NTNU – Trondheim
Norwegian University of
Science and Technology

Master Surgeon

A Serious Game for Enhancing Ultrasound Interpretation

Astrid Kløve-Graue

Master of Science in Informatics

Submission date:	June 2015
Supervisor:	Alf Inge Wang, IDI
Co-supervisor:	Frank Lindseth, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Abstract

In this thesis we set out to examine theories from game design and gamification to create a serious game that teaches ultrasound interpretation. The motivation behind this research has been the need for surgeons, technicians and students to be able to practise this skill outside the operation room. To be able to read ultrasound images, a lot of practice is required, and we hope that through this serious game we will make this experience easier to obtain and available for a larger group of people.

The focus of this thesis was to discover success factors for serious games and find theories from game design that affects how engaging a gaming experience is. This information was used to developed a serious game prototype named *Master Surgeon,* primarily for smart-devices.

After the final prototype was implemented the game was evaluated by surgeons, technicians and students. The results from the evaluation revealed that *Master Surgeon* does present a fun way of practising ultrasound interpretation, but that there is still room for improvement regarding how engaging the game is.

Sammendrag

I denne oppgaven ønsket vi å undersøke teorier fra spilldesign og spillifisering for å videre bruke disse til å skape et nytting-spill som kan brukes til å lære ultralyd tolkning. Motivasjonen bak denne undersøkelsen var behovet kirurger, teknikere og studenter har for en mulighet til å videreutvikle denne ferdigheten utenfor operasjonsstuen. Det kreves mye trening for å kunne lese ultralydbilder og vi håper at vi gjennom dette nytting-spillet vil vi gjøre det lettere å opparbeide denne typen erfaring for en større gruppe mennesker.

Hovedfokuset for denne oppgaven var å oppdage suksessfaktorer for nyttig-spill og teorier fra spilldesign som påvirker hvor engasjerende en spillopplevelse er. Denne informasjonen ble brukt til å utvikle en prototype av et nytting-spill kalt *Master Surgeon*, først og fremst for smart-enheter.

Spillet ble evaluert av to brukergrupper, både ved bruk av touch-enheter og en stasjonær datamaskin. Resultatene fra evalueringen viste at *Master Surgeon* presenterer en morsom måte å trene på ultralyd tolkning, men at det fortsatt er rom for forbedringer når det gjelder hvor engasjerende spillet er.

Assignment Text

The national competence center for ultrasound and image guided therapy was established in 1995 and is a collaboration between SINTEF, NTNU and St Olav. Within the center ultrasound-based navigation technology is being developed and used in various clinical fields in order to guide surgical instruments safely into the human body (similar to GPS navigation). One of the main challenges in order to make use of this technology is sufficient knowledge about interpreting ultrasound images. Each week unique data material is collected within the center. If presented in an easy and knowledgeable way, selected parts of the anonymous material can be used to teach relevant groups to interpret ultrasound images in an entertaining manner. A serious game for mobile devices (iOS and maybe Android) has huge potential in this regard.

The possibilities are almost endless but some basic suggestions would be to learn ultrasound interpretations of brain images by presenting:

- Anatomy – The user sees an ultrasound image as well as names of anatomical structures and is challenged to drag each name onto the correct structure.
- Pathology – The user sees ultrasound images of various tumors and is asked to identify the tumor and state the type (multiple choice).
- Tumor borders – Some tumor types have diffuse border to normal tissue. The user sees such images and is asked to draw the border.

A scoring system should be developed and based on the available data material random questions, tutorials and skills validation can then be generated. This can be made very advanced with global user profiles and benchmarking but it's probably vise to start easy.

It should be easy to generate and upload new material and when new cases become available the users are notified and asked it they would like to download the new material.

Acknowledgements

This document is the master thesis of Astrid Kløve-Graue, and concludes my master's degree in Informatics.

I would like to thank my supervisor Frank Lindseth for his guidance during the creation of this thesis and all his collages at SINTEF and St. Olav Hospital who helped out with evaluating the final prototype.

I would also like to give a special thanks to my family and friends who provided invaluable moral support throughout this project. Especially you Vinny.

Astrid Kløve-Graue
Trondheim June 15th, 2015

Table of Contents

List of Tables

List of Figures

Part I

Introduction and Research

Chapter 1
Introduction

1.1 Background

Ultrasound imaging is one of our most popular and least expensive medical imaging modalities[1]. If offers great safety to both operator and patient and shows promise as an intra-operative imaging modality because of its real-time imaging and portability. However, interpreting ultrasound images requires years of experience, and currently there is no good alternative for students to acquire this experience apart from attending surgery.

Serious games are games designed for a serious purpose, in addition to being fun. With its initial foothold mainly in the military area, serious games have during the second half of the 20th century expanded through digitalization to education and business and have grown into a substantial industry and research field[2,3]. To create a serious game that can successfully teach ultrasound interpretation would be a great advantage to ensure the future of ultrasound as an intra-operative imaging modality.

With the introduction of smartphones people now have the opportunity to play games whenever and wherever they go. As they have become more popular and more capable as a gaming platform they offer great potential for game developers to distribute their game in an inexpensive way to a large audience.

1.2 Project goal

The goal of this project is to use the collected image data from the national competence center for ultrasound and image guided therapy(USIGT), together with theories from game design and gamification to create a serious game prototype that is both entertaining and educational. The purpose of the game will be to give surgeons, technicians and students a safe environment to improve their ability to interpret ultrasound images. Further, the application will need some support for creating new content and sharing it with users. After development the application should be evaluated by users, consisting of both players and the people responsible for creating new content, to determine its educational value and general usability.

An earlier version of this application has already been developed, but supports only iOS-devices. To redevelop that application with a cross-platform framework would expand the range of possible users, and make updating and further work on the application easier.

The application that the USIGT wants consists of two parts. The first part should handle storing and displaying their data material in a convenient way, and this is currently in development by a different project. The second part, and the focus of this project, will be the educational game section of the application. A future merge of these two projects is planned and should be considered throughout development.

The name of this serious game will be *Master Surgeon*.

1.3 Reader's guide

The thesis has been divided into four parts:

- Part I contains the introduction of the thesis where the background and goal for the project is described. The research method and process are also presented here.

- Part II contains a prestudy where information regarding gamification and serious games are presented. In addition a short analysis of relevant games and an introduction of the Qt framework are also included here.

- Part III presents the contribution made by this thesis. The final game will be explained and both the game design and implementation process is described in detail.

- Part IV consists of the prototype evaluation and a discussion based on the results from these tests. A conclusion is made where the research questions are discussed and finally some suggestions for future work are made.

Chapter 2
Research

The main purpose of this thesis is to design and develop a serious game that can improve players ultrasound interpretation skills. In light of this we aim to discover the secrets of “fun” and how to create it in games. In this chapter we will define research questions that can help answer these questions, and both the research method and process are described.

2.1 How to make serious fun

Most game developers aim to make a game that is fun. Even if one is in it just for the money, it is necessary to get the player to buy the game and play it for an extensive period of time. A game which is boring or poorly made will quickly be noticed as unfunny, and put away after a few minutes of play or simply never bought. But what is it that makes a game fun? What can be learned from previous games and game design about creating a fun user experience?

Using a game for educational purposes offers its own challenges. It is possible to play a educational game that is not fun, either because one has to or one ought to, but this defeats the purpose of creating the game in the first place. A serious game should aim to be engaging, in fact so engaging that the player feels that what they are doing is not work, but play. Can this be achieved? Is it possible to balance the seriousness and the fun?

Many of today's most popular games are multiplayer games, i.e. games where people play with or against each other. Some well known ones are Minecraft, League of Legends, Counter Strike, World of Warcraft and FarmVille. The rules and structure of these games, or game play, are vastly different, but they all offer some form of competition and cooperation between players. Even game franchises that traditionally were single player games have started to add support for several players in some of their content, as seen in the new Assassin's Creed Unity[1] and Grand Theft Auto V[2]. Is it possible that competition and cooperation are important factors for the success of game? How do they affect engagement?

[1] http://assassinscreed.ubi.com/en-GB/games/assassins-creed-unity/new-multiplayer/index.aspx

[2] http://www.rockstargames.com/V/info/GTAOnline

2.2 Research questions summarized

Based on the questions above, four research questions have been created:

RQ1	What makes a game fun?
RQ2	Which factors affect the success of an educational game?
RQ3	Can theories from gamification and game design be used to better tailor an engaging learning experience?
RQ4	Do competition and cooperation contribute to how fun a game is?
RQ5	Can we create a game that teaches ultrasound interpretation and at the same time is perceived as fun?

Table 2.1: Research question

2.3 Research method

The engineering method was chosen as the research method for this project since it is well suited for prototype development. In addition, a literature study will be conducted to attempt to find answers to some of the research questions from previous work. The engineering method is described by Marvin V. Zelkowitz[4] as “A solution to a hypothesis is developed and tested. Based upon the result of the test, the solution is improved, until no further improvement is required”.

In this project however, we are limited by two factors, with the first one being the time limit for this project. Creating a finished game with all its mechanics and artwork requires time, and with the addition of the literature study, there will only be room for so many iterations during the time period allowed for this project. The second factor is limited access to a relevant group of people that can take part in evaluation of the prototype. Given the specific educational content of the application it is desired that the people who evaluate it have some experience with ultrasound, and can be able to give feedback on the educational value of the application. It has however, been requested from a representative from USIGT that medical students and personnel will only be included in any evaluation when a working prototype has been developed. This is not unreasonable, as evaluation may be a time consuming process. To solve this, the representative from USIGT will instead take part in each iteration, and help evaluate the progress of the prototype.

2.4 Research process

The research process will be divided into four steps: research, design, implementation and evaluation.

2.4.1 Research

A literature study will be performed to find relevant articles to help answer the selected research question and make decisions concerning design. In addition some research will be done on medical imaging, the selected framework, and related games. Learning more about medical imaging will give a greater understanding of the purpose of the application and can be used to better design *Master Surgeon* to fit the educational content. Secondly, the prototype is being developed in an unfamiliar framework and some research into how it is structured will help make decisions concerning implementation. Lastly, we will look at some popular smartphone games to get some inspiration for the design of *Master Surgeon*.

2.4.2 Design

In the design phase the information from the literature study will be used to suggest solutions for the design and implementation of *Master Surgeon*. For an initial evaluation of the design, a paper prototype will be created, for later to be tested and discussed with the representative from USIGT. This paper prototype is a quick and cheap way to check how the design will work, and to get some valuable feedback early in development.

2.4.3 Implementation

When the paper prototype has proven successful, functional requirements will be created and the the implementation may begin. During implementation, the prototype will be evaluated and discussed with the representative from USIGT, and it will be considered finished and ready for evaluation when all functional requirements have been met, or when there is no more time.

2.4.4 Evaluation

A final evaluation will be done at the end of this project. The goal is to find volunteers fitting the expected user group to participate in experiments. During the experiments players will be given different tasks to complete within the game. After the task has been completed they will be asked to fill out a survey based on their impression of the prototype. If necessary short interviews might be performed at the end to determine things that were unclear or discuss future improvements. The tool used to create new content for the game should also be evaluated with the same type of experiments, but mainly to determine its usability.

2.5 Development tools

2.5.1 Qt

As mentioned in the introduction of this thesis, a future merge of *Master Surgeon* and a separate application is planned, and working with the same framework is a clear advantage for this to be an easier process. This other application is currently in development, and the creators have selected Qt as their main framework. In light of this decision, the chosen development tool for *Master Surgeon* will also be Qt. This is a framework that has been designed for easy creation of UI-centered[3] applications which is a good fit for this project.

2.5.2 Adobe Photoshop

To create the graphical content for the game, a drawing tool is needed. With some previous experience in Adobe Photoshop, this will be the preferred tool for creating this content. Photoshop is an advanced image editing tool which supports a vast number of file formats.

[3] http://en.wikipedia.org/wiki/User_interface_design

Part II

Prestudy

Chapter 3

Medical Imaging

The national competence center for ultrasound and image guided therapy is a collaboration between surgeons, engineers, and scientists, with the goal of improving the methods and techniques that are being used in surgery today[5]. Their main area of expertise is image-guided-therapy, the use of medical imaging to plan, perform and evaluate surgical procedures. Medical imaging is the process of scanning the inside of the human body, and several different technologies exist that create different kinds of images. The images used in *Master Surgeon* are a fusion of data from different imaging modalities. To better understand the need for *Master Surgeon* and its educational content, we provide a short introduction on how and why these images are made.

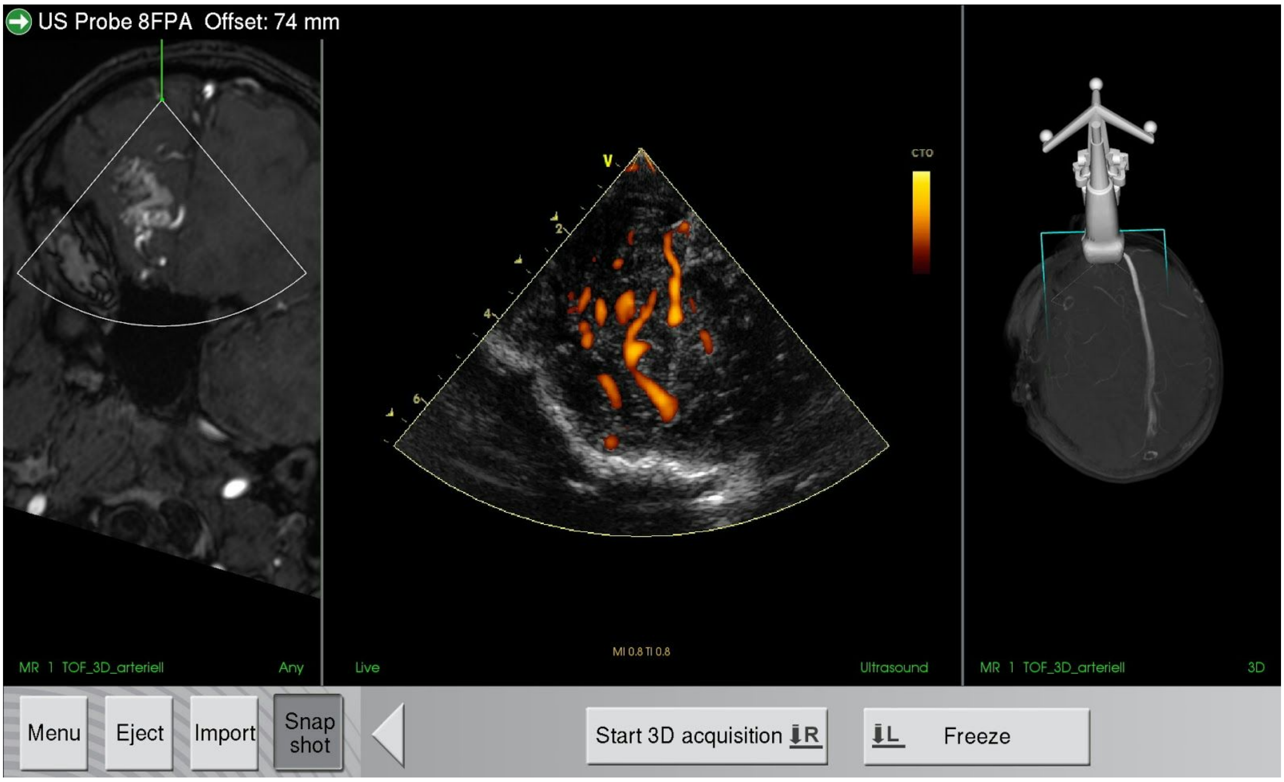


Figure 3.1: Combined MRI and Ultrasound image

3.1 MRI

Magnetic resonance imaging, or MRI, is one of the medical imaging techniques used by USIGT. These scanners use the principle of magnetic fields and radio waves to create high quality images of the inside of the human body[6]. This gives MRI a safety advantage over the much used computerized tomography scan which exposes both the operator and the patient to ionizing radiation. MRI also has higher sensitivity to tissue contrast, making it the tool of choice for neurological cancer[7]. The leftmost part of Figure 3.1 shows a MRI slice of the brain while the rightmost part shows a 3D representation of several slices.

Despite the detailed images and safety of MRI it also has some drawbacks. In comparison with other imaging modalities it has a higher cost and more time is needed to produce the images. Any movement from the patient or the presence of metallic objects will cause distortion in the images. The standard MRI scanner is shaped much like a oblong donut, and the patient is required to lie inside the center. This might cause patients to experience psychological distress, such as anxiety, claustrophobia or panic attacks[8].

3.2 Ultrasound

Ultrasound imaging is currently one of the most popular and least expensive medical imaging techniques[1]. No insertion into the body is required and it can be made portable. The technique is based on high frequency sound waves that are sent into the body from an ultrasound probe, and when these waves hit tissue boundaries, an echo pulse is reflected. The time interval between sending and receiving can be used to calculate the distance to the position of the tissue boundary. The middle part of figure 3.1 shows an image slice from an ultrasound scan.

Although ultrasound is considered a safe image modality the wave energy might cause some bioeffects that one should be aware of. Thermal bioeffects are caused by the ultrasound energy heating nearby tissue and this is more applicable the larger the pulse lengths are. Mechanical bioeffects are caused by sound pressures large enough to tear tissue apart. This is more likely to happen with shorter pulse length. There are, however, regulations on the output intensity levels, and most modern systems come with indicators enabling the operator to control that the pulse length is within acceptable range[1].

A major challenge with ultrasound images is the readability of the images. As seen in figure 3.1, the ultrasound image lacks the detail shown in the MRI on the left. For a surgeon to be able to make accurate decisions concerning something as delicate as the brain based on ultrasound images, it is required that this person is well trained in reading this kind of images.

3.3 Intra-operative imaging and ultrasound-based guidance

In connection with surgery medical imaging is used in three stages; before, during and after an operation. The first step, called pre-operative imaging, is used for general diagnosing and planning the surgery. The second step, called intra-operative imaging, is used during an operation for guiding purposes . Lastly, post-operative imaging is used to evaluate the result after surgery. The images used in *Master Surgeon* are used during surgery and are a combination of pre-operative MRI images and intra-operative ultrasound images.

By using both image modalities the surgeon has the advantage of the high detailed MRI images, while the real time images from the ultrasound shows the current state of the body. The reason the images from ultrasound is so important for the surgeon to have, is that as a surgery progresses, the pre-operative image may no longer accurately describe the current state of the body. For example in the cases used in this project, where the goal is to remove tumors from the brain, there is a possibility that tissue around the tumor changes positions as the tumor is being removed. This phenomenon, called brain-shift, is a major source of error in neurosurgery based on pre-operative imaging[9].

Chapter 4
Serious games and Gamification

This project aims to design an engaging learning experience, that can help students and surgeons alike to better read ultrasound images. In the research chapter of this thesis, we stated that this was to be done using theory from game design and gamification. Gamification as a term was first used in 2008[2] and as it is a relatively new concept, a clear definition is yet to be established. In this chapter we will discuss what gamification is, what serious games are and see how theories from game design may be used to improve *Master Surgeon* as a learning tool.

4.1 Gamification

As mentioned there is still some dispute on what the actual definition of gamification is. In a study done in 2011[2], where they investigate gamification and discuss whether or not it describes a truly new phenomenon, they propose the definition “the use of game design elements in non-game contexts”, and clearly distinguishes it from serious games as they refer to those as “computer-based game software”. Zichermann[10], however, defines the term gamification as “the process of game-thinking and game mechanics to engage users and solve problems.”, and proceeds to call it an umbrella term that unites concepts like “serious games, advergaming, and games-for-change”. Callan[11] in his article suggests using gamification instead of simulations for some serious games and uses the definition “the addition of elements commonly associated with games (e.g. game mechanics) to an educational or training program in order to make the learning process more engaging.”

From these three definitions one might get an idea of what gamification is about, but for the sake of clearance the definition used in this thesis will be the one presented by Zichermann. This definition is broader than the two others, but highlights what is truly important; using what is known about game design to “engage users and solve problems”. Gamification is based in traditional game design, but with the single focus on the psychology that makes games so appealing. The goal is to take this “fun” and use it to create engagement in situations where there normally is none.

4.2 Serious games

Games used for serious purposes is nothing new and can be traced back several millennia[2]. With its initial foothold mainly in the military area, serious games have during the second half of the 20th century expanded through digitalization to education and business. It has grown into a substantial industry and research field and in 2010 held a market worth of 1.5 billion €[3].

A digital serious game is defined by Deterding[2] as “any form of interactive computer-based game software for one or multiple players to be used on any platform and that has been developed with the intention to be more than entertainment”, and is by this definition required to be a game. This distinguishes it from gamification where no game or “game play” is necessary.

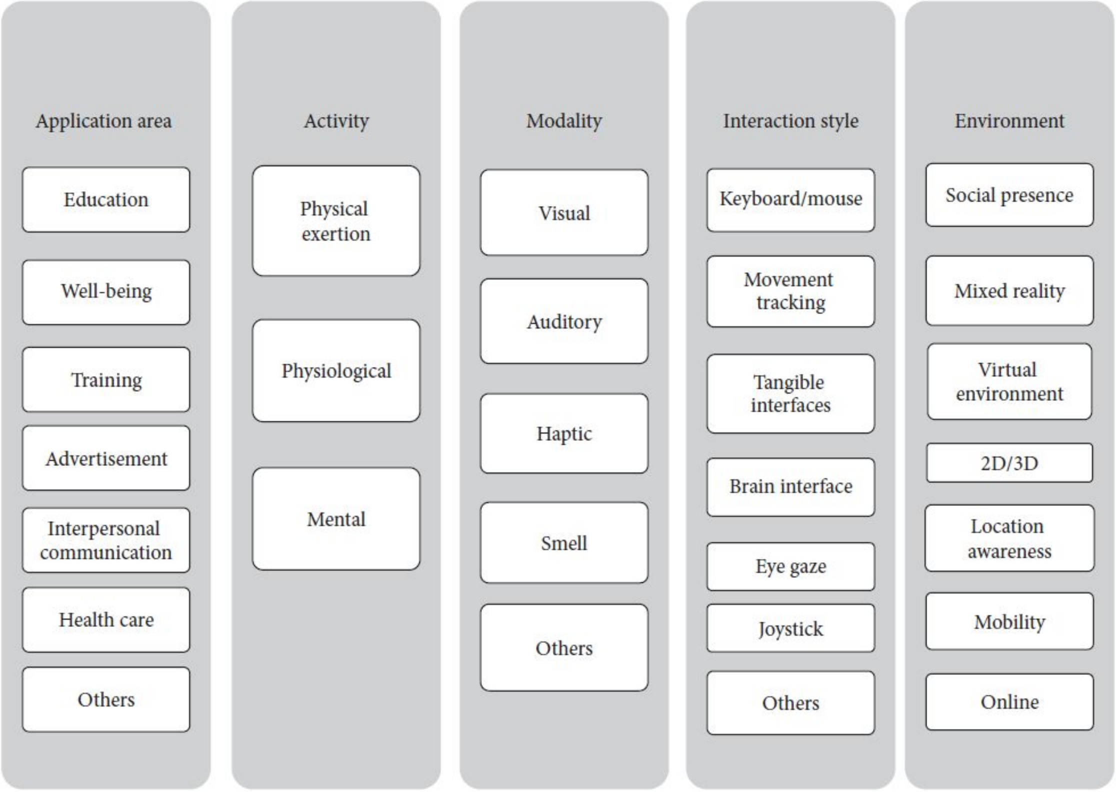


Figure 4.2: Taxonomy of serious games[4].

In an attempt to classify serious games Laamarti[3] defines five characteristics that were found to be important when designing serious games; application area, activity, modality, interaction type and environment. These are all displayed in figure 4.2. Selecting the correct modality, interaction style and environments for the purpose of the game is important in order to enhance the user’s experience, and thereby its success.

Based on their study of previous serious games Laamarti also outlines some guidelines for creating successful serious games. The more general are; providing guidance in the game, avoiding negative feedback as result of low performance, using multiplayer collaborative exercises, and offering challenges. They also suggest that educational games created for classroom use should be based on the curriculum to better be accepted by players.

Callan[11] analyses how casual social games and social network sites can be used for learning and suggest the following guidelines:

- Feedback and rewards must be immediate. This is an important motivational element.
- Rewards must be matched to the difficulty of the completed task, and feel achievable.
- Rewards must be explicitly recognizable in the social context to make them meaningful.
- High usability is crucial to ensuring that the game is played.

4.3 Why we play

Zimmerman[12] presents four general reasons for why people are motivated to play games:

- For mastery
- To destress
- To have fun
- To socialize

By him they are described as “a good working theory”, and a more detailed version of these four reasons are presented by in a paper by Lazzaro[13]. She describes four ways that games create emotion in players:

- Hard fun: Players enjoy challenge, strategy and problem solving. The emotions frequently generated by this are frustration and personal triumph.
- Easy fun: Players enjoy intrigue and curiosity. They display the emotions wonder, awe and mystery.
- Altered States: Players enjoy the way a game makes them feel, and play to achieve excitement or relief from their thoughts and feelings.
- The People Factor: Players use the game as a social platform. The emotions generated from competition includes amusement of and gloating over the misfortune of others, as well as social bonding and personal recognition.

She also found that emotional display dramatically increased when people were playing together. In groups people expressed emotion more frequently and with more intensity, and new behavior and rituals were created, which in turn made playing more exciting.

Bartle[14] in his study of players from Multi-User Dungeon(MUD) games[4] found four different categorisations of why and how people play these kinds of games. His player types will be described separately later in this chapter as they will be an important tool for tailoring *Master Surgeon* to fit the selected user group.

4.4 Flow

For any game to be successful - serious and normal games alike - the developers aim to fully absorb the player in their gaming experience. This state of mind is by the psychological professor Mihaly Csikszentmihalyi described as *flow*, and describes a player's experience in between boredom and anxiety[12, 15]. Figure 3.3 displays mental states in terms of challenge and skill level. Arousal, flow, control, and relaxation are considered positive, while the remaining states are all negative. Owen Schaffer[16] describes the flow experience with these points:

- Intense and focused concentration on the present moment
- Merging of action and awareness
- Loss of reflective self-consciousness (loss of awareness of oneself as a social actor)
- A sense that one can deal with the situation because one knows how to respond to whatever happens next
- Distortion of temporal experience (typically a sense that time has passed faster than normal)
- Experience of the activity as intrinsically rewarding, such that often the end goal is just an excuse for the process

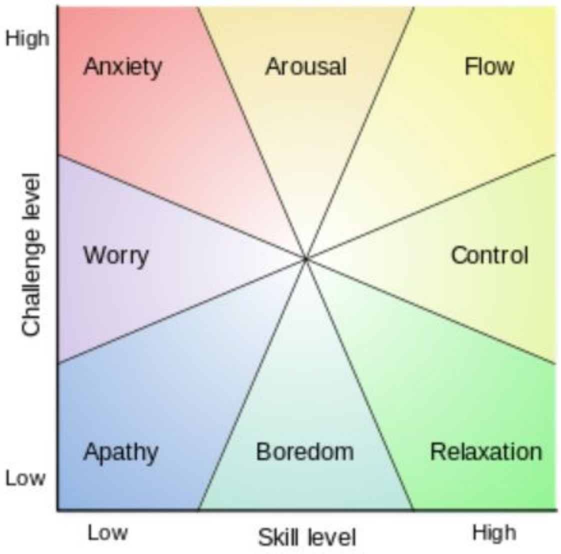


Figure 4.3: The Experience Fluctuation Model showing the 8 channels of experience.[16]

[4] http://en.wikipedia.org/wiki/MUD

He also points out that how flow is experienced stays consistent across different activities, and research done across such diverse activities show how positive this state of mind is. In his paper Schaffer discusses how the ability to achieve flow can be measured, and suggests seven conditions that are required by an activity for participants to be able to achieve flow:

- High perceived challenges
- High perceived skills
- Knowing what to do
- Knowing how to do it
- Knowing how well you are doing
- Knowing where to go(where navigation is involved)
- Freedom from distractions

One of the challenges of maintaining flow is to adjust difficulty level according to the player's skill level. If both the difficulty level and the player's skill level are low we produce the state apathy, where the player will lack interest in continue playing. If the difficulty is low while the player skill is high, we get the state boredom where the player will feel that higher challenges are needed. Lastly, if the player level is low while difficulty level is high we produce anxiety, causing the players unease or distress. Flow will only be achieved when skill level and difficulty level match[15].

Laamarti[3] present some factors that they have found helps serious games be successful. Among them is the suggestion of "providing guidance to player within the game" which would "prevent them from feeling lost", showing that knowing what to do and how to do it is an important factor also for serious games. In addition they list "Keeping the challenges at the right level is key in keeping the players interest in the game." This highlights what has been said of the importance of adjusting difficulty level to make the players feel like they know what they are doing, and that they are doing well.

4.5 Player types

Zichermann[12] suggest Bartle's player types as a rubric which can help developers shape the gaming experience to better fit current or future players. With better knowledge about who one is developing for, the easier it is to drive desired behavior. Based on his studies on players of Multi-User Dungeon games, Richard Bartle[14] developed a well known and still used rubric to group people based on how they play games. He found that he could categories them into four different player types; achievers, explorers, socializers and killers. These types can be displayed on a graph according to how they act or interact with either players or the world, see figure 4.4. Achievers and Explorers are players who prefer acting on the world, or in this context, the game itself. Socializer and Killer types are only visible in multiplayer games, as they describe some form of interaction between human players. Bartle types can be described as follows[12, 14, 17]:

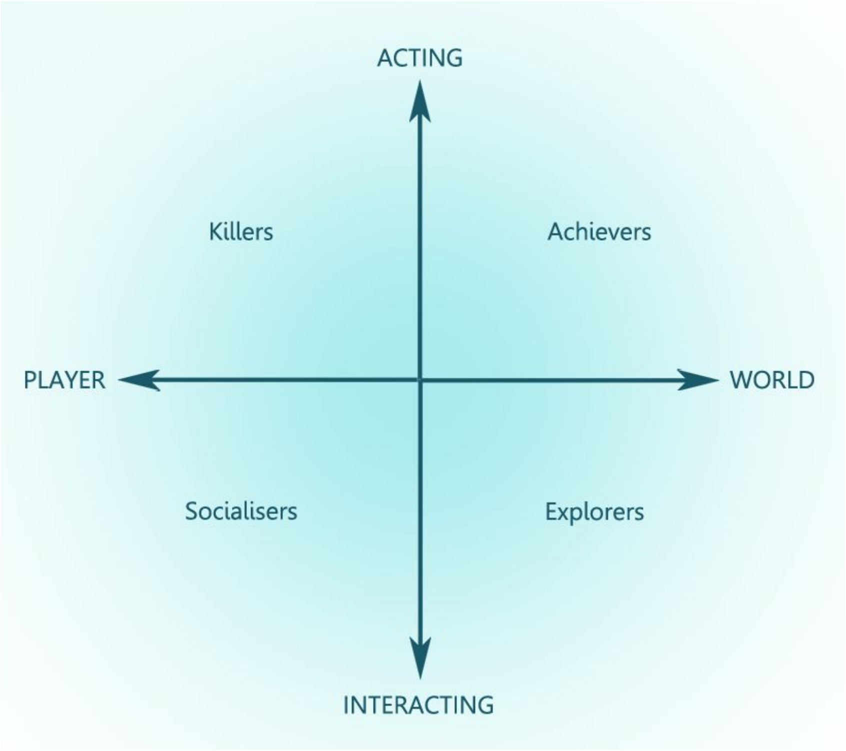


Figure 4.4: Bartle's Player Types

Achievers: Players who regard gathering points, rising in levels or otherwise obtaining concrete measurements of success in games. The challenge of designing multiplayer games for achievers is that not everyone can win. Achievers who feel that they are losing at the game might no longer be interested in playing. Multiplayer games does however give Achievers the opportunity to show off their achievements to other players and enjoy the feeling of holding an elite status.

Explorers: Players who like discovering hidden secrets and places, creating maps and learning about other interesting game features. Explorers might feel restricted when they are forced by a game to move in a certain direction or complete their task in a certain time. Gathering points does not feel interesting to them unless it unlocks new content to be discovered.

Socializers: Players who choose to play games for the benefit of a social interaction. The game is only a means to an end, a place where they may come to talk and hear what others have to say. Single-player games have little or no appeal to Socializers, but this type of people might play popular titles to later use the experience to socialise with other players outside the game.

Killers: Players who enjoy competing with and beating other players. Like the Achiever, killers enjoy winning, but in addition someone else needs to lose and defeating computer controlled opponents is not rewarding enough. Mostly a Killer is a result of a friendly competitive spirit, but there is also those who are in it for the feeling of power and the ability to hurt others. Killers thrive on being considered someone one should watch out for and wants to be admired or respected.

It is important to mention that these player types are not mutually exclusive, one would usually be scored with a percentage in all four player types with one or more type weighted higher. An interesting observation noted by Zichermann is that if the player types where exclusive, as much as 75% of us would probably be categorized as socialisers. People who lean more towards being achievers and explorers might also gain from multiplayer games. Achievers gets the opportunity to show off their skills and achievements, while explorers will be able to share their wisdom about the game with other players. Laamarti[3] points out in one of their guidelines for creating better serious games that collaborative exercises in general are more motivating and engaging than single player games, which would support the theory that most players enjoy playing with others.

The number of player types has later been expanded to include even more player categories and the Bartle's test used to categorize players does have some room for improvement[18]. It is however the four initial player types that appear to be the stickiest[12]. But in the context of serious games there is another type of players that should be considered. In the article by Heeter[19] they address the challenges with what they call Self-Validators. These players are described as "players who enjoy being a winner, but dislikes hard challenges" and "worry about the distress of failing". With traditional games, these players are not present, since if one do not enjoy a game, one does not play it. With educational games however, or other games people are required to play, this player type needs to be addressed for the game to be a successful learning tool. Heeter suggest carefully crafted feedback, gameplay hints and practice sessions as solutions to better facilitate Self-Validators. This agrees with the guidelines for serious games presented earlier by Laamarti.

4.6 Game Mechanics

With the player types established, it is time to see how they can be used to improve a gaming experience. Different game mechanics fit different player types, and one should consider this to either tailor the experience to specific player groups, or if one is developing for a broader audience, make sure the game offers something to all players. The number of mechanicks used in games are huge, and vary in complexity and the effect they have on the player experience.

Below is a summary of some of the most used game mechanics and their possible strengths and weaknesses[20, 21, 22].

Points

Points, or scores, are probably one of the most well known mechanics from games and is also a well used metric in the real world. We find them in schools as grades, goals in football, and even our pay checks may be considered some form of scoring. In games points are commonly displayed on screen during gameplay to show the player how they are doing. Zichermann presents several basic options for handling points, the most important one being experience points. The player is rewarded with such points as they do activities within the game, and experience points do in general never go down, nor can they be redeemed. Collection of experience points can be structured into levels, a benchmark that helps players see their progress, set subgoals and may be required to unlock additional content.

Developers can use experience points to encourage desired behavior by assigning higher rewards to important task, and lower to lesser tasks. Scores and points are an important mechanic for achievers, as they enjoy completing goals and acquiring high scores.

Leaderboards

The concept of leaderboards is also something that is known outside of the gaming world, and most people will not need an explanation when shown one. Leaderboards offer a simple way for players to compare their scores or progression with other players. Zichermann describes two models for leaderboards that are largely used today:

The no-disincentive leaderboard: Leaderboards from old arcade games suffered from the problem that new players were presented with the most incredible high scores, and would initially consider it impossible to ever reach that kind of level themselves. To discourage new players in this way is an obvious disadvantage. With facebook and social games came a new leaderboard model that would solve this issue. Regardless of the actual position of new players, they will be placed in the middle of the leaderboard. This way, it is easy for the players to see how much they need to do to beat the next player on the list, while still feeling good about having someone below themselves. A disadvantage with this design is that if players actually is in the top 20, they do not get the recognition for this. To make the board able to show these scores when relevant is a possible solution.

The infinite leaderboard: This model allows the player to show different parts of the leaderboard as they see fit. An example might be showing only players from their workspace, city or country. This way the players may decide for themselves on what level they want to compete, and is also an advantage for games with many users, as it allows developers to handle and display shorter player lists.

A challenge with leaderboards pointed out by Zichermann is that some information might be sensitive to the user and should not be shared with other players. They use the example of a gym where people are trying to lose weight, which is a score that not everyone is willing to share with others. Choosing appropriate scores for the leaderboard is an important factor for it to be successful. A well designed leaderboard can be a powerful motivation tool for a game, but may offer little to socializers and explorers. Leaderboards also highlights the problem with designing competitive games for achievers, only one can be on the top of the board.

Levels and onboarding

Levels, not to be confused with levels from experience points, are different sections of the game a player can go through. A real live comparison is math classes in school. The first class starts out easy, but with each year they get harder. The new classes do however build on the knowledge taught in previous ones which, if one paid attention last year, will make the class not so difficult after all. Using levels to create a seamless learning curve is commonly used in games. Starting out with only the basic features on the lowest levels and then add more specific or harder challenges as the player progress helps to keep the player interested and ensures that they know what to do.

Structuring the game in this way can also be used to help with a concept called onboarding. Onboarding is described by Zichermann as the “act of bringing a novice player into you system”, and highlights the importance of the first few minutes the player spends in a game. Players make most decision during the first minute of gameplay, and developers should try to maximise this experience without being overwhelming.

This two mechanics does not appeal to any of the player types in particular, but can be an important tool to achieve *flow*. As mentioned to achieve the mental state flow, the player needs to know what to do and how to do it. To ease a player into a game with either easier challenges or less features will help achieve this. *Flow* also requires players to feel what they are doing is challenging, but that they are doing well. Levels does not automatically guarantee that difficulty is matched with player skill, but high difficulty and low player skill is less likely.

Badges

Badges, or achievements can help developers in several ways depending on how they are designed. They are usually awarded to a player when completing a difficult task, completing a task in a specific way or completing a set of tasks. They can signal status, as a player has completed something that most other players have not. Some may prefer them for collecting purposes or simply enjoy the surprise of a unsuspected reward.

Similar to experience points, badges can be a measure of progress in the game or to be used as sub goals, i.e. “Won 50 games” or “Received maximum score in a game”. This makes badges an appealing feature for achievers. They also differ from leaderboard in the way that generally everyone can get the same achievement, which will make it possible for more than one player to be considered winning. Badges can also be designed to motivate explorers, either by rewarding actual exploration in game, “Discovered 5 hidden items”, or by having vague achievements. Achievements like ”Search your feelings… for your heart” give little indication as to how to obtain it, and may motivate exploration.

Timers

Timers are an easy way to increase activity level. They force the player to complete a task within a given time frame, and so adds another level of challenge. This will motivate achievers, but may be a disadvantage for explorers as it forces players to rush through the game as fast as possible, and leaves little room for exploration.

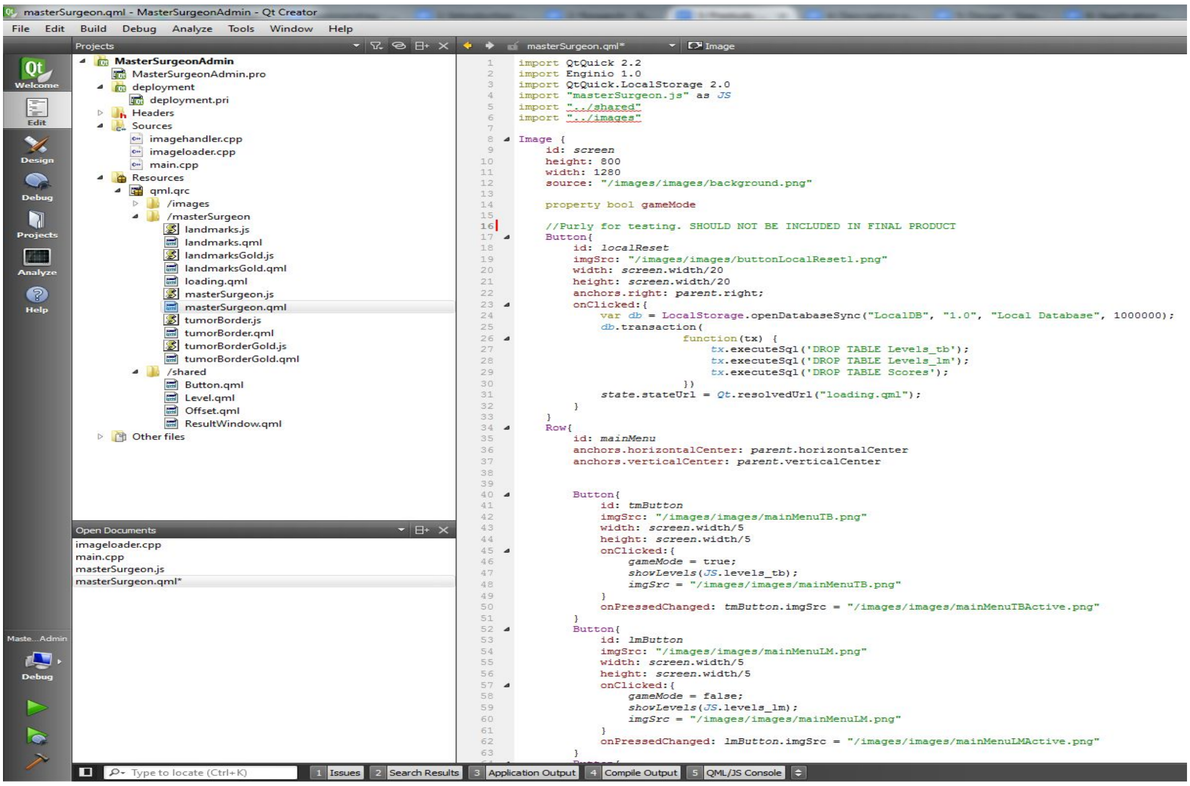
Lottery

The effect of chance in game is an extremely powerful in regards to rewarding players. Zichermann explains that by rewarding players at a fixed interval or fixed amount, the player will always know when or what to expect, and during the time in between will remain passive. Combining varied ratio and interval will keep the user playing in hope that next time will be the time when they win big. This is the exact model used in slot machines and most other gambling models. Zichermann recommend it to be used with caution, but within a broader game experience it is a powerful force for shaping player behaviour. This mechanic applies to all player types, with it’s tendency to be addictive to humans.

Chapter 5

Qt

Qt is a cross-platform application and UI framework that supports over a dozen leading platforms[23]. It is currently being developed by the Qt Company, a subsidiary of Digia who owns the Qt trademark and copyright. It was initially developed to support cross-platform desktop applications, but after it was bought by Nokia the focus shifted to mobile development. First when Digia created the Qt company was Qt able to evolve into the cross-platform framework that it is today[24]. It comes with its own cross-platform IDE[5] called Qt Creator, displayed in figure 5.1. This includes examples, tutorials, and a several tools to ease development, among them a drag-and-drop GUI[6] creator called QtDesigner.



```
import QtQuick 2.2
import Enginio 1.0
import QtQuick.LocalStorage 2.0
import "masterSurgeon.js" as JS
import "../shared"
import "../images"

Image {
    id: screen
    height: 800
    width: 1280
    source: "/images/images/background.png"

    property bool gameMode

    //Purly for testing. SHOULD NOT BE INCLUDED IN FINAL PRODUCT
    Button{
        id: localReset
        imgSrc: "/images/images/buttonLocalReset1.png"
        width: screen.width/20
        height: screen.width/20
        anchors.right: parent.right;
        onClicked:{
            var db = LocalStorage.openDatabaseSync("LocalDB", "1.0", "Local Database", 1000000);
            db.transaction(
                        function(tx) {
                            tx.executeSql('DROP TABLE Levels_tb');
                            tx.executeSql('DROP TABLE Levels_lm');
                            tx.executeSql('DROP TABLE Scores');
                        })
            state.stateUrl = Qt.resolvedUrl("loading.qml");
        }
    }
    Row{
        id: mainMenu
        anchors.horizontalCenter: parent.horizontalCenter
        anchors.verticalCenter: parent.verticalCenter


        Button{
            id: tmButton
            imgSrc: "/images/images/mainMenuTB.png"
            width: screen.width/5
            height: screen.width/5
            onClicked:{
                gameMode = true;
                showLevels(JS.levels_tb);
                imgSrc = "/images/images/mainMenuTB.png"
            }
            onPressedChanged: tmButton.imgSrc = "/images/images/mainMenuTBActive.png"
        }
        Button{
            id: lmButton
            imgSrc: "/images/images/mainMenuLM.png"
            width: screen.width/5
            height: screen.width/5
            onClicked:{
                gameMode = false;
                showLevels(JS.levels_lm);
                imgSrc = "/images/images/mainMenuLM.png"
            }
            onPressedChanged: lmButton.imgSrc = "/images/images/mainMenuLMActive.png"
        }
```

Figure 5.1: Qt Creator.

[5] http://en.wikipedia.org/wiki/Integrated_development_environment

[6] http://en.wikipedia.org/wiki/Graphical_user_interface

5.1 Qt Framework

Qt uses the standard C++ language with some extensions. This includes a built in support for the observer pattern[7], called signals and slots. The extension allows communication between GUI widgets, such as buttons or sliders, or it may be used to handle asynchronous I/O, event notification or associate timeouts. In addition Qt offers it's own framework, Qt Quick, which includes a declarative scripting language called QML. This language gives the developer the option of using JavaScript for logic, but native language is also an alternative and may be used to achieve best possible performance. QML supports importation of C++ classes, which makes it possible to combine all these languages. The result is a framework which facilitate quick and easy GUI creation, using QML and JavaScript, but that is still able to deliver good performance with C++ and native languages.

5.2 Data storage

Qt offers several options for data storage[25, 26]. It contains importable classes for file and datastream handling in C++, with the addition of several APIs[8] for both C++ and QML:

- Local Storage: The Local Storage API[27] offers the developer an quick and easy way to do local storage. The API can be used from JavaScript, and creates a SQLite database. The database is user- and QML-specific, but can be accessed by any QML application. SQLite has however some restrictions regarding multiple users or transactions.

- Qt SQL: The Qt SQL module, used with C++, offers driver plugins to communicate several different APIs. Developers have the option of using an existing databases or embedded MySQL or SQLite.

- XML: Qt provides support for reading, parsing and writing to and from XML streams with C++. It also offers APIs for querying a XML source and schema validation.

- Enginio Data Storage: Qt offers their own cloud services which is free to use for anyone who creates a developer account. It is limited by storage space, but if necessary it is possible to scale this for a fee. Enginio combines a non-SQL object approach with JSON and is supported with APIs for both QML and C++. It also includes easy to use support for any files and users[28, 29].

[7] http://en.wikipedia.org/wiki/Observer_pattern
[8] http://en.wikipedia.org/wiki/Application_programming_interface

5.3 Smartphone development

Although Master Surgeon will be developed with as a cross platform framework, that can run on both desktop and mobile, the request from USIGT was for a mobile game in particular. Developing for smartphones presents additional challenges, both for usability and performance, that should be considered. Some of these challenges are known also from desktop development. Different machines uses different operating systems and screen size and hardware specifications will vary. Using a cross-platform framework will help solve some these challenges, but these are still things that should be considered as some devices might perform worse than others based on hardware or graphical objects may appear different on particular screens.

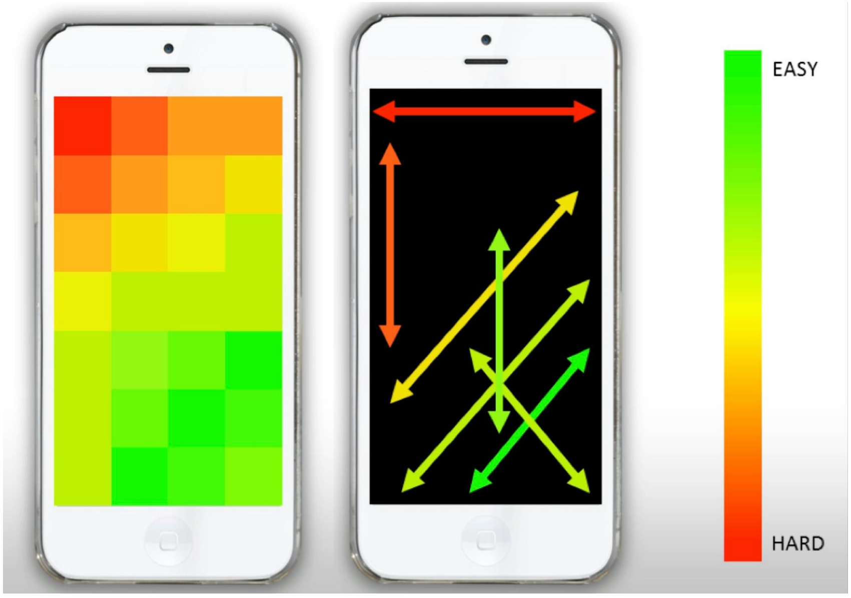


Figure 5.2: Ease of use.[30]

There is also the question of whether one should develop using a cross-platform software like Qt or use native language. With native language one is guaranteed best possible performance as this gives developers direct access to all features of the device. However, creating an application for several different operating systems and keeping it updated will require significantly more work. For applications that is not performance independent, made by smaller businesses or projects who wants to reach a broader audience, cross-platform will appear to be the better choice.

Smartphones also offer a different interaction style between game and player with their touch screens and orientation sensors. This does give developers new options as to how they can design their applications, but also limits them in the number of possible inputs[31]. An important factor for gestures is also that they differ from the traditional keyboard and mouse in the way that users can not see the options they have for interaction. Some gestures are more commonly used, like press, press and hold, swipe, pinch and unpinch and so the average user can be expected to have some knowledge of these.

The screen size of phones have been growing over the years and especially with the introduction of tablets. Still there is a large difference between an average desktop and smartphone screen which restricts smartphone developers in the amount of screen space available. This requires some extra consideration as to how this space is utilized, and the size of graphical objects. Figure 5.2 show a graphical description of how easy the different parts of the screen is to reach and do swipe gestures on for a right handed person.

Chapter 6
Related Games

Lastly in this prestudy we will look at some relevant mobile games for inspiration for *Master Surgeon*. We analyse the game mechanics used to determine their value for the different player groups and compare the design solution to those recommended for serious games. As pointed out previously, player types are not mutually exclusive, and they are used here only as a tool to validate that a game holds value to different kinds of player styles. Most players will find some aspects of all the games below amusing. For a game to be selected for this analysis it needed to:

- Be made for smartphones
- Be considered popular (determined by the number of downloads)
- Offer some single-player content
- Offer some element of competition or cooperation

6.1 Farm Heroes Saga

Farm Heroes Saga[32] is a match-3 game, where the goal is to match three or more similar objects in a line to score points. As the names suggest, in this game the objects are vegetables and fruits. The interaction between the user and game is extremely simple, the basic gameplay consist of sliding on fruit from one square on the board to another, limited only by how many times this can be done during each level. The game is however riddled with mechanics that enhance engagement and creates *flow*.

- Levels: The game is structured into levels. The first level is constructed as a tutorial where the player is forced to take a certain combinations of actions to complete the level. The first levels start of with only a few different objects on the board, and the requirement to win is to collect one type of fruit. As the player progresses through levels both the number of different objects and requirements to win increase. Levels needs to be completed in linear fashion, as the next is unlocked by finishing the previous.

- Points: The game use several different point systems. It has points for each level, where the player is rewarded one to three stars depending on how high a score they got. The game also contains points in the form of currency. One is obtainable in game, the other bought with real money. These are used to buy advantages during a level or hearts which allow players to play more games.

- Lottery: The game also have some element of chance. The the location of objects on the board appears to be random each game, and this is the same for how new objects enters during play.

- Bonus: Stacking five or more objects will reward the player with a bonus, where all fruit on the board of the same type is collected immediately. In addition, if one finishes the objective of a level with less moves than one were allowed to use, the game enters "Hero mode" and the gameboard begin to glitter, displayed in figure 6.1 b. The amount of points received in this mode is increased and the visualization is signaling to the player that they did especially good this level.

- Leaderboard: For each level, a leaderboard is displayed containing the players that did best on that particular level. The map showing all available levels also hold thumbnails of the player's friends, showing how far they have progressed through the game.

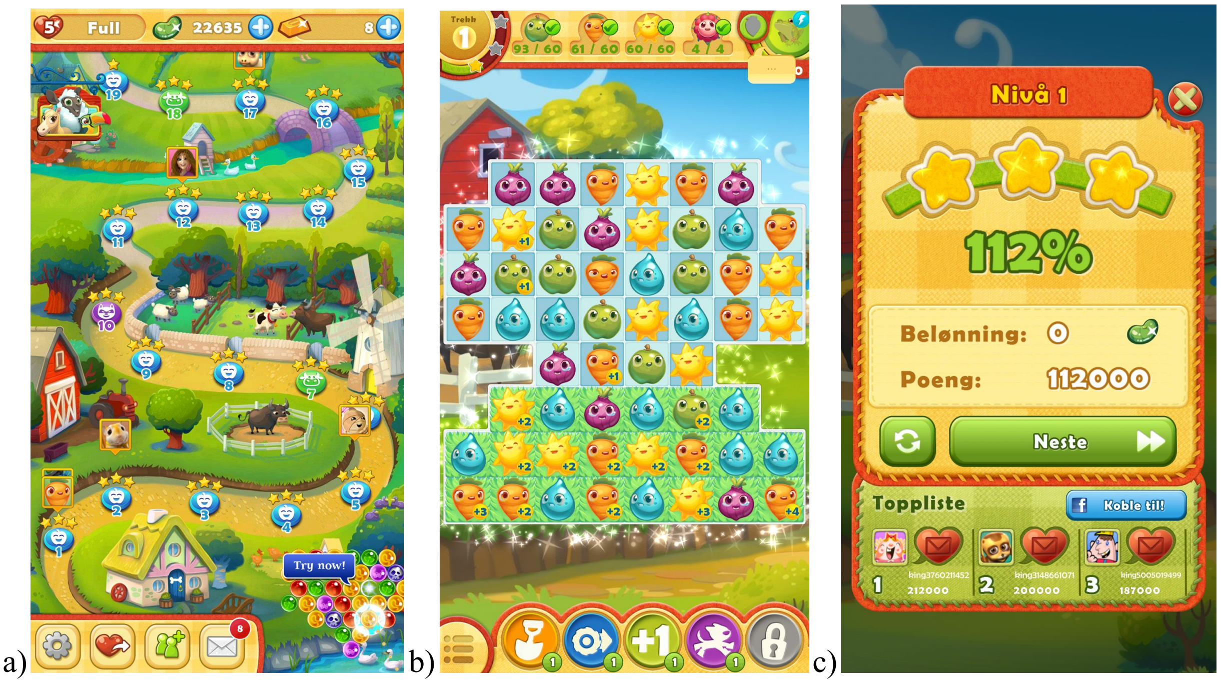


Figure 6.1: Farm Heroes Saga

- Social interaction: The game supports logging in with facebook to enable users to play with friends. If a players choose not to log in they will be placed in a random player group. Friends may send each other help to unlock levels or in form of hearts.

In conclusion Farm Heroes Saga is a game well suited for achievers and socializers. Gameplay is simple, but addictive and offers a pleasing graphical style. Although the social interaction is limited, the support for adding friends from facebook makes this game ideal for players who just want something to do with friends. For advanced achievers the game might not offer enough challenge for an extensive period of time, but the mechanics used will most likely sustain their interest for a while.

6.2 Brain Wars

Brain Wars[33] is a real time quiz battle game, where players compete against each other doing puzzle exercises. This game is not structured in levels, but in several mini-games. During a duel some of these will be selected at random, instructions presented and then played right way, see figure 6.2. A duel consists of four rounds, each round lasting only a few second. At the end of the duel the results of both players are displayed and compared. The results from all duels are used to create a skill-map and decide your rank shown in figure 6.2 c.

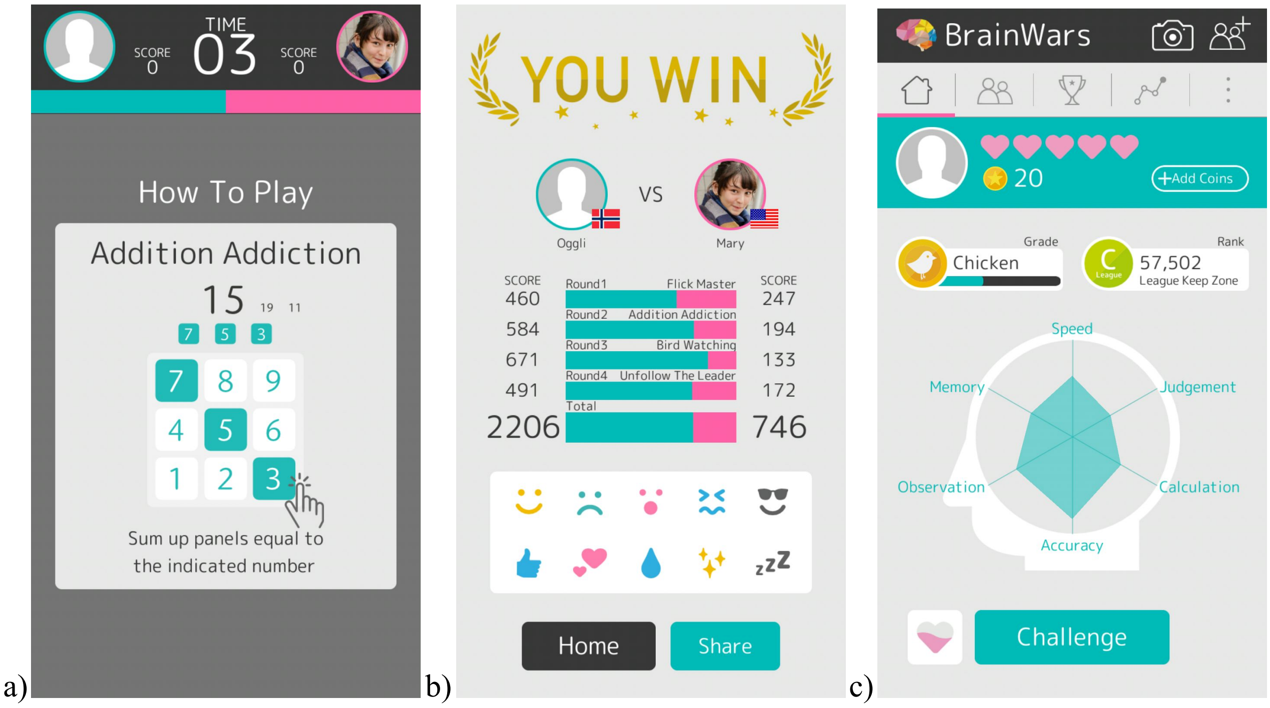


Figure 6.2: Brain Wars

- Points: This game also contains different forms of points. Each duel rewards experience points which is used to increase in rank. In addition players is rewarded points in each of the skill categories. These may drop or increase depending on performance, and are also required to increase in rank. The game also operates with a currency that can be bought for real money. This currency is used to by hearts or determine what mini-games will be used in duels.

- Timer: Everything during duels are done with timers. Even reading the instructions, which one would assume is not part of the competition, must be done within five seconds. This helps increase the challenge level and keep the relatively simple mini-games interesting. In addition it keeps the duel running seamlessly, with no waiting for the other player.

- Leaderboard: There are several leaderboards in this game, and both no-disincentive and infinite leaderboards are used. It is possible to show only players that have the approximately same score as you, players according to country and an list of all players. Also countries are ranked in one leaderboard.

- Social interactions: During duels players may communicate between minigames using the emoticons displayed in figure 6.2 b. There are also possibilities for adding friends and communication with other players outside of duels with an included message system.

Also here the first duel is a tutorial duel, where one competes against a programmed player. However the time allowed to read and understand the instructions are limited to only a few seconds, and since there are so many different mini-games, the learning curve is quite steep. Each game is however simple in regards to amount of possible actions, and are quickly understood. There is also a training option where one can practice all available mini-games as a single-player, this is however hidden away in a side menu and offers no rewards whatsoever.

The high amount of points gathering, scoring, and ranking makes this an excellent achievers game. The way results from duels are displayed and the amount of leader boards is clearly appealing to killers, as it’s focus is how good one is to beat other players. With the support for battling against friends and the built in chat the game also adds some value to socialisers. However, considering self-validator and what has been mentioned about negative feedback in serious games, the design in this regard is not that strong. The constant focus on players scores, regardless if they are good or bad, would clearly have negative effect on players who perform poorly.

6.3 Draw Something

Draw Somthing[34] is a multiplayer game based on drawing. It is structured different from the other games in the way that the players provide the content themselves. A game is played with one other person, and the two switch between drawing and guessing what the other person has drawn. Draw Something provides only a set of potential words to draw and the means to do the drawing. The game do not have levels as such, but the words that can be selected come in three degrees of difficulty. This gives new players the option of selecting their own perceived skill level. A second feature added recently is a single-player mode where one guesses on old player content for rewards, see figure 6.3 c.

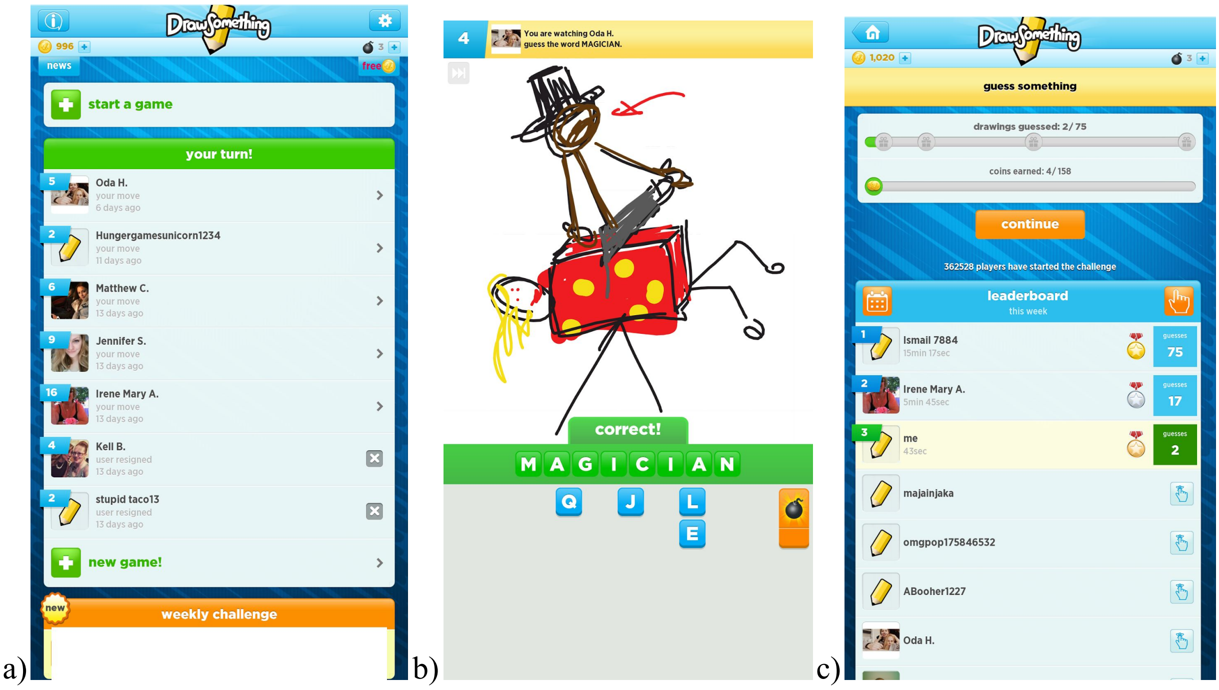

a)
DrawSomething
996
3
news
free
start a game
your turn!
5 Oda H. your move 6 days ago
2 Hungergamesunicorn1234 your move 11 days ago
6 Matthew C. your move 13 days ago
9 Jennifer S. your move 13 days ago
16 Irene Mary A. your move 13 days ago
4 Kell B. user resigned 13 days ago
2 stupid taco13 user resigned 13 days ago
new game!
new
weekly challenge
b)
4
You are watching Oda H. guess the word MAGICIAN.
correct!
MAGICIAN
Q
J
L
E
c)
DrawSomething
1,020
3
guess something
drawings guessed: 2/75
coins earned: 4/158
continue
362528 players have started the challenge
leaderboard
this week
1 Ismail 7884 15min 17sec guesses 75
2 Irene Mary A. 5min 45sec guesses 17
3 me 43sec guesses 2
majainjaka
omgpop175846532
ABooher1227
Oda H.

Figure 6.3: Draw Something

- Points: The main point system of this game is gold, which is a currency. Players are rewarded gold both for guessing and drawing, the size of the reward depends on the difficulty of the word. Completing the single-player challenge does also reward gold. This is used to buy colors, change word options in a game or to buy bombs which helps with guessing. In addition the number of correctly guessed drawings between two players are shown as a score on the main screen, see figure 6.3 a.

- Badges: Some of the available words in game are part of badges. Drawing all words in a badge will reward the player with “big rewards”. These are not displayed to other players, and therefore hold no value as a status icon.

- Leaderboards/Timer: The single-player part of the game have a leaderboard for correctly guessed images. In addition, after guessing a drawing correctly, the time used will be compared with a player who did worse. If no player did worse only the players own time is displayed.

- Social interaction: Players can log in using facebook or create a Draw Something account. These allows users to invite friends to play with. There is also a random-button that matches the player with others players looking for a partner. Messages between players can however only be sent between drawing sessions.

People who enjoy drawing are possibly more inclined to enjoy Draw Something, but high drawing skills is absolutely not a requirement to play. The game is likely at it’s best when players are struggling to draw what they want and have to come up with creative ways to express their word. Draw Something offer nothing to killers as the main mechanics are based around cooperation. Little is also provided for achievers since the measuring of progress and points collection is only a small part of the game. Socialisers and explorers however are inclined to like it as it offers a casual way to spend time with friends and offer an experimental gameplay and hidden badge-words. In regards of feedback it was interesting to discover that the game only gives positive feedback regarding timers, which as discussed is an important factor for serious games.

6.4 Special Case: Master Neuro Surgeon

Master Neuro Surgeon[35] is a previous version of the application developed in this thesis. The gameplay consist of drawing the border of tumors in ultrasound images.

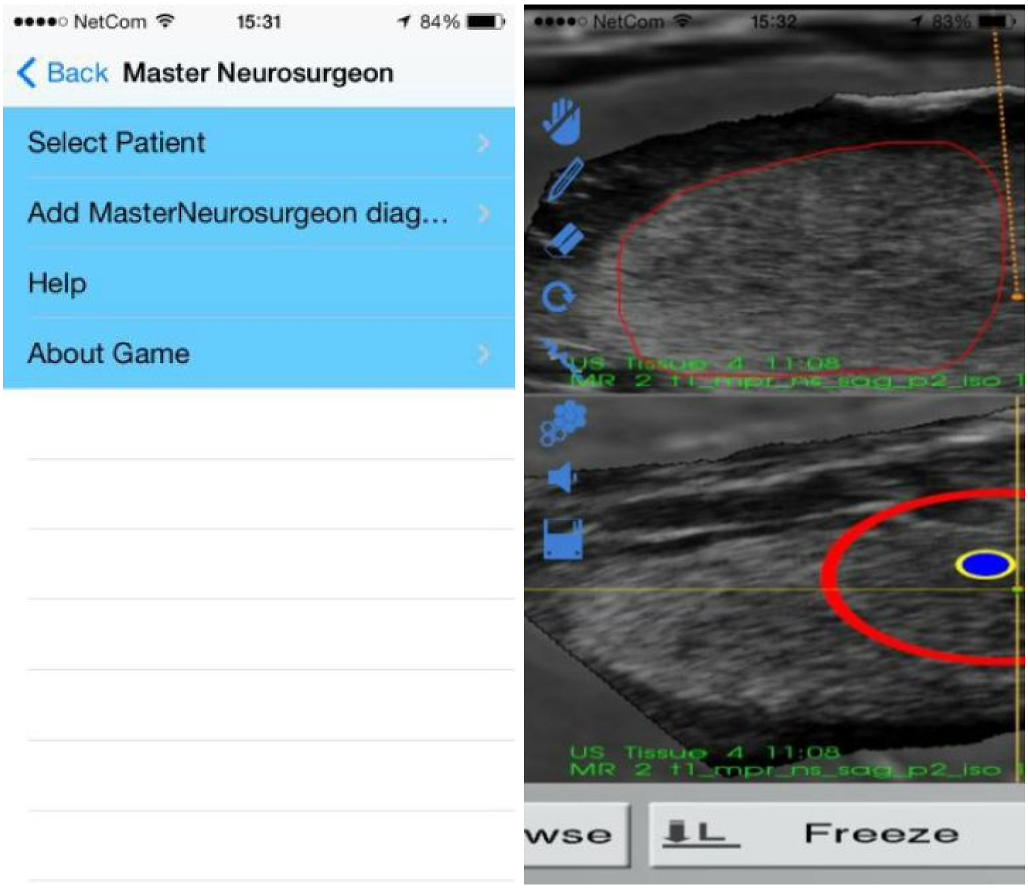


Figure 6.4: Master Neuro Surgeon

- Level: The game offers the players different patients to practice on. They have different difficulty, but are not structured in any way that will help adjust difficulty level.

- Points: The players are given a score after a level is completed. The score is given both as a number and the average distance between the players suggestion and the solution.

This game differs from the others by being a prototype. This means it is still in an experimental phase, and has some room for improvment. As it is today the game would appeal only to achievers, as it contains no mechanics suited for the other player types.

Chapter 7
Summary

In the beginning of this prestudy we did a short introduction on how and why the images used in Master Surgeon are made. We now know why teaching ultrasound interpretation is needed and the importance of this ability for modern surgery. This understanding of the educational content of our serious game will be helpful during the design process of this project.

Secondly, we found some valuable material regarding gamification and serious games. We now know much about why people play and how to make games fun. From articles concerning serious games we saw some important success factors for educational games, and from gamification theory we have discovered how to achieve flow and consider player types and mechanics in the design process.

To better understand the selected framework we also had a look at Qt. Since the framework offers different options for languages and data handling, several decision will have to be made regarding the implementation of the project. The information that was found will be used before and during implementation and will be further discussed in the implementation part of this thesis.

Lastly we looked at relevant mobile games to try and find inspiration for *Master Surgeon*. We found that the games adopt different mechanics, and as a result favors different types of players. The analysis will be used to select appropriate mechanics for *Master Surgeon* and make sure that the intended user group is fully valued by the design of the game.

Part III

Method

Chapter 8

Description of the final application

In this chapter we will describe the the final design and gameplay for *Master Surgeon*. This is done to give the reader a basic understanding of how the application works, so to better follow the design choices made in the next chapter.

8.1 Master Surgeon

The game contains two different game modes. In the first, called *Tumorborder,* the player will attempt to locate and draw the border of a tumor on an ultrasound image. The second game mode, called *Landmarks*, uses the same type of images, but here the user is given the name of a landmark and is asked to locate and mark it in the image by clicking it.

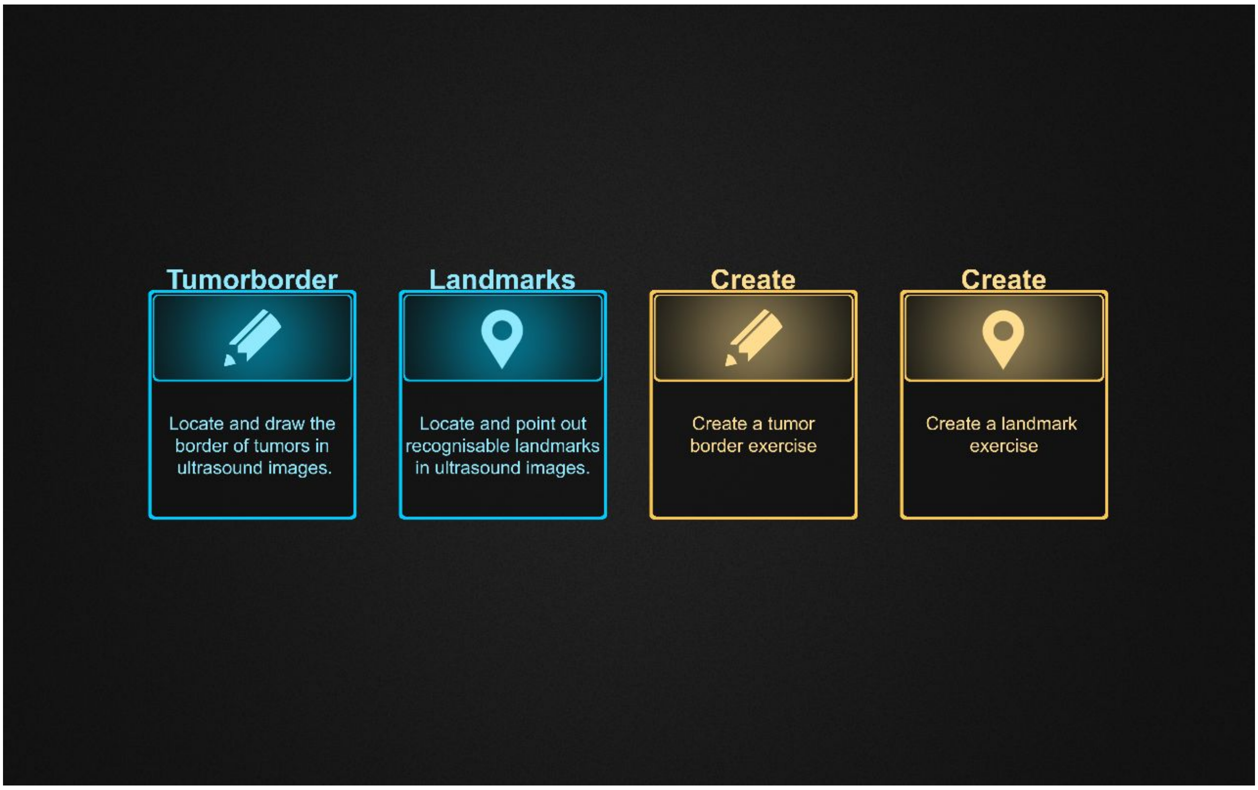
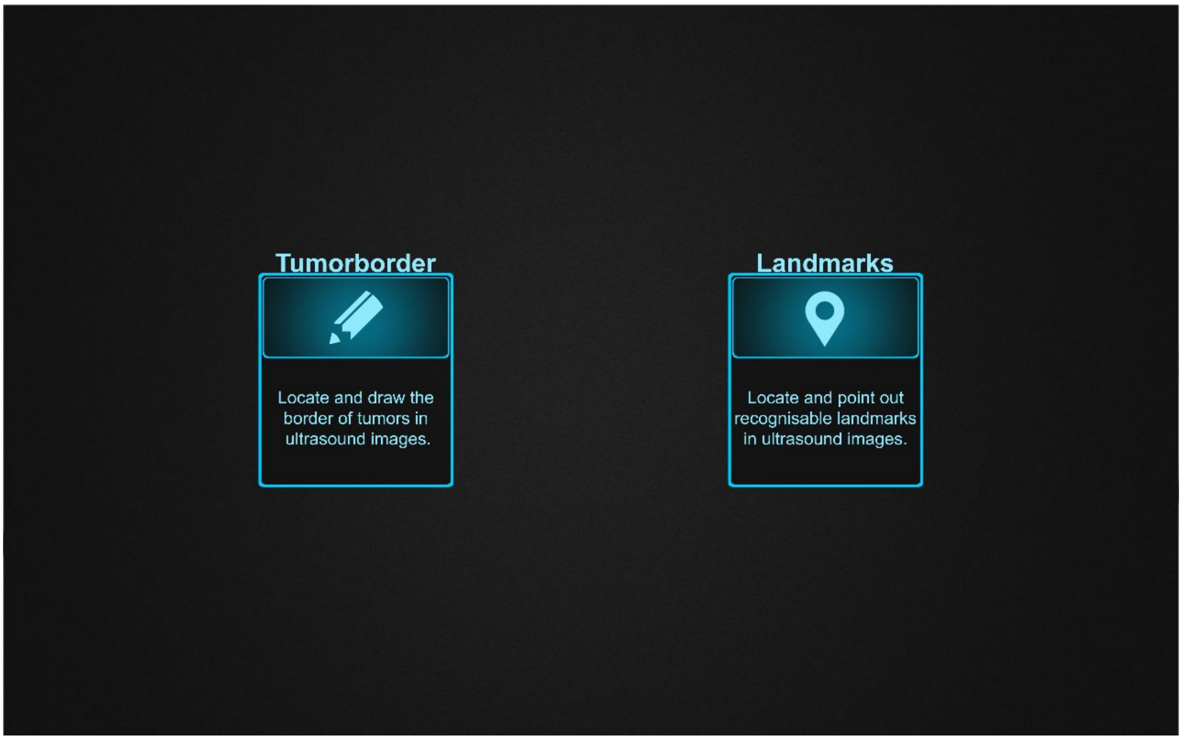

Tumorborder
Locate and draw the border of tumors in ultrasound images.
Landmarks
Locate and point out recognisable landmarks in ultrasound images.
Create
Create a tumor border exercise
Create
Create a landmark exercise

Figure 8.1: Main menu - Admin user

In addition to these two game modes, as an admin user, one will have the option of creating new levels. These will only be available to qualified personnel who is deemed worthy by the owners of the application. Figure 8.1 and figure 8.2 show the different main screens for an admin user and a normal user respectively.



Tumorborder

Locate and draw the border of tumors in ultrasound images.

Landmarks

Locate and point out recognisable landmarks in ultrasound images.

Figure 8.2: Main menu - Normal user

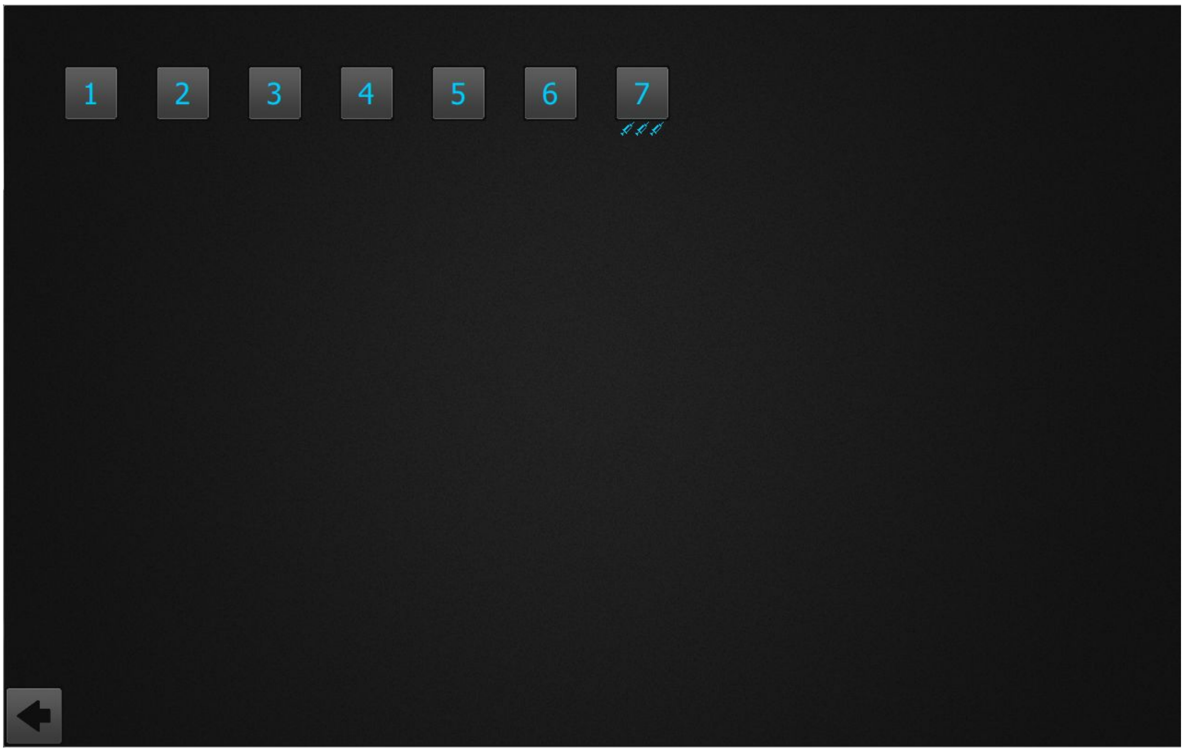


Figure 8.3: Level menu

By selecting a game mode the players will be taken to a level menu where they can chose a specific case to play. In this menu users are able to see if they have played a specific level before and how well they did, indicated by syringes below the level buttons. The syringes are rewarded after a level is finished, and rates the player performance on a scale from one to three. Figure 8.3 show a level menu where the player did excellent at level seven.

8.2 Tumorborder

The *tumorborder* game mode is meant to teach the player to recognise different tumors in ultrasound images, the layout of the screen can be seen in figure 8.4. To draw the solution the player has three buttons to work with:

- Pencil tool: Using swiping gestures or a mouse it creates a 1 pixel thick line in blue color.
- Erase tool: Using swiping gestures or a mouse it removes a square of drawn pixels. The size of the square is screen size dependent, but will for a the average smartphone be smaller than the size of a finger.
- Thrash tool: Removes everything that has been draw. Requires only to be clicked, but the player will be asked to confirm the deletion.

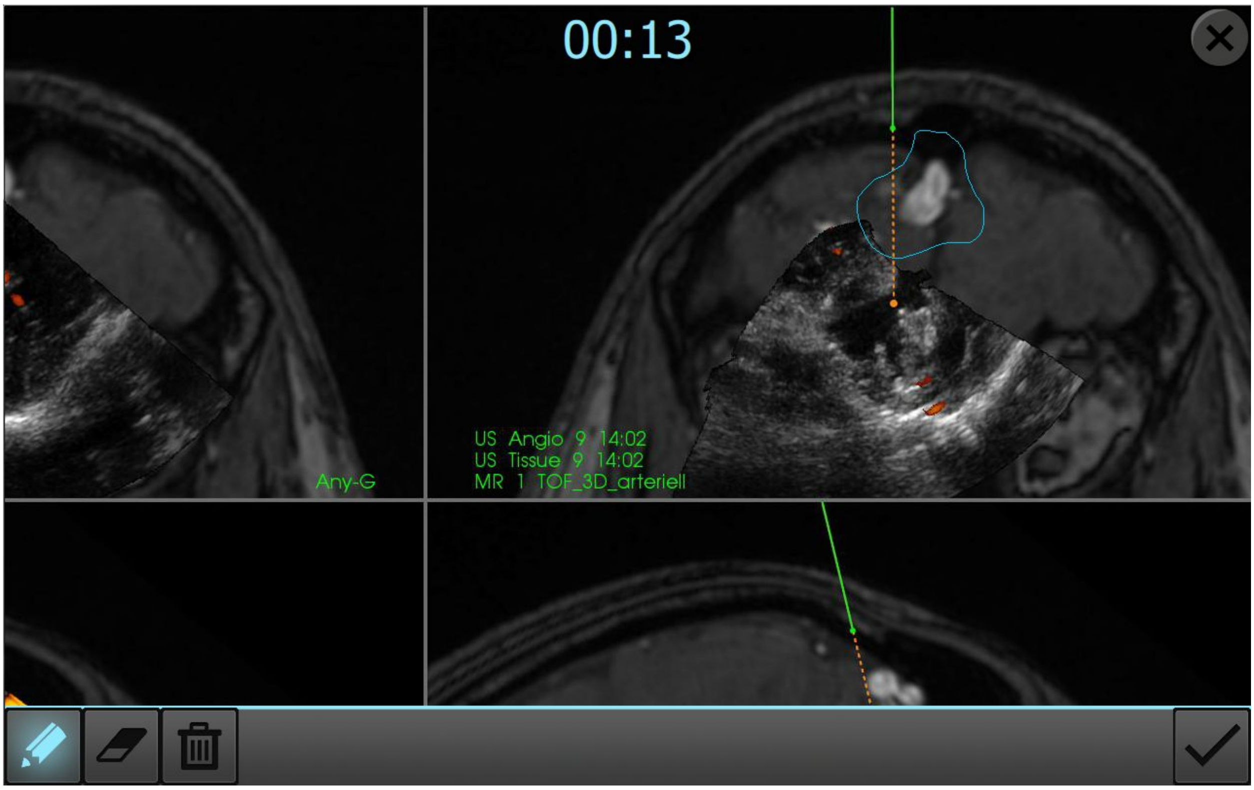
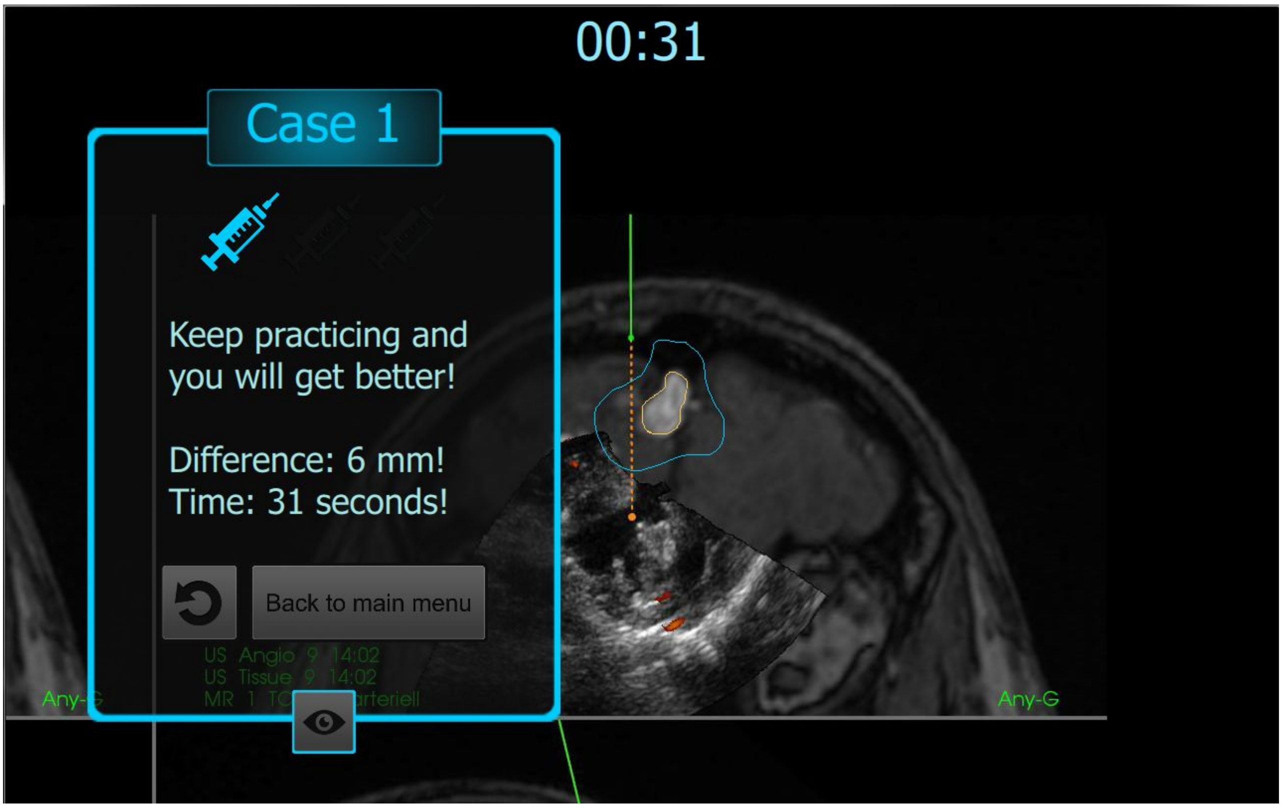


Figure 8.4: Tumorborder

If none of these are selected the image is movable, which means that using swiping gestures or by dragging it with a mouse the image can be moved freely. In addition to these tools the player can either exit the level by clicking the exit button or hand it in by clicking check button. In both cases the player will have to confirm the action.

Once the image is sendt the results window will be show. The part of the image containing the solution will be centered on the screen, and if the player wants to inspect it further the window can be hidden by clicking the eye button. If the player wants to try again he can do so by clicking the redo button or exit the level by clicking back to the main menu. Figure 8.5 show a result window, where the player's suggestion is drawn in blue and the solution is drawn in orange.



Case 1

Keep practicing and you will get better!

Difference: 6 mm!
Time: 31 seconds!

Figure 8.5: Tumorborder Result

8.3 Landmarks

The *landmarks* game mode is meant to teach the player to locate known landmarks in ultrasound images. Figure 8.6 show the layout of this screen. The current landmark is given at the bottom of the screen, and there is only one button that is used to play:

- Marker tool: Needs to be selected to mark a point in the image. The placement of a marker is not permanent and can be moved as long as the tool is selected.

As with the other game mode, when no tool is selected the image is movable and the function for the exit and send buttons are the same. The result window is also identical to the one in tumorborder, but as seen in figure 8.7 the solutions are different. A landmark is either an area or a point in the image.

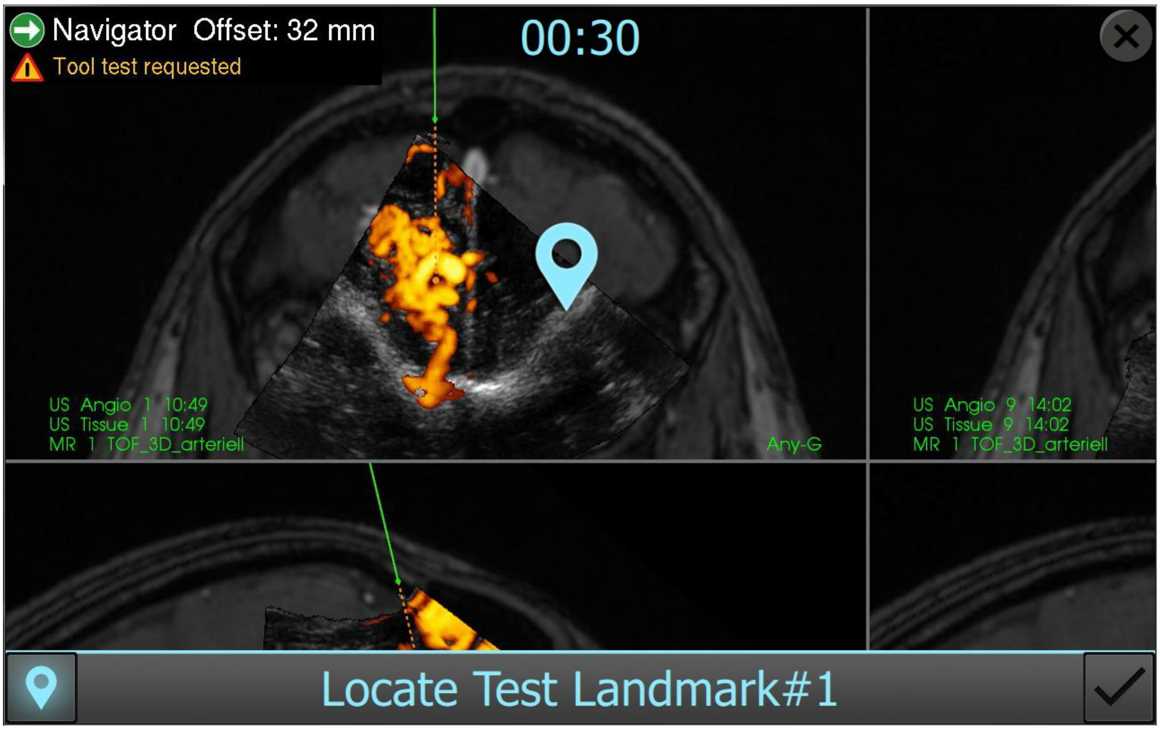
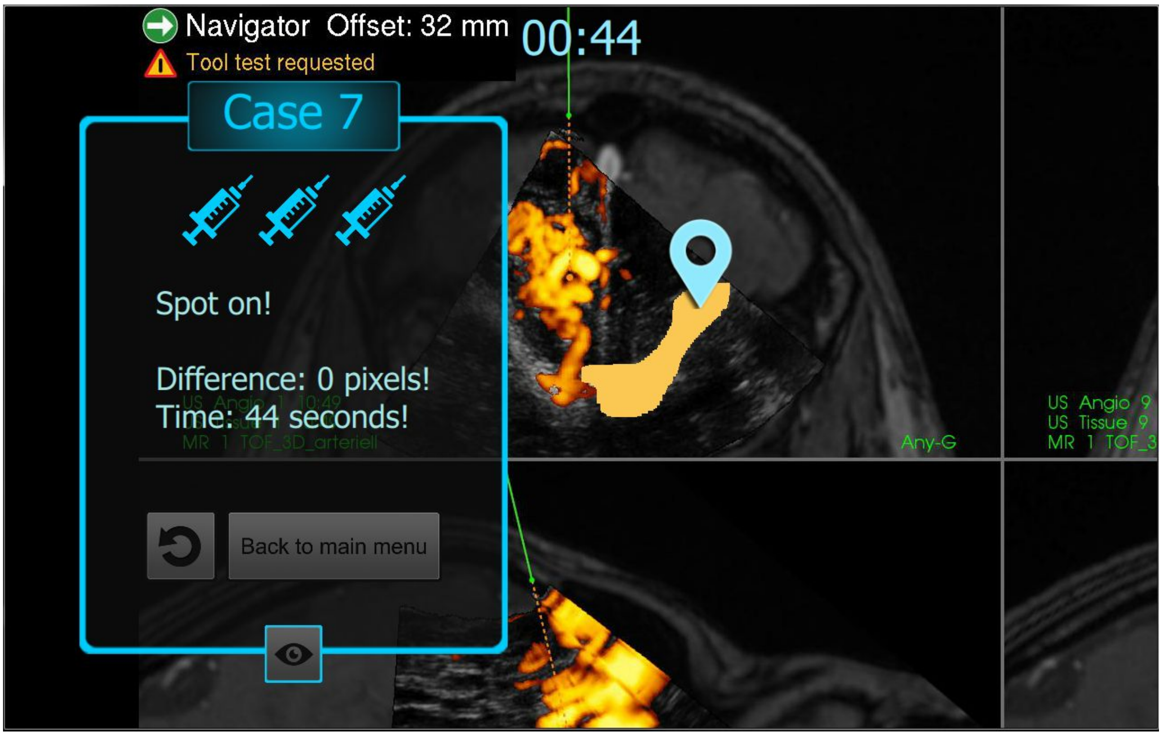


Figure 8.6: Landmarks



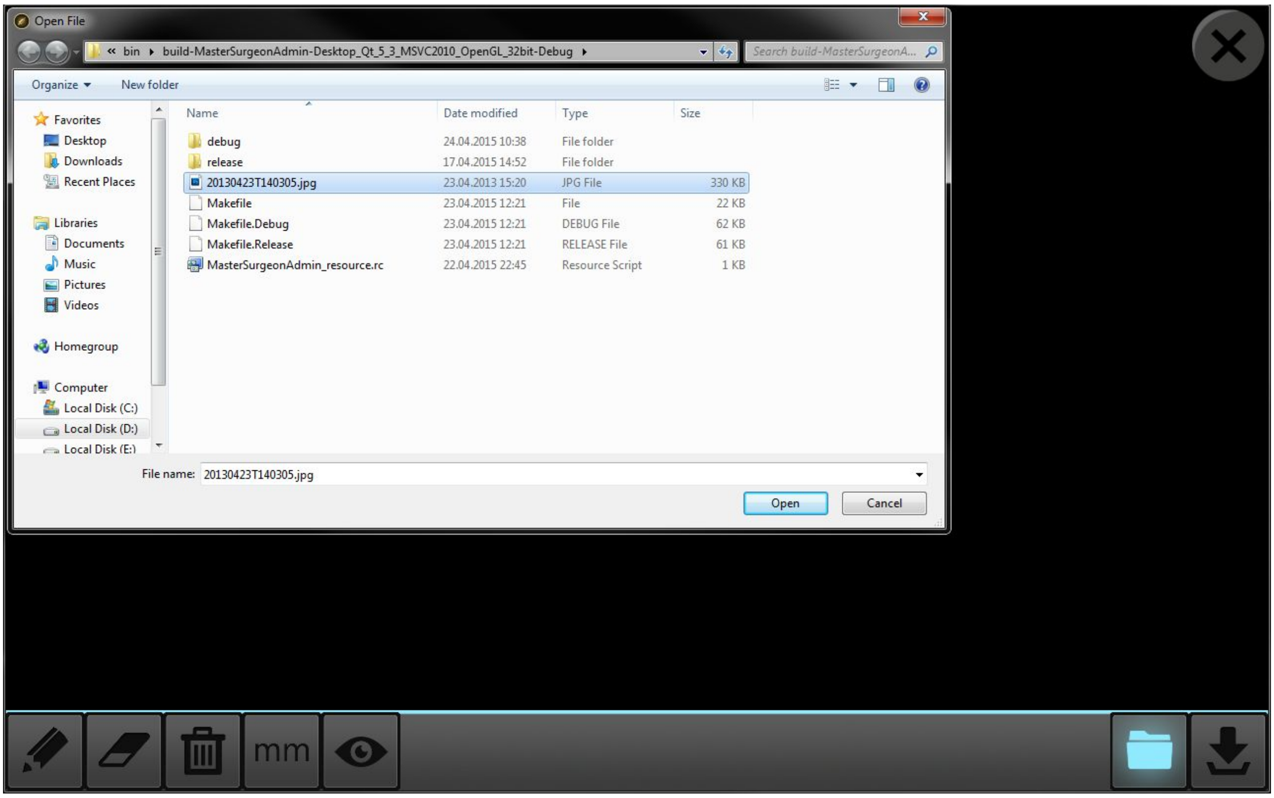
Spot on!

Difference: 0 pixels!
Time: 44 seconds!

Figure 8.7: Landmarks Result

8.4 Create new tumorborder level

To be able to create a new level it is required that an image is uploaded to the application from some local storage. When the users open the creation window they will initially be asked to select an image. After a selected image is loaded, it can be changed by clicking the folder button again. Figure 8.8 shows the folder button selected and the file dialog. In addition to the the pencil, erase and thrash tools explained earlier, the user has two new tools to work with:



Name	Date modified	Type	Size
debug	24.04.2015 10:38	File folder	
release	17.04.2015 14:52	File folder	
20130423T140305.jpg	23.04.2013 15:20	JPG File	330 KB
Makefile	23.04.2015 12:21	File	22 KB
Makefile.Debug	23.04.2015 12:21	DEBUG File	62 KB
Makefile.Release	23.04.2015 12:21	RELEASE File	61 KB
MasterSurgeonAdmin_resource.rc	22.04.2015 22:45	Resource Script	1 KB

File name: 20130423T140305.jpg

Figure 8.8: Upload image

- mm tool: Used to set the pixel-to-millimeter ratio. Once the button is active one may place points in the image by clicking it. After putting down two points with a known distance in between them, this distance is put by the user in the text field on the button of the screen. This is not required, but may later be used to calculate actual millimeter distance for results. Figure 8.9 shows the screen when the mm-tool is selected and the two marks have been set.

- Eye tool: Used to hide the canvas layer to better see the image beneath. This is a feature inspired by the hide layer function in Photoshop, and is necessary when the image drawn on is no longer visible behind the colored area.

When the user has finished drawing the solution if can be saved by clicking the save button.

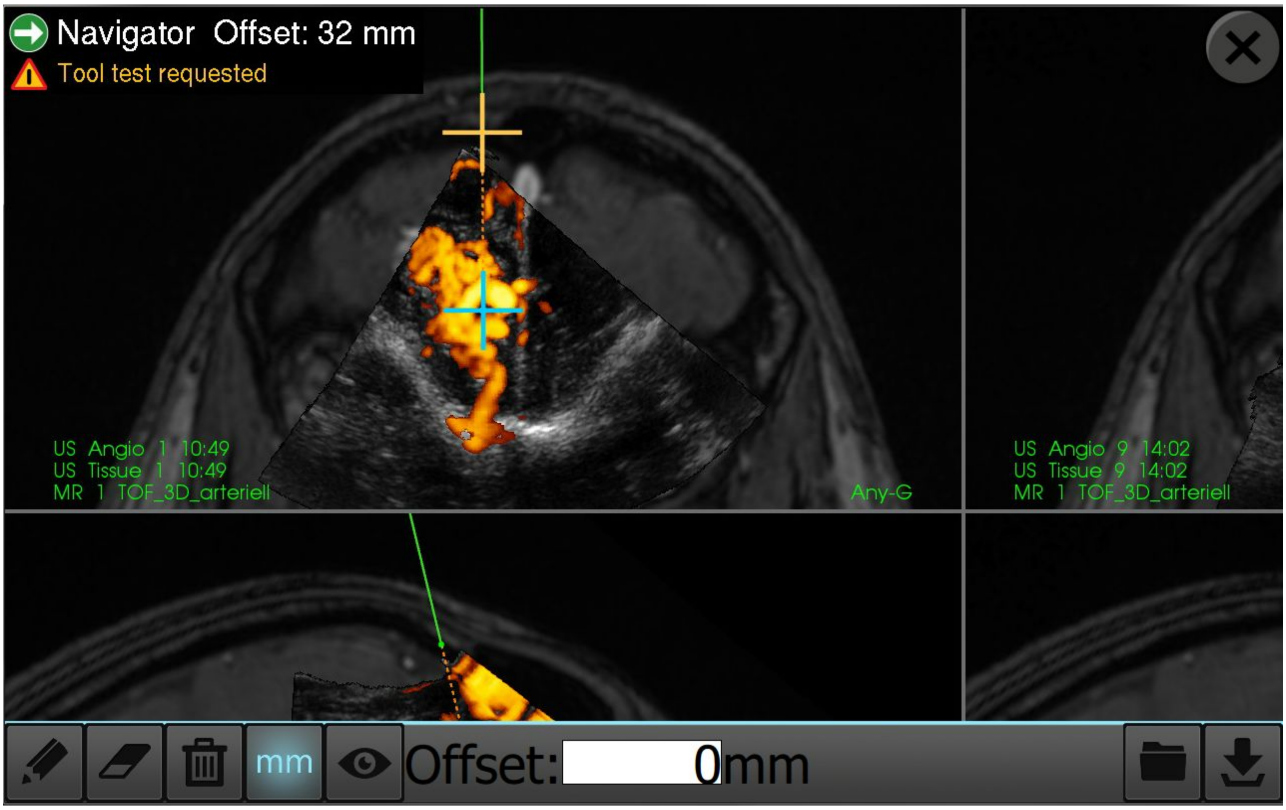


Figure 8.9: Tumorborder - Create level

8.5 Create new landmark level

The core functionality of this mode is almost identical to the one implemented for creating tumorborder levels, however the pencil tool is here replaced by the point and brush tool. The point tool works the same way as in the landmark game mode, when selected the user may click the image to set a point.

- Brush tool: Used to draw larger areas on the image. A screen size dependent square will be colored blue.

While borders will always consist of just one line, landmarks can be anything from a point to an area. It is not possible for the user to do both a line and an area as the solution in one image. If the user tries to draw an area after creating a point, the point and the marker will be deleted and visa versa.

8.6 Final game versus the initial prototype

As will be shown in the next chapter, substantially more content was designed for this game during the design process. Limited by time and an unfamiliar framework, it was suspected at the beginning of this project that an entire game would not be implemented in time. This made creating and documenting a viable design even more important, as the game will have to be completed by different developers.

Chapter 9
Game Design

In this chapter we will go through the design decisions made for *Master Surgeon.* Here we attempt to use what has been shown concerning gamification and the guidelines for serious games to tailor a game experience based on the user group and educational material. To illustrate all game elements, including those that did not make it to the final prototype, we will use images from the last paper prototype created in collaboration with the USIGT representative. It is important to note that these images where used as an informal tool to discuss design decisions, and are only shown here to help readers better understand what is discussed in this chapter.

9.1 Target player types

Making assumptions about what player types will be represented by users of *Master Surgeon* is difficult without any investigation into who they are, what games they play or how they play. Based on what is know about Bartle's four player types there is however one guideline that can be deduced. It was found that most players can be considered socializers, while killers are the least represented type. This suggest that having socialisers in mind when developing for a broader audience, or an unknown player group, is a safe solution.

Further, after finishing the previous version of *Master Surgeon,* Khosravifard[35] completed an evaluation of her game together with surgeons, technicians and students, which is the intended user group for this project. She observed that that participants enjoyed taking part in competition and wanted to announce their results to others, which can be considered both achiever, killer or socializer behaviour. In light of this observation *Master Surgeon* should also contain some game mechanics with has value to these players.

Lastly the player type self-validator should be considered. Heeter[19] suggested that this type of player will be common in games where one has to or ought to play, since they offer no selection of genre or which game to play. Considering the high level of education the general user of Master Surgeon has, it is however possible that self-validators is less represented here than in the general population. Regardless the guidelines for facilitating self-validators is in line with those suggested for serious games and achieving flow, and will therefore be incorporated in *Master Surgeon*.

9.2 Game Mechanics

Considering the assumptions above the following mechaniks are suggested for *Master Surgeon*:

Levels

It has been established that levels can be designed to contribute both to onboarding and flow. Master surgeon will use levels to structure the gaming experience, where a level will consist of one image and one task for the player to complete. The images used in *Tumorborder-mode* have a natural variety in difficulty, dependent on the placement and shape of the tumor or if parts of it has already been removed. The same is true for *landmarks-mode* where the size and clarity of landmarks will vary. This can be used to order levels according to difficulty when presented to players. To ease implementation this will be done by ordering them from easy to hard, and present the easy levels first, inspired how Farm Hero Saga presents their levels. The solution will differ from Farm Hero Saga by having all levels available, since it is necessary for players to be able to pick levels with their preferred difficulty. If the player traverse these levels from beginning to finish, the difficulty level should follow the players skill progression, but if the players feel they need more challenge they have the option of skipping easier exercises. The difficulty of the level will be shown in an startup message displayed before the level starts, see figure 9.1.

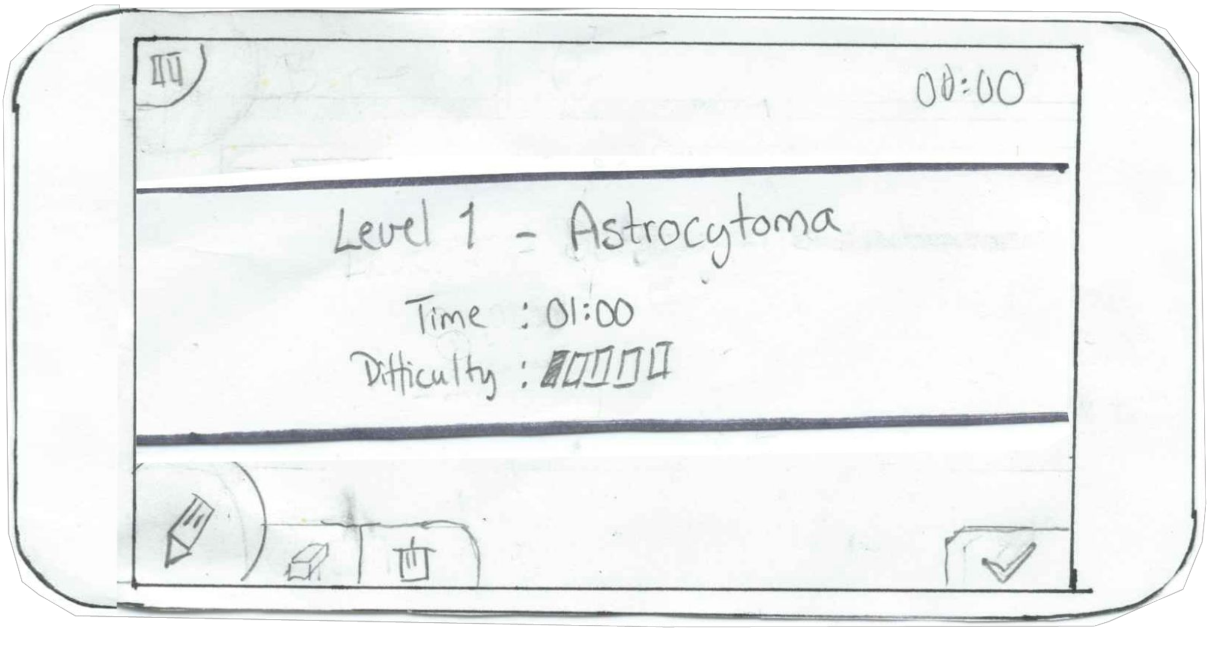


Figure 9.1: Startup message for a Tumorborder level

A possible solution better suited for serious games would be to use algorithms to automatically present levels that has an appropriate difficulty, based on previous player performances. In this way the player will not know if they are completing easy or hard levels. This alternative will mask low performance from players by making them feel they are mastering all challenges that has been presented to them. However such algorithms will have to be complex as they must account for player progression, and should be thoroughly tested to make sure they provide appropriate content. This design is therefore deemed too complicated to be implemented during this project, but is highly recommended for future development.

Points

Experience points will be used to track players progress as they complete levels in any game mode. The points are rewarded by completing levels, and the size of the reward will vary depending on the player's performance. Players will be awarded experience points regardless of their performance to avoid negative feedback, and if players manage to improve on their previous score for a level, they will be rewarded the difference in experience points. This will motivate them to keep practising on levels where they performed badly. The experience points will be used to increase the player's level, and unlock content and receive achievements. As of today the only planned unlockable content are avatars used as profile images, but others might be added at a later point like titles or customisation options.

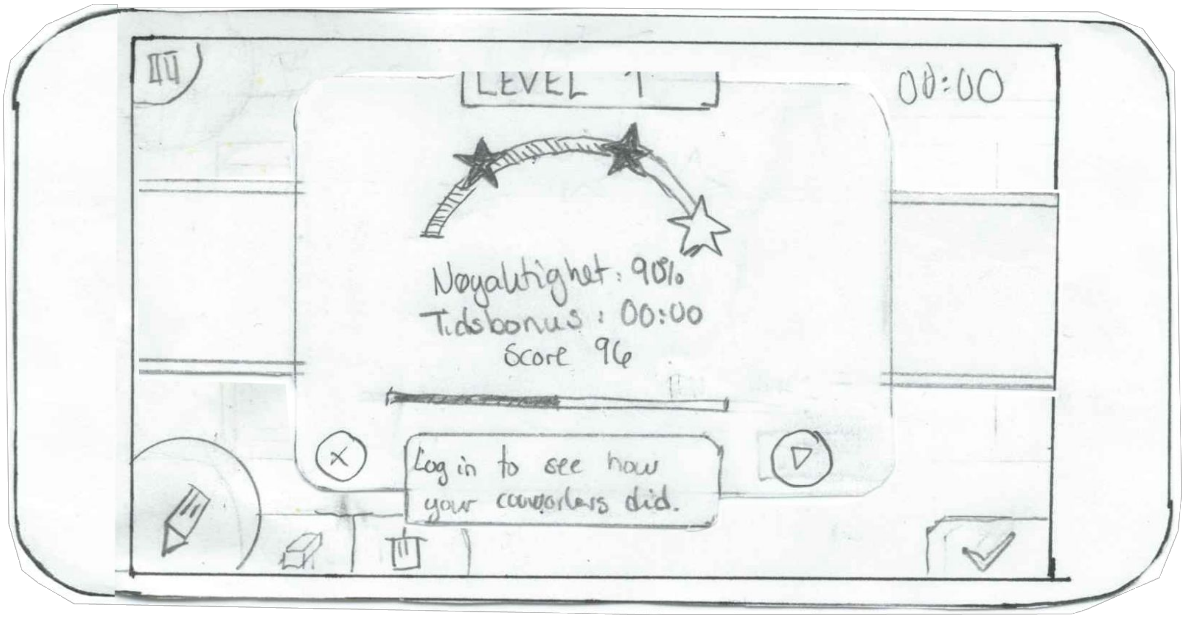


Figure 9.2: Result window.

In addition to experience points players will receive a score between one and three on each level they complete. This is inspired by the star-points in Farm Heroes Saga, and as with experience points the player is rewarded at least one for completing. This reduces negative feedback on low performance and helps the players keep track of what levels they have completed as such levels will be marked with at least one point.

Timer

Timers will be used to give the players a stronger feeling of challenge. Initially, as displayed in figure 9.1, timers were meant to be used as a countdown to make levels harder by forcing the player to finish in time. After some careful consideration this was later changed to count upwards and changed the timers into a bonus score. This gives players who enjoy the extra challenge an additional score to achieve in, while those who enjoy solving the task accurately can do this without the timer stopping them in the middle of their game.

Leaderboards

Master surgeon will include three different leaderboards after the infinite model. They will be available through the player profile. The leader boards will, as explained in greater detail later in this chapter, be center around a workplace or classroom setting, and will all be available from the player's profile. The first board will include all players within a players workgroup or class. This is a traditional leaderboard in the sense that all players will be shown in accordance to their experience points. In an attempt to mask sensitive information, the leaderboard will show their level, which indicates time spent in the application, rather than the star-score which shows how well players actually perform. This will hopefully also motivate players to spend more time in the application to rank higher.

The next leaderboard differs from the two others in that it shows how the workgroup or class is doing. For now it is also limited to the player’s city to attempt to create some friendly competition between classes or workplaces that are known to each other. This leaderboard will use experience progression. This is done not only to protect sensitive information, but because making the ranking depend on actual level-scores would discourage introduction of new, low level players. Idealy it should also favor the number of active players as this could encourage workplaces and classes to introduce new players, but this should be carefully considered as it might be unfair to groups that are smaller in size.

The last leaderboard is an overall leaderboard and will include all players. This is a board meant mainly for competitive players that are interested in their world rank. To make it more interesting to the average player it is possible for the board to adopt the no-disincentive model, meaning that the view of the list presented to the players will show a few players above and below their own score. Based on what has been shown about player types there is however no guarantee that this will work, as people who are not as competitive will most likely enjoy the two previous lists more, and the board will loose it’s value to competitive players.

In addition to these main leaderboards, a smaller one will be used on each level. This list uses only scores from players own class or workplace, and will offer more people the opportunity to win at something and give players scores to compare their own to. A drawback with this leaderboard is that the longer people play, the more likely it is that this list will fill up with max scores. To keep them interesting, they should only keep track of scores achieved over a given period of time, i.e. during the last month, and display only the three best players on a given level.

Badges

To add an additional level of challenge and facilitate achievers the application will include badges, here called achievements. Certain player behavior will be tracked and as the values exceeds defined thresholds or a special action is taken, players will be notified and the new achievement displayed on screen. Achievements can be rewarded for a single event; “Achieve 0mm difference while playing Tumorborder”. Or they may be recurring; “Complete 5 games”, Complete 10 games” or “Complete 50 games”.

For achievements to be most effective they should be made visible to other players. Because they require a great deal of screen space, they will be shown in a separate view, accessible through the player profile and only the number of achievements will be shown in the actual profile. Players will then be able to access other people's achievement thought their profile.

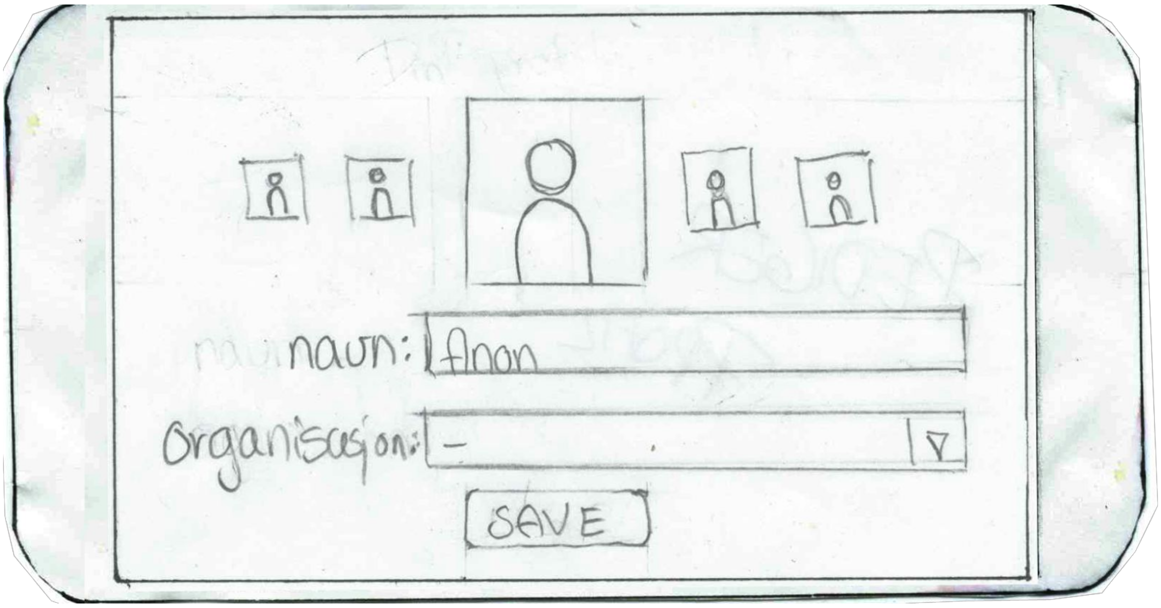


Figure 9.3: Create account screen. Avatar selection in the top of the screen.

In addition to achievements, profile pictures called avatars, will be used as a measure of progress or performance. A new player will be presented with a set of starter-avatars to choose from, but others can be unlocked by increasing in level or winning competitions. These avatars are a quick and effective way for players to show of their status to other users and should be used in all leaderboards.

9.3 Flow

Flow, being an important factor in any game, should also be facilitated in *Master Surgeon*. To achieve this Schaffer's list of factors required to achieve flow will be used and each point addressed to make sure they all are considered in the final design.

High perceived challenges and skills

To achieve high perceived challenges and high perceived skills it is required that the difficulty of the game and the skill level of the player match. As explained earlier in conjunction with levels, this will be done in *Master Surgeon* by ordering the levels by difficulty, and in that way attempt to create a functional learning curve.

However, as more levels are created for the game, it is not necessarily the case that players will have to complete all the easy levels before they can move on the more difficult ones. Given that all levels are always available, players will have the option of skipping those they find too easy, but as mentioned a better solution would be to implement algorithms to present the player with appropriate content.

Knowing what to do, how to do it and where to go

A seen in two of the games reviewed in the prestudy, *Master Surgeon* will also use the first level as a tutorial for each game mode. This level will consist of a few forced steps that the player will have to complete to be able to continue to the next level. Each step will have an explanation that is displayed on the screen and any opportunity for the player to not follow the instructions will be made inactive or hidden.

To encourage players to choose the tutorial-level it will be marked with some some text, glow or other effect to make it appear more important to click. The remaining levels will stay locked until the tutorial has been completed. To avoid confusion created by trying to start locked levels, players will be informed that they need to complete the first level to unlock the others.

Knowing how well you are doing

As mentioned several types of feedback will be used in the game and they provided different ways of showing players how they are doing. The star-points, experience points and achievements will show players their progression and how good they are doing according to the standard set by developers and solution-creators, while leaderboards will show players how they progress according to others.

A third form of feedback not yet discussed as it would not qualify as a separate game mechanic is personal highscores. By saving all time best scores for each level the game can inform the player when said score has been beaten, and in that way show they player how they did according to their own previous performance.

Freedom from distractions

Preventing external distractions cannot be done through the game, as we cannot control where or how the players chooses to play the it. Making sure the system is user-friendly and free of technical bugs is however a way that the game can help avoid distraction. Great thought should be put into the symbols used and layout of the GUI, to make the game as intuitive as possible to the user group. Minimising the risk of bugs and crashes will be done through testing and writing lucid code.

9.4 USIGT requests

Given that the game is developed for USIGT, it was natural to ask for their inputs at the start up of this project. They had two concrete request concerning the design of the game.

The first and most important request from USIGT was that it should be possible for users to start playing right away, without it being necessary to log in. As mentioned in the prestudy, onboarding is an important concept to get novice players to engage in the game, and allowing the player to skip the creation of a profile will not only lower the threshold for new players to use the application, but also help with onboarding by giving the player access to the “fun” immediately after startup.

The second request was that the main purpose of the game would be weekly or monthly competitions where the players would play specific level, hand in their solutions, for the results to be calculated and returned later, either by mail or via the application. The argument for this was that the results or scores from these competitions would be more valued by the players as it offers no second chance to improve the score, and would better display the player's actual level of skill. This does however go against one of the main advantages that games have in an educational context which is instantaneous feedback and rewards, which has been established is an important motivational element. The argument made in this features favor is however valid, the scores from this kind of competition could hold more status as they display skill, rather than time spent in the application. Taking this into consideration a compromise was made, and Emergencies was introduced.

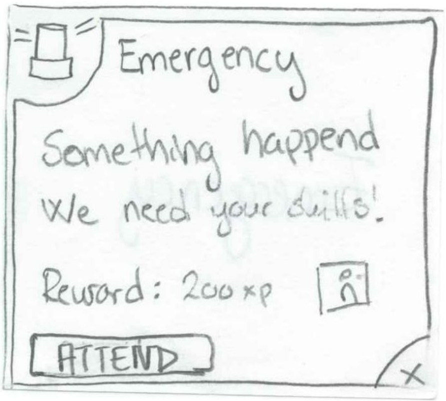


Figure 9.4: Emergency window

Emergencies will be a separate feature that comes in addition to the normal gameplay. They will work as USIGT requested, being released weekly or monthly, and should be active only one at the time. To encourage users to participate special rewards will be provided for participating and additional rewards to the winner. The results will be given after a sufficient period of time to allow as many as possible to partake without players forgetting about the competition. This feature could also be a great opportunity to introduce new contentend. When a new level is created by a qualified surgeon, they have the option of making it an emergency and selecting appropriate rewards, and after the competition is finished it is added to the list of available levels in game.

In addition to these concrete requests from USIGT, they had some ideas regarding the playable content of the game. It was recommended that the game mode from the previous *Master Neurosurgeon* was implemented, as this had been proven successful despite some technical difficulties. Further they suggested a game mode where players would try to locate known landmarks in ultrasound images, to better be able to orientate in such images. A general ultrasound knowledge mode could also be interesting, where players would have the opportunity to learn general information about the image modality.

9.5 Collaborative exercises

It was suggested in the prestudy that collaboration exercises is a great motivational tool in serious games. However creating a game that requires other people to play can be a gamble in regards to number of users. Unless players are encouraged to play during certain time periods, for example during class, there is no guarantee there will be anyone to play with if the number of active players using the application is low. Because of this the main challenge with implementing collaborative exercises for this game was finding a way to do it so that the main game play could stay single-player.

The solution was inspired by Charlie Kim[36] and his desire for his employees to exercise. To achieve the result he wanted he facilitated teamplay by allowing co workers to work together and added a leaderboard for workout hours. In three years he increased the percentage of employees working out twice a week from 5% to 80%.

In *Master Surgeon* logged in players will be registered to a working place, a school or a different organisation. A leaderboard after the infinite model will be available in the every player's profile, showing not only their own score, but how their organisation rank according to others. This way as players progress and gather points from levels, their score will contribute to how their workplace or organisation are doing. By creating this kind of collaborative leaderboard actual gameplay is allowed to stay single-player and available at any time, while people within an organisation will be motivated to do better not only for themselves, but for their team. For any collaborative exercise to work however it is important that the teams are genuine. The size of a team might be also be a factor for the rate of it's success. Both these things should be explored in the final evaluation.

9.6 Offline mode

As mentioned logging in with an account will be optional in Master Surgeon to lower the threshold for new players and help with onboarding. To get the motivational value from the competition and cooperation created by the leaderboards it is however preferred that players create accounts. To motivate users to do this, messages will be displayed in the location where competition and cooperation elements would normally be shown, containing information as to what will be visible here if the player was logged in, as well as a button that will take users to the create account view. An example is shown in figure 9.5, where the best ranking players would be shown if the player was logged in.

It should also be considered whether or not the game should require players to be online. Today we have access to internet almost anywhere we go, but when developing for smartphones it is still an disadvantage to force players to rely on a steady internet connection. Restricting single player content with the requirement of being online makes little sense unless it is used for preventing piracy or to protect vital game data, for example to limit cheating. Since none of these are significant factors in Master Surgeon it will be possible for players to continue playing when offline.

9.7 Ultrasound game-mode

As requested from the representative from USIGT the game should contain some way for players to learn general knowledge about ultrasound. A suggestion for this game mode is to use a story driven series of questions. The player is presented with some information regarding ultrasound though the story, and then has to answer a question regarding some of that information. This is repeated until the story is finished.

For example players are tasked with scanning a patient and are presented with images and an explanation of how the ultrasound probe works. They then have to answer a question regarding this information, before the story continues and the players are shown how the images are created analysing the signals that were reflected back to the probe. The information concerning this mode and the vage definition of "general ultrasound knowledge" results in a broad design for this mode. It is also considered an additional feature as the main purpose of this application is to teach ultrasound interpretation.

Chapter 10

Final Design

Here we will show how the game elements discussed in the previous chapter were put together to form the final game. Each screen and the connection between them will be described. Figure 10.1 shows an overview of the different screens and how they may be navigated by the player.

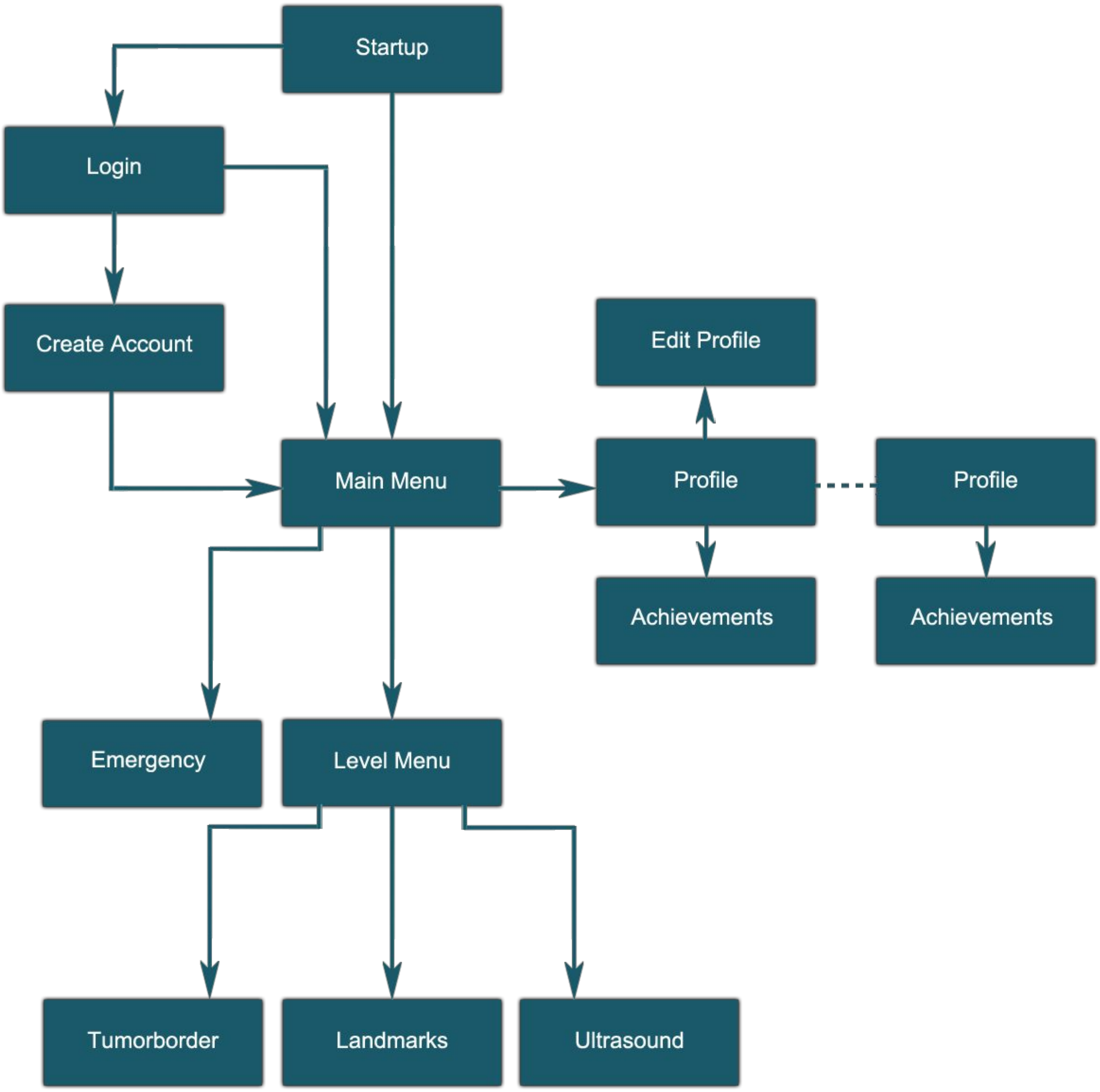


Figure 10.1: Game screens

10.1 Screen Flow

In the *startup screen* the player will have the option of sign in or to start playing directly, see figure 10.2. To have a button here for creating an account was deemed redundant as new players will most likely want to play the game first and will be able to navigate to the *create account screen* from several other screens later in the game. For those interested in creating an account before starting the game or want to sign in with an existing account may do so from the *login screen.*

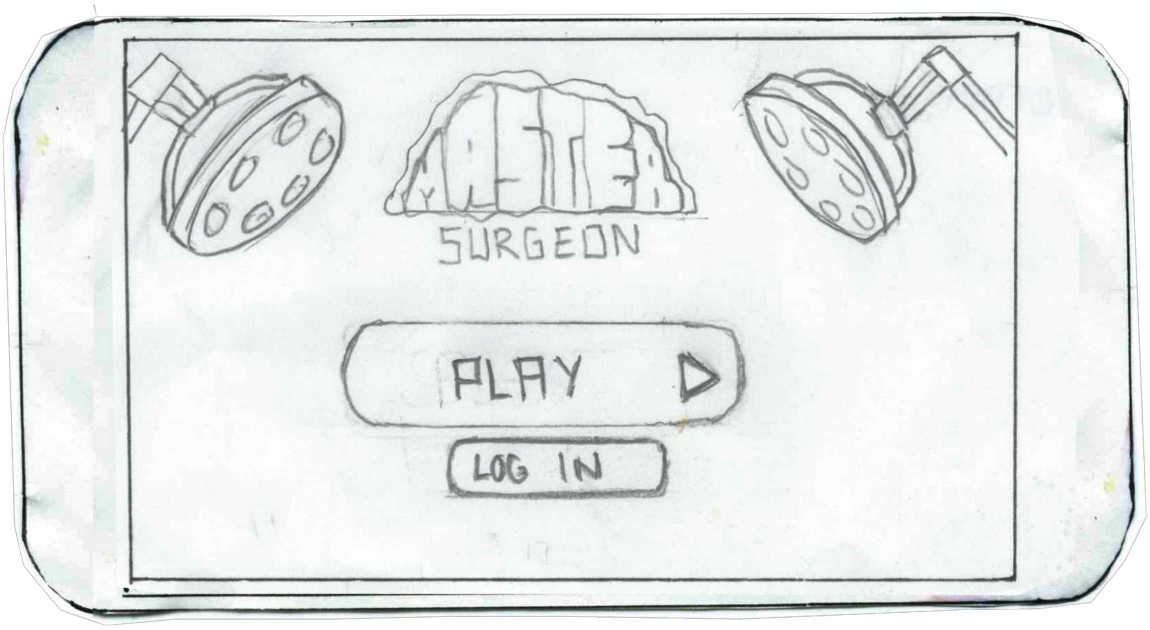


Figure 10.2: Startup screen

The *create user screen* allows users to create an account. They are required to sign up with a email address and custom password. As mentioned the player will also have to select an avatar, a user name and an organisation. Both avatars and organisation will be selected from a prepared list, while username will be created by the user. A quick way for players to suggest and for the system to validate new organisations should be considered as it could be hard to know before hand what organizations will take part in the application.

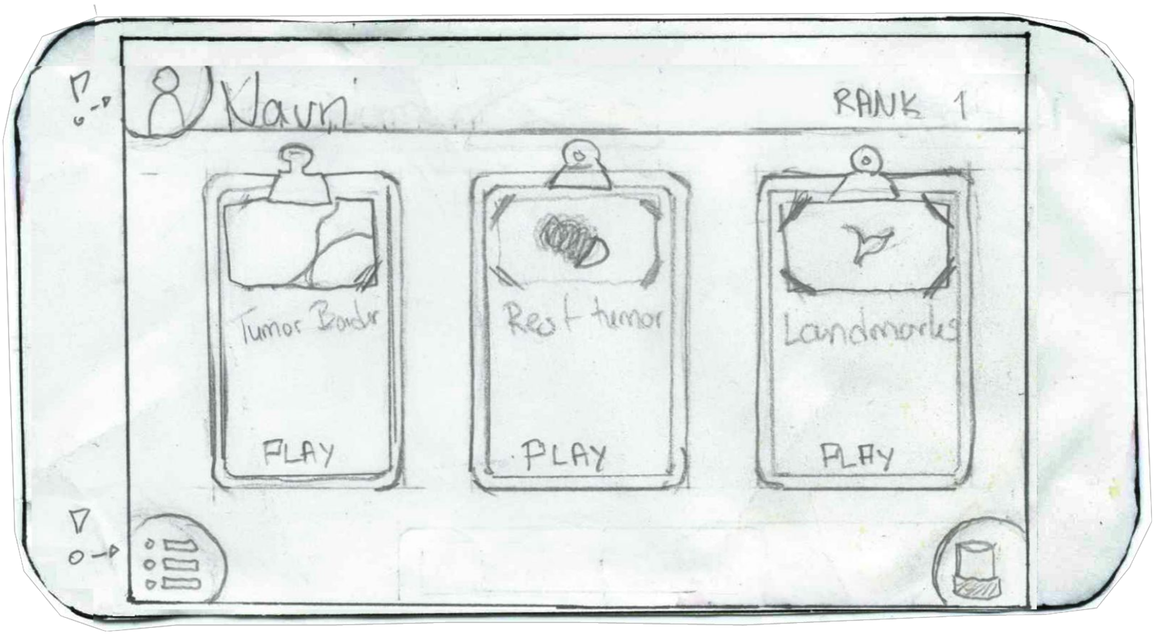


Figure 10.3: Main menu screen

The game-modes, player profile and emergencies will all be available from the *main menu screen*, see 10.3. If an emergency is currently active, there will be a rotator light in the lower right corner, which should signal users that something important is going on. If no emergency is active the light will be turned off, but will still display an explanation of the feature if clicked. In figure 10.3 three game modes are available, among them one called rest-tumor. This was removed as the content overlapped with the tumorborder mode, and the menu should instead hold a button for the general ultrasound knowledge game mode. A thumbnail of the player's avatar and their experience points progress will be displayed at the top of the screen. This area is clickable and will take the players to their profile.

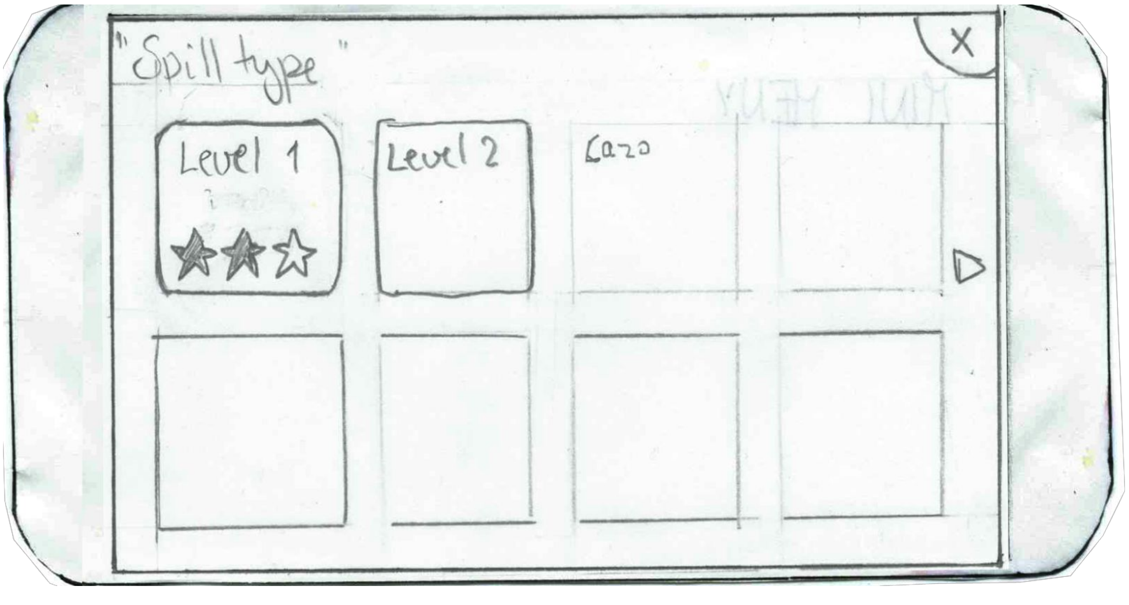


Figure 10.4: Level menu screen

In addition to the main menu screen there will be an additional menu for each game mode-containing all available levels, the *level menu screen* shown in figure 10.4. As mentioned these will be ordered by difficulty, and marked with the star-score.

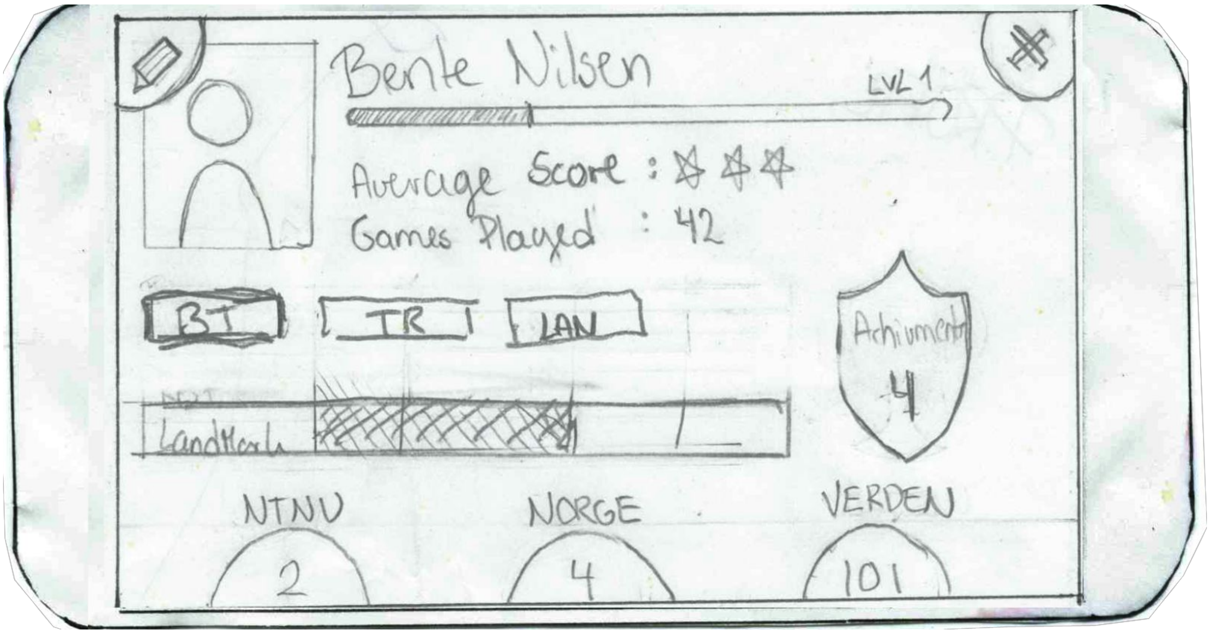


Figure 10.5: Player profile screen

The *player profile screen* will contain all information on how the player is progressing and performing. There will be an average star-score for all game modes and an option for players to select a specific game mode and see the average score for that mode. There will be a button to access the achievement screen and for editing the profile. In the bottom of the screen are three buttons that can be clicked to show the different leaderboards.

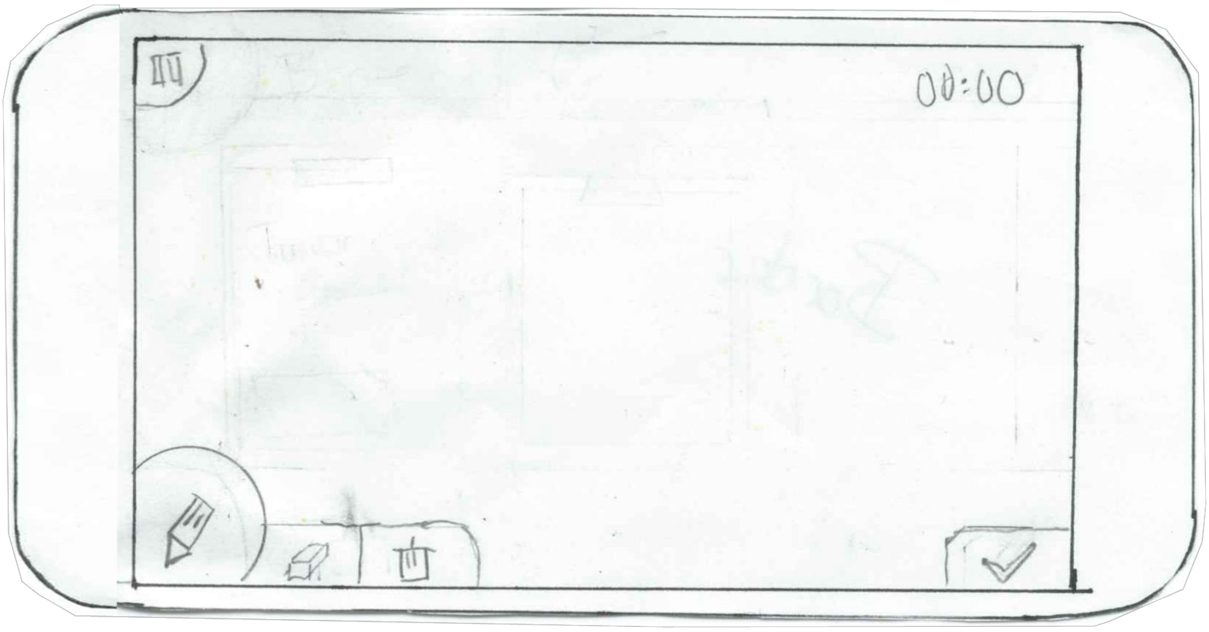


Figure 10.6: Tumorborder screen

The different game modes will have different options available to the user depending on what mode the player is in and these have been thoroughly explained in chapter 8. Common for all of them is however the possibility to pause the game and return to the menu.

10.2 Windows

In addition to these screens the game contains two smaller windows. These are the result window that appears after a game, and the drop down leaderboards in the player profile. They are displayed in figure 9.2 and 10.7 respectively.

The *result window* provides the player with information concerning how they scored on a given level. In figure 9.2 a player has earned two stars out of three, received 96 experience points and somehow used zero seconds to complete the level. The players level progression is also displayed, represented with a bar. From this window the player will have three options to choose from; try again, play next or go back to the main menu.

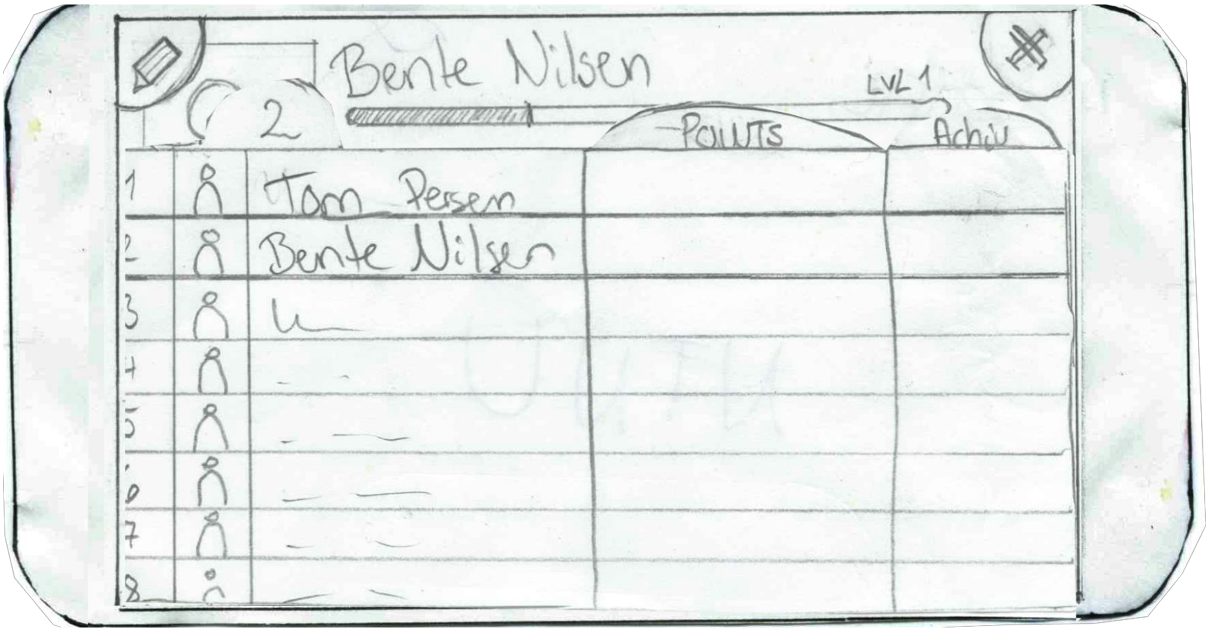


Figure 10.7: Leaderboard dropdown in player profile.

The drop down leaderboards located in the player profile are accessible through buttons at the bottom of the screen, see figure 10.5. The buttons display the player or the player's workspace/class position in the leaderboard. In figure 10.5 the player is ranked second at her workplace and 101th in the world, while the workplace rank fourth in the city. From the leaderboards concerning individual players, the row for each player in the list will be clickable and will take users to the selected players profile, allowing users to inspect the competition's profile. For the sake of hiding sensitive information only players level and achievements should be viewable.

10.3 Final design critique

It was pointed out in the prestudy that choosing the right interaction style is an important success factor in serious games. Drawing precise tumor borders with one's finger could present a challenge as this is not a precise interaction style in itself. Using a custom pencil for touch screens may help this, but not many have those laying around. Allowing more room for error in the drawings and distance between points when scoring players may help keep them motivated, but this does not solve the problem of inaccuracy with touch screens.

A more general criticism is one concerned with the gamification approach used in the design process. Defining player types and adding game mechanics to tailor a motivating experience external to the actual game, rather than making a fun game in itself is suggested by Schaffer[16] as a poor way to create flow. He further states that attempting to motivate players with rewards in this way "can make them feel controlled and less like they are playing a fun game".

10.4 Final design and implementation

The final design as it is described here is ambitious and would require more time to implement than what has been allowed for this project. As mentioned is was however important that a fully functional design was created so that it may be used by other developers to further improve the prototype. To make sure that the prototype is both usable and possible to evaluate at the end of this project, some important elements from the final design will be selected for implementation. The main goal is to have a playable game and the elements selected will be prioritized as follows;

1. The Tumorborder- and Landmark-mode will have the highest priority, and should both have playable levels at the end of this project. In addition to this, the scoring from each game should be finished and saved locally so that there exists some way for players to determine how well they have done, and what levels they have completed.

1. For the game modes to be playable the tool used to create solutions needs to be completed, and a way to share these new levels with players should also be implemented.

Ideally the player profile and handling of users should be implemented to allow testing of leaderboards, but as this is not needed for the game to be playable it will be lower prioritised.

Chapter 11
Requirement Specification

Based on the final design and prioritization of game elements, we here present the functional and nonfunctional requirements established for the prototype. We first describe the functional requirements specified for the game before moving on to the requirements for the tool.

11.1 Functional requirements

These are the functional requirements for the overall system. For easier use during development the requirements for the two game modes; tumorborder and landmark, are listed separately.

#	Description	Priority
FR1	The system should be able to automatically check for new levels on the server	High
FR2	The system should be able to download and store new levels locally	High
FR3	The system should track information about the user, what levels have been completed and the score and time for each level, and store this locally	High
FR4	The system should be able to show all levels for each game mode and possible star-scores	High

Table 11.1: Functional requirements

11.1.1 Functional requirements: Tumorborder

These are the functional requirements specified for the tumorborder game mode:

#	Description	Priority
TFR1	The player should be able to draw on and erase from the image	High
TFR2	The player should be able to clear a drawing completely	Medium
TFR3	The user should be able move the image around using the “flicking”-gesture	High

TFR4	The user should be able to scale the image by using the “pich”- and “unpich”-gesture	Medium
TRF5	The system should be able to draw the solution border in the image	High
TFR6	The system should be able to compare two drawings and calculate the average pixel distance between drawn points	High
TFR7	The system should be able to calculate the average distance in millimeter if millimeter-to-pixel ratio is given for the image	Medium
TFR8	The system should display the results from a level after it has been completed, showing star-score, pixels, time and millimeter if available.	High
TFR9	The system should track the time used to complete a level	Medium
TFR10	The system should give the user the opportunity to move the image around after the results have been presented	Medium

Table 11.2: Functional requirements, Tumorborder

11.1.2 Functional requirements: Landmarks

These are the functional requirements specified for the landmarks game mode:

#	**Description**	**Priority**
LFR1	The system should be able to display the current landmark-name	High
LFR2	The user should be able to select a position in the image by clicking it	High
LFR3	The user should be able move the image around using the “flicking”-gesture	High
LFR4	The user should be able to scale the image by using the “pich”- and “unpich”-gesture	Medium
LFR5	The system should be able to display the solution on the screen, both as an area and a point.	High
LFR6	The system should be able to calculate the distance between the user selected point and the solution; both area and point	High
TFR7	The system should be able to calculate the average distance in millimeter if millimeter-to-pixel ratio is given for the image	Medium

TFR8	The system should display the results from a level after it has been completed, showing star-score, pixels, time and millimeter if available.	High
TFR9	The system should track the time used to complete a level	Medium
TFR10	The system should give the user the opportunity to move the image around after the results have been presented	Medium

Table 11.3: Functional requirements, Landmarks

11.2 Creation-tool functional requirements

This requirements are specified for the tool used to create solutions. The users who interact with this part of the system can be considered admin users, and the functionality described here should not be available to the average player.

#	**Description**	**Priority**
CFR1	The user should be able to open a image from local storage and load it into the application	High
CFR2	The user should be able to draw on and erase in the image	High
CFR3	The user should be able to draw and erase in the image. For landmarks is should also be possible to set a point in the image, depending on game mode	High
CFR4	The user should be able to clear the drawing completely	Medium
CFR5	The user should be able to set millimeter-to-pixel ratio in the image	Medium
CFR6	The user should be able to save the image and type in a landmark name for landmark levels	High
CFR7	The system should be able to save levels, both locally and on the server	High

Table 11.4: Functional requirements, Creation-tool

11.3 Non-functional requirements

Some non functional requirements are also selected for the application:

#	**Description**
NFR1	The application should run on the latest versions of iOS, Android OS and Windows
NFR2	The game should be easy to use, and require less than 5min to learn for each game mode.
NFR3	The code written for the application should be easy to read and reusable for future development
NFR4	The tools used to create solutions should be made unavailable for other than qualified surgeons

Table 11.5: Non-functional requirements

NFR1 will be achieved by using the Qt framework. There will however always be platform specific settings that should be considered to make the application run smoothly on all devices and keep a decent looking GUI. NFR2 is strongly connected with how attractive the game will be to new players, and only a small amount of time and actions should be needed for a player to get from startup to playing one of the game modes.

As was shown in the design chapter of this thesis that there exists several options for this prototype to be improved, and for it to become a fully engaging serious game some further work is necessary. NFR3 would help new developers build on existing code or use parts of it for further development. NFR4 is important for the authenticity of the levels in the application. This could be done by having admin accounts that makes this content available to specific players or by creating separate applications for the game and the creation-tool.

Chapter 12

Implementation

In this chapter we will discuss implementation of the application. We briefly describe architectural patterns and models used, before moving on to implementation of the functional requirements. A short description of the Qt framework was done in the prestudy and it was found that it offers several language options for implementing the same functionality. Deciding what languages and APIs to use for the game was therefore the first challenge that had to be faced. Because of this we describe how each language were used and why. A more thorough description of QML is done since it is not as common as the others languages.

12.1 Architecture

To ease further development of the application and the possible merge with the other part of the system, writing lucid code and having a modular architecture will be a great advantage. To achieve this, an attempt to follow the model view controller pattern was made. It was also necessary to consider how the levels used in the game should be saved and shared with players.

Model View Controller pattern

The model view controller pattern is an often used pattern for separating UI and logic. The user interacts with the controller and receives updated information from the view. The model receives, processes and sends information between the controller and the view. Working with QML, this is not as easily achieved as with other languages, since a QML element can be considered both the view and the controller. As an example, the canvas element both displays the drawing by the user and contains a mouse registration element to receive input from the user. This structure of elements within elements will be explained later in this chapter and is the most effective and intended use of the elements. Instead QML code will be divided into “screens”, where each file will be concerned with the context of one state of the application, i. e. one for loading, one for main menu, one for each game mode and so on.

It is still possible, and in our interest to keep the logic separated. Both C++ and JavaScript files may be imported and used by QML, and this will be done in the same way as with QML-files. Each “screen” will have it’s own QML-file that communicates with its own JavaScript code. Downloading new content and saving it locally should also be done separately, but as this requires element from both JavaScript, C++ and QML, the entire process will be contained within a separate file.

Client–server model

For this application using peer-to-peer was never truly an option. The main reason for this is mentioned in the design chapter, and concerns one of the major challenges with peer-to-peer games, i. e. the number of active players. To acquire new content, using peer-to-peer would make players depend on other online users to already possess the necessary data. Knowing that we have no garanee for how many players this game will have, and when they will be playing it, it is too much of a risk to share levels using the peer-to-peer model.

To use the client-server model was therefore decided quite early in the project, and the important question for implementation was then how and where data should be stored. As mentioned the other part of this application is developed to store and display data for USIGT, and during the design phase of this project it was discussed whether or not their database could also hold the data for this project. As the development progressed it became however apparent that this would not be achievable during the limited time period allowed for these projects. Since it had already been decided that the applications would be implemented using the Qt Framework, selecting the Qt Cloud service as the main data storage was therefore a natural choice. As mentioned in the prestudy Enginio, one of the services offered by Qt, supports easy-to-use handling of files and users which was a good fit for this project.

12.2 QML

Qt Quick and QML is designed to provide the means to create dynamic and easily implemented user interfaces. The game developed in this project is heavily UI-centric, and much of its required functionality can be easily implemented using QML. Being a relatively new and uncommon language, the tutorials provided by Qt Creator[9] was a great help in deciding how to structure the code for the application. As described earlier, this is done by creating QML files with a corresponding JavaScript file for logic. Before the selected solutions for implementation is discussed, a short introduction to the basics of QML is presented.

Elements

The building blocks of the language are called *elements*. An *element* is declared with its name and contains predefined functions and variables that can be changed both within or outside the element. Below is an Image element with its size and source set.

[9] http://doc.qt.io/qt-4.8/tutorials.html

```
Image{
      id: myImage
      height: 100
      width: 100
      source: "/images/image.png"
}
```

Elements are accessible from other code by using their id. The source of the Image element above might be set some place else in the code with:

```
myImage.source = "/images/image.png"
```

Elements may also be nested within each other to create a form of heritage. In the code below the Image element refers to *parent*.height, which in this case is the height of the Item. Custom elements can be made by using this nesting and writing them in a separate QML-file. The code below represents a Button element that contains an Image element, a MouseArea to register mouse activity and three properties that are used for logic. The signal is used to connect mouse or gesture clicks registered by the MouseArea to the Button and allow usage of an onClicked function in the Button element.

```
Item {
  signal clicked
  property alias imgSrc: image.source
  property alias pressed: mouseArea.pressed
  property bool active: false

  height: image.sourceSize.height
  width: image.sourceSize.width

  Image {
      id: image
      height: parent.height
      width: parent.width
      anchors.horizontalCenter: parent.horizontalCenter
  }

  MouseArea {
      id: mouseArea
      anchors.fill: parent
      onClicked: parent.clicked();
  }
}
```

By saving this in a separate QML-file and name it Button this element may be used in other QML-files like shown below:

```
Button {
      id: button
      imgSrc: "button.png"
      width: screen.width
      height: 100
      onClicked:{
            gameMode = false;
            showLevels(JS.levels_lm);
      }
      onPressedChanged: button.imgSrc = "activeButton.png"
}
```

Canvas

An important QML element used in this application is the Canvas. This element provides a 2D area that can be drawn on using JavaScript. It is used together with the Context2D API, that includes premade functionality that is used to manipulate the content of the canvas.

Below is a representation of how the canvas element was used. It is nested within a Flickable and an Image element, and in doing so it inherits their properties. The canvas will scale to the same size as the image, and move with it as it is flicked. Doing this ensures that the pixel-position registered when drawing will always correlate to the pixel position in the image.

```
Flickable {
      .
      .
      Image{
            Canvas {
                  MouseArea{
                   onPositionChanged:
                       if(!flick.interactive){
                           JS.setPosition(Math.floor(mouseX),Math.floor(mouseY))
                           canvas.requestPaint()
                       }
                  .
                  .
                  .
```

The canvas also contains its own MouseArea that detects input from the user. As the user input changes the onPositionChanged function is called, which directly communicates with JavaScript logic. To show the connection between the QML element and the logic, the function that is called by the MouseArea is shown below.

```
function setPosition(xNew, yNew){
   mouseX = xNew;
   mouseY = yNew;
}
```

It is mentioned in the functional requirements that zooming would be an important functionality for players to be able to draw accurately. The initial plan was to use the Context2D APIs own functions to draw lines and points in the image, and then extract the necessary information later. This would allow usage of the canvas elements premade scaling functionality to implement zoom. It became however apparent that this would be harder to achieve than expected. Extracting the necessary information from the canvas was not as easily done as first anticipated, and there was no longer any correspondence between coordinates given in the canvas and those in the original image.

To be able to compare the drawing made by players with the level solution a manual registration of drawn points and lines where implemented. This will be explained later in this chapter. By handling the input points directly, zooming would also have to be implemented manually. To scale both the image and the canvas is easily implemented, but if the players scales mid-drawing the pencil thickness would also have to be scaled. As will be better explained later in this chapter, comparing a drawing with the solution is done using one-pixel thick lines, so for the actual comparison the line would have to be scaled back down to one pixel thickness.

After assessing the time required to implement this versus the impact it would have on the game, it was decided to leave zooming out of the final product. Much time had gone into creating a solution that did not work, and there was little time to implement an even more advanced solution for zooming.

Threads

To handle calculations of results separately and to free the main GUI thread the QML element WorkerScript was used. This elements starts a separate thread that will complete a JavaScript function with a set of given parameters. Below it is shown how the WorkerScript element was used in the project.

```
WorkerScript {
       id: myWorker
       source: "tumorBorder.js"

       onMessage: {
           result =  messageObject.result;
```

The element is used via its id and the function sendMessage. No external variables can be accessed after the JavaScript method has been called, so every value required for the calculations needs to be sent with this function.

```
myWorker.sendMessage({ height: canvas.height, width: canvas.width,
      solution: Input.level.solution, points: JS.getPoints()});
```

A WorkerScript function is located in the JavaScript file, and redirects the call made from the element to the function that is to be executed.

```
WorkerScript.onMessage = function(message) {
   var calculatedResult = send(message.height, message.width,
      message.solution, message.points);
   WorkerScript.sendMessage({result: calculatedResult});
}

function send(h, w, solution, _points){
      //calculate distances
      return (totalDistance/tempPoints.length);
}
```

Lastly the result is returned to the WorkerScript element and in this case saved as the variable result to help determine what scores that should be displayed on the screen and saved in the local database.

12.3 JavaScript

As explained and shown above JavaScript has been the main language used to implement the logic of the application. In addition to the adoptions shown above JavaScript is used to calculate lines drawn and removed by players, store all registered points, compare these with the solution and in addition updating the local database. Below is a short description of the more noteworthy usages of this language.

Drawing and solution calculation

When a player starts a new tumorborder level the *setUpCanvas* function creates an array with the same dimensions as the image and canvas called points and fill it with zeroes. As the MouseArea registers input from the user the x and y position is sent to a javascript that registers this position and the canvas is updated. When this update happens, if the point registered is the first position drawn this point is registered as the oldest. This will happen after a new level has been started, the canvas has been reset using the thrash tool or if the user paused in the drawing of a line. Regardless of when the point was registered it is also set as the latest. A function is then called to draw a line between the oldest and the latest positions in the canvas, before setting the latest point to equal the oldest.

```
function paintLine(){
      //If this is the first point, set this as the oldest
      if(oldX == -1 && oldY == -1){
            oldX = mouseX;
            oldY = mouseY;
      }
      //Draw the line between old and latest
      drawPoints(oldX, oldY, mouseX, mouseY);
      //Set oldest to latest
      oldX = mouseX;
      oldY = mouseY;
}
```

This is done to compensate for delay in input detection. Even the faster devices would create dotted instead of continuous lines when considering only the actual input from the user. The function used to draw lines is shown below and created based on the Bresenham's line algorithm[10]. This calculates what the values for each point in the line should be, and the corresponding coordinates in the points array is set to one and drawn on the canvas using the *ctx.fillRect* function. This way, when a drawing is created, the points array will have zeroes where no points has been registered and one where there is currently a drawing.

```
function drawPoints(x0, y0, x1, y1){
      var dx = Math.abs(x1-x0);
      var dy = Math.abs(y1-y0);
      var sx = (x0 < x1) ? 1 : -1;
      var sy = (y0 < y1) ? 1 : -1;
      var err = dx-dy;

      while(true){
            points[x0][y0] = 1;
            ctx.fillRect(x0, y0, 1, 1);
            if (Math.abs(x0-x1)<0.0001 && Math.abs(y0-y1)<0.0001) break;
            var e2 = 2*err;
            if (e2 >-dy){ err -= dy; x0  += sx; }
            if (e2 < dx){ err += dx; y0  += sy; }
      }
}
```

The reason for storing an array with the same size as the canvas, instead of simply storing them in a one-dimensional list was that implementing the erasing tool would be both easier and take shorter time to use. When erasing, the area around the point given by the user can be directly set to zero, instead of searching through a list of coordinates. It was also the intention that this solution would later work better for implementation of zoom.

[10] http://en.wikipedia.org/wiki/Bresenham's_line_algorithm

When a player ends a level by handing it in, a WorkerScript is used to traverse the points array and locate each point that has been set to one. Each coordinate is then put in a separate list that is compared with the solution array. This is done using euclidean distance[11], and comparing each point from the suggested drawing with each point in the solution array.

```
//For each point in the suggested solution
for(var i = 0; i < tempPoints.length-1; i++){
        //distance between this and first point in solution array
        distance = Math.sqrt(Math.pow(solution[0].x - (tempPoints[i].x), 2) +
        Math.pow(solution[0].y - (tempPoints[i].y), 2));
        //For each(-1) point in the actual solution
        for(var j = 1; j < solution.length-1; j++){
                //Check if distance between points are smaller
                tempD = Math.sqrt(Math.pow(solution[j].x - (tempPoints[i].x), 2) +
                Math.pow(solution[j].y - (tempPoints[i].y), 2));
                if(tempD < distance) distance = tempD;
        }
        totalDistance += distance;
}
return (totalDistance/tempPoints.length);
```

The average distance is returned to the WorkerScript. This solution for calculating scores requires that players draw the full border. Since it compares the user's drawing to the solution, the player is rewarded equally for drawing only a part of the border as long as it is accurate compared to the solution. A a fix to this problem could be finding the difference between number of points drawn by players and the number of points in the solution. This value may then be used to adjust the score players receive.

This implementation also assumes that the drawing by the user consist of one single line, with the thickness of one pixel. To implement zoom the lines drawn after an image has been scaled would need to be adjusted for this solution to work.

Local Storage

JavaScript is also used to handle all connections with the local database. As mentioned in the prestudy Qt includes its own API for Local Storage, and this can be used through JavaScript. The storage created is a SQLite database, and below it is shown how the database used for Master Surgeon is created.

```
db = LocalStorage.openDatabaseSync("LocalDB", "1.0", "Local Database", 1000000);
```

[11] http://en.wikipedia.org/wiki/Euclidean_distance

The parameters used are the new database's identifier, version, description and estimated size if the database in bytes. *Master Surgeon* use three separate tables to store information relevant for the game. Two contain information regarding levels, one for each game mode, while the last table list all the scores achieved on the device running the application. The tables are connected through the key attribute id, which is the same key used for each level in the Engenio cloud storage. This identificator contains both numbers and letters, and is therefore saved as TEXT in the database.

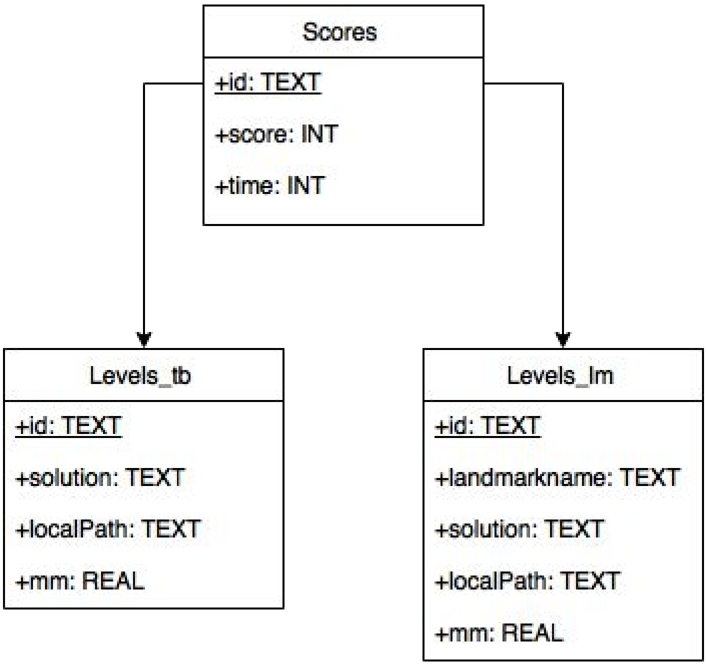


Figure 12.1: Local Storage

Today Master Surgeon is still a single player game, and the local database could in theory just consist of one table containing information for each level, including the scores. As discussed earlier it is however a future goal that the application will include player accounts, and to better facilitate the implementation of this functionality the tables where separated.

Below is an example of how the content downloaded from the cloud is put in the local database.

```
function insertLevels_tb(id, solution, localPath, mm) {
        db.transaction(
                function(tx) {
                        //Insert into db
                        tx.executeSql('INSERT INTO Levels_tb VALUES(?, ?, ?, ?)', [id,
                        JSON.stringify(solution), localPath, mm]);

                        //Insert into JS list
                        JS.levels_tb.push({id: id, solution: solution, localPath:
                        localPath, mm: mm});
                }
        )
}
```

This function is used to create a *tumorborder* level in the *levels_tb* table, and this function is only called after is has been confirmed that the level does not already exist in the database. Further, all information from the local storage is loaded into JavaScript lists. These lists are later used to display the level menu, and later play the different levels.

12.3 C++

Although most of the logic were implemented using JavaScript, some of the systems functionality concerning downloading of files and platform specific features needed to be created using C++. To be able to access C++ classes from QML they do have to be registered as a Type. This is done in the main C++ file:

```
#include "imageloader.h"
qmlRegisterType<ImageLoader>("ImageLoader", 1,0, "ImageLoader");
```

It is also necessary to import the class, but then it may then be declared and used as any other QML element:

```
import ImageLoader 1.0
//Imported C++ class
ImageLoader{
      id: imageLoader
}
```

Saving images

To be able to access the images saved locally through the application it were desirable to save them in a location that could be accessed in the same manner on all devices. Being a cross-platform framework this is of course supported by Qt, and can be done using the QStandardPaths class[12]. This class contains set of standard paths that can be used across several platforms. For this application the QStandardPaths::AppDataLocation were deemed the most fitting location as it is described to return a “directory location where persistent application data can be stored“. This did however appear to not be working properly in Qt5[13], and QStandardPaths::DataLocation were suggested and used instead, as it returns the same value as AppDataLocation.

```
void ImageLoader::prepareImage(QString id, QString name){
      Image tempImage;
      tempImage._name =
```

[12] http://doc.qt.io/qt-5/qstandardpaths.html

[13] http://comments.gmane.org/gmane.comp.handhelds.sailfishos.devel/5131

```
      QStandardPaths::writableLocation(QStandardPaths::DataLocation) + "/" +
      name;
      tempImage._id = id;
      imageList.append(tempImage);
}
```

The code above is used to prepare new images for download, where the id and the name of all new levels are temporarily stored in a list. The usages of QStandardPaths is shown here and included to already determine the path where the images should be saved too.

Downloading images

When attempting to download files using QML, the larger images would not fully dowload as the calls made to the cloud returned faster then the images were being downloaded. To solve this issue C++ were used to store a list of everything that needed to be downloaded, and then do the download while making sure each image were allowed to finish downloading.

```
void ImageLoader::saveImage(int time, Image image){

   QEventLoop loop;
   QTimer timer;
   timer.setSingleShot(true);

   imageHandler->saveImage(image._id, image._name);

   connect(&timer, SIGNAL(timeout()), &loop, SLOT(quit()));
   connect(imageHandler, SIGNAL(imageSaved()), &loop, SLOT(quit()));
   timer.start(time);
   loop.exec();

   if(timer.isActive()) loop.quit();
   else saveImage(time + 1000, image);
   if(time > 10000 ) loop.quit();
}
```

To do this the classes QEventLoop[14] and QTimer[15] were used together with the Signal and Slot functionality that is included in Qt framework. After each level has been checked by QML, the list of new leaves will be traversed and each corresponding image downloaded. The timer and loop is used to give each image some time to finish before starting the download of the next one. If the image is finished before the timer runs out, a signal is emitted and the loop stops. If it is not finished, the image is given some extra time to finish.

```
void ImageHandler::onDownloadFinished()
{
   QByteArray imageData = _reply->readAll();
   QImage *image = new QImage();
   image->loadFromData(imageData);
```

[14] http://doc.qt.io/qt-4.8/qeventloop.html

[15] http://doc.qt.io/qt-4.8/qtimer.html

```
   image->save(_filePath);
   emit imageSaved();

   _reply->deleteLater();
   _reply = 0;
}
```

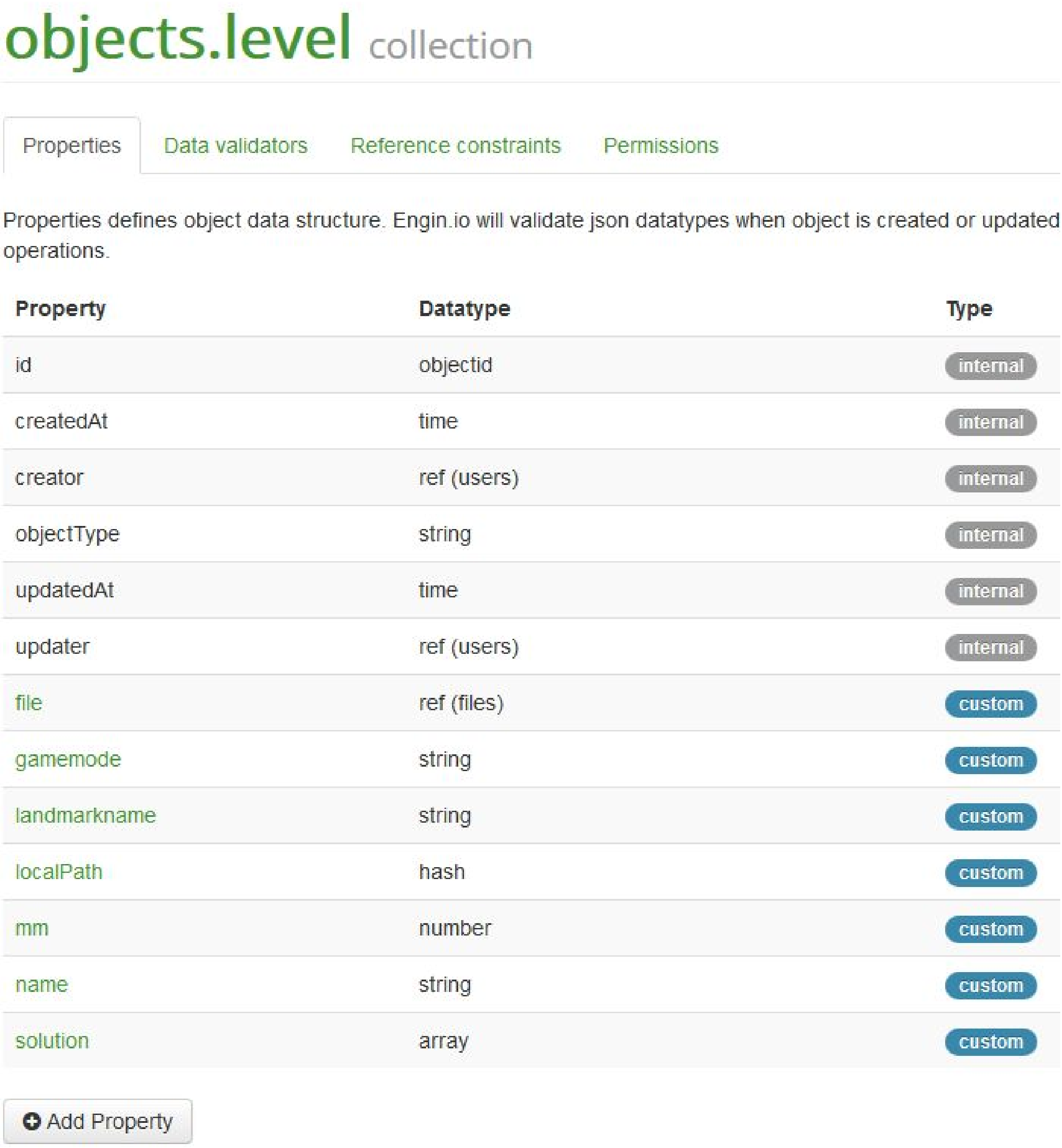
This implementation has one major drawback. There is always a risk when using loops as they could potentially block the application from continuing. QEventLoop will be stopped when an image finish downloading, but if the connection to the database is cut during download, the application could potentially freeze. There is a check to see if the timer has been going for too long, and the download will never start in the first place if there is no connection to the cloud at startup, but solving this issue without loops could be a potential area of improvement for future work.

12.4 Enginio

As discussed the Enginio data storage were selected as *Master Surgeons* online storage. Engino uses JavaScript Object Notation(JSON)[16] to gather the data in objects, which can be created and edited manually using Enginios own browser based interface, see figure 12.2. Below is an example of one such JSON-object for a landmarks solution where the answer is a point.

```
{
 "mm": 8.50005744465883,
 "file": {
      "id": "55475e58d45dfe6cd5030a95",
      "objectType": "files"
 },
 "gamemode": "lm",
 "landmarkname": "Sulci",
 "localPath": {},
 "name": "OK4_Course_Land_Sulci.jpg",
 "solution": [
      {
      "x": 469,
      "y": 100
      }
 ]
}
```

[16] https://en.wikipedia.org/wiki/JSON



objects.level collection

Properties defines object data structure. Engin.io will validate json datatypes when object is created or updated operations.

Property	Datatype	Type
id	objectid	internal
createdAt	time	internal
creator	ref (users)	internal
objectType	string	internal
updatedAt	time	internal
updater	ref (users)	internal
file	ref (files)	custom
gamemode	string	custom
landmarkname	string	custom
localPath	hash	custom
mm	number	custom
name	string	custom
solution	array	custom

Figure 12.2: Enginio object screen

Initially the different game modes were saved separately as *object.level_tb* and *object.level_lm*, but this was changed as the information contained in the objects where very similar, and combining them made downloading levels easier to achieve with QML. A JSON-object containing a *tumorborder* level does therefore look much like the one above where the *solution* will contain significantly more points.

Part IV

Results

Chapter 13

Evaluation

In this chapter we will present the results from the application evaluation. The evaluations were completed individually, where one player would attempt to complete a given set of tasks. Two groups of participants were used for the evaluation, one where the application was running on a tablet and later one where it was executed using Mac computers. This was done to see how the game performed on different platforms and how much the distinct interaction styles affected how the game was perceived. The participants of the last evaluation were all from the intended user group, i. e. surgeons, technicians and students, and had some level of experience with ultrasound imaging. The feedback from this group will therefore be more valuable to determine the educational content of the game, while the information from the first evaluation will help determine how the game performed on a touch-device.

13.1 Usability

To evaluate the usability of the system each participant were asked to complete a System Usability Scale(SUS)-questionnaire after using the application. The SUS-test is a simple questionnaire consisting of ten question regarding the usability of a system and is used to give an overview of how user friendly a system is[37]. It was explained to the participants of the evaluation that the questionnaire were to be anonymous, so to get the most honest and accurate description of the application's usability.

The ten questions presented are rated by participants on a scale from one to five, and these values are further used to calculate an overall score that would fall somewhere between 0 and 100. As seen below in figure 13.1, a score above 70 can be considered acceptable.

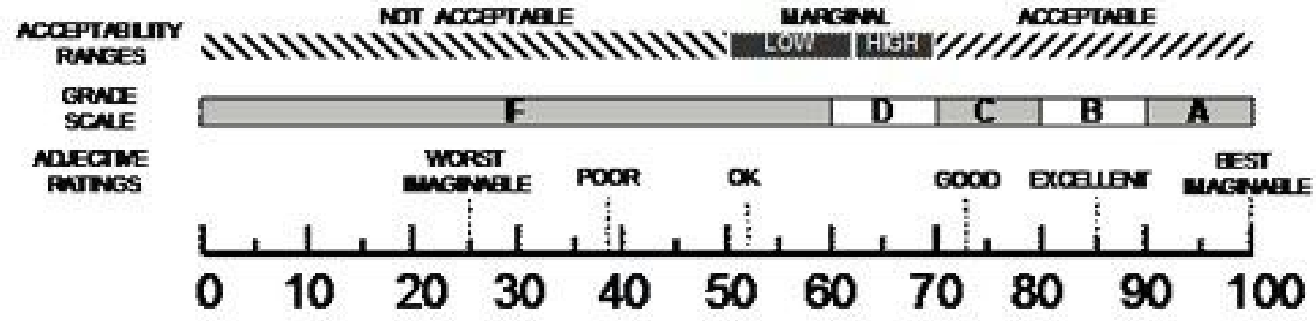


Figure 13.1: SUS Score[38]

Medical personnel, including both qualified surgeons and students, gave the application the highest score of 85,63. The technicians that work with medical imaging and ultrasound on a regular basis gave a lower score of 81,87. Looking at the scores individually, it also appears that the score is lower for participants with higher ultrasound interpretation experience and age.

The overall score from the last evaluation ends at 83,75, which compared to figure 13.1 can be considered an acceptable result. However, given that the amount of available functionality in the application is still quite low, a high result on this type of scale was to be expected.

The first evaluation executed with a tablet gave a SUS-score of 81,5. This value is still high and similar to those achieved using a Mac computer, which indicates that the game is definitely playable on both platforms. It should be noted that during the evaluation on tablets a touch-pencil was used instead of drawing with a finger. Still, some participants struggled with achieving what they perceived as acceptable accuracy using this type of interactions style.

Although the difference is marginal it is also interesting to note that the scores achieved by this system was lower than or equal to that of the previous version of the game which scored at 85.12. The evaluation of the previous version also suffered from some minor bugs and crashes during testing, while no such problems occurred during any of the sessions done with this application. This could indicate that usability vice the application has not been improved with the new implementation. However, since both groups used for testing were relatively small in size, these values are hard to draw decisive conclusions from.

13.2 Educational value and engagement

To better understand the difficulties players had with the game and answers questions concerning its educational value, an interview where done after some of the test. Players were asked questions about whether or not they found the game fun, it they thought they would get better at ultrasound interpretation from using the application or if there were elements in the game they found had room for improvment.

From both observation and the feedback from participants it was clear that *Master Surgeon* was not as engaging to play as initially hoped. Too much time was spent trying to figure out how or where to draw, and even as those necessities were explained or discovered the feedback from the application appeared to weak to encourage some excitement or sensation of winning. Most players did however appear to enjoy solving the levels, as they took their time and tried to be as accurate as possible. Most did also express excitement when told that future improvements for the application are planned.

That the application had educational value, was agreed upon by everyone that was asked. It was however mentioned by more experienced ultrasound interpreters that the application would better suit people who are less experienced with the image modality. Some of the more experienced users were more concerned with the validity of the level solution, rather than with their own performance and the feedback from the game. It was also noticed by observation that the younger participants who had less experienced interpreting ultrasound appeared more engaged while playing the game.

Participants were also asked what they thought about the idea of implementing leaderboards, competition and cooperation. The general feedback for this idea was surprising as they argued that since the game modes are single player, adding external competition would not contribute to how fun the game would be. This contradicts what Khosravifard found in her evaluations; that players would share their scores with other participants after completing a level. No such behavior was observed during this applications evaluation.

A lot of interesting feedback regarding the UI solutions were also received both during play and the intervju. This is valuable information to consider both to help evaluate why the application were perceived as it was and to help future work on the game. An issue experienced in both evaluations was that the images used in the application are abnormally dark. The tablet testing was done during some exceptionally sunny days, and the daylight made locating buttons, understanding icons and examining the ultrasound images hard. The same problem occurred during the second evaluation, as participants ask that the lights be turned off for them to better read the ultrasound images. Buttons appeared to be easier to see on a computer screen than on the tablet. Another issue that many commented on was the the uncropped images that were used during evaluation. It was difficult for participants to understand where in the image they were supposed to draw, as one image consists of several other images, both from MRI and ultrasound. The lack of a zooming functionality were mainly mentioned by participants testing on the tablet, which is probably connected with the difficulty to draw using touch.

Chapter 14
Discussion and Future Work

In this chapter we will discuss the result from the evaluation and how the information found both during the prestudy and the evaluation can be used to answer the research question selected for this thesis. Based on this discussion we will present suggestions for future work on *Master Surgeon*.

14.1 Discussion: Research Questions

RQ1: What makes a game fun?
To be able to create an engaging learning experience it was desirable to determine what it is that make games fun, for later to use this information to gamify the educational content of *Master Surgeon*. During the prestudy we found four general reasons for why people play game; for mastery, to destress, to have fun and to socialize. Considering player types, we also saw that what people actually find fun, varies with their personality and preferences. Knowledge about what different people like can be used to help a game facilitate as many people as possible, or to tailor the experience to a specific user group. We also learned of the concept of flow, a state of mind associated with fun, where one is so absorbed in the current activity that time flies and the outside world no longer matters. Lastly, we briefly touch of the effect of lottery-elements in games. Although this mechanic can be considered borderline addictive rather than fun, using this mechanic can help shape wanted user behavior and be a powerful tool to create engagement.

A side note that can be deduced from this information, that was not anticipated at the start of this thesis, is that the reason people play games is not always just to have fun. As mentioned, fun is only one of the general reasons that people play. Additionally, as long as the list of required factors for flow are fulfilled it is possible to achieve flow in situations that generally would not be considered fun, like taking an exam or reading a master thesis. Possibly, striving to make a serious game fun is not the only way to facilitate engagement.

RQ2: Which factors affect the success of an educational game?
Several guidelines and possible success factors for serious games were found during the prestudy. It was shown that selecting suitable modality, interaction style and environments for the purpose of the game is important to enhance the user's experience. Providing guidance in the game, avoiding negative feedback, using collaborative exercises and offer challenge are all suggested guidelines found to achieve success with serious games. A short list of additional factors, concerning social games as a tool for learning, states that feedback must be imminent and rewards must feel satisfying, achievable and be recognised by other players.

RQ3: Can theories from gamification and game design be used to better tailor an engaging learning experience?

During the prestudy we did find tools and information that was helpful to design for a particular player group. In this thesis we have made an attempt to use player types and selected game mechanics to create a game that would appeal to the given user group. However the feedback from the evaluation was not as promising as initially hoped, in regards to how engaging the game was to play.

The fact that the complete design for the game was not implemented could be one reason that the game did not reach our expectations. A second factor that could have affected the quality of the design is that player types on is own not necessarily is enough to define the user group. To better understand what a group of users want, other characteristics of the group should also affect design, for example age, experience with the educational content and experience with games.

RQ4: Do competition and cooperation contribute to how fun a game is?

One of the guidelines found for serious games were to create cooperations exercises. In addition it is suspected that the largest portion of the human population lean towards the socializer player type. This would indicate that designing for multiple players and cooperation in a serious game should increase its entertainment value and how engaging the game is for most players.

From the information found when presenting player types, competition however appears to be a double edged sword. Competing against other players could add an additional level of challenge and a way to achieve status. But the competition should be designed to allow for more than one winner as achievers, who like to win, can be discouraged from not being able to do so. Competition against human players could also make it harder to adjust the difficulty of the game to match the players skill level, which has been shown is an important factor to achieve flow.

RQ5: Can we create a game that teaches ultrasound interpretation and at the same time is perceived as fun?

The results from the evaluation suggest that we were only semi successful in completing a game that is both fun and educational. Although all participants of the evaluation felt the game had educational value, the entertainment value appeared to be too low. Many seemed excited by the actual game, taking their time with drawing and being as accurate as possible, but because of minor flaws in the UI design and weak feedback, the players were not as excited and motivated by the gamified elements of the game as initially hoped.

This said, the participants that took part in the evaluation were excited and supportive of further development of the game. Many claimed that with a few adjustments, the game would definitely be useful as an educational tool for ultrasound interpretation.

14.2 Future Work

We have established both in the design chapter of this thesis and by discussing the feedback from the evaluation of the application, that *Master Surgeon* has room for improvment.

Zoom
To make it easier to draw accurately and to better examine the ultrasound images, especially for tablets, a zoom function should be implemented. This feature was initially planned for this project, and valued as a medium important functionally, but after having some challenges creating this with the selected framework, the idea was temporarily scrapped. For future work however, adding a zoom-functionality should be one of the first additions to be evaluated for implementation.

Undo
A second functionality, not considered in the initial design, is an undo tool for drawing. In some of the evaluation sessions, participants would attempt to move the image with the pencil tool-selected or otherwise accidentally draw unwanted lines in the image. This would result in a lot of erasing and some minor frustration. To keep track of the last move made by users and give them the ability to undo this move, is a commonly used and known functionality that appears to be missing from *Master Surgeon.*

Images
Given that the images in the game consist of several other images made it difficult for most participants of the evaluations to determine in what image to draw. The first option considered to solve this was to crop the images beforehand, to only display the section of the original image that is to be drawn on. A problem with this implementation is that the other sections of the image contain information that is a great help and often is necessary to be able to draw the correct solution. In light of this, cropping the image to achieve more difficult levels could be acceptable, but a better solution would be to in some way mark an approximate area where the tumor in question is located. This was a functionality commonly mentioned during evaluation and should definitely be considered for further development.

Scores
From feedback received during evaluation it appears that scores received after completing a level was not clear enough on how well the player did. The millimeter-score was the most emphasized by players when discussing their score, which is natural as this is the value they would normally operate with. Many also had positive feedback concerning the textual message received when completing a level. However, the syringes, or star-score, were unclear to many. This was intended to provide an easy and quick way to determine if one did good or not. Because syringes that was not received were marked with gray, in most cases participants did not see it. This made it difficult to understand what the syringes were for. In addition, the way scores are calculated today appear to vary too much between levels as to what is accepted as a bad, decent or good answer. Giving the player scores that are meaningful and consistent is something that should be tested in future work.

UI

Today the GUI is dark and does not give enough contrast to the already dark ultrasound images. To update the game with brighter, more colorful GUI could help users, especially tablet users, to detect and understand buttons.

Most of the icons selected for buttons appeared intuitive to users, as no explanation were needed for players to understand what their functionality were. However, the eye button, used to hide the result window to better examine the solution, went undetected by many. It appears that this functionality where not as interesting to players as expected. What they did ask for however, was a functionality to hide the solution area for landmarks. If the area marked was large enough it was hard to see what the solution was actually drawn on. This is a serious flaw to the educational value of this type of level, as it makes it impossible to see what the landmark looked like in the image after receiving the result.

The marker-tool used to set a landmark also appeared redundant during testing. The tool is used to make the marker draggable to make repositioning of the mark easier. From observation it was however clear that user would just re-click the image if the first position for the marker was off. To use dragging motion to move the image and clicking to reposition the marker, the marker-tool is no longer necessary. All participants asked about this solution where positive to removing the marker tool.

Tutorial

Looking at SUS-scores, it is clear that participants in general found the game easy to understand. The answers to question 9 however, asking if the player felt safe using the application, have some room for improvment. From observation during evaluation it was also clear that players needed some time to understand what they were supposed to be doing. Concerning onboarding and flow, implementing some simple tutorials might have a positive effect on how the game is perceived. Looking at the answer of SUS-question 10, there is not much additional information needed as participants in general found they would be able to learn the game quickly. But implementing a short and concise explanation to each game mode should be considered by future developers.

Editing solutions

Today there is no option for editing or deleting saved solution. This feature is easily implemented using Enginio, and should be considered for future development. Today, changing an already saved level requires the user to exit the application, log in to the Engenio browser based interface to delete the level for so to redo the entire thing. This is not a good solution for a final release of the application.

Chapter 15
Conclusion

In this thesis we have studied theories from game design, gamification and guidelines for serious games to attempt to create a serious game that teaches ultrasound interpretation. The motivation behind this research has been the need for surgeons and surgeons in training to be able to practice this skill outside the operation room. To be able to read ultrasound images, a lot of practice is required, and we hope that through *Master Surgeon* we will make this experience easier to obtain and available for a larger group of people.

The core of this thesis has been based on games, and we have tried to find answers to why we find them fun and how this fun can be transported and used in situations that are normally perceived as boring or difficult. A large part of the project has been concerned with discovering success factors for serious games and theories from game design that affects how engaging a gaming experience is.

Although there has been a great focus on collaborative exercises throughout this thesis, the limitations regarding time and experience with the selected framework, resulted in the final game ending up as a single player. The feedback from the evaluation indicated however that *Master Surgeon* would not necessarily benefit from adding such exercises. A reasons why the planned multiplayer aspect of the game did not appeal to participants of the evaluation could be that the suggested design is to weak to motivate collaboration. The focus on this feature is not strong enough in the game and it comes off as a second-rate and unnecessary addition. A second factor could be the unique characteristics of the user group. The participants of the evaluation are, with their high education and position, most likely achievement driven people that thrive on doing well and love the sensation of winning. This is easier to facilitate with a single player game where the player competes against themselves or the machine, while creating a multiplayer that allows for more than one winner can be challenge.

The feedback concerning the educational value of the game was overall positive. Participants enjoyed looking for tumor borders and put great effort into the accuracy of their drawings, which indicates that the game is successful at presenting the relevant image material and teaching interpreting ultrasound in a fun way.

Bibliography

[1] Holm S. *Medisinsk ultralydavbildning*. Fra Fysikkens Verden. 1. (1999). http://heim.ifi.uio.no/~sverre/papers/99_FFV.pdf. [Accessed 01.05.2015]

[2] Deterding, S., Dixon, D., Khaled, R., & Nacke, L. *From game design elements to gamefulness: defining gamification*. In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments. (2011).

[3] Laamarti F., Eid M., and Saddik A. *An Overview of Serious Games*. International Journal of Computer Games Technology. (2014).

[4] Zelkowitz, M.V., Wallace, D. *Experimental models for validating computer technology*. IEEE Comput. 31 (5), 23–31. (1998).

[5] The National Competence Center for Ultrasound and Image-Guided Therapy. http://www.usigt.no/index.php [Accessed 02.05.2015]

[6] Hornak J. P. *The Basics of MRI, Introduction, NMRI or MRI?* (1996). http://www.cis.rit.edu/htbooks/mri/inside.htm [Accessed 02.05.2015]

[7] Ebel, Klaus-Dietrich; Benz-Bohm, Gabriele. *Differential diagnosis in pediatric radiology*. p. 538. (2014).

[8] MRI safety, *Safety Information Article*. (2015). http://mrisafety.com/SafetyInfov.asp?SafetyInfoID=253 [Accessed 03.05.2015]

[9] Ingerid Reinertsen I., Lindseth F., Askeland C., Iversen D. H., Unsgård G. *Intra-operative correction of brain-shift.* Acta Neurochirurgica. (2014).

[10] Zichermann, G. and Cunningham, C. *Introduction* in *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly, Sebastopol, (2011).

[11] Landers C. *Casual Social Games as Serious Games: The Psychology of Gamification in Undergraduate Education and Employee Training* in *Serious Games and Edutainment Applications.* Springer London. (2011).

[12] Zichermann, G. and Cunningham, C. *Player Motivation* in *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly, Sebastopol, (2011).

[13] Lazzaro. N. *Why We Play Games: Four Keys to More Emotion Without Story*. In Proceedings of the International Game Developers Conference, San Francisco, (2005).

[14] Bartle R. *Hearts, Clubs, Diamonds, Spades: Players Who Suit Muds.* Journal of MUD Research, (1996).

[15] Nakamura J., Csikszentmihalyi M. *The concept of flow* in *Handbook of positive psychology*. Oxford University Press. (2005).

[16] Schaffer O. *Crafting Fun User Experiences - A Conversation with Owen Schaffer*. White Paper, Human Factors International. (2013).

[17] *Personality Types*. Gamification Wiki. https://badgeville.com/wiki/Personality_Types [Accessed 26.04.2015]

[18] Bartle R. *Players* in *Designing Virtual Worlds*. p.146. New Riders. (2003).

[19] Heeter C., Magerko B., Medler B., Fitzgerald J. *Game Design and the Challenge-Avoiding, Self-Validator Player Type*. International Journal of Gaming and Computer-Mediated Simulations. Volume 1, Issue 3. (2009)

[20] Zichermann, G. and Cunningham, C. *Game Mechanics: Designing for Engagement (Part I)* in *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly, Sebastopol, (2011).

[21] Zichermann, G. and Cunningham, C. *Game Mechanics: Designing for Engagement (Part II)* in *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly, Sebastopol, (2011).

[22] *Game Mechanics*. Gamification Wiki. https://badgeville.com/wiki/Game_Mechanics [Accessed 26.04.2015]

[23] *Qt Framework.* http://www.qt.io/qt-framework/ [Accessed 05.05.2015]

[24] Wilcox M. *Cross Platform Apps – Qt vs. HTML5.* http://www.developereconomics.com/cross-platform-apps-qt-vs-html5/ [Accessed 05.05.2015]

[25] *Data Storage*. Qt Documentation. http://doc.qt.io/qt-5/topics-data-storage.html [Accessed 05.05.2015]

[26] *Introduction to Enginio Data Storage*. Cloud Services https://developer.qtcloudservices.com/eds/introduction [Accessed 05.05.2015]

[27] Qt Quick Local Storage QML Types. Qt Documentation. http://doc.qt.io/qt-5/qtquick-localstorage-qmlmodule.html [Accessed 05.05.2015]

[28] *Files*. Enginio Data Storage. Cloud Services
https://developer.qtcloudservices.com/eds/key-concepts/files [Accessed 05.05.2015]

[29] *Users and User Groups*. Enginio Data Storage. Cloud Services
https://developer.qtcloudservices.com/eds/key-concepts/users-usergroups
[Accessed 05.05.2015]

[30] Aarabi P. 5 tips for creating great mobile app user interfaces. Venturebeat. (2013).
http://venturebeat.com/2013/04/08/5-tips-for-creating-great-mobile-app-user-interfaces/
[Accessed 10.05.2015]

[31] Ruiz J., Li Y., Lank E. *User-Defined Motion Gestures for Mobile Interaction*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.Vancouver, BC, Canada. (2011).

[32] *Farm Heroes Saga*. Google Play.
https://play.google.com/store/apps/details?id=com.king.farmheroessaga&hl=no
[Accessed 18.05.2015]

[33] *Brain Wars*. Google Play.
https://play.google.com/store/apps/details?id=jp.co.translimit.brainwars&hl=no
[Accessed 18.05.2015]

[34] *Draw Something Free*. Google Play.
https://play.google.com/store/apps/details?id=com.omgpop.dstfree&hl=no
[Accessed 18.05.2015]

[35] Khosravifard M. *MasterNeuroSurgeon: A mobile game/app for enhanced ultrasound interpretation.* Master of Science in Information System. NTNU. (2014).

[36] Kim C. *Changing Employee Behavior Without Force: Health and Wellness at NextJump*
http://www.slideshare.net/gzicherm/charlie-kim-final [Accessed 23.05.2015]
Presentation can be seen at https://www.youtube.com/watch?v=RBLmlGaYGHg

[37] Brooke, J. *SUS - A quick and dirty usability scale*. In P.W.Jordan, B. Thomas, B.A. Weerdmeester, and I.L. McClelland (Eds.) *Usability Evaluation in Industry* (189-194). London: Taylor and Francis. (1996).

[38] A. Bangor, P. Kortum and J. Miller, *Determining what individual SUS scores mean: Adding an adjective rating scale. Journal of Usability Studies,* 4 (3): 114-123, (2009).

Part V
Appendix

Appendix A: Acronyms

2D - Two dimensional
API - Application Programming Interface
GUI - Graphical User Interface
IDE - Integrated Development Environment
JSON - JavaScript Object Notation
MUD - Multi-User Dungeon
MRI - Magnetic Resonance Imaging
SUS - System Usability Scale
UI - User Interface
USIGT - The National Competence Center For Ultrasound and Image Guided Therapy

Appendix B: SUS Questionnaire

#	*Question*	***Strongly disagree*** ***1***	*2*	*3*	*4*	***Strongly agree*** ***5***
1	I think that I would like to use this system frequently.					
2	I found the system unnecessarily complex.					
3	I thought the system was easy to use.					
4	I think that I would need the support of a technical person to be able to use this system.					
5	I found the various functions in this system were well integrated.					
6	I thought there was too much inconsistency in this system.					
7	I would imagine that most people would learn to use this system very quickly.					
8	I found the system very cumbersome to use.					
9	I felt very confident using the system.					
10	I needed to learn a lot of things before I could get going with this system.					

Table B: SUS Questionnaire

Appendix C: Evaluation Raw Data

Participant	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*	*Q6*	*Q7*	*Q8*	*Q9*	*Q10*
1.1	3	3	4	4	3	3	4	1	3	3
1.2	4	2	4	1	3	3	5	2	4	1
1.3	4	1	5	1	4	1	5	1	3	1
1.4	4	1	4	2	5	1	5	1	4	1
1.5	5	1	5	1	5	4	5	1	5	1
2.1	3	3	3	1	3	2	3	2	3	1
2.2	4	2	4	1	4	2	4	2	4	2
2.3	4	1	4	1	4	2	4	1	3	2
2.4	4	1	4	2	3	1	5	1	3	1
2.5	4	2	5	2	5	2	5	1	4	1
2.6	4	1	4	1	4	2	5	1	4	1
2.7	5	1	4	1	4	1	4	1	5	1
2.7	4	1	5	1	5	1	5	1	5	1

Table C: SUS - Raw data