented. Later, this was a starting point for the *Make use of tangible pieces* guideline. One alternative that we discussed was to make it fully digital with two numbers representing the players' population. A second alternative was to put the tracker around the playing area along the edges like several other tabletop games such as *Ticket to Ride*.

Finally, the "home" component was implemented. This component was copied straight from the original game and turned into a digital version. With the same motivation as with the population tracker we decided to still use the original markers on the screen. However, with all trackers we have, the actual value was displayed by marking the corresponding step. The previous value was also marked with a white border.

The last step of this iteration was to test the hi-fi prototype. We asked interaction design students, who are familiar with user testing, to see if they could follow the tutorial and finish the introduction of the game without any help from us. However, if they had any questions or

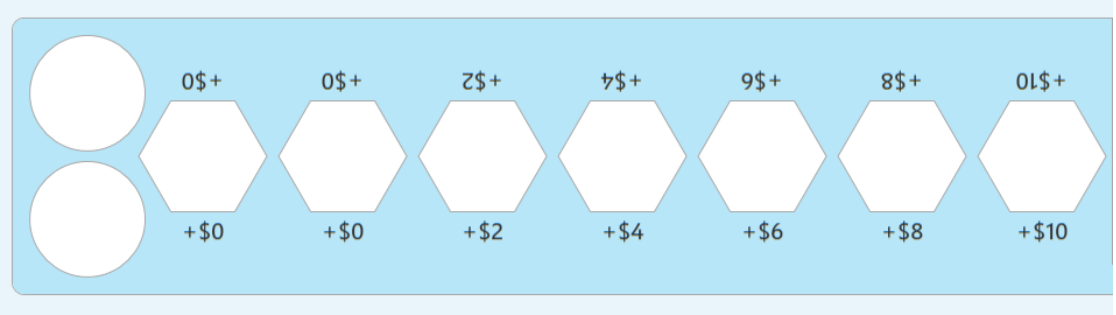


Figure 6.7: An early version of the market and the public goal slots. The market slots are hexagons and the goal slots are circles. Note: The image is rotated 90 degrees.

comments about the design we actively discussed it with them during the test.

A good discussion arose from the tests. The testers came with a lot of new ideas and ways to improve the prototype. The main issue that was hard to understand was to connect the result of placing a tile to the effects it have. Suggestions were given to draw connecting arrows or have images of the tiles with the relevant parts highlighted. This was similar to the feedback we got from the lo-fi prototype tests. Some texts in the dialogs were too long so some of the testers suggested that we split some of them into several dialogs. We could also use bullet lists when there were long texts, but we decided to split them instead.

Overall, the participants were positive of the prototype. We asked some control questions about how to calculate scores of placing tiles. Even though the participants admitted they had not looked on the tiles while placing them, they were able to explain the effects of tiles they had not seen before. The participants were also positive of our use of symbols and stated that it helped them.

6.2.3 The market and goals

Now that the introduction of the game was implemented for one player we could start to add some of the rest components of the game, mainly the market and the goals. An early version of these can be seen in Figure 6.7. The goal of the second iteration was to implement support for tiles to be bought from the market and to add a part to the tutorial to teach about the rest of the introduction (fill the market with tiles and add public and private goals) and the first turn. This work was done a lot in parallel with the next iteration *Visualizing the effects of a tile* (see subsection 6.2.5) because they are not dependent on each other and could be developed in parallel.

Already in the ideation phase, we had decided to place the market between the players. Since it is important that both players keep track of it and it is central to playing the game, the choice was obvious. One important goal when designing the market was to keep it robust and flexible. For example, we wanted to make it possible to move the tiles freely between the slots

to change prices and not enforce when it was updated. The motivation for this goal was to keep it close to the original game. This reasoning is part of the inspiration for the *Allow rules to be broken*-guideline.

To this end, the market simply loads any tile placed on it and adds the corresponding market price to it. This tile will still be tied to the slot even if it is lifted and placed in the playing area. The association is only removed when a new tile is placed on the slot.

While it was decided early on that the market should be placed between the players, there was discussion about its orientation. It could either be oriented vertically, standing between the player areas like a barrier, or it could be horizontally oriented, lying along one of the edges of the screen. The choice fell on a vertical orientation since it created a natural division between the players.

Late in this iteration, we came up with the idea to highlight which tiles in the market the player can afford. This was implemented by adding a border to the slots. The border is colored green when a tile can be afforded, red when a tile is too expensive and grey when the game is in a state where a tile cannot be bought. This highlighting was an important inspiration to the *Show available options*-guideline.

We decided to create slots for the goals, similarly to the market slots. Instructions were added to the tutorial for how to distribute the goals. It was decided early on that we would not digitalize the goals by putting tags on them or automatically calculate who achieved the goals. Two reasons motivated this choice. The first were time constraints, we chose to use our time for features we found more important. The second reason was that it is a relatively simple calculation performed at the end of the game. By that time, we figured the players had gotten used to the rules.

If tags had been placed on the goals, they would have been placed at the back side of the goals. The public goals would be scanned as soon as they were placed face-up in their slots. However, the private goals would be a bit more problematic, since they are private and therefore need to be placed face-down. This means the tags would not be facing the screen of the *PixelSense*. Our solution to this problem would have been to wait with scanning the private goals until the end of the game, when they are revealed, by turning them face-up.

6.2.4 Help dialogs

In *Suburbia* there are a lot of components and symbols to keep track of. During early user tests, questions were asked about these components such as the income and reputation trackers.

Since *Augmented Suburbia* is played on a screen, we decided to place question marks near some components. When tapping these question marks a help dialog is opened, with information about how the component works. Initially, help dialogs were created for income, reputation and population. An example of an early help dialog is shown in Figure 6.8. These dialogs were the inspiration for the *Allow players to get information on demand*-guideline.

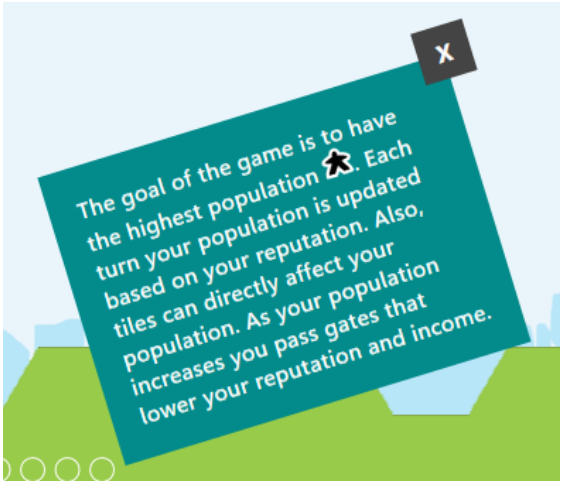


Figure 6.8: An early version of a help dialog explaining how population works. It can be moved and rotated by the user.

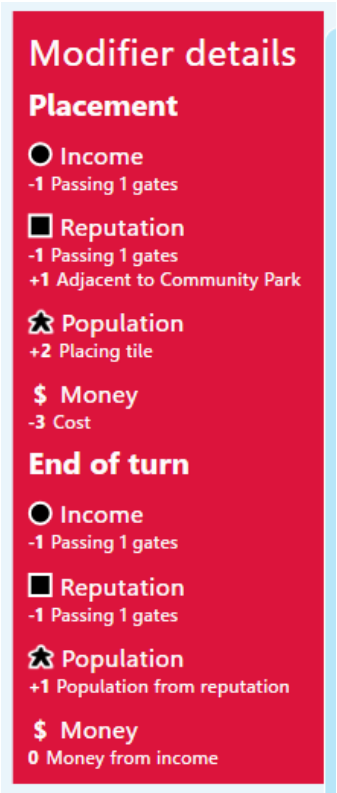
A problem with these dialogs was where to position them. Initially, they were placed near the components. However, this is a position where tiles are often placed, therefore occluding the help dialogs and also interfering with the ability of the *PixelSense* to detect tags. The first attempt of solving this issue was to make it possible to move the dialogs by dragging them if they were occluded. However, dragging the dialogs was not easy for the players to understand because there was no affordance indicating this possibility. Therefore, we ended up placing them at the top of the screen, below the population tracker, where tiles are not placed.

6.2.5 Visualizing the effects of a tile

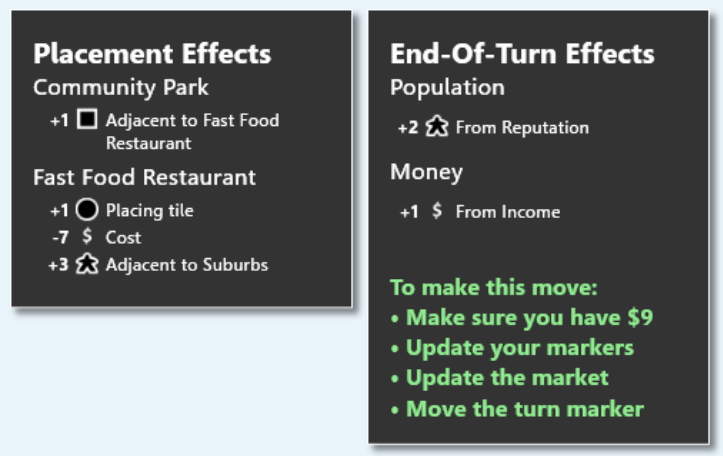
To make it easier to make a choice of where to place a tile, two components were developed. They are the *effect details* and the *effect table*. The effect table gives a quick overview of what a tile placement leads to while the effect details give a detailed explanation of where the effects come from.

The effect details went through several major redesigns during the prototyping phase, as shown in Figure 6.9. Throughout the redesigns it basically contained the same information. The difference is mainly in layout and categorization. Early on, the effects were separated into placement effects and end-of-turn effects. This was to indicate that placement effects were applied first, while end-of-turn effects were calculated based on what had happened in the placement phase. Initially, the effects were sorted by their type, as shown in Figure 6.9a.

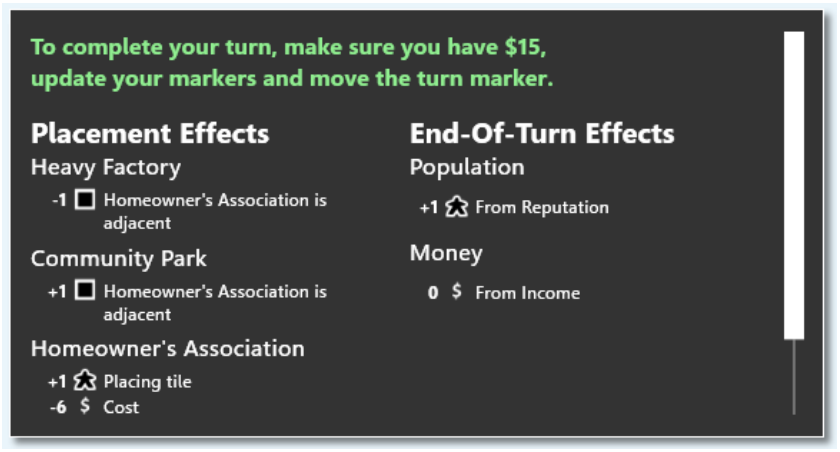
It turned out that explaining effects created by more than one tile could become complicated when categorized by effect type. Therefore, the placement effects were re-categorized according to which tile the effect was written on. This made it possible to list all tiles that triggered one tile's effects in one place. In the same stage, placement effects and end-of-turn effects were



(a) An early version of the effect details. Placement effects were sorted by type.



(b) Eventually the effect details were split into two components. Also, the placement effects are now sorted by the tile that caused them.



(c) At the end of the prototyping phase the effect details were merged into one component with two columns. This made it possible to give the status message a better position. Also, scrollbars were added to constrain the height.

Figure 6.9: The evolution of the effect details during the prototyping phase.

Figure 6.10: An early version of the effect table.

split into two separate components, to make the difference between them clearer. Also, a status message giving instructions or informing about an error was added. These changes are shown in Figure 6.9b.

Separating the effect details into two components was problematic. Since the status message was often related to both, it was hard to choose which component to place it in. Therefore, the components were merged into one again, as shown in Figure 6.9c. However, they were still kept in separate columns. This made it possible to let the status messages span both the placement and end-of-turn effects. Since the variable height of the effect details sometimes led to overlap with other components, a scroll bar was added. To make sure status messages were not missed if players did not scroll down, they were moved to the top. These were the last changes made to the effect details during the prototyping phase.

To make it easier to get a quick overview of what a tile placement leads to, the effect table was created. As shown in Figure 6.10, it gives an overview of how the income, reputation, population and income are affected when placing a tile. It gives the total change as well as the changes during the placement and end-of-turn phase. Unlike the effect details, the effect table did not go through any major redesigns, only a change of color.

Instead, the main discussion around the effect table was about its position. There were two alternatives discussed. The first was to keep it in a static location. This would make it easy to find for the players. The alternative was to place it near the tile that was placed. This would make it clearer that it visualized the effects of a tile. The first alternative was implemented first since it was easy. Eventually we decided we wanted the second alternative and started implementing it. However, the implementation of this feature was only partial since it turned out to be hard to place it in a good position near the tile being placed without being occluded or occluding other components. Therefore, the second alternative was abandoned in favor of the first.

Another way of visualizing the effects was the markings on the different trackers. Whenever a tile has been placed, all the trackers change value according to the new effects.

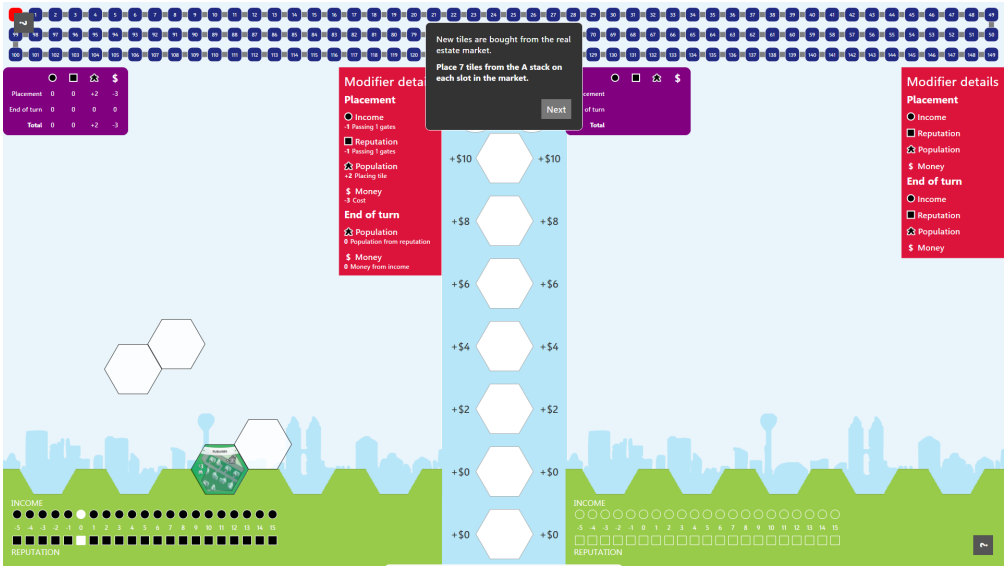


Figure 6.11: The player area was copied once and placed next to the first. The tutorial dialog is positioned in the middle.

6.2.6 Introducing two players

When work had been done to implement each of the main components in the game, what was left was the turn handling. Until now, nothing had been dependent on two players so the first thing to do was to copy the first player area and put the copy next to it, see Figure 6.11.

One of the largest decision in the prototype design was taken in this iteration. At first we thought it was best to have the players facing each other, sitting on each of the two short sides of the table. We designed the lo-fi prototype with this in mind, but already then we had some concerns about what the rotation of the market slots should be so both players could read the tiles.

Players facing each other is common in competitive tabletop games, such as *Magic: The Gathering*. However, we decided to place the players so they sit next to each other instead, while also rotating everything on the screen so it was readable to both players. This layout allowed the players to more easily see the board of the other player. This could be helpful in the communication between them, for example making it possible to point at something and explain it. While the game is competitive, we wanted the learning process to be cooperative. When sitting next to each other, the possibility to help each other to learn is enhanced because the players can watch the other player's turn. This thought was the main inspiration for the guideline *Encourage learning by observing* that we designed later.

The population tracker needed to change a bit when adding a new player. Before, the tracker had only one border to show the position of the marker, but now there was a need for two. In most cases, the two players would not be on the same position so then we could simply have

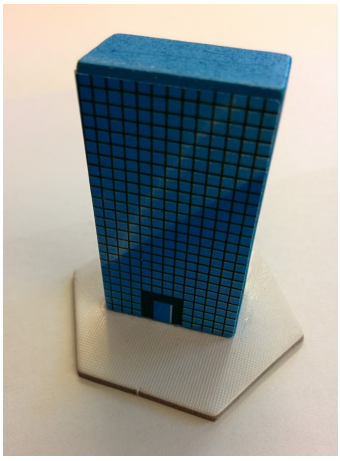


Figure 6.12: The turn marker placed on a blank tile with a tag underneath.

two single borders. However, when the players have the same population value it becomes trickier. This was solved by adding a second border inside the first. Another solution that we thought about as well was to have half the border in one color and half in the other, but the first alternative was chosen because of the form consistency with the single borders.

During this iteration we thought about how to change turn to the next player. Early ideas were to change when the player had put down a tile in his or her play area, change when the player pushed a button to confirm his or her choice, or using a tool to determine whose turn it was. Most turn-based tabletop games have some kind of token to mark whose turn it is and so does *Suburbia* as well. We decided to keep this marker and augment it to be recognized by the system. By putting it on a blank tile with a tag underneath (see Figure 6.12), we could use it to mark the current player. Putting the marker on one player's side of the screen indicates that it is his or her turn. When the turn is over, the player puts the turn marker on the opponent's side of the screen. By using a tangible marker we keep one of the manual tasks in the game, allowing the players to know what happens to the game state (Xu et al. 2011) (consult the guideline *Make use of tangible pieces* for more information about this).

The tutorial also needed to be updated to support two players. At first there was only one tutorial dialog, positioned in the middle of the screen so both players could see it. However, one dialog for each of the two players was needed so they could read the texts more easily. We decided to implement a new dialog that is basically a copy of the first dialog, but each property of it can be changed like the description and action texts. That allowed us to give different instructions to the two players.

One problem with having two different dialogs is that the players can easily get out of sync. If one player is faster than the other, that player will get further in the tutorial. A decision was taken to implement a mechanism to get the players synchronized because if they need to wait, the hypothesis was that they will start to help each other, making use of the guideline *Encourage learning by observing*. When one player pushes the OK button, a text would appear

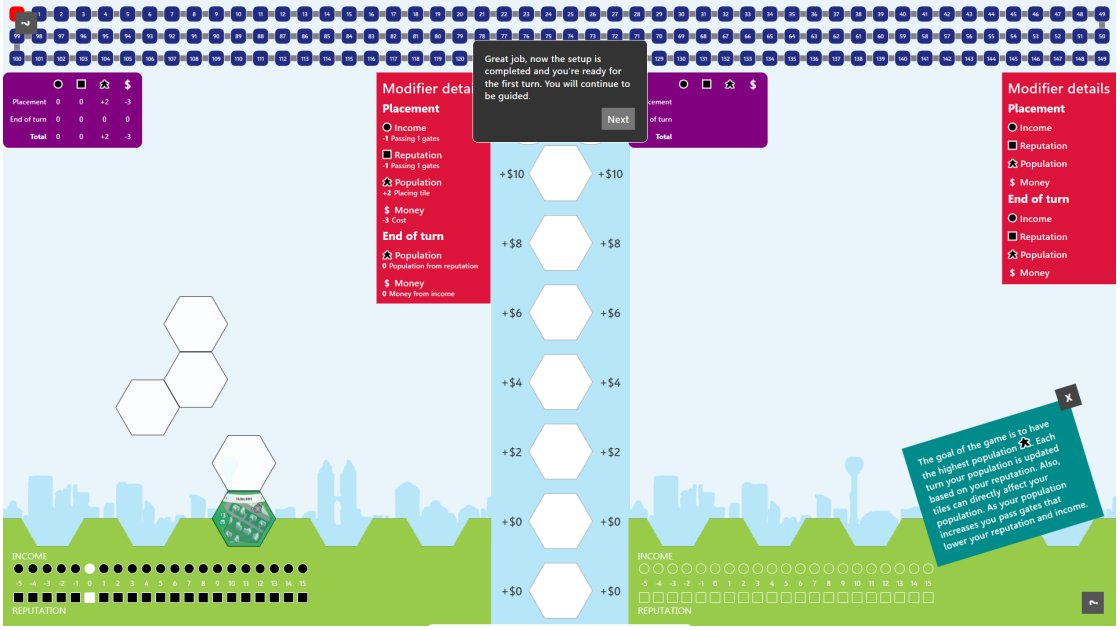


Figure 6.13: A lot of the graphical elements at an early stage of the project. There are a lot of inconsistencies.

next to it telling the player to wait for his or her opponent. Only when both players have pushed their own buttons, the tutorial would continue.

After the split of the tutorial into two dialogs we could position them more closely to the related components. This allowed us to still position dialogs to point at components without making it difficult for the other player to read. For example, when the tutorial introduces the population markers, one dialog is positioned near the population marker on the left side of the screen while the other is positioned in the center of the right side. In this way, both players could read the text. The texts could also be different to let the players help each other more.

6.2.7 Graphical design

The goal with the graphical design of *Augmented Suburbia* was to have a simple and clean design that did not distract from the main goal: playing and learning *Suburbia*. Reaching a high level of polish was not a high priority.

The graphical design is loosely based on Google's *Material Design*-guidelines [1]. Examples of *Material Design* can be seen in the flat design, the choice of bold colors and the use of drop shadows to create the appearance of some components being closer than others. An important motivator for choosing *Material Design* is our prior experience with using it.

[1] Available at `http://www.google.com/design/spec/material-design/introduction.html`.

Based on early sketches and prototypes, the graphical design was gradually improved throughout the development of the prototype. Early on most components used different colors, paddings and font sizes. Some components had rounded corners while others did not. These inconsistencies are shown in Figure 6.13. To create a more coherent experience this was eventually reduced to a limited set of standard styles that were reused in different components.

6.2.8 Changes from the original game

While the focus of the prototype was to stay as close to the original game as possible, several changes have been made to the game. The reasons for these changes range from not having enough time to technical problems to misunderstandings of some rules.

In the original tabletop game, the players play through three piles of tiles named A, B and C The tiles in the later piles are more powerful and expensive than those in the earlier tiles. When making *Augmented Suburbia*, we decided that the first pile was enough for learning the game. The tiles in later piles have the same type of effects and we believe the learning experience could be improved more by spending time on other things than implementing more tiles. A problem with having fewer tiles is that it changes the balance of the game and what strategies work well. While this may irritate experienced players, we disregarded this since our target audience is beginners and teaching strategy is not a primary goal of the project. While many tiles are not implemented, the application is prepared for the addition of more tiles, meaning it would not require too much work.

In addition to skipping the tiles from the B and C tiles, some A-tiles were not implemented as well. These tiles were highly specialized tiles that would have required custom logic. The skipped tiles were *Fancy Restaurant* and *Waterfront Realty*. Instead of implementing them, other features were prioritized.

The real estate tiles of *Suburbia* have lakes printed on the opposite side. When low on money, one can get some money in the short term by simply turning a real estate tile from the market into a lake by turning it over. This presents a problem when implementing the game on a *Microsoft Pixelsense*. Either, tags need to be placed on both sides of the tiles or the tiles can be separated into two types. The former alternative has the benefit of remaining close to the original game. However, it means covering important information on the tiles. The latter alternative requires changing the rules somewhat. Instead of turning tiles from the real estate market over, they are discarded and a lake tile can be taken from a separate pile. This is similar to how the base tiles already work in the game. For technical reasons, we chose the latter alternative. The unused tiles from the B and C piles were used as lakes instead.

An unforeseen consequence of modifying how the lakes worked, was an unintentional change to the rules. In the original game the cost of placing a lake has to be paid immediately, before getting income. However, when discarding a real estate tile after placing a base tile, the cost of the discard is paid after the income. When lake tiles were modified to work more like base tiles, the cost of a discard was also moved to the end of the turn. This decision was not planned, it simply was implemented. It has merits, it would be hard for a beginner to understand why

a different mechanic is used for lake tiles when they otherwise work so similarly to base tiles. However, this rule change does not help reaching the goal of testers being able to play the original goal. Had this unintentional rule change been found earlier, it might not have been included in the prototype that was tested.

One rule was completely ignored when designing *Augmented Suburbia*. In the original game, players have the option of placing an investment marker instead of placing a tile in the suburb. This marker is placed on an already placed tile and doubles all of its effects. The cost on the tile needs to be paid again when placing an investment marker. Since the *Microsoft Pixelsense* can only detect components directly on its surface, some workaround would have been needed to implement it. For example, lifting the tile and tapping the space below it could have been used to indicate wanting to place an investment marker. However, a workaround like that would have been complicated both to implement and play the game with. Instead, we simply chose to skip investment markers in our prototype. While they add some depth to the game, they do not significantly change the game and are easy to explain to someone who has played the game once.

Suburbia has a complicated setup where many components need to be prepared before playing the game. This is described in the rules booklet in an illustrated step by step guide. Since we believed making this guide digital would not give many (if any) benefits it was not implemented in the prototype. Instead, the testers were introduced to an almost complete setup where only the player home and the missions needed to be setup. Skipping the initial setup allowed us to focus on parts of the game we believed would benefit more from being ported to a *CATG*.

6.3 User testing

When the prototype had reached a playable state, the time had come to test it thoroughly. The first step was to have a pilot test to see if our plan worked out. During this test, the prototype crashed a lot so the main focus of this test became identifying bugs. In spite of the crashes, the testers still managed to play the game, even though they were confused by the incorrect information shown by the prototype. This is one of the drawbacks with using a hi-fi prototype (Benyon 2010). However, the pilot test made it possible for us to avoid a lot of issues during the actual tests.

Testers were invited to a *Facebook* event where the project was briefly introduced. The requirement put on testers was that they should not have played *Suburbia* before. Also, they were told not to look up any information about the game before the test. Those interested were asked to enter the times they were available in a *Doodle*[2]. The text from the event is available in appendix A.

For the tests we wanted a broad audience of both tabletop gamers, computer gamers and people who don't play much games. This was taken into account, when choosing which persons to

[2] *Doodle* is a service for scheduling where participants can tell which of the available time slots they are able to attend. It is available at `http://doodle.com`.

invite.

As people responded, we booked them for the tests in pairs. Our goal was to book around 10 user tests. After some discussion and one cancellation we ended up with a total of 10 tests booked. The testers were notified of the test as soon as we booked the session. They were also reminded about the test one day before it.

Of the 20 testers, 6 were female and 14 male. The age of the testers varied between 21 and 33 with an average of 25.3. 8 of the testers are students of the *Interaction Design & Technologies*-masters programme. The testers' experience with tabletop games and computer games varied from casual gaming to playing several times a week. Since the testers were invited through *Facebook*, all of the testers are our friends or acquaintances.

When the testers arrived, they were welcomed and asked to sit down next to each other by the *PixelSense*. They were given a short introduction to the project and how the *PixelSense* works. An important part of this introduction was stating the goals of the test, which were:

- Find inspiration and ideas for how to make tabletop games and games in general easier to learn.
- Identify weaknesses and strengths in the prototype.

By stating the goals, we wanted to remind the testers that we had a broader goal than just evaluating the prototype. Then, they were told about the structure of the tests. Before using the prototype they were instructed to:

- Follow the instructions on the screen.
- Try to figure out problems together if needed and communicate freely with each other (for example to discuss problems or ask each other questions).
- Ask us for help only as a last resort and remember that the goal is to learn from the prototype.
- Feel free to speak out whenever they had ideas or suggestions for improvements.

At the end of the introduction they were asked if they were OK with having their hands filmed while playing and having audio recorded. They were also asked if they had any questions. The script used for the introduction is available in appendix B.

Next, the prototype was started as well as the recording. As the testers played the game, we observed them and took notes on what they did and said. We tried to minimize the interaction with the testers, for example refraining from commenting, unless it was needed for the test to proceed. The observation method "fly on the wall" (Blomberg et al. 1993) was used during the entire test sessions. Figure 6.14 shows the layout of the tests. The test consists of first setting up the game with the help of a tutorial, then playing the first rounds with a lot of guidance. For the last turns the testers played without a tutorial as shown in Figure 6.15. Because of time constraints, the game was ended after a third of the game was played. The time the spent with the prototype varied between 30 and 90 minutes.

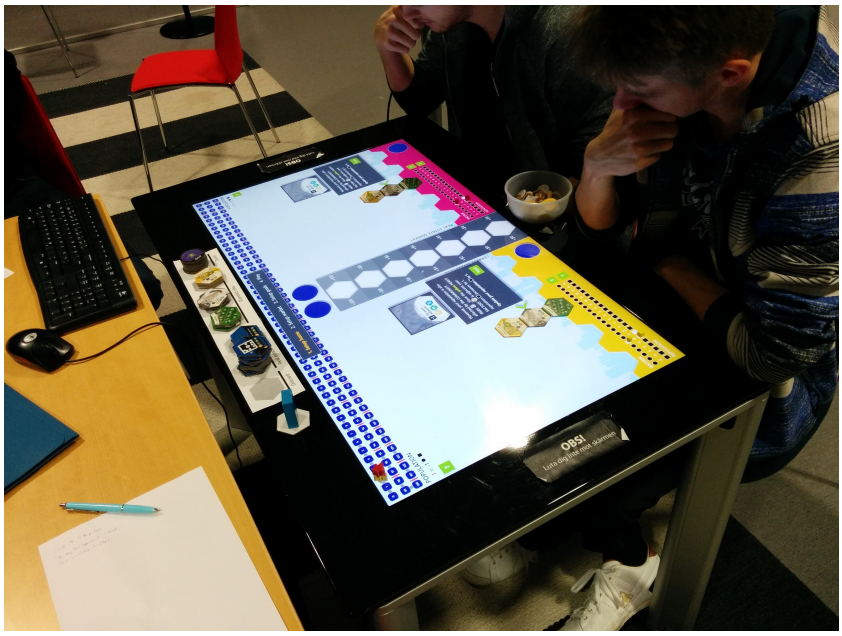


Figure 6.14: The setup of the user tests. We sat at the table to the left, observing and taking notes. The testers sat next to each other on the right. This photograph is from an early part of the tutorial.

Figure 6.15: A photograph from a later stage of a user test, where the tutorial is over and the testers play on their own. In the grey box the effects of a tile placement are shown.

At first, the time in front of the table was over after they had played through the game. However, in later tests, we asked the participants to summarize their experience while still sitting in front of the table. The purpose with this change was to get opinions and ideas while the participants were near the table. If needed this made it possible for them to point at things and explain using the components of the board.

Before the interview, the participants were offered coffee and pastries. Then the participants were individually interviewed. The questions are available in appendix C. The interview was semi-structured, following a loose script but allowing and encouraging follow up questions.

After each tests, the notes from the observation and the interviews were gathered. The data was analyzed to come up with ideas for concepts, changes to the prototype and guidelines.

6.3.1 Insights from the tests

In this section the ideas for concepts and guidelines from the tests will be presented. Feedback that actually led to changes of the prototype will be presented in subsection 6.3.2.

One of the questions from the interview was "Do you feel that you would be able to start playing Suburbia without the prototype? Motivate why or why not." All testers answered yes to this question although some had reservations. The reservations included not knowing everything about how to setup the game, feeling unsure about the calculations and wanting to keep the rules as a reference. Whether this is actually true has not been extensively tested and should be treated with some scepticism. As mentioned before there have been some alterations to the game and the system may have led the participants into thinking they know more than they actually do (Linderoth 2012b). Still, we believe this statement indicates an overall improvement of the experience of learning *Suburbia*.

Many testers questioned why they had to updated the markers when the application already calculated the status updates for them. This was a conscious design decision which would eventually lead to the *Make use of tangible pieces*-guideline. There are two reasons for not changing this. The first is that we observed that many players realized something unexpected had happened when they saw that they needed to move the markers. This led them to trying to find out what had happened and starting to make calculations. We do not believe this would have happened to the same extent if we just showed the updated game state. The second reason is our goal of teaching the original tabletop game, where no help is given when updating the markers.

6.3.2 Changes made between the tests

Since the study was formative and did not involve comparing to a baseline or having the exact same conditions between tests, changes could be made between the tests. Even if we did not use a specific method, our process is similar to the RITE Method (Medlock et al. 2002). Therefore,

Figure 6.16: The modifier table showed how a tile placement affected the different phases of a turn.

we were able to incorporate some feedback from earlier tests into updates before the later tests. These changes ranged from bugfixes and small changes in texts to interface changes with big effects on the user experience. When the suggested changes were deemed too big because of time constraints, they were instead added to an updated version of the concept which is presented in chapter 7. In this section the most important changes between the tests will be discussed.

Already during the pilot test, a lot of ideas for changes surfaced. The most significant change made after the test was the removal of the modifier table, pictured in Figure 6.16. This table gave a clear and concise overview of the outcome of placing a tile. During the pilot test, it received a positive reaction along the lines of "It makes it easy to make a good decision quickly." The problem we noticed is that this resulted in less attention paid to what the effects on the tiles actually meant. Therefore, the modifier table was removed just because the testers saw it as a convenient feature. This decision follows the reasoning of Linderoth (2012b), who argues that giving the player a tool that makes it easier to progress not necessarily leads to the player becoming better or learning something.

Other changes made after the pilot tests were an overhaul of the texts, a lot of bug fixes and changes to make the interface clearer. One issue discovered was that the application still reacted on tiles that had already been placed. By marking placed tiles as committed, the problems this created (such as tiles being placed twice) were solved. Another issue was confusion about what different parts of the interface were for. Therefore a text label was added to the real estate market and help dialogs were added to the public and private goals and the population tracker.

A tutorial legend was also added to clearly display in which phase of the tutorial the players were, see Figure 6.17. This was added after some testers had a problem knowing if they played the game for real or just played through a tutorial. The legend is displayed during the tutorial and is located above the market in the center of the screen. This area is used for the population tracker but during the tutorial there is no way to get your population so high up on the tracker. Thus it is not a problem that the legend covers another component. The legend disappears

Figure 6.17: The tutorial legend with the four phases of the tutorial.

(a) The effects are highlighted by fading out the periphery.

(b) The effects are highlighted with red circles.

Figure 6.18: Higlighting of the effects of tiles was updated to make it clearer.

when the players begin to play without guiding. After adding this, the testers seemed to get less confused even though some of them did not see it. However, if they would start to wonder, it was there to see.

Some testers thought it was hard to understand the difference between the two phases. Therefore we decided to change the names of these to “Tile placement phase” and “Collection phase”. We also added numbers to the phases to hint that they are calculated in order and not simultaneously.

A minor thing we also added was more ways to trigger the next step of the tutorial. From the beginning we only had two ways: clicking the OK button and placing a specific tile in a specific position. Two new ways were added at this stage: placing any tile on any position and moving the turn marker to the other side of the screen. These triggers are all optional and can be set for each individual dialog to ensure the next step is not triggered accidentally.

In the tutorial there are several images of tiles where effects and other things are highlighted. From the start, the effects were highlighted by fading out the rest of the tile, as shown in Figure 6.18a. During the tests, it was observed that some testers had a hard time understanding these images. We believe it was because they did not realize that the subtle fade effect was not interpreted as a highlighting. Therefore, the fade effect was replaced with a red circle around the effect. Figure 6.18b shows an updated image. While not as aesthetically pleasing, this made

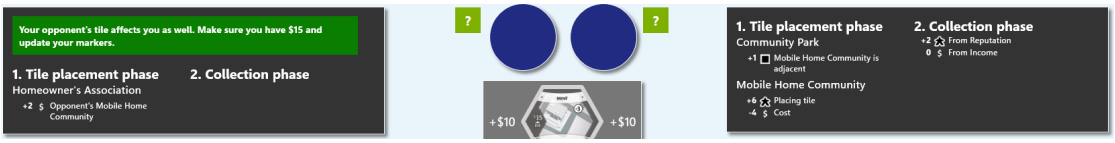


Figure 6.19: When an every-effect is triggered during the other player's turn, two effect detail windows are shown.

the images clearer.

Another problem identified during the user tests was how the every-effect was explained. The tiles with an every-effect are affected by both player's tiles. Therefore, they may have an effect on you when your opponent makes a move. During the first tests, these effects were not shown until it is your turn again. This was both incorrect and confusing. Therefore, the behavior of the effects details was changed so that, when this happens, both players were shown an effects details window with effects that affect them. Figure 6.19 shows an example of this where the first player placed a *Homeowner's Association* tile. When the second player places a *Mobile Community Home* tile, the every effect of the previous tile is triggered. Then each player is shown an effects details window. This helped some of the later testers realize how the every-effect worked.

6.4 Finalizing concepts and guidelines

After all the tests had been done, we began to look at the categories and ideas that we had from the early brainstorming sessions. We saw that some of these were conflicting so we started to organize them into dimensions.

When the dimensions were created we recognized some of them from the dimensions mentioned by Bergström et al. (2014). Even though the dimensions from Bergström et al. (2014) were created for computer-augmented games as well, they were not designed with a learning perspective. Therefore some of them differed from our dimensions.

Based on the dimensions we started to design guidelines and find examples of when they could be used. The template for the early version of our guidelines was as follows:

- *Description* where the guideline was described in summary.
- *Example* where one example of a CATG that uses this guideline was presented.
- *Conflicts with/Similar to* where a relationship was presented.

This template was later improved to include a more thorough description of the guideline, meaning a general description in addition to examples of how the guideline is used in computer games and tabletop games today. We also added more examples of CATGs that use each guideline because it is good to see how the guidelines can be applied in different ways. One reason why we chose to add information about the usage for computer games and tabletop games

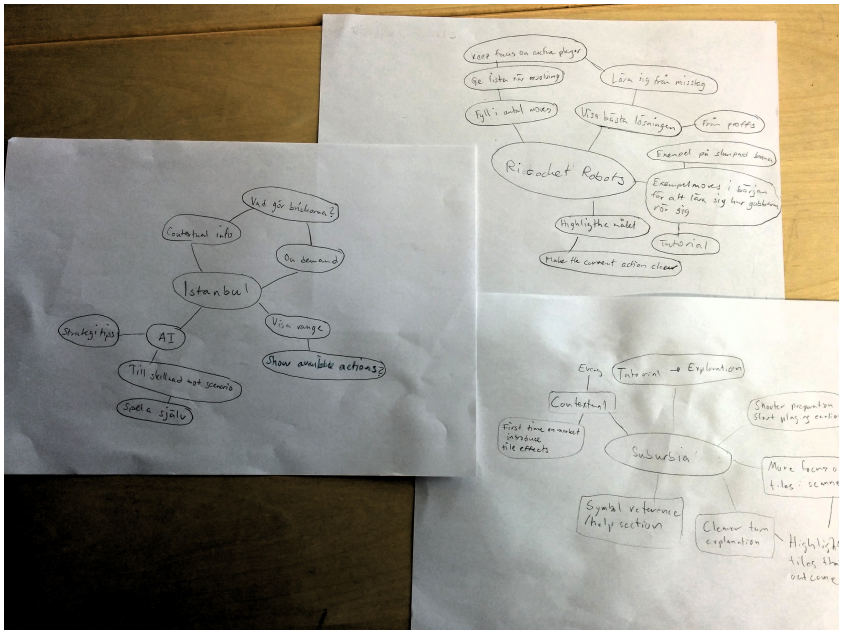


Figure 6.20: Mind maps of our three concepts.

is that we recognized that most of our guidelines were very general, not only for when designing CATGs. Therefore we tried to add more generalization to the guidelines and to be clear about the general use. Also, presenting a contrast between computer games, tabletop games and CATGs makes it easier to understand how computers can be used when augmenting games to make them easier to learn. We also changed the attribute *Similar to* to *Complements*, because we found that the guidelines were not so similar to each other. They however sometimes work well if used together.

During the development of the guidelines, we also started to design new concepts of CATGs based on the guidelines we had created up to that point. Two entirely new concepts and a new concept of our prototype were designed. A mind map was created for each concept, relating main features to guidelines, see Figure 6.20. Even though the guidelines can be used when creating new games, we chose to use existing tabletop games to augment because the main focus was to create concepts, not new games. Also, because we chose existing games, we did not have to explain all the details of gameplay, but rather focus on the parts that could be augmented.

Not only did we design the concepts based on our guidelines, the opposite worked as well. When designing the concepts, we came up with several additional guidelines. The development of the concepts and guidelines came hand in hand with each other and knowledge from one part influenced the other.

The end result was twelve guidelines and three concepts whereof one is a new and relatively different concept of our prototype.

7
Results

There are three main results in this thesis project:

- Guidelines for designing CATGs that are easy to learn.
- *Augmented Suburbia*, a prototype of a CATG created to explore learning.
- Three concepts of how other tabletop games can be combined with electronics to facilitate learning.

While the results are presented in this order, they have been developed in parallel, affecting each other. For example, some guidelines were created based on the result of user testing the prototype. The guidelines are presented first, so they can be used while explaining the prototype and the concepts.

7.1 Guidelines

In this section, the twelve guidelines that have been created will be presented. The goal of these guidelines is to support the design process when creating CATGs or games in general to make them easier to learn.

Each guideline has been given a short title. In some cases, simplicity has been prioritized over having long but more correct descriptions. To understand the guidelines fully it is not enough to just read the title. However, the titles are a convenient way to refer to the guidelines and give a reminder of what they are about.

The guidelines have been created with CATGs in mind. However, they may be of more general use. This is reflected by the structure of their descriptions. Each guideline has a description consisting of three paragraphs. The first paragraph describes the general motivation and application of the guideline. The second gives examples that match the guideline in existing computer games and tabletop games. The third paragraph discusses the specifics of using the guideline when creating a CATG.

After the description, some examples of how the guideline has or can be applied are given. The examples are taken from existing CATGs, our prototype and concepts and in some cases from

regular tabletop games. A guideline typically has two examples.

Some of the guidelines may be contradictory and give conflicting advice. When this is the case, these conflicts are listed for each guideline under the *Conflicts with* heading. Also, some guidelines are similar to each other or work well together. In these cases, they are marked as *Complements*. To find out which of two conflicting guidelines are suitable in a certain situation, the examples in the guidelines and the concepts in section 7.3 can be consulted.

The guidelines may be used both when designing a game or evaluating how a game enables learning. When designing a game, reading them may give ideas for how to solve problems and introduce the game. When analyzing an existing game, they can be used as a checklist. They may be used to both find how learning is facilitated and how well the guidelines are followed. There is a big difference between ticking off that a guideline was used and analysing how it was used and what the results of using it is. Simply using a guideline does not guarantee success, a bad use of a guideline may even hamper learning.

Make the current game state clear

Description It should always be clear what is happening in the game to the player. This includes knowing whose turn it is and what the game expects the player to do. If a turn has phases, what phase the game is in can also be important. When keeping track of the state is important, the player should be given reminders of what is happening.

In computer games this is done in most games, at least to some extent. One common way is to have a HUD (Heads Up Display) where information about the game's state such as the player's health and current equipment is shown. Also, many computer games use highlighting, for example to show what in-game objects that can be interacted with and where the player should go next (Linderoth 2012b). In tabletop games, this is not as easy to do. Typically, visualizing the game state requires manual excise such as moving a turn marker or updating markers.

When designing a CATG, information such as whose turn it is or what phase of a turn the game is in can be tracked. If such state information is tracked it is suitable to make it available to all players in a prominent location such as a screen or project on top of the game.

Examples In *Carcassonne* a turn consists of taking a tile, placing it in a valid position and then choosing whether to place a meeple or not. In a computer augmented version of the game, the player should get a clear reminder of what phase of a turn the game is in. Assuming the game is played around a table, a suitable placement of this information is in front of the player whose turn it is. After a tile has been placed, it can be suitable to show the information near the tile that was placed.

In *XCOM: The Board Game* one player is given the role *Central Officer*. It is the *Central Officer*'s responsibility to use the game's smartphone application. The game does not automatically keep track of the game state but instead relies on the *Central Officer* entering the needed data in the app. The app guides the *Central Officer* through the many complicated steps of a turn to ensure

no status updates are forgotten. It is the *Central Officer*'s responsibility to relay information from the application to the players with other roles.

Avoid overwhelming the players with new rules and mechanics

Description When a player is new to a game, he or she may put the game aside if the game is perceived as confusing or too complicated, as discussed by Koster (2004). To avoid this, the player should be given a gentle introduction with only the most important concepts introduced first. It is often better to hold the player's hand and only give a limited set of the available possibilities first. Later, when the player has gotten more comfortable with the game, its more complex rules and mechanics can be introduced. Two principles of learning (Gee 2007) are the *Incremental Principle* and *Subset Principle*. They both relate to how learning should be divided into smaller steps instead of introducing a lot of new things at the same time.

A computer game has many options of how to introduce new elements to the game. The *Super Mario*-series is often lauded for its level design, where enemies and obstacles gradually become more complicated (Parish 2012). Other methods to manage the learning curve in computer games include unlocking new abilities later in the game and walking the player through an interactive tutorial. Tabletop games have a harder time doing this, instead they have to rely on players using the rule booklet as a reference while playing. While rule booklets can be pedagogical and well written, this still often leads to missed and misinterpreted rules. Some tabletop games have a beginner setup where only a subset of the rules are used the first time the game is played.

In a CATG, a lot can be borrowed from computer games. It is possible to have an interactive tutorial, delay the introduction of some elements and to make choices for the player. If something is omitted, care should be taken to avoid a situation where the player feels he or she was tricked to make a bad move. Make sure the players are given enough information to make a good move. The *Provide contextual advice and information*-guideline gives one way of managing the order things are introduced in.

Examples When teaching the players about the first turn in *Suburbia* it is not necessary to explain all the available options, but instead guide the players through one of them. Alternative options can instead be explained during later turns. In *Augmented Suburbia* only tiles from the market is introduced during the first turn. Standard tiles and lakes are saved for the second turn. This is suitable since players almost always want to build something from the real estate market during the first turn.

While *XCOM: The Board Game* has a tutorial tutorial it requires reading a lot of text before starting the first turn. This is very similar to how tabletop games requires reading the rules first and takes a lot of time. To avoid this problem, rules and roles could instead have been introduced while playing with more details added later. In *XCOM*'s defense, this is done to some extent.

Complements *Provide contextual advice and information*

Allow players to get information on demand

Description When it is hard to decide what information is currently needed by the players, it can be suitable to allow the players to decide what information they are interested in. By letting the player make the choice of what to show, it is more likely to be relevant information. Also, information that was requested is less likely to be percieived as annoying or in the way. Three principles of learning (Gee 2007) that this guideline is related to are the *Explicit Information On-Demand and Just-In-Time Principle*, *Probing Principle* and *Discovery Principle*. The first states that the learner should be given information when it is needed. The second principle states that the learning is a cycle of initiating an action and reflecting on the results. The last principle states that overt telling should be kept to a minimum in favor of allowing the players to experiment and discover on their own.

The user interfaces of many computer games allow access to more information on demand. Helpful information can be shown when clicking or hovering a game element. Help can be shown in a tool tip or a separate help window. It is possible to offer dedicated help buttons next to components. Some games include help sections, where the players can browse and search for information. When playing tabletop games, players can always look up rules in the rules booklet while playing. This is less convenient than in computer games since it takes more effort.

In a CATG a lot of the components are tangible. Therefore, it is harder to make information available on demand than in a computer game, where everything is part of the GUI. The CATG needs both a way to react on something happening and somewhere to show the information. Ways of triggering and event can be scanning a component with for example a camera. If the game is played on a touch screen the area near different components can be used for example by placing buttons next to them. If the components have sensors, like RFID, another component can be used to scan them. To actually show the information requested a screen or projector can be used. A smart phone can be a good choice since it can be used both to scan components with the camera and then display information on the screen.

Examples *Golem Arcana* uses a custom pen-like peripheral called the *TDI Stylus*. It allows players to scan miniatures, map tiles and action cards to get more information and perform actions. The information is shown on the screen of a smartphone or tablet that is connected to the stylus with bluethooth.

The board of *Istanbul* consists of place tiles representing different buildings. What tile the player decides to land on during a turn determines what action the player can take. If the game was implemented on a *Microsoft Pixelsense*, help icons could be placed on the tiles. When tapped, they could provide more information about what actions the tiles enable.

Make use of downtime

Description Games often have downtime, meaning times when a player needs to wait before being able to act. In a turn based game players have to wait when it is another player's turn or,

if turns are parallel, other players often need to wait for the slowest player to finish. This is a time were players may turn their attention to other parts of the playing area or even something outside of the game. When downtime happens, it is useful for players to be able to read up on the rules and learn more. This can include reading up on the rules, planning their strategy and get more information about the game state. Therefore, a player that has downtime should still be able to act in some way and have something to do. If following what the other players are doing is of vital importance this guideline should be disregarded.

Many computer games have downtime when loading a new level or area. Often, this screen is used to give strategy tips or more information about the game world. Computer games seldom allow much interaction during downtime. One exception is recent games in the *FIFA* series such as *FIFA 2014*, where the player can perform various practice activities such as penalties while the game is loading. Unlike computer games, there is often a lot of downtime in tabletop games. This time is usually spent observing the active player, reading the rules or planning a future turn. *Catan*[1] minimizes downtime in two way. It allows players with downtime to offer trade deals to the active player. When the dies are rolled by the active player, all players have a direct interest in the result, since it may lead to them gaining resources.

If a CATG has help facilities, such as the ability to scan components to read about them, this should not be disabled during another players turn. The interface should remain responsive to input from all players (of course the game should still adhere to the rules).

Examples In *Augmented Suburbia*, help buttons are placed next to many of the game components, such as the real estate market. The inactive player can interact with them while the other player is having his or her turn.

In *Golem Arcana* each player can have a *CDI Stylus* and a smart phone of their own. When this is the case, each player can scan components to get information of the game state while waiting for the other players.

Provide contextual advice and information

Description Instead of showing all information in an introduction to the game or in a tutorial during play, give advice contextually based on what the players choose to do. In that way, the information will be more easily associated with the action, than if the information was given earlier. This is closely related to one of the principles by Gee (2007) called *Explicit Information On-Demand and Just-in-Time Principle*. Advice can also be given to players when needed, for example when some time has passed since the last action was performed or whenever the game state changes to help hesitating players choose what to do. To related principles of learning (Gee 2007) are the *Explicit Information On-Demand and Just-In-Time Principle* and *Discovery Principle*. The former states that good games give information just in time, when it is actually needed. The second discusses that players should be allowed to explore instead of being guided.

[1] The game is more famous by its old name, *Settlers of Catan*.

Computer games often give information when finding new items. For example, during exploration in the *Zelda* series, when the player finds a new item in a chest, the game gives information about how to use it. In large strategy computer games like *Sid Meier's Civilization V*, there are a lot of choices for the players to do. *Civilization V* contains an advisor panel where several advisors exist to help the player know what to do next. These advisors are teaching the player about different type of gameplay, like military, economy and science. In many card games, explanations are printed on the cards to give information of the cards' effects. Also, in some tabletop games there are an event deck with cards that the players draw at the start or end of a turn. These cards will direct the players what to do next. However, it is hard to give contextual advice for players in tabletop games.

A CATG can use the same principles as when giving contextual advice and information in computer games. Because CATGs often use a tangible board that can be hard to put text on, advice can instead be given to a player's smartphone. Additionally, if the smartphone is private, the amount of advice can vary depending on the experience of the player. If the game is played on a screen, such as a tabletop computer, it is suitable to place contextual advice near the components it is related to. If possible, the amount and complexity of the advice should be related to the player's skill level. Also, make sure to put the advice as close to the related component as possible to avoid misunderstandings.

Examples When playing the updated concept of *Augmented Suburbia* (subsection 7.3.1), information about how gates work will be given to the players when a player passes one of them, instead of teaching about gates beforehand. This reduces the need to give a lot of information before it is relevant, in accordance with the *Discovery Principle*. Other examples of contextual advice in the updated concept are explanations of new effects and that lakes are not introduced until the players are low on money.

In the *Istanbul AI*-concept, when a player lands on a "place tile" that has not been used before, the game gives information on how to use the corresponding action. If the game is played against the AI, the AI players' turns are also used to teach the human players about the game. For example, the if the AI player visits a "place tile" that has not been explained yet, the game will pause and give an explanation.

Complements *Avoid overwhelming the players with new rules and mechanics*

Make use of tangible pieces

Description Make sure to use tangible pieces to indicate a change in the game state. When a change occurs, a manual task then needs to be performed. This is to make sure that the players will notice what is happening to the game state. According to Xu et al. (2011), these manual tasks also maintain the focus of attention between the players. Tangible pieces are also good because they can be easy distinguished on a game board, making the new game state clear. A related principle of learning (Gee 2007) is the *Multimodal Principle*, that states that learning is built through multiple modalities such as images, texts, sounds and interactions. While tangibles are not explicitly mentioned in the principle, they can be viewed as a new modality.

Another related principle is the *"Material Intelligence Principle"*. It states that thinking and knowledge is stored in material objects in the environment, freeing the learner from having to memorize the knowledge without help.

In computer games, usage of tangibles is not common. The only tangible component is usually the game pad or the mouse and keyboard used to control the game. Tabletop games, on the other hand, make heavy use of tangibles. Examples include dice, pawns, cards and markers. Manipulating these components is usually a big part of the experience of playing a tabletop game.

In CATGs, tangibles can be augmented in ways that are impossible in regular tabletop games. They can be automatically tracked so that a computer can keep track of the game state. On the other hand, the use of screens and computers make it possible to remove some of the excise associated with handling tangible components. For example score trackers can be updated automatically and a pair of dice can be simulated. According to Magerkurth et al. (2004) doing this is not necessarily beneficial. He states that using digital dice removes some of the social interactions associated with them. When designing a CATG, the designer has the power to choose what is digital and what is tangible. Since the choices of what is made tangible and not can have big repercussions, they should be made carefully. Automating something is not necessarily a good thing.

Examples The population, income and reputation trackers in *Augmented Suburbia* has to be manually updated with tangible trackers, even though the application knows where they are supposed to be. During our user tests, players often realized how effects worked when updating the trackers and starting to wonder why some values had change. We believe that this would not have happened to the same degree if the trackers had been fully digital.

Another example is *XCOM: The Board Game* where the *Central Officer* instructs the other players where to place UFOs and other tangibles. These tangibles are easy to distinguish from the rest of the the game board since they physically protrude from it. Because the players need to put them on the board themselves instead of having it digital, it is also more likely that they remember the new threat in further actions.

Show available options

Description When it is time for a player to act, make sure to show the currently available options. In this way, the players will have a limited set of options to choose from, making it clear that not everything can be done at the same time. This will make the players less confused and overwhelmed if there are a lot of options to choose from. There is a trade off between how much information to show, for example providing a list of the best choices may both hinder learning and remove the challenge from the game. Also, as the game progresses it may be suitable to show less information and instead make the player responsible for remembering the rules (see *Give players responsibility* for a strategy of how to do this).

Linderoth (2012b) talks about how to support exploratory actions by using visualization tech-

niques in computer games. In computer games, there are several ways to visualize what can be interacted with and how. One way is using a vision mode (Linderoth 2012b). A vision mode is a built-in feature of a game where highlighting is used to indicate what is interactive in the entertainment. In tabletop games, cheat sheets are often used to indicate the general options each turn. These are typically dealt out to each player at the beginning of the game. Sometimes these sheets are combined with a role card, showing individually assigned role options as well. A problem with these sheets are that they may not necessarily match the current game state. For example, a player may not have the required resources for some of the actions displayed.

In CATGs, available options can be dynamically presented to the players based on what they can do right now. Options can, for example, be highlighted on a game board or they can be presented in a smart phone app.

Examples In *Augmented Suburbia*, when it is time for a player to buy a new tile from the market, the border of each market slot will be colored green or red depending on whether the player can afford it or not. This ensures that player does not try to place a tile that is too expensive. This also teaches players how the cost calculation works if they compare their money to the tiles that are green.

In the tabletop game *Carcassonne*, a player draws a tile and places it on the playing area. In an augmented version of this game, positions where where the tile fits can be highlighted. This ensures that the player understands what to do with the tile, and narrows the choices the player needs to make. In the mobile and tablet versions of the game, this type of highlighting is used. However, some players may feel that a big part of the game is to find these available spaces, therefore viewing it as the removal of an important aspect of the game.

Complements *Make the current game state clear*

Encourage learning by observing

Description Having a mentor, teacher or role model is a classical way of learning. This applies to gaming as well, it is common to observe an experienced player. This can be a shortcut to learning the basic rules and more advanced strategies. One way to do this is to show alternative solutions calculated by the computer after the game.

In some computer games, such as *Starcraft*, the best players have fans and the tournaments they participate in are televised. Another example of how experienced players can give knowledge are *Let's Play's*, where players record their experience with a game and post them online. However, it is not necessarily a human player that has to be observed. For example, in many games, such as chess, computer players have surpassed humans. In these cases a human may learn from observing a computer playing. For tabletop games, the preferred way to learn for many players is to have an experienced player present at the table. From our interviews, learning from an expert was the preferred way to learn a tabletop game.

There are alternatives to having an experienced player present in a computer augmented tabletop game. An artificial opponent may show how to perform a move and give insights into what a good first strategy is. A problem with this approach is that if tangibles are used they cannot easily be manipulated by the computer controlled players. One workaround is to let the computer play with digital representatives of the tangibles. Another approach is to let the human players move the tokens for the computer player. An alternative to observing a computer play is to let a group of beginners learn together by observing each others' mistakes and strategies. To accommodate for this, it is suitable to make it easy for the other players to follow what the active player is doing. All players should be able to follow what is happening, especially when the move directly affects them. Xu et al. (2011) write that if the active player moves tangible pieces, the other players are automatically directed to his or her actions.

Examples In *Augmented Suburbia* players are seated next to each other so that they could read each others' messages and easily point at each others' tiles. This resulted in much interaction between the players, often with both players taking turns being the expert and explaining to the other player.

In the *Ricochet Robots Expert Helper*-concept, the best solution can be calculated by the computer based on the robots' positions and the current goal. When the best player has presented a solution, the players have an option to show a better solution. By observing the computers improved solutions the players can learn new tricks for how the robots can be moved. If a player has managed to find the very best solution, it will be a reward to have this confirmed.

Give players responsibility

Description As players become more proficient in a game, they do not need as much hints and guidance any more. What previously was helpful can become an annoyance. While knowing exactly when it is time to stop giving advice is hard, this will be helpful for players that have progressed and do not need reminders of basic rules and mechanics. If turning advice off automatically is impossible an alternative is to provide an option of inactivating or changing the level of help to the players. Being conservative with how much help is offered can also be good for learning. By giving the players the responsibility of remembering the order of they will also gain more understanding of the rules. Instead of telling the players what will happen, they can be reminded of what was supposed to happen if something was done in the wrong order. If a player is given more responsibility, he or she is also more likely to start reflecting and thinking strategically.

As Linderoth (2012b) discusses, computer games sometimes go too far in how much they railroad the experience of playing a game. This leads to little learning occurring while the player progresses in the game. In *Bioshock: Infinite*, the player can always ask for an arrow to show the direction they should walk in for their next objective. This means the player never needs to learn how to find his or her way in the game world. In contrast, *Don't Starve* places the player in a a generated world with the advice that food and a fire may be needed to survive. After that, it is up to the player to devise a strategy for their survival. If the player dies, they have

to start in a newly generated game world, only equipped with the knowledge from their previous sessions. In tabletop games, players are usually given more responsibility. While players sometimes have handy references of what action are available or what phases a turn has, a lot more responsibility is given to the player than in most computer games. If a player disagrees with a rule it is usually easy to change it or ignore it.

In a CATG, a good way to handle responsibility is to give much advice to the player early in the learning process. Later, some reminders can be given. Eventually, it may be suitable to stop giving advice at all, instead only telling the player when a rule has been broken. An interesting aspect of CATGs when it comes to responsiblity is that the designer can choose where on the spectrum between no player responsibility and full player responsibility is placed. One dimension of computer-augmented games (Bergström et al. 2014) that can be related to this guideline is *"Manual" vs. "Automatized" excise*. By automating excise the player is given less responsibility. By choosing to let more excise be manual, the player will instead have to take more responsibility and therefore learn more.

Examples In the *Istanbul AI*-concept the players are given a lot of explanation and a suggested move in the first round. As the game progresses less information is given putting more responsibility on the players. For example, only if the player forgets to perform an action, such as rolling a die, are they reminded of how the place tile they have landed on works.

In *Augmented Suburbia* players are given a lot of advice and help in the beginning. After the tutorial, they are only told when they are performing an incorrect move and reminded to update their markers. It was also discussed, but not implemented, to later remove some of these hints as well, to ensure that the players know what is going on.

Another example in *Augmented Suburbia*. At first, players are reminded after each turn to update. After a few turns, the reminders were removed. During the user tests, the participants never forgot to update the market, even though the reminders were removed.

Conflicts with *Make the current game state clear*, *Enforce the rules*

Complements *Allow rules to be broken*

Be transparent about rules and mechanics

Description It is important for most games that players have at least a basic understanding of their rules and mechanics. If it is unclear how a game works, it will be harder and require more trial and error to find a good strategy. Therefore, game designers should make sure that enough of a games inner workings is explained to make it possible for players to make informed decisions. While it may be convenient to automate things, it may be suitable to let players perform status updates and calculations manually in the beginning for pedagogical purposes. As players become more experienced and prove that they understand how things work, these status updates can become automated leaving only the higher level strategies to the players.

Computer games often include complicated status updates such as physics simulations that are automated by the computer. Since few have the expertise required to understand the inner workings and these rules usually are similar to the real world, there is seldom a point in giving a detailed explanation of them. It is when computer games diverge from how the real world works that it becomes important to explain them. For example, many computer games simulate physics but make it possible to break some of the laws of physics. Examples can include letting players jump high, fly, travel in time and teleport. Another important aspect of computer games is their controls. While many games have similar control schemes, most computer games teach the controls in some way in the beginning. After that, players are often left exploring the game on their own, only getting new descriptions when they encounter new obstacles or get more abilities. Tabletop games rely on players updating the game state. Therefore, the rules need to be clear and non-ambiguous. If this is not the case, players will be confused and have a hard time playing the game. Since players are responsible for enforcing the rules, the rule booklets of tabletop games tend to be comprehensive and include examples of situations that may arise.

A CAGT offers the opportunity to automatically handle some rules, like in computer games. While this may be convenient for experienced players it is not necessarily a good thing for learning. By walking a new player through how a state update is performed they will gain more understanding than by simply giving the result. This is especially true when translating a tabletop game into a CATG. Automation may significantly change how the game is learnt. After players have shown that they know how to perform state updates and how the rules work, it may be suitable to start automating things.

Examples One goal with the design of *Augmented Suburbia* was that players should be able to play the original tabletop game after using the prototype. Therefore, it is of high importance that the rules are clear even when they are automatized. One area that was simplified too much is how the effect of placing a tile is calculated. Players were given a summary of the calculation but no explanation of how to perform the calculation. To further complicate the situation, the effects were not always presented in the order they were calculated. While some players managed to reverse-engineer how the calculation was performed this led to a lot of confusion and questions during the user tests.In the updated concept for *Augmented Suburbia* a suggested solution to this problem is given.

In an early version of *Augmented Suburbia*, the players were given a table (the effects table) with information about the sum of all the tile effects when placing a tile. After user testing this feature we found that players relied on this feature when making decisions. They did not really look at the more comprehensive list of effects and the text on the tiles. Since we wanted to created more awareness of how the effects work, the effects table was removed after user testing it. In later user tests, players seemed to grasp how the effects work faster and think more before choosing a tile.

Linderoth (2012a) discusses how the tabletop game *Kingdoms* is an excellent way of learning addition, subtraction and multiplication. In the tablet version these calculations are performed automatically, thereby removing this aspect and creating a very different game. This changes also contributes to hiding how the game actually works from the player. While this may be

convenient for an experienced player this likely makes it harder for beginners to find good strategies instead simply trying out different placements until they find what is best in the short term. Instead, this automation encourages players to make decisions quickly without reflecting on the long term consequences.

Enforce the rules

Description The more complicated rules a game have, the more likely it is that they will be misinterpreted, forgotten or ignored. For learning purposes this may not be desirable, since it can be harder to unlearn something than to learn something new. To prevent this from happening, one option is to make sure the rules cannot be broken. This makes it more likely that a players will not have misunderstood or made their own assumptions about how a game works after playing it. Also, if rules are enforced automatically, it is possible to have more complex rules, since it is no longer up to the players to enforce them.

Strict enforcement of rules is the norm in computer games. When rules are enforced, there is less need to explain them, since they cannot be broken anyway. Two exceptions to this are bugs or glitches, which may allow players to perform actions that were not intended by the players. Some computer games also allow modification of the rules, from an options screen or with mods (Bergström et al. 2014). In tabletop games it is up to the players to enforce the rules. This allows for house rules, where the group playing a game have agreed to change the rules. However, if all players are beginners, it is easy to completely forget or misinterpret a rule. In this case, tabletop games have no mechanism that corrects these mistakes.

Game designers can make the choice of how strictly to enforce the rules when making a CATG. For beginners it may be suitable to be strict, so no rules or mechanics are forgotten. By clearly alerting the players when an incorrect status update has been made, the players will be made aware of the rules of the game. This makes them more confident they are playing the game in the "right" way. Enforcing rules in a CATG works in the same way as in computer games. The exception is tangibles, which the players may place in incorrect postions. In this situation rules can not be enforced as strictly, it is only possible to alert the players that an incorrect move was made. When rules are enforced and limit the options a player has, it can be suitable to visualize this to the player (see *Show available options*). This guideline affects the *"Low-effort" vs. "High-effort" modification of rules* dimension of computer-augmeted games (Bergström et al. 2014). By enforcing the rules automatically, a higher effort will be required to change them, making the game more similar to a computer game.

Examples In *Augmented Suburbia*, the players are informed about whether their tile placement is correct or not. A common question was whether you were allowed to place two tiles. If this is attempted, the players will be informed that only one tile can be placed. The players are also informed when they cannot afford a tile. When these rules are broken, it is not possible to change turn. This ensures that the rules are enforced.

Golem Arcana has an app that enforces the rules. For example a miniature's range and line of sight is calculated by the application. All moves need to be registered in the app. While it is

physically possible to move a miniature to a disallowed position this will never be reflected in the application, which always has the final say when it comes to the game state. Since *Golem Arcana* has complex rules, similarily to many computer games, it is a suitable design choice to let the computer take care of enforcing them.

Conflicts with *Allow rules to be broken*, *Give players responsibility*

Allow rules to be broken

Description Learning often takes place in a safe environment where different rules and conventions apply. For example, a pilot may be trained in a simulator where a plane crash is not fatal. In a game some rules may be skipped in the beginning to make it easier in the beginning. For training purposes it may be suitable to for example allow moves to be taken back and changed, start with more resources or to make all information public.

A computer game usually makes sure rules are enforced. Some games make the game easier in a tutorial chapter or provide a handicap system to beginners. Tabletop games, on the other hand, allow a lot of flexibility. Since the players are responsible for updating the game state, a beginner will often be allowed to take back a bad move. Also, the game state can easily be modified to create a certain situation, which can help when explaining something. Another example is house rules, were players disagree with a rule in the rule booklet and create their own rules.

When designing a CATG, it is tempting to enforce the rules, in accordance to the *Enforce the rules*-guideline. This is usually technically easier to do when the game state is at least partially controlled by a computer. However, doing this removes the possibility to do many of the modifications to the rules and game state that tabletop games allow. This also means that the pedagogical uses of these modifications are removed. Like in the *Enforce the rules*-guideline, the *"Low-effort" vs. "High-effort" modification of rules* dimension of computer-augmented games is affected (Bergström et al. 2014). However, this guideline moves a game in the opposite direction, making the effort required to change rules lower.

Examples When playing *XCOM: The Board Game* the player with the *Central Officer* role is responsible for entering what happens into the smartphone application. One example is being asked whether the base was destroyed each turn. Since the application has no way to verify whether what was entered is true, the *Central Officer* may enter that the base was not destroyed even if it was. This can allow a group of beginners to keep playing even if the difficulty turned out to be too high.

One feature discussed for *Augmented Suburbia* was the ability to at any time move placed tiles to a new position. When moved, all the markers would update to reflect the new state. While this was never implemented because of technical reasons, it would have allowed a player that made a bad move to take it back.

For being a CATG, *Alchemists* is fairly lenient when it comes to its rules. The app will allow you to scan as many cards as you like, even if you have not assigned enough tokens to your

experiments. This makes it possible to change rules, to give beginners an advantage and to take back a bad move.

Conflicts with *Enforce the rules*

Complements *Give players responsibility*

7.2 Augmented Suburbia

A hi-fi prototype of an computer-augmented version of the tabletop game Suburbia was designed and implemented during this project. The prototype was implemented on the *Microsoft PixelSense* and uses the actual tangible pieces from the game. The players are introduced to the game through a tutorial, detailing the basic rules. After the tutorial, they continue to play the game by getting information displayed on the screen about the effects of each tile. The prototype was designed to help players to learn the game rules together while playing instead of learning everything beforehand.

Two players use the prototype at the same time and the screen is split to allow one side for each of the players. Everything in the prototype is rotated so both players can read while sitting next to each other along one of the long sides of the table.

The screen of the prototype (see Figure 7.1) features two player areas which split the screen in two halves. Each player area has a "home" where the players will keep track of their income and reputation, and a tile grid where tiles will be placed. The tile grids are hexagonal and are not displayed until a player puts down a tile. Additionally, the screen has a common population (score) tracker at the top where both players will track their population during the game. In the middle of the screen is a real estate market with seven tile frames for placing real estate tiles that can be bought. Finally, the blue circles near the market and the player "homes' are slots for the public and the private goals respectively, but these slots have not been made interactive by putting tags on them. The different components of the prototype will be described in more detail in the following sections.

7.2.1 Player area

The player area is divided into two parts, the tile grid and the player "home". The tile grid is where the player puts all of his or her tiles that have been bought during their turn. This grid consists of hexagonal slots that are designed to match the size of the physical tiles of the game.

On each turn, the active player have their currently available tile slots marked with a black border on the grid. The player can choose a tile from the real estate market or take a lake or standard tile from the supply and place it onto the grid on one of the highlighted slots. This is an application of the guideline *Show available options*. When a physical tile is placed on a tile slot, it will be marked with an image of that tile so the player can easily see that the tile is

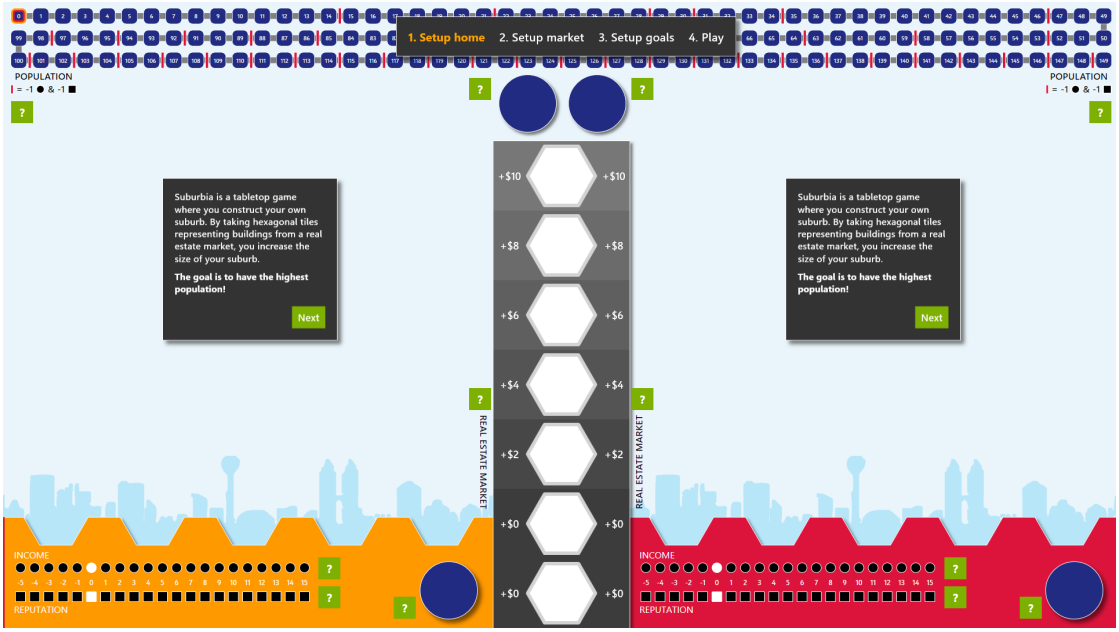


Figure 7.1: The start view of the hi-fi prototype.

recognized by the system. The player is free to move around the tile to test different positions on the grid. The prototype also supports to undo a tile choice and take another tile during the same turn. If the player chooses to take two or more tiles and put them on the grid at the same time, the system will give an error.

The player "home" consists of two trackers: one for the income and one for the reputation, see Figure 7.2. Both of these trackers are similar in appearance, but they represent different values. The income tracker is made of circles and the other is made of squares, matching the markers put on them and the symbols on tiles and elsewhere. The current value is marked by a full white symbol. When the value changes, the previous value is marked by a white border. The players use tangible markers put on the currently full white symbols. If they are currently standing in the wrong positions, they serve as reminders to the players that something has happened. Even though the system keeps track of the level of income and reputation, players need to move the markers to encourage them to reflect on what happens in the game. This is an example of how to use the guideline *Make use of tangible pieces.*

After a tile has been placed on the grid, and the player is happy with the choice, he or she updates the markers, updates the money, and finishes the current turn by moving a turn marker to the opponent's side of the screen. If the placement is legal and the tile was bought from the market, it is now the opponent's turn. If instead the tile was a lake or a standard tile then the player needs to discard a tile from the market before moving the turn marker. Additionally, after each turn, the real estate market needs to be updated by moving down all the tiles and place a new tile from the real estate pile.

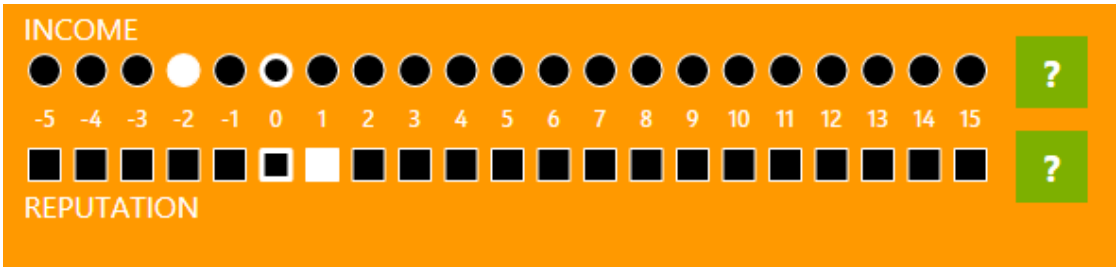


Figure 7.2: The player "home" with income and reputation trackers. The full white symbols are the currently active values. The white borders mark the previous value.

Figure 7.3: The real estate market displaying tiles. Green borders mark tiles that can be afforded and red ones that are too expensive. This is a screen shot, meaning the actual physical tiles are not visible, just their digital "shadows". Note: The image is rotated 90 degrees.

7.2.2 Market

The real estate market is positioned in the middle of screen within easy reach of both players. It consists of seven hexagonal market slots, the same size as the tiles, see Figure 7.3. The background of each slot is different, indicating the price difference between them. However, the two bottom ones are the same color because they have the same cost.

When a physical tile is put on one of the slots, that slot will get a digital representation of the same tile in as an image, indicating that the tile was recognized by the system. That physical tile in the future contain information about the cost of that market slot. When the tile is removed nothing will happen with the image. The image then serves as a reminder where the player needs to put the tile back if it is not wanted anymore. If another tile is placed on an existing image of another tile, then that will be overwritten with the new.

The market component is really flexible and can be adjusted in almost anyway. The slots can take any tile in the game even if they cannot normally be bought from the market such as the lakes and standard tiles. It is even possible to take several tiles and put them back in any order. If it is suitable for the learning of the game, a tile can become cheaper by moving it down to one

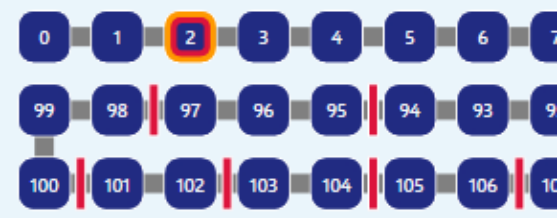


Figure 7.4: One part of the population tracker where both players have the same value, marked with borders in both players' colors.

of the lower slots. This is one way that the guideline *Allow rules to be broken* can be used.

In the beginning of a turn the borders of the real estate market's slots will be updated based on the player's money. If the player is able to afford a tile, that tile's slot will be marked green, otherwise it will be red. When one of the tiles are placed onto the grid, the borders will be grey to indicate that no more tile can be bought at the moment. If the player still tries to place a tile that cannot be afforded, the system will display an error, making use of the guideline *Enforce the rules*.

7.2.3 Population tracker

At the top of the screen is the population tracker. This tracker functions like the two other trackers in the players' "home". The tracker highlights the current value for each player with borders in the players' colors. This highlighting is an application of the *Make the current game state clear*-guideline. If the current values for both players are the same, both borders will be displayed on the same slot, however one of them will be inside of the other, see Figure 7.4.

The population tracker contains the values 0-149. At some points of the tracker gates marked as red lines appear. The frequency they appear is higher at higher values of population.

When the population is changed, the players need to move the population marker to the new position. The players are encouraged to remember to move the markers, even though the application is aware of the score anyways. This is an application of *Make use of tangible pieces* and *Give players responsibility*. It would still be possible to play the game without the markers.

7.2.4 Tutorial

The tutorial in the prototype introduces the players to the game and the rules. It starts with teaching about the basic terms of Suburbia, e.g. reputation, income and population and continues to teach about the first few turns. Finally, the tutorial brings up more advanced features of the game such as lakes, standard tiles, and discards, before it ends and the play mode of the prototype starts. The tutorial is an example of how the guideline *Avoid overwhelming the players with new rules and mechanics* can be used.

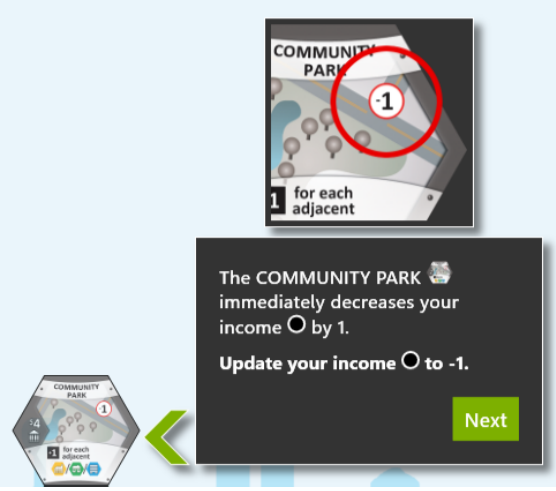


Figure 7.5: A dialog with a help image that highlights the tile's effect.

The tutorial dialog texts are split into a description and an action. They are independent of each other so one could be used without the other. Icons are associated with each of the basic terms and were used throughout the texts where the terms are mentioned in order to help the players to remember the meaning of the terms. Above or below the text dialogs is an option to put a descriptive image to further enhance the learning experience. Images of the tiles with circles around the details are used in the beginning to try to get the players to look at the physical tiles when learning the effects, see Figure 7.5.

At the top of the screen is a tutorial legend to keep track of where in the tutorial the players currently are. The four steps of the tutorial are "Setup home", "Setup market", "Setup goals" and "Play".

For the most part, both players have the same text to read but in their own tutorial dialog placed on their side of the screen. However, at several times of the tutorial the dialogs are split and individually designed texts are shown to the players. For example, when asked to put their population marker on the population tracker, the player to the right is informed to ask the other person to help. The tutorial dialogs can also be placed separately from each other and it is possible to only show only one dialog at any step. This was meant to get both players to read the same text and do the action together and is one way to make use of the guideline *Encourage learning by observing*.

The tutorial is synced between the two players and both has to press their OK button to continue. When one player has pressed the button the text "Waiting for opponent..." will appear (see Figure 7.6). The next step of the tutorial can also be triggered by placing the correct physical tile on the corresponding place (see Figure 7.7), or if the turn marker is moved to the other player's side of the screen.

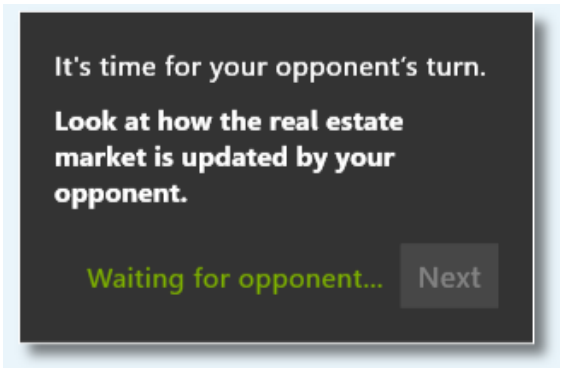


Figure 7.6: A dialog when the button labeled "Next" has been pressed.

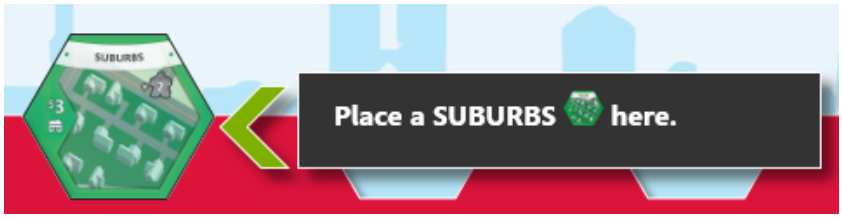


Figure 7.7: A dialog without a button that is triggered by placing a physical tile on the indicated position.

7.2.5 Effect details

After a tile has been placed onto a player's tile grid, a window appears at the top of the screen with details about the tile's effects, see Figure 7.8. The effects are split into two phases: Tile placement and Collection. In the tile placement phase every effect that is part of the currently placed tiles on the players' grids are presented, while the collection phase only presents the players' population and money change from reputation and income respectively. When this change causes a gate to be passed on the population tracker, the effects are also presented here.

All the tile placement effects are separated and sorted by the tile that is associated with it. The placement effect and the cost of the tile are always presented. If the tile placement causes an adjacent tile to trigger its effect, that will be presented connected to that tile's effects. This is also true when a tile is triggered by any of your tiles as well as when triggered by your opponent's tiles. When the later effect is triggered, the effect text will begin with "Opponent's".

If your tile placement triggers an update of one of your opponent's tiles, a similar detail window will appear on the opponent's side of the screen in order to make sure that the player updates markers and money, see Figure 7.9.

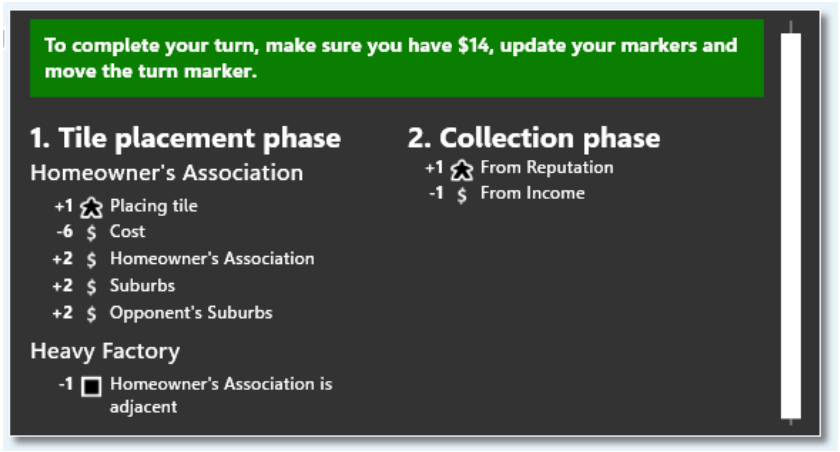


Figure 7.8: Details of the effects from placing a "Homeowner's Association" tile.

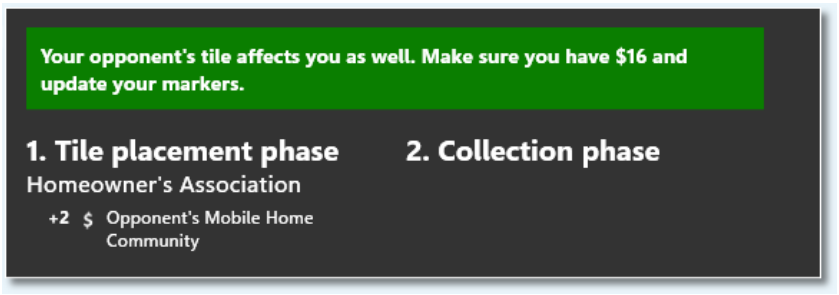


Figure 7.9: Details of the effects when your tile triggers an effect of your opponent.

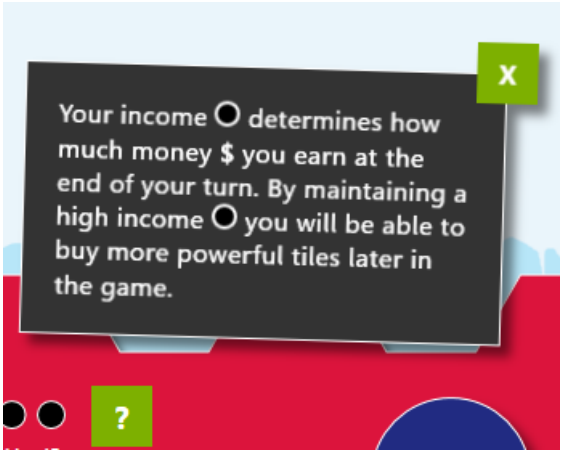


Figure 7.10: A help dialog for the income component of the prototype.

7.2.6 Help dialogs

In addition to the tutorial, players can press the question marks that are positioned near the different components. The question marks work as toggle buttons to toggle the visibility of help dialogs, see Figure 7.10. The help dialogs can be moved around and be rotated on the screen if necessary. Help dialogs exist for the following components:

- Population
- Income
- Reputation
- Real estate market
- Public goals
- Private goal

These help dialogs can be accessed at all times, even if it is not your turn. This is an example of how both the guideline *Make use of downtime* and *Allow players to get information on demand* can be used.

7.2.7 Tile and goal supply

The tile and goal supply is the only component that is not on the screen of the *Microsoft Pixelsense*. It is pictured in Figure 7.11. Instead, it is put on the edge of the screen, above the rest of the game. The tile and goal supply is simply where tiles and goals are stored until they are put in play. Each of the piles in the supply have labels explaining what what they contain. In the tutorial, the purpose of the various types of tiles is explained more clearly. The tile and

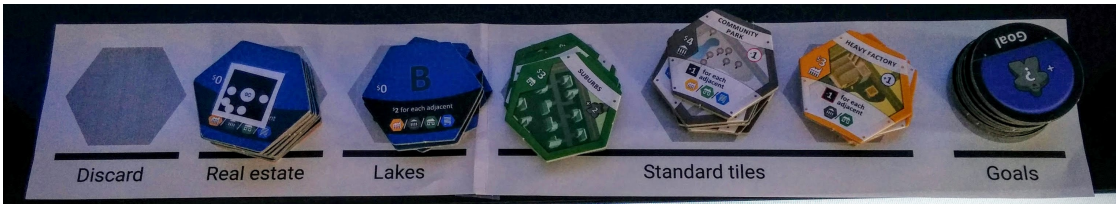


Figure 7.11: The tile and goal supply is the only component that is not on the screen.

goal supply was not put on the screen to save space and since there was no need to make it interactive.

7.2.8 Technology behind the prototype

The prototype was implemented on a *Microsoft PixelSense* with the framework called *Windows Presentation Foundation* (WPF). The code language of WPF is C#. This framework enabled us to quickly get access to user interface components such as dialog windows and buttons.

In the project, resource dictionaries that are defined in XAML are used to separate pure content from code logic. Each step in the tutorial is defined separately, and they are organized in five different files for different parts. At the end of each part the next is started.

Code from a sample project called *Tag Visualizer Events*[2] found from the Microsoft Developer Network was adapted to enable the tiles to be recognized by the *PixelSense*. The tags that are used to do this are called *Byte Tags* and are defined by an identifier number ranging from 0 to 255. The tile definitions are defined in a resource dictionary, containing information like name, cost, image, placing effects and a list of additional effects. Each tile definition is then combined with a unique tag. Doing it in this way, we do not have to copy the tile information for each copy of the same tile in the original game.

7.3 Concepts

In parallel with the creation of the guidelines, concepts were developed both to give new ideas for guidelines and to show how the guidelines can be used. The first concept is a further development of the prototype with features that are not yet included in the prototype. The other two concepts are translations of two other tabletop games into CATGs.

7.3.1 Augmented Suburbia revisited

While our prototype of *Suburbia* works, a lot has been learnt in the process. Therefore, we have created a concept that reflects our vision if we would start from scratch with a new pro-

[2]Retrieved from https://msdn.microsoft.com/en-us/library/ff727918.aspx (2015-05-21)

totype. In this concept, only the aspects that are significantly different from the prototype are mentioned.

The biggest change in the new prototype is a shift from having a long tutorial before playing to using *Avoid overwhelming the players with new rules and mechanics* and *Provide contextual advice and information*. An example is giving information about how the "for every"-effect works when interacting with a tile that has that type of effect. Another example would be to introduce lakes when the player has little money instead of always giving a lot of information during the second turn. A third example is to suggest trying a different position when a tile has been placed in a position where it has a negative effect.

The goal with this shift away from a long tutorial is to avoid showing too much and irrelevant information to the players. During our testing, players often missed or forgot important information only to be confused when it was really needed. By lowering the amount of information given in the beginning and instead introducing rules when they become relevant, we hope to lower the burden put on the players' memory. By not overwhelming the player with information that may not be relevant at the moment, it becomes more likely that the player actually reads the information instead of dismissing it.

Another important change is to make the order of effects during a turn clearer. Since we failed to follow *Be transparent about rules and mechanics*, some testers were confused about how to calculate the score updates after a turn. To clarify the rules and how to make the calculation, two changes have been made in the concept. The first is to show the calculation in steps the first time. This would make it clearer to the players how to make the calculations on their own (which several players attempted). The second change would be to show more information about an effect when needed (*Allow players to get information on demand*). This would include a textual description and highlighting the tiles that have participated in creating the effect.

At later stages in the game it may even be suitable to stop showing the calculation of effects. By letting the players perform the calculations themselves, they would gain a deeper understanding of the game. This is explained in more detail in the *Give players responsibility*-guideline. If the players make an incorrect calculation, they could be informed about their mistake before ending the turn.

Since players' often had questions about how a specific tile works, we have introduced a tile scanner in this concept. It is a hexagon placed in the home area. When placing a tile on the tile scanner information about the tile and all of its effects are given. This application of *Allow players to get information on demand* would help answering the player's questions about the tiles in the market and in other places. Another problem this helps to solve is to put more focus on the tiles, in many of the tests the tester's barely looked at the tiles. Instead they relied on the information from the effect summary, which often confused them.

Another improvement is some sort of symbol reference. There are a lot of symbols in the game many which the current prototype expects the players to figure out on their own. While some symbols are explained by question marks with help dialogs, many symbols do not a have a suitable place to put a help dialog. Therefore, we would like to move the explanation of all

symbols to one central place where a short overview of each symbol is given on demand. If more information is needed about a particular symbol its information can be expanded.

To summarize, the updated concept of *Suburbia* would focus more on exploration and being more intelligent about when to show information and hints.

7.3.2 Ricochet Robots Expert Helper

Ricochet Robots features a grid system where robots need to be moved around in order to arrive to a goal. In an augmented version of this game, the board is digital but the robots are still tangible pieces (*Make use of tangible pieces*). The game will be played as normal. However, when the time has run out and a solution is presented by the player who picked the shortest path, the players have an option that the best solution available will be presented digitally in order to learn all the players how they can improve their play. After several turns, more strategies will start to appear. Looking at another solution can be very efficient because it can open up the mind to think differently. This is an example of the guideline *Encourage learning by observing*.

This concept also contains a small tutorial (*Avoid overwhelming the players with new rules and mechanics*) at the start of the game. The players start with putting down the robots in random places on the board. Then the game will begin showing animations of how the robots can be moved according to the rules. These animations take into consideration the robots initial positions to get examples that are easy to grasp rather than if the examples were general animations.

To make it clear what to do when a turn begins (*Make the current game state clear*), the game will highlight the specific goal that was previously drawn on the board. This will make all players have the same chance in finding the shortest path without having to locate the goal first.

The final feature of this concept is a digital tracker of player solutions where the players will fill in the number of moves of their best solutions when they discover them. This will keep track of who submitted the solutions, the number of moves and the order in which the players submitted them. When it is time for the players to present their solutions, the tracker will tell them who is first to present. This can be done by the players themselves, but as a large part of this game is remembering, it can be good to have it written down. Then everyone will look at the best solution together and learn from each other (*Encourage learning by observing*).

7.3.3 Istanbul AI

This concept explores how the board game *Istanbul* could be taught to a single player playing against the computer.

Istanbul is a game with relatively quick and simple turns. Its complexity comes from the fact that during a turn a choice has to be made from around ten possibilities. More complexity is

added from bonus cards and other players' moves.

A good strategy is to identify a route which the player can repeat several times, to be able to pick up spent pawns while completing their goal. To a new player, realizing that this is useful and how to find these routes is hard. Typically, it either requires help from an experienced player or browsing online forums. With one or several opponents controlled by artificial intelligence, the process of learning to find strategies and even how the different "place tiles" work could be accelerated.

At the start of the game the player is given a short introduction to the game's core mechanics. However, the various "place tiles" are not explained in detail at this stage (*Avoid overwhelming the players with new rules and mechanics*).

Next the player is told to observe as the AI opponents make their moves. Their moves are shown in a slow pace so the player can follow what happens. When they enter a place tile that has not already been explained an explanation of the tile is given. During the first round, the AI players visit the types of tiles that are the most important in the beginning. This is an application of *Encourage learning by observing* in combination with *Avoid overwhelming the players with new rules and mechanics*.

When it is time for his or her first turn, the player will be familiar with the most relevant actions and can make a good first move. The GUI highlights which place tiles the player can move to (*Show available options*), making it clear what the options are.

If the player wants information about a tile that has not yet been introduced or wants a reminder, the tile can simply be tapped for information about it. (*Allow players to get information on demand*).

After a few rounds, the player will have learnt about all the different place tiles. If the player is observant, he or she may also have noticed that the different AI players follows different patterns. This is because each AI player has been assigned a different strategy. By seeing how the different moves can be chained over turns, the player will learn how to use similar strategies. This is an example of how *Encourage learning by observing* can be used in a more subtle way.

8

Discussion

As shown in the previous chapter, the project has resulted in guidelines, a prototype of a CATG and three concepts of CATGs, all with the focus of creating a good learning experience. In this chapter the process, result, the project's general applicability and potential ethical considerations will be discussed.

8.1 Process reflection

In general, the project plan has been followed. The deadlines have been met and the project has always been progressing without major interruptions. One explanation to this is that the plan was not very detailed to start with, it allowed for a lot of freedom and was improved as the project progressed. Another explanation is that the plan, already from the beginning, took into account that there could be delays. When there were delays and interruptions, there was always something else to work on.

The project was divided into three major phases with different main tasks. This might make it sound like a waterfall process was used, which is true to some extent. However, the phases themselves were divided into smaller iterations, were improvements were gradually made based on feedback. This is closer to an agile process (Fowler et al. 2001). Furthermore, the phases were not strictly limited to only one activity. As can be seen in chapter 6, code was changed between the user tests and early prototypes were user tested as early as in the planning phase. Therefore, the process was more agile than the division into phases makes it seem. The phases should rather be viewed as a part of the project with a main activity, complemented by other activities as needed.

There was one big delay in the project, the prototype was finished two weeks later than planned. Since the user testing phase became one week shorter than planned, the end result is a delay of one week. There are several reasons for this. Since a hi-fi prototype was produced, there was a lot of coding involved. While no major problems were encountered, some features took longer than expected to implement. Also, we tested more than we initially planned, which also took extra time. Since the quality of the prototype was important for the user tests, we did not want the focus to be on bugs or how unfinished the game felt, the game needed to be fully playable

and appear finished. However, some compromises were made as presented in subsection 6.2.8. In the end, the delay did not have a major impact on the project, since the later phases went more smoothly.

When working with the prototype, one issue was to decide when the prototype was good enough, that is ready for testing. At what point would the result from the user tests not become better from improving the prototype? This relates to design being a wicked problem. Thanks to our agile process and the fact that we were performing a formative study, we were able to incrementally improve the prototype between the user tests. These rapid iterations made us more confident in starting our testing phase early on.

In hindsight, the process was much more planned for the prototype development and user testing than for the creation of guidelines and concepts. This can be seen in chapter 6, where much more text is devoted to the prototype than the guidelines. Perhaps, the project would have benefited from planning the process for the guidelines in more thoroughly. However, as has been stated before, the goal of the design of *Augmented Suburbia* has been to inspire and give ideas for guidelines and concepts.

Rogers et al. (2011) list three aspects that guidelines[1] can be derived from. They are:

- Theory-based knowledge
- Experience
- Common sense

Looking at the design process, all three were used when creating the guidelines. While all three were used, there is a need for more of all of them. There is currently little research in the area of how games are learnt. Also, the guidelines are based on the experience of designing only one CATG. The extent of common sense in the guidelines is hard to quantify, but more experience would likely lead to more of it.

8.2 Result reflection

Even though the guidelines is the most important result from this project, the prototype is what made it possible to come up with the guidelines. The prototype took a lot of time to implement but without first hand experience in what it means to design a CATG, we think that it would be harder to understand what possibilities and limitations that technology has when designing games. By designing a CATG ourselves we found out what can be done and what cannot. While *Augmented Suburbia* is not the most interesting result of the project in itself, it was an integral part of the process that contributed a lot to the guidelines.

Also many of our guidelines were designed by investigating the prototype through user testing. The testers came up with lots of ideas after having tested a hi-fi prototype. It was important for us to take the time it took to bring the prototype to a hi-fi level, otherwise it would be harder

[1] Rogers et al. (2011) refer to guidelines as principles in the text.

for them to think about what else can be possible with technology. If the prototype would be lo-fi instead, the testers would probably not have behaved in the same way when using it. Also, a game that makes use of a computer is hard to test with a lo-fi prototype because of the sheer amount of possibilities than can arise. A lo-fi prototype, with for example the emphWizard of Oz-technique, would make status updates and calculations of scores to slow for a game like *Suburbia*. To summarize, we believe a hi-fi prototype was necessary to get our testers to behave as if using a real and complete product and become more immersed in the experience.

The guidelines were designed to be used by other game designers when designing new CATGs. Even though the guidelines are based on the experience of designing only one CATG, we believe the guidelines are applicable when designing other games. They are also further based on existing game design theories, previous knowledge in designing games and experience in designing additional concepts in this project. These concepts were designed both based on our guidelines, but they also generated new guidelines. Working with the design makes you understand how designers work and therefore we could take that into consideration when designing our guidelines. The two-way use of the concepts and guidelines was interesting to see.

The concepts are the third result of the project. They serve as a complement to the guidelines, to show how they can be used when creating a CATG. They were also a useful tool when creating the guidelines. By developing them in parallel with the guidelines, they helped us figure out how to phrase the guidelines. They also inspired some new guidelines. A third important aspect of the concepts is that they could be used as examples in the guidelines themselves. Since there are not a lot of examples of CATGs, they were able to serve as examples when no suitable examples were found in already existing CATGs.

An interesting result from this project is that learning is not always synonymous with making things easier or more approachable. While some testers wanted to just have a digital version of *Suburbia* that calculated everything for them, we wanted them to learn by making the calculations themselves. This creates an interesting conflict between players, that may want convenience, and designers, that may want players to learn their game well. An example of this is that when user testing *Augmented Suburbia*, some users remarked that it would be nice if they did not have to update markers and handle the money themselves. Since our goal was to get the users to reflect (in accordance with *Make use of tangible pieces*), this suggestion ran contrary to our goals. It should be noted that learning is not necessarily a goal of the game designer or at least the publisher of a game. Another way of viewing this conflict of interest is that new players want to learn a game as quickly and thoroughly as possible, while experienced players want to have a game that is more efficient and convenient.

There are conflicting views of how learning in games work. On one hand, Gee (2007) thinks that games are good learning tools because players become engaged and can experiment in a controlled environment to mention some of his principles. On the other hand, Linderoth (2012b) argues that games often involve a mechanisms that help players feel that they are learning and progressing, but instead makes the games easier and less demanding, creating an illusion of learning.

One of the concepts we designed after the implementation of the prototype was another take

on *Suburbia* with the same technology, the *PixelSense*. While this may seem like we think that we failed with the prototype, that it not the case. Instead, we wanted to investigate how the two components could be combined differently after we had come up with our guidelines. The new concept is also more demanding to implement. Because we wanted to fully implement a hi-fi prototype in time, we needed to make some choices that was not in line with our guidelines. However, it is important to note that we did not have the guidelines prior to designing the prototype and therefore we wanted to design a new concept that redesigned some of the features found in the prototype. It would be interesting to implement the new concept and compare it to our prototype to get a clearer image of how the two approaches to learning *Suburbia* could be received by players. Another important point is that having a more advanced and complete prototype would not necessarily have been more helpful when creating the guidelines. Many of the guidelines are based on mistakes we made in the process of creating *Augmented Suburbia*.

8.3 Generalization

Throughout the duration of the project, the focus has shifted several times. From the beginning, the goal was to produce guidelines for how to make tabletop games easier to learn by translating them into CATGs. This was eventually broadened to how CATGs should be designed to make them easy to learn. The last shift was to make the guidelines even more general, so that they can be applied to tabletop and computer game design as well.

In addition to how the guidelines should be used in CATGs, they also include examples of how they are already used in tabletop games and computer games. After some of the guidelines were designed, we discovered that other types of games already use them. We added this because we wanted to make it clear that our guidelines are not specifically designed for CATGs, even though we focused on them. This marks a departure from the initial plan, which was to narrow the scope of the project in later phases.

8.4 Future work

In the future we would like to test the usefulness of the guidelines we designed. It would be interesting to let other designers that are not familiar with our project look at the guidelines and see if they can use them in any way to introduce a better learning experience in their games.

We would also like to get feedback from designers and players on the guidelines. Having a session where people can give critique and feedback is really important to make the guidelines as good as possible. This is something that we lack in this project, we only had a test session with our prototype, not the guidelines themselves. However, the guidelines were designed at the end of the project.

In their current form, the guidelines are long and comprehensive. The reason for this is to make them clearer, by giving a lot of information and examples. When learning the guidelines we believe this is a good thing. However, when using the guidelines in an actual project, they may instead become problematic, since it is hard to get an overview of them at a glance. A solution to this could be to create a summary of each guideline that is brief enough that they can fit on two pages. These summaries could be used as a check list when evaluating a design. A good starting point for the summaries is the first paragraph of the description in each guideline.

Also, we would like to implement our new concepts into prototypes, much like the prototype we implemented during this project. While doing this, we would be able to improve the guidelines further by adding, removing or altering them, based on our new experience. After the implementation we would be able to test these different concepts and compare them between each other. There is a chance that some of the guidelines are better to use than others, but that has not been a part of this project.

This thesis' subject is interaction design. However, the field of learning is studied by pedagogues and psychologists. An interesting future direction would be to analyze the guidelines and the prototype from these perspectives.

8.5 Ethical considerations

While working with this thesis, we have not found any ethical issues with CAGTs and learning that we consider critical. The issues we have identified will be discussed here.

When adapting a tabletop game into a CATG, there is a risk that players of the original game think too much has changed. An example is how the tablet version of *Kingdoms* also removed one of the challenges of the game (Linderoth 2012a). If original players do not like the changes to a game, there is a risk of losing some players, even if new players are attracted.

While what happens after learning a game is outside of the scope of this thesis, most games are designed to be played more than one time. Therefore, it is important that features that facilitate learning do not get in the way as players become more experienced. Help features should be non-intrusive or be inactivated at later stages of the game. Ideally, the CATG itself should recognize when players do not need help any more. An alternative is to let the players inactivate help features when they do not need them any more.

Another potential issue is how using electronics affects the social aspects of playing a tabletop game. Some testers wondered whether they became less social by using a partly automated system with a screen. As previously discussed, there is a risk of removing the flexibility of tabletop games when using electronics. Examples of this is the possibility of having house rules and cheating. These aspects are hard to quantify, but should be taken seriously.

In our user tests we found that the screen often distracted the testers from looking at the tiles. When using a digital system, it typically becomes a center of the activity. When playing a

computer game or using an application, we are used to have all our focus on the screen. If a screen is used in a CATG, this may not be desirable. Instead, the components or other players may be where the main focus should be. While screens and other distracting electronics can pose a problem in the short term, it might be a habit that will change, as players become more used to CATGs.

As mentioned in chapter 6, the participants of our user tests were friends and acquaintances. This has likely affected how they have acted during the tests and what answers they have given during the interviews. If we had performed a summative study this would have been a big problem since it introduces a lot of bias in the tests. Since our study was formative, with the goal of forming a set of guidelines, we do not think this has had a major impact on our study. Also, we believe that the fact that the testers were comfortable with us may have made them more honest. This is supported by the fact that many of the testers gave criticism and disagreed with the choices we had made.

A potential risk, when computer-augmenting a tabletop game is constraining the player to fewer possible actions and ways of playing. An example is the tutorial of *Augmented Suburbia*, where we have made choices about how and in what order to introduce the rules in. When learning the original tabletop game the user has a lot more choices and freedom. One example of where we have attempted to preserve the choices is the real estate market. Instead of enforcing how the market should be updated, we allow the players to update it whenever they like in any way they like, even if it means breaking the rules. The updated concept, *Augmented Suburbia revisited*, takes these concerns more seriously, by attempting to preserve the exploration and freedom from the original game. It should also be noted that two guidelines (*Give players responsibility* and *Allow rules to be broken*) help ensure that these qualities of tabletop games are preserve when making a CATG.

9

Conclusion

The purpose of this project has been to look into how computer-augmented tabletop games can be made easier to learn. If the experience of learning CATGs is improved, it could be possible to attract more people to play these games. More specifically, the research question of this project is:

> *How can computer-augmented tabletop games be designed to make them easier to learn?*

To answer this question, a prototype of an augmented version of *Suburbia* on a *Microsoft PixelSense* was designed, implemented and evaluated. The prototype is called *Augmented Suburbia*. *Augmented Suburbia* features an introduction tutorial where two players go through the game rules while playing the first few turns of the game. After these turns, the prototype continues to guide the players by showing information when they try different tiles and positions on the player grids.

From the experience of designing and user testing *Augmented Suburbia* and existing research, we developed 12 guidelines for how to design CATGs that are easy to learn. These are as follows:

- *Make the current game state clear*
- *Avoid overwhelming the players with new rules and mechanics*
- *Allow players to get information on demand*
- *Make use of downtime*
- *Provide contextual advice and information*
- *Make use of tangible pieces*
- *Show available options*
- *Encourage learning by observing*
- *Give players responsibility*
- *Be transparent about rules and mechanics*
- *Enforce the rules*

- *Allow rules to be broken*

We also designed three concepts where we use the guidelines we created earlier. From these concepts, new guidelines arose as well. The three concepts were *Istanbul AI*, *Ricochet Robots Expert Helper* and an updated version of *Augmented Suburbia*.

When using the guidelines, care should be taken when deciding how to follow them. Some guidelines conflict with each other and others may not be applicable in all situation. Therefore, the examples and concepts should be used to identify where they are suitable to use. Since design is a wicked problem, it should not be taken for granted that games designed by using the guidelines will automatically be easy to learn.

The guidelines are just a first exploration of the area of learning in CATGs. Since the guidelines were finalized late in the project and are based on experience, they need to be further evaluated and refined. Whether game designers will find the guidelines useful or not needs to be investigated.

After working with learning in games, we have found more similarities between tabletop games, computer games and CATGs than we expected. This is reflected in the guidelines, which we believe have a more general applicability than just CATGs. These similarities need to be investigated further in the future.

Bibliography

Andersen, T. L. et al. (2004). "Designing an Augmented Reality Board Game with Children: The Battleboard 3D Experience". In: *Proceedings of the 2004 Conference on Interaction Design and Children: Building a Community*. IDC '04. Maryland: ACM, pp. 137–138.

Bakker, S. et al. (2007). "Weathergods: Tangible Interaction in a Digital Tabletop Game". In: *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*. TEI '07. Baton Rouge, Louisiana: ACM, pp. 151–152.

Benford, S., Magerkurth, C., and Ljungstrand, P. (2005). "Bridging the Physical and Digital in Pervasive Gaming". In: *Commun. ACM* 48.3, pp. 54–57.

Benyon, D. (2010). "Envisionment". In: *Designing Interactive Systems*. Addison-Wesley. Chap. 8, pp. 177–197.

Bergström, K. and Björk, S. (2014). "The Case for Computer-Augmented Games". In: *Transactions of the Digital Games Research Association* 1.3.

Blomberg, J. et al. (1993). "Ethnographic field methods and their relation to design". In: *Participatory design: Principles and practices*, pp. 123–155.

Bono, E. de (1999). *Six Thinking Hats*. 2nd ed. Back Bay Books.

Buxton, B. (2007). *Sketching User Experiences: getting the design right and the right design*. Morgan Kaufmann.

Caillois, R. and Barash, M. (1961). *Man, Play, and Games*. Sociology/Sport. University of Illinois Press.

Casey, M. M. (2014). *Apps bring new possibilities to board games*. BoingBoing. URL: `http://boingboing.net/2014/10/15/apps-bring-new-possibilities-t.html` (visited on 02/05/2015).

Costikyan, G. (2002). "I Have No Words & I Must Design". In: *Proceedings of Computer Games and Digital Cultures Conference*. Tampere: Tampere University Press.

Courage, C. and Baxter, K. (2005). *Understanding your users: a practical guide to user requirements: methods, tools, and techniques*. Gulf Professional Publishing.

Cusumano, M. A. and Smith, S. A. (1995). "Beyond the waterfall: Software development at Microsoft". In:

Djajadiningrat, J. P., Gaver, W. W., and Fres, J. W. (2000). "Interaction Relabelling and Extreme Characters: Methods for Exploring Aesthetic Interactions". In: *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*. DIS '00. New York City, New York, USA: ACM, pp. 66–71.

Eriksson, D., Peitz, J., and Björk, S. (2005). "Enhancing board games with electronics". In: *Proceedings of the 2nd International Workshop on Pervasive Games-PerGames*. Citeseer.

Floyd, C. (1984). "A Systematic Look at Prototyping". English. In: *Approaches to Prototyping*. Ed. by R. Budde et al. Springer Berlin Heidelberg, pp. 1–18.

Fowler, M. and Highsmith, J. (2001). "The agile manifesto". In: *Software Development* 9.8, pp. 28–35.

Fox, D., Sillito, J., and Maurer, F. (2008). "Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry". In: *Agile, 2008. AGILE'08. Conference*. IEEE, pp. 63–72.

Fullerton, T., Swain, C., and Hoffman, S. (2004). *Game Design Workshop*. CMP Books.

Gee, J. P. (2007). *What Video Games Have to Teach Us About Learning and Literacy*. New York: Palgrave MacMillan.

Graser, M. (2014). *Legendary Buys Felicia Day's Geek & Sundry*. Variety. URL: http://variety.com/2014/digital/news/legendary-buys-felicia-day-geek-sundry-1201275126/ (visited on 01/30/2015).

Hasbro (2015). *About Monopoly*. URL: http://www.hasbro.com/monopoly/en_US/discover/about.cfme (visited on 02/03/2015).

Huizinga, J. (1955). *Homo ludens; a study of the play-element in culture*. Boston: Beacon Press.

ICv2 (2014). *Hobby games market hits $700M*. URL: http://icv2.com/articles/games/view/29326/hobby-games-market-hits-700m (visited on 02/05/2015).

Jones, J. C. (1992). *Design methods*. 2nd ed. John Wiley & Sons.

Koster, R. (2004). *A Theory of Fun for Game Design*. Paraglyph Press.

Lee, W., Woo, W., and Lee, J. (2005). "Tarboard: Tangible augmented reality system for tabletop game environment". In: *2nd International Workshop on Pervasive Gaming Applications, PerGames*. Vol. 5, pp. 2–1.

Linderoth, J. (2012a). *När tekniken stjälper lärandet*. URL: http://spelvetenskap.blogspot.se/2012/03/nar-tekniken-stjalper-larandet.html (visited on 05/21/2015).

– (2012b). "Why gamers don't learn more: An ecological approach to games as learning environments". In: *Journal of Gaming & Virtual Worlds* 4.1, pp. 45–62.

Magerkurth, C. et al. (2004). "Towards the Next Generation of Tabletop Gaming Experiences". In: *Proceedings of Graphics Interface 2004*. GI '04. London, Ontario, Canada: Canadian Human-Computer Communications Society, pp. 73–80.

Mandryk, R. L. and Maranan, D. S. (2002). "False Prophets: Exploring Hybrid Board/Video Games". In: *CHI '02 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '02. Minneapolis, Minnesota, USA: ACM, pp. 640–641.

Medlock, M. C. et al. (2002). "Using the RITE method to improve products: A definition and a case study". In: *Usability Professionals Association* 51.

Michael, D. R. and Chen, S. L. (2005). *Serious games: Games that educate, train, and inform*. Thomson Course Technology.

Müller-Tomfelde, C. and Fjeld, M. (2010). "Introduction: A Short History of Tabletop Research, Technologies, and Products". In: *Tabletops - Horizontal Interactive Displays*. Ed. by C. Müller-Tomfelde. Human-Computer Interaction Series. Springer London, pp. 1–24.

Orbanes, P. E. (2007). *Monopoly: The World's Most Famous Game–and how it Got that Way*. Da Capo Press.

Parish, J. (2012). *Learning Through Level Design with Mario*. 1UP.com. URL: `http://www.1up.com/features/learning-level-design-mario` (visited on 05/27/2015).

Patton, J. (2002). "Hitting the target: adding interaction design to agile software development". In: *OOPSLA 2002 Practitioners Reports*. ACM, 1–ff.

Prensky, M. (2005). "Computer games and learning: Digital game-based learning". In: *Handbook of computer game studies* 18, pp. 97–122.

Rogers, Y., Sharp, H., and Preece, J. (2011). *Interaction design: beyond human-computer interaction*. John Wiley & Sons.

Scott, S. D. and Carpendale, S. (2010). "Theory of Tabletop Territoriality". In: *Tabletops-Horizontal Interactive Displays*. Springer, pp. 357–385.

Smed, J., Hakonen, H., et al. (2003). *Towards a definition of a computer game*. Turku Centre for Computer Science Turku, Finland.

Smith, R. T. and Piekarski, W. (2008). "Public and Private Workspaces on Tabletop Displays". In: *Proceedings of the Ninth Conference on Australasian User Interface - Volume 76*. AUIC '08. Wollongong, Australia: Australian Computer Society, Inc., pp. 51–54.

Song, P., Winkler, S., and Tedjokusumo, J. (2007). "A Tangible Game Interface Using Projector-Camera Systems". English. In: *Human-Computer Interaction. Interaction Platforms and Techniques*. Ed. by J. Jacko. Vol. 4551. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 956–965.

Spool, J. M. (2004). *The KJ-Technique: A Group Process for Establishing Priorities*. User Interface Engineering. URL: `http://www.uie.com/articles/kj_technique` (visited on 02/03/2015).

Sy, D. (2007). "Adapting usability investigations for agile user-centered design". In: *Journal of Usability Studies* 2.3, pp. 112–132.

Wilson, C. (2013). *Brainstorming and Beyond: A User-Centered Design Method*. Morgan Kaufmann.

Xu, Y. et al. (2011). "Chores are fun: Understanding social play in board games for digital tabletop game design". In: *Think design play: The fifth international conference of the digital research association (DIGRA)*. Vol. 16.

Appendices

A

Facebook event

This appendix contains the text from the Facebook *event used to invite testers. In addition to this text a date, an address a picture of the box of* Suburbia *and a picture of the* Microsoft Pixelsense *was available on the event page.*

We are two students at Interaction Design at Chalmers and we are doing a master thesis on how to make board games easier to learn by using electronics. We have developed a prototype that we would like to test to get feedback and new ideas.

The test will involve two persons at a time and is planned to take max 2 hours. We would like to get going with the tests in week 16 and 17 (April 13-26). You will get to play a board game on the Microsoft Pixelsense, a 40" touch table. The test will be in Swedish or English depending on your preferences. Also, there will be fika!

Our prototype is based on the board game Suburbia. Therefore, we need people who have not played and are not familiar with the rules of the the game. Please, don't google it!

If you want to help us, please tell us when you are available with the link below. Mark as many times as you like and we will attempt to pair you with someone. If the available time slots don't fit your schedule, write in the event and we'll see if we can find another time.
[Doodle link removed for privacy reasons]

Feel free to invite people you think might be interested in this.

B

User test script

The following list of keywords was used during the introduction of the user test.

- Make tabletop games easier to learn with the help of electronics.
- Suburbia on the Microsoft Pixelsense
- Today you will:
  - play the game
  - have some coffee
  - answer some questions
- Our goals:
  - Find inspiration and ideas for how to make tabletop games and games in general easier to learn
  - Identify weaknesses and strengths in the prototype.
- While playing:
  - follow the instructions
  - try to figure out problems together if needed
  - ask each other questions
  - as a last resort: ask us for help
  - feel free to speak out whenever you have ideas, suggestions for improvements or find things confusing.
- Permission to record
- Questions?
- Good luck!

C

User test interview questions

These questions were asked individually to all participants after the test. In parentheses are suggestions for things to bring up if the interviewee has a hard time answering the question.

- Name, Age, Sex, Date, Score
- Do you play board games? How do you usually learn these games? Give some examples of games you play.
- Do you play computer or video games? How do you usually learn these games? Give some examples of games you play.
- What are your thoughts on learning Suburbia with our prototype?
- Do you have any ideas for improvements to the prototype?
  - (remind of learning aspect)
- Do you feel that you would be able to start playing Suburbia without the prototype? Motivate why or why not.
- Do you have any ideas on things in board games that would benefit from using electronics?
  - (Try thinking about the games you usually play)
  - (Examples: score summary, keep track of the game status, AI)