[[1]](#endnote-1)

FORSIDE FOR INNLEVERING

TILGJENGELIGHET

|  |  |  |  |
| --- | --- | --- | --- |
|  | Fri |  | Begrenset |

|  |  |  |
| --- | --- | --- |
| Emnekode og emnenavn: | Hovedprosjekt (PJ6000-13) | |
| Tittel norsk: | Hovedprosjekt | |
| Tittel engelsk (Hovedprosjekt): | Hovedprosjekt | |
| Eventuell oppdragsgiver: | Syncrotec [[2]](http://www.syncrotec.no/) | |
| Utleveringsdato: | 6.01.2014 | |
| Innleveringsdato: | 21.5.2014 | |
| Antall sider: |  | |
| Antall ord: |  | |
| This is the project report to group 10 at NITH. It contains detailed information about our current bachelor game project. The main focus of this document is to keep information about the game (rules, graphics and gameplay) and how we work on the project up to date as well as easily available. | | |
| Gruppenummer: | 10 | |
| Studentnavn: | Studentnummer: | Signatur: |
| Andreas Engebretsen | 700542 | 9ldGmIwqjIdGtRteFmGl_XsOeKbeWQGIEoyG80Yau57PIYHxWFJ4D5p3CIjUyVpdjDcFEI4BgbGscehgzr664GqGr22J5ANUoM8nXkwWzCY-f7QehQpg_QFTfhsP8w |
| Aleksander Hougen | 700633 |  |
| Geir Haugen | 700579 |  |
| Geir Inge Hillersøy | 700458 |  |
| Jonas Dyrkorn | 700116 |  |
| Studentens signatur er også en bekreftelse av at hun/han har gjort seg kjent med, og fulgt, NITHs retningslinjer for intellektuell redelighet (tilgjengelig på intranett). | | |

Table of content

[Hovedprosjekt (PJ6000-13) 1](#_Toc388281843)

[1.0 Introduction 3](#_Toc388281844)

[2.0 Contacting the company 4](#_Toc388281845)

[3.0 Company Description 4](#_Toc388281846)

[4.0 Vision 5](#_Toc388281847)

[4.1 Goals 6](#_Toc388281848)

[4.2 Concept 7](#_Toc388281849)

[4.3 Concept analysis 7](#_Toc388281850)

[5.0 Methods 8](#_Toc388281851)

[5.1 JIRA 10](#_Toc388281852)

[5.2 Research method 10](#_Toc388281853)

[5.2.0 Product reviews 11](#_Toc388281855)

[5.2.1 Photoshop 12](#_Toc388281856)

[5.2.2Paint.net 12](#_Toc388281857)

[5.2.3 Mindmap 12](#_Toc388281858)

[5.2.4 Unity 13](#_Toc388281859)

[5.2.5 GitHub 13](#_Toc388281860)

[5.2.6 Google Docs 13](#_Toc388281861)

[5.2.7Prime 31 14](#_Toc388281862)

[5.2.8 Parse 14](#_Toc388281863)

[5.2.9 Audacity 14](#_Toc388281864)

[5.3 Technical analysis 14](#_Toc388281865)

[6.0 Balancing 15](#_Toc388281866)

[6.1 Balancing graph explanation 17](#_Toc388281867)

[6.1 Challenge highlights 18](#_Toc388281868)

[6.2 Prototyping 19](#_Toc388281869)

[5.3 Prototypes 20](#_Toc388281870)

[6.4 Design choices 21](#_Toc388281871)

[6.5 Flowchart of the menu navigation 22](#_Toc388281872)

[6.6 Testing 23](#_Toc388281873)

[6.7 Testing Analysis 28](#_Toc388281874)

[7.0 Multiplayer logic 30](#_Toc388281875)

[8.0 Sound design 30](#_Toc388281876)

[9.0 Market Analysis 31](#_Toc388281877)

[9.1 Store guidelines 33](#_Toc388281878)

[10.0 Workflow analysis: 34](#_Toc388281879)

[11.0 Final product review 35](#_Toc388281880)

[12.0 Future development 35](#_Toc388281881)

[13.0 Conclusion 36](#_Toc388281882)

[14.0 Post Mortem 37](#_Toc388281883)

[15.0 Figure list 37](#_Toc388281884)

[16.0 Reference list 38](#_Toc388281885)

[17 Attachements 39](#_Toc388281886)

[17.1 Sprint Reviews 40](#_Toc388281887)

[17.2 Rules 42](#_Toc388281888)

[17.3 60 seconds of gameplay 43](#_Toc388281889)

[17.4 Drawings: 43](#_Toc388281890)

[17.5 Visual themes 50](#_Toc388281891)

# 

# 1.0 Introduction

Through NITH we have gotten the opportunity to develop a game for a company in Oslo by the name of Syncrotec. They wanted us to develop a release ready game for the mobile market. Developing this was a step by step process that contained frequent user tests and a high amount of testable prototypes.

Frequent tests position us to make the required changes efficiently as development proceeded.

We used the SCRUM methodology and tools such as Jira, Google drive and GitHub for planning, version control and documentation. We had meetings with both the external and the internal counselors when required, as well as giving them all access to relevant documentation for feedback. An important part of our design process was to not only create a game that could be monetized, but at the same time we spent time analyzing the current market.

We designed our game with the intent to improve on current trends for design and business models, implementing different ways of testing.

What we envisioned was a tower defense game with something we consider to be a new twist. The game supports multiplayer, and has several new gameplay elements mixed in as an attempt to reinvent the genre. This document contains test reports, analysis of individual components within our game as well as process methodology and testing procedures.

# 2.0 Contacting the company

The process of contacting Syncrotec started when one of our lecturers mentioned the company in class. We proceeded with sending a mail to Ståle Engen about setting up a meeting with them. In the meeting, they presented themselves as a company and what they could offer us as a project.

We also talked about how the process of developing the product for them and how the bachelor project works in general. We received an NDA that they wanted us to sign if we were to accept the project.

After this meeting we kept in touch through mail and kept each other updated on ideas as well as starting to plan the project. After accepting the project we had another meeting in their new offices where we talked more in detail about the project. We also discussed different ideas the group had for games. After a brainstorming session we eventually had a voting process where we eliminated ideas until we only had one left. Finally we decided to go for the tower defense game “Towers of Valhalla”.

# 

# 3.0 Company Description

Syncrotec is a company that focuses on back end solutions. They have connections within the industry and have worked with large companies such as Funcom. Their main focus is creating back end solutions and outsourcing the front end. They are currently working on a new online gaming platform called TapasPlay.

TapasPlay is a selection of smaller multiplayer games where people can bet money. They wanted to expand into mobile gaming. This is the reason they hired us to develop a game for them. The company did not give us any restrictions when it came to the description of the game, but offered us any help needed throughout the project.

# 4.0 Vision

The initial plan for the game was to implement it for the platform Tapas Playthat Syncrotec owns. This idea was abandoned in favor of a standalone game. The game was meant to be a source of income for Syncrotec, where the revenue comes from in game purchases and/or advertisement included in the game.

We intended to keep the game free and entirely playable without making a single purchase, but to make in game purchases enticing to the player by offering them an opportunity to ease the amount of time investment required. Therefore we looked at the more successful business models and tried to create a similar model tailored to our game.

When we developed this game for mobile devices we wanted the gameplay to be inherently simplistic, yet give the player complexity through choices and options in the Meta game.

We spent a lot of time analyzing the current trends in the mobile marked in order to have an idea of what was currently popular.

We planned on designing and creating all of the assets ourselves. We had several different types of tests to ensure that we were making progress throughout the project. In the early phases we used quantitative testing for a better overview of the genre. The later testing was qualitative testing for the specific elements of the game such as font, graphical details and sound.

In between the external testing we performed internal group testing within the group. This was important for the sake of keeping everyone up to date with the current version of the game.

### 4.1 Vision analysis

Our vision was based on the desires of the client as well as our market analysis. Considering the huge growth of today’s mobile game market, particularly ones that are responsive and easy to play, we felt that this was the most appropriate market to focus on.

It was important to be able to create our own assets so that we could shape the visual style of the game the way we wanted. Keeping in mind that mobile games are usually played in shorter sessions, we wanted to focus on making the game enjoyable when playing for limited stretches of time.

We designed the multiplayer to be turn based. We settled for this approach because it allows the players to play on their own time. It was important for us to keep the single player functioning the same way as the multiplayer component. The reasoning for this is that we wanted the player to be able to jump right into multiplayer without having to learn new things.

We designed the single player to function as a tutorial for the multiplayer component. The first few levels in this mode have some short tutorials in them in order to help the player grasp the game.

##### 4.1**.**0 Goals

Our goal was to design and develop a fun and innovative game experience where skill based gameplay is the main focus. We performed consistent testing and iterations to create as good a game as possible.

We intended to create our own assets using tools such as Photoshop and

Paint.Net for a more visually consistent art style. Another goal that we had made ourselves was to innovate within the genre parameters. This was an attempt to change the basic formula of the tower defense game type. Another goal for this project was to develop our game so that it could be played by two people competing against each other without having to do everything in real time, in other words a turn based multiplayer experience. This allowed the players to take their time planning their next move before ending a turn. A difficult yet important aspect was to keep the game balanced. This was an extended process throughout the entire development.

We were prepared to cut certain aspects of the game if necessary in order to finish the game. Some elements of the game were put on hold. We did this until we could find a way to implement them into the game and at the same time keep everything feeling balanced. By being realistic and analyzing the amount of time we have and how much time the different parts of the project will take, completing the game within the set goals is more realistic.

## 4.2 Concept

Our concept is a Tower defense game with base building and strategy elements. We wanted to take the “defending your base gameplay” to a new level. You start off with a main building (Town Hall) that represents your health. If the opposite player manages to destroy this building, you will lose the game.

The tower defense part of the game is composed of placing towers next to a road leading into your base and stopping the enemies from advancing too far. The city building part is simplistic as to not confuse the player. The player will have slots for buildings around his/her town hall that are used for placing buildings in the base. These buildings will have functions ranging from income upgrades to unlocking different units which the player can send out to the battlefield.

The player can customize his or her game strategy by personally choosing what buildings to invest in, each with its own benefits.

We introduced a competitive multiplayer system where instead of responding in real time, you will queue up your soldiers for attack, construct your buildings and place your towers before telling the game that you are ready by clicking the “End Turn”-button. When both players have completed their preparations, the game carries out the necessary actions, which can be viewed by either player at any time, before the next round begins. This type of asynchronous multiplayer mimics popular games like “Draw Something” and “Word Feud”.

 The first player to destroy the enemy Town Hall will win. Our game will be set in the Viking Age and the design and theme will reflect this in a multitude of ways, such as churches, Viking armor, towers and weapons. Even though we will follow this particular style we had the creative freedom required to add our own touch to old designs and themes. The reason for this is that we wanted the game to be thematically consistent, while also having an art style that evokes a casual and funny atmosphere.

## 4.3 Concept analysis

We chose to base our concept around Tower defense as well as base building because of the fact that these are two very popular types of gameplay mechanics in the different stores on mobile devices, but combining them is still a new genre that hasn’t been tapped into significantly. We intend to take the basic aspects from both styles of gameplay and merge them together into an innovative as well as entertaining experience.

By not having synchronous multiplayer we avoid a large number of complications related to latency that would otherwise severely inhibit the gameplay experience. Latency had been an issue in games that Codewise had worked on earlier, so this was an important consideration for us from the very beginning.

Looking at the mobile market we also decided on a heavily stylized approach to the art style in order to appeal to as large an audience as possible. This approach extends to GUI design as well. We wanted to avoid copying other art styles or GUI designs, and instead rather draw inspiration from various sources.

While the primary goal of the game was to provide a fully functional multiplayer experience, we also wanted to have a single player mode which lets the player familiarize him/herself with the game mechanics. In this mode the player will be put up against a relatively simple AI player, instead of another human being.

5.0 Methods

Throughout this project we have been using different methods to manage the project and to keep the workflow going. One of the things we have been relying on the most is the SCRUM method, we felt like this would be the most ideal way for us to manage the project. The SCRUM method allows us to know what is being done and at the same time it keeps people focused on the task at hand.

The reason for choosing SCRUM as a project management method was because we recognize it as the best method for our type of project.

When it comes to designing and implementing features in our game, we always focused on functionality first then design. By doing this we always knew that aspects of the game were testable before we designed it. This made it easier to design after function and made the design fit the overall style of the game better.

We created a way of working where we had one team focusing on the multiplayer portion of the game while the other part of the team focused on the single player. By doing this we managed to keep both modes of the game up to date and could test them both at the same time.

Figure 1 [[Scrum]](http://www.codeproject.com/Articles/704720/SCRUM-explained)

[[2]](#footnote-1)

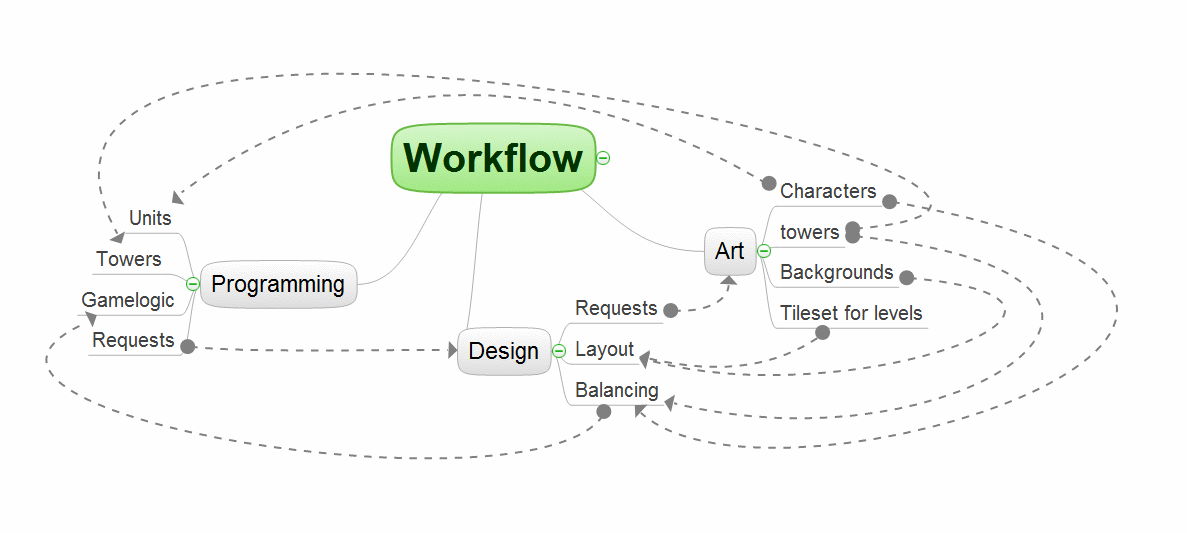


Figure 2 Mindmap

Workflow

This chart shows the relations in the group and where requests come from and where the assets end up. This method worked great for us as it streamlined the work flow and made it possible for everyone to work simultaneously. No part of the group had to wait on requests (as this chart infers), since we had a starting pool of task through our scrum board.

## 5.1 JIRA

As a supplement to fulfill SCRUM, we used a tool called JIRA. This tool helps the project team by fulfilling the SCRUM requirements.

We found JIRA to be a useful tool for managing a project using SCRUM. JIRA gives us, amongst other things, the opportunity to manage all the tasks in a project. Everyone included in the project may see which tasks that are needed to be done, in progress or tasks that are completed.

## 5.2 Research method

An additional method we considered using for the development process is the Waterfall Model. The first formal description of this method is often cited as a 1970 article by [Winston W. Royce](http://en.wikipedia.org/wiki/Winston_W._Royce) [[Article]](http://leadinganswers.typepad.com/leading_answers/files/original_waterfall_paper_winston_royce.pdf) Even though he did not use the term Waterfall in this article he did present it as a flawed , non-working model. This method is commonly used in the software development process in a way that makes the progress look like it is flowing down steadily.

The model has five “levels” starting with Requirements, design, implementation, verification and then maintenance. We chose to not use this method because we felt it would be difficult to finish a phase of the product’s lifecycle perfectly before moving to the next phases.

Another problem would be if the client decides to make big changes to the project, it would mean invalidation of a good amount of working hours.

We found this article while researching different methods of working and came to a conclusion that it would not be suitable for our project, but provides a good example of an alternative work method.



Figure 3 waterfall[[3]](#footnote-2)

### 5.2.0 Product reviews

This section has descriptive and detailed reviews of the different products that we used during the project. It is intended to be both positive and negative and will give our overall impression of the different products.

### 5.2.1 Photoshop[[4]](#footnote-3)

For this project we used several different tools to develop our game. The main tool we have been using for creating the main assets and maps is Photoshop. The reasoning for us using Photoshop is that it is one of the most powerful and best suited tools for drawing and editing our assets in detail. With the wide range of tools in Photoshop we had no limitations, but this also makes it quite complicated software to learn. At the start of the project we had a very shallow knowledge of Photoshop, but quickly learned the necessary skills to create what we wanted with it. In the beginning of the project we hadn't completely decided the style of the game. Rather than deciding the style in the starting phase, it evolved throughout the project. Due to not deciding on the style right away, we had to improve or remake many of the assets we had already made. But we did not see this as a challenge due to Photoshop’s flexibility.

### 5.2.2Paint.net[[5]](#footnote-4)

We have been using Paint.net as a way for group members who are not familiar with Photoshop to help produce game content. This has allowed us to develop more maps and assets at a quicker pace allowing us to develop more maps.

Even though Photoshop has been our main method of creating content, having Paint.net has been incredibly useful even though it has a lot more limitations.

An issue with using something that is more limited than Photoshop is the fact that we could only use it for the more simple assets such as shadows and slight modification of older assets. Overall it was a crucial addition when it came to the level design portion of the game because it allowed us to work faster when building the maps.

### 5.2.3 Mindmap[[6]](#footnote-5)

To structure our flowcharts we used a program called Mindmap. This tool is meant as a brainstorming tool, but has some nice features that work well in a flowchart.

The upside of this piece of software is its quick use and ability to clearly show relations, and has a nice visual style that is easy to comprehend. The downside is its intended use, it wants to create charts out of a single point and only point inward to it. This was not an issue the way we used it, as we had to break down our application into smaller parts for the flowcharts for better understanding.

### 5.2.4 Unity[[7]](#footnote-6)

When it came to actually developing the game we chose to use Unity as our framework and game engine. We chose Unity as our framework for developing the game mainly because of the ability to port the game to “all” platforms. Unity was a great tool for developing fast prototypes of ideas. It also gave us the ability to discard ideas that might not function without having wasted too much time on the development.

We were running Unity 4.3.2f1. The 4.0 update opened up a new way of doing 2d games. This was crucial when it came to developing quick prototypes.

### 5.2.5 Visual Studio

We chose not to use Unity’s built in programming tool Monodevelop for what we consider to be a better tool, Visual Studio. Monodevelop, while easy to use, lacks a lot of customization that Visual studio offers, while also being much more prone to crashing, stuttering and similar problems.

### 5.2.6 GitHub[[8]](#footnote-7)

For our main storage we used GitHub. This was a great tool when it comes to version control and as a backup solution. Although we were not able to use this ability when working with Unity. The reason for this is that GitHub does not work very well with some of the binary files that Unity produces. Overall we have been fairly happy with GitHub. Except for us finding it not to be very stable when the repositories are growing in size. To solve the problem with space, we managed to keep only the two last versions in the repository. When a new version was added, we archived the older version.

### 5.2.7 Google Docs[[9]](#footnote-8)

For documentation, our main tool has been Google Docs. This excellent tool provided by Google gave us the opportunity for us to edit the documents together at the same time. The documents are in the cloud, and therefore backed up and available at all times.

The one issue we seemed to have with this product is the fact that when working with larger documents, containing several images and precise formatting the product would often remove images as well as ruin formatting forcing us to redo it. We decided to abandon Google Docs towards the end of the development and focused more on using Microsoft Word and LibreOffice writer. We could then use Github for backup.

### 5.2.8 Prime 31[[10]](#footnote-9)

From day one, it was always part of our vision to have Facebook support integrated with the game, in order to make it as easy as possible for the user to register, authenticate and connect with his or her friends. First we looked at Facebook’s own API, but quickly realized that this was incredibly unwieldy and unnecessarily difficult to apply to our game. This is when we discovered Prime 31’s “Social Networking Plugin”. This plugin let us write Facebook code that would function on both Android and iOS, while also providing a slew of practical and easy to use methods that let us easily authenticate users through Facebook, get their profile image and friends lists.

### 5.2.9 Parse[[11]](#footnote-10)

The next piece of the multiplayer puzzle was a database system. We played around with the idea of building one ourselves, but quickly realized that this would be much too time consuming. After doing some research we came across Parse. Parse offers a range of services connected to database and online functionality. First and foremost it let us easily convert Facebook users into users on our own system, tying them together. Then it had a Unity API that made it exceedingly easy to query the database from the scripts we had created.

On top of all this, Parse will be instrumental in further development, as it offers analytics, a database dashboard, and the ability to send push notifications to all users.

### 5.2.10 Audacity[[12]](#footnote-11)

For our sound design we have been using Audacity. This is a free recording program with all of the basic tools needed for editing sound in a simple yet effective way. It is incredibly easy to use, even for those who have little to no experience with such programs. This allowed us to work efficiently without having an extensive learning process.

There was no point in spending money on an expensive sound recording program when we decided on making the sounds for the game simple. Overall Audacity is a great program for simple sounds, but if we were to have more flexibility we would look into better software with more effects and editing choices.

## 5.3 Technical analysis

This chapter describes technical information about the platforms used for developing as well as the reasoning behind the different systems.

# LEGG INN iOS HER ANDREAS!!!!!!!!!!!!!!!

Android

Android has a maximum of 48 MB of memory available to an application. If we were to use more memory than this we would risk our application to be terminated in order to free more memory.

There are a lot of different phone producers that use the Android OS and all of them have the opportunity to use custom resolutions for their phones. The phones we used for our testing and developing were:

* Samsung Galaxy s4:1920 x 1080 at 5 inches
* Samsung Galaxy s3: 1280 x 720 at 4 inches
* Sony Experia Z1 Compact: 1280 x 720 at 4.3 inches

These phones were our main testing phones because of them all having different technical specifications such as ram, processor and screen size. By adapting our game for the different phones and specifications we were able to design something that would work on a lot of different phones.

We also performed testing on tablets so that we could be able to see if our game would scale well with the higher resolutions.

* Samsung Galaxy Note 10.1 2014 edition 2560 x 1600 at 10.1 inches
* Google Nexus 7 2012 1280 x 720 at 7 inches

The Samsung Note was our main tablet for developing as it allowed us to see how our game would function at a very high resolution with the highest specifications available for tablets

We spent time testing on the Nexus seeing as it was an older tablet with a lower resolution and weaker specifications. This gave us a clear distinction between the two different tablet generations.

Next to the mobile development we also spent some time testing the game in a web build through the Unity Web Player, in order to keep that avenue of release open, as well as making sure our game was as ubiquitous as possible.

Our assets are all hand drawn in 2d which means that we can run them in a very high resolution without the game having frame rate issues even when there are many elements occupying the screen at once.

Even on modern lower end phones frame rate was never an issue unless we manipulated the game and placed more units than possible. We also remove units when they hit a certain point on the map to make sure that their numbers will not increase and stack up over time.

6.0 Balancing

When it came to balancing, there were several things that we had to do. The towers needed to be balanced against the waves of enemies and to be correctly priced when they are bought or upgraded, and how much money the player will earn per round. The player must not feel cheated by the game. The balancing itself will start from the first external testing.

The balancing itself is something that was an ongoing process throughout the entire project. And finding a way for the spy to function in a way that does not add a dominant strategy was be a difficult task. We were not entirely sure on how to design the spy. We had to decide if this is to be a killable unit or a type of active ability. We were working hard on removing any kind of ability for the player to manipulate stats in order to dominate a game.

Making the game balanced became a bigger hurdle than first thought. It started to add up in complexity fast and it could not compare unit to unit or tower to tower, but we had to make groups. We did this by making groups of elements work by the rock-paper-scissors idea and making these groups balanced against each other.

The grouping we did was two units and a tower; this means that one tower can specialize in stopping two units. This made it easier for us to balance as we did not need to focus on how it does against the rest of the units. Since we have three towers these towers can then be balanced against each other in a rock-paper-scissors scheme, and since the units are grouped to the towers it will somewhat balance out. The new step is the upgrade paths, and as long the unit’s stats increase is not too fast to overpower the towers.

A basic unit has 100 health points and a basic tower is able to hit it three times as it passes by in a straight line, killing it on the third hit.

This is a flow chart covering how the balancing of the game is done. It is based on the circle diagram below, but written in a more readable and easy to understand way.

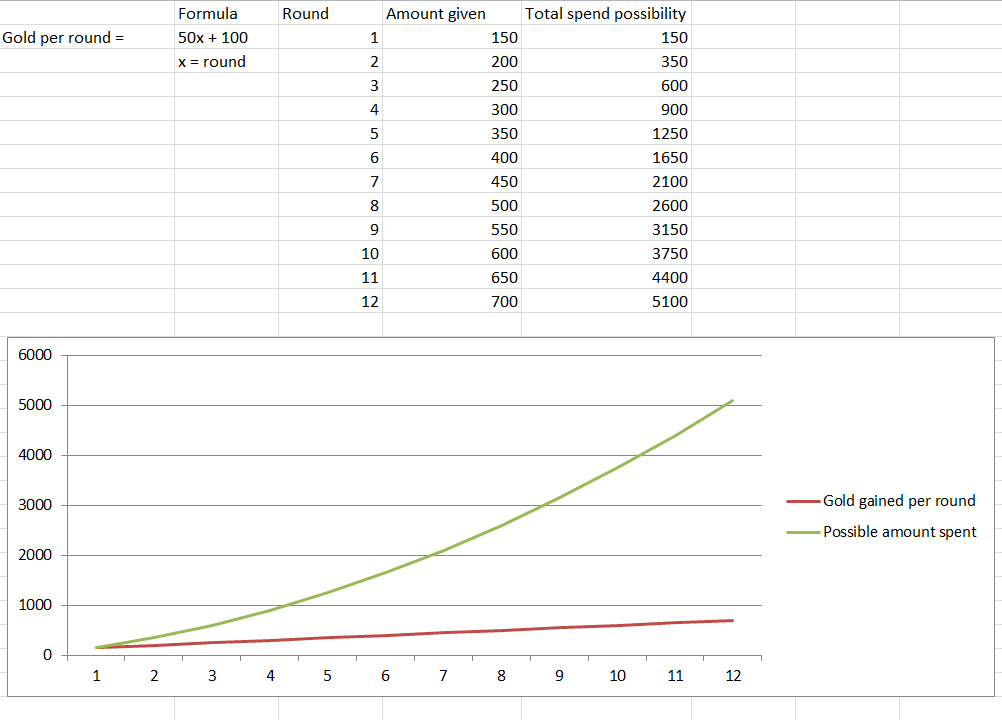


Figure 4 Gold balancing

## 6.1 Balancing graph explanation

This is the logic behind the gold we give the player.

As demonstrated the possible amount the player can use increases fast, this is planned as we want the player to have a small start and have many possible responses and tactics to use as the game goes on. Since we have done the math behind it, it was simpler to balance the game as we knew the restrictions we had put on the player.

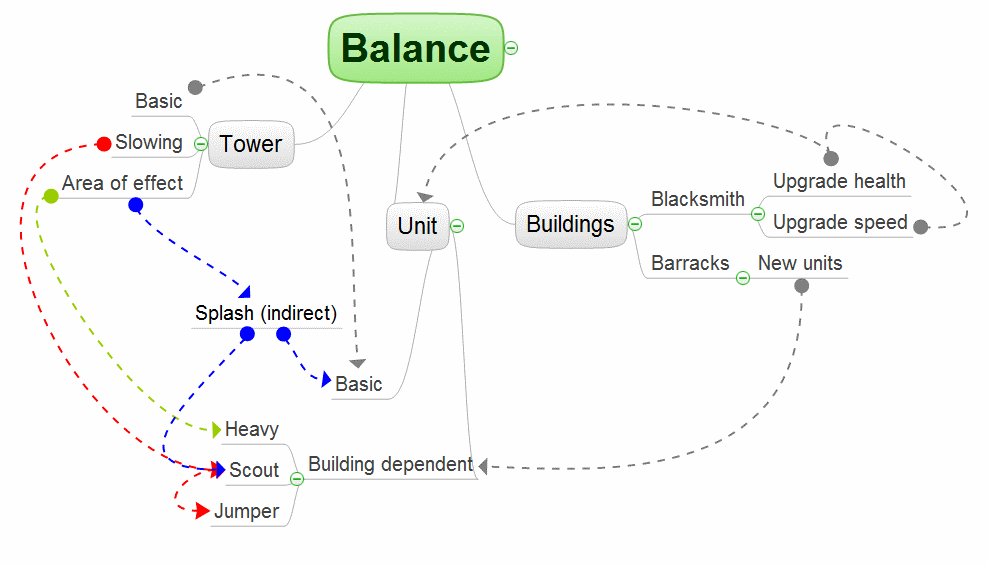
Figure 5 Balancing.



Figure 6 Balance loop

This flowchart shows the relationship between the elements in the game while a user plays. The colored arrows on the left side show what tower is effective against which type of unit. The right side shows how buildings have an effect on the units. This flowchart also shows that there are two directions to impact units.

The base idea is to have basic unit and tower to start of the game, and then make advancements as the game moves on. Since we have three more units that have a radical difference to the basic unit we wanted the other towers to be answers to the new units. As the towers then are effective against the new units there is also a way to upgrade the units, this makes the game into an arms race and gives the users more depth as to how to play the game.

## 6.1 Challenge highlights

An important aspect of the design was to have the player feel like his choices matter when it comes to building and have an effect on the gameplay itself. Not forcing the player to follow a certain strategy and allowing them to create own strategies. The challenge in this is that the player will have to figure out what strategy works the best for him through trial and error. At the same time we wanted to make any strategy viable when applied correctly.

It was important for the design of the game not to force the player down a certain path, but instead the player tries to forge a path of his own.

The city building element of the game allows the player to choose how he wants his towers and units to work. The way the player decides to build will eventually reflect back on the results from the round.

We have 3 main units and 1 specialty unit that the player will have to defeat. The challenge from this will be that some tower will be efficient against certain units while less effective against others. Another aspect that will challenge the player is that he has to defend against several different paths on some levels. The player will have to place his towers in a way so that he can be somewhat effective on both paths. By focusing on only one path the player will let some enemies slip through.

The units have certain attributes that make them unique. The basic unit is easily defeated by the basic towers. The jumper unit needs a special tower that can bring him down from the jump. In this lies the challenge considering that the anti-jumper tower is only effective when it comes to bringing down the jumper.

The player has to balance how he builds the towers so that he can find a balance between damaging and slowing down enemies. The player will also find a way to balance how he wants to upgrade his towers.

When it comes to the multiplayer component to the game the challenge will have similarities, but will also require some strategic thinking. The main difference between multiplayer and the single player component is that the other player chooses what enemy types you will face. The challenge here will be to try to find a balance between defending against the enemy’s troops as well as sending your own special troops.

Another important aspect is to try to find your opponent’s tactic when it comes to building troops and then create your own tactic to overcome him. The players will have to try to get as much done as possible before ending a round to maximize his defense as well as the offense.

Another challenge was adapting to the different kind of players that you will face. Some will play more aggressively and spend their money on units while other may be more defensive and spend most of their money on developing buildings and towers. Finding strategies against all the different types of opponents will be the key to winning.

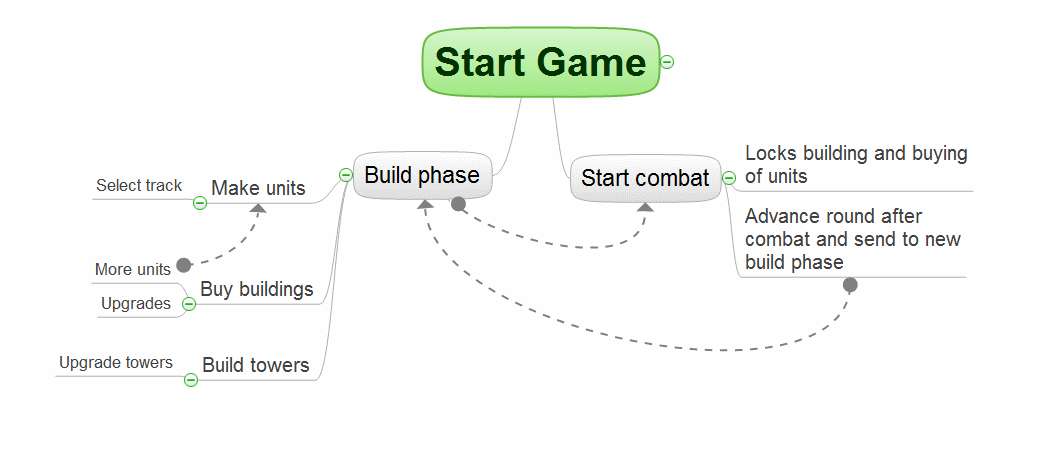


Figure 7: Game loop

This diagram describes how the in game actions work. The game has two phases, the build phase and the combat phase. These phases toggle and one locks out the other. We chose this setup to make the multiplayer easier to develop.

We could have the multiplayer in real-time, but this would make the single- and multiplayer work differently from each other, as well as bringing with it a plethora of issues regarding latency and synchronization. We wanted the game to function the same way in both formats so the user only had to learn one setup for the game.

The biggest difference from single player to multiplayer is that the game doesn’t start before both have ended their build phase and the application has synchronized the game state.

## 6.2 Prototyping

In the first two weeks of developing an idea for the game, we went through several concepts. We also had several meeting between the group and the client to discuss what we wanted and what they wanted.

We made prototypes for the concepts that we liked the best. While looking at the different prototypes and the concepts, we chose one concept with the consent of Ståle and Eirik.

We made complete prototypes of two of the concepts. One of the games we made a prototype for involved you shooting a mole down into the ground. While the mole digs its way downwards, you have to steer it to collect treasures, avoid obstacles and eventually reach the goal at the bottom.

We also created a detailed prototype for a game we called “Brong” which is a mix between Pong and Breakout for two or more players. We left this concept because real-time gameplay between players is not possible in the way we designed the concept.

After going through five prototypes we had a meeting at the end of the week where we voted for which game we were going to develop. We settled down on creating a tower defense game with a new twist. We could by doing this get around the obvious lag problem.

## 6.3 Prototypes

Our first Tower Defense prototype was a simple attempt to get the basic functionality going. We wanted to see how difficult it would be for our programmers to grasp Unity and its features. After a short learning period we had a functional prototype with basic graphics and basic functionality. This gave us a chance to see what would possibly become a challenge later on in the project.

What the group felt was a difficult part of the project was the network coding because of its complexity with Unity. At the same time we felt that AI could have taken time, but we had taken these things into consideration and planned extra time when it came to developing them.

We tested the first prototype internally on a phone in order to see if scale and animation was functioning as intended. We had to tweak the path finding somewhat, but overall the test was a success and everything felt visible and simple to understand. When it came to our externally tested prototypes we went for a more complete design. We designed a map up to what we consider production quality to avoid having the tester focus on the look of the game over how it played.

We decided not to add all the gameplay elements in this prototype and instead focus on the basic tower building part of the game. Things such as sending units to the enemy base and city building were iterated for later prototype testing. This allowed us to focus our tests at certain aspects of our game at one time instead of having several half-finished parts of the game at once. This also allowed us to do internal testing in more detail when it came to the new gameplay elements before we did the external testing.

Our first main test consisted of changes that we made based on the feedback we got on our Google Docs tower defense based questionnaire. The changes were subtle, but important as it gave us an overview of how a tester felt about our current build.

The build for sprint three was a lot closer to what the final build were to look like. It had all of the basic gameplay elements completed and was played on a phone. This prototype gave us an even clearer goal when we started working on the next iteration of our game.

The prototypes for sprint four and onwards were focused on gameplay over design, considering that we at this point had the design completed. We had several quantitative tests where we gave the testers tasks that were related to playing a single round within the game. We chose to do tests like this to see if the player understood the rules of the game and to see if the gameplay itself would be clear without help.

After these tests it was clear that we needed to create a tutorial level with different text boxes explaining how to do the different aspects in a round. After adding the tutorial level, the testers showed a better understanding of the game and it helped us in developing the final version of the game.

## 6.4 Design choices

When designing a game like this choosing the right colors and fonts is an important task. There are certain areas of the menus and in-game that we want the players to notice naturally without having to guide the player. By doing a detailed marked analysis we chose a design that we feel is a combination of current trends as well as choices based on different design articles.

UX design refined by Ritch[[13]](#footnote-12) Macefield[[14]](#footnote-13) gave us a way to approach design with the intention of making it more user friendly. It was very useful for finding out how to balance the different kinds of design methods. This gave us a way to balance usability with information and design.

Using these methods were a way for us to learn about different ways to think as designers. It gave each and every member of the group a way to affect the design of the different buttons, colors and fonts that we used. The end product is a representation of the combined ideas of the group that has been planned and tested throughout the entire process.

6.5 Flowchart of the menu navigation.

This is a representation of how you would navigate through the menus in the game. We based our early navigation design on the “Three Click Rule”, namely that a user should not have to go through more than three clicks in order to access anything. As the process went on, we discovered through some research that studies have shown[[15]](#footnote-14) that this does not necessarily result in a satisfied user. Nevertheless, we think that when applied correctly, the Three Click Rule can be useful in making sure that information and actions are easily available, even if it is a potentially misguiding rule.

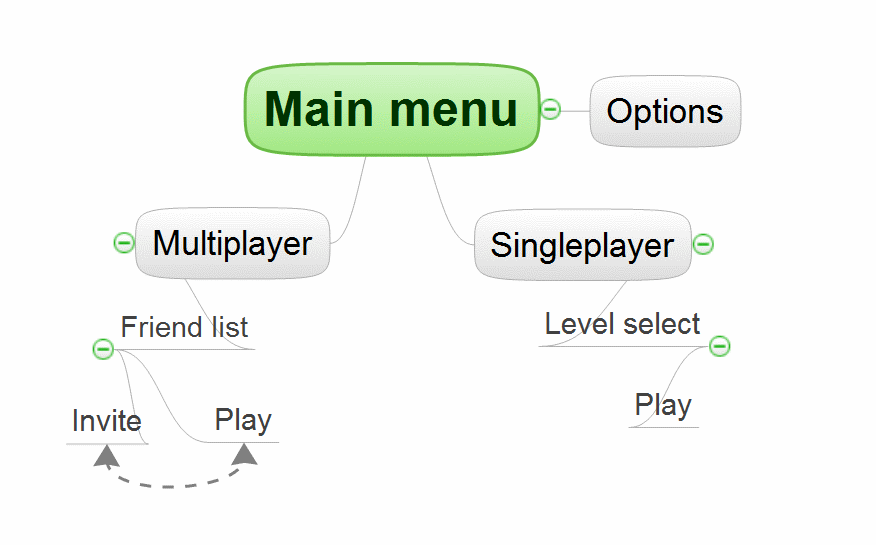


Figure 8: Menu navigation.

## 6.6 Testing

We performed several different tests throughout the project to test the gameplay and design of the game as well as getting feedback from a wide range of potential users. Our initial test mostly focused on basic gameplay and graphics. We found that it is important to start off with this kind of test to see if the player found the game fun and that it was not confusing or difficult to understand.

We designed and created as close to a production quality map for the first proper external testing to avoid having too much feedback about it not looking finished. At the initial testing only the basic gameplay functions were present so that we could focus on testing new things in the later iterations.

We detailed questions in a questionnaire for the testers to fill out where the questions were focused on how the player felt that the gameplay was moving forward and if he or she picked up some of our visual cues. We also did tests on both mobile and a pc, but we felt that if we only could find time to do the initial test on a computer it would have sufficed for this iteration.

When it came to selecting testers for this prototype we tried to get a variation of people. Ranging from those who play these kinds of games on a daily basis to people who have little to no experience playing games. We intended on creating a game that anyone can play without having to read through a long tutorial.

The testing was performed with one of the developers sitting next to the tester, observing as well as giving aid if the player gets stuck or has a specific question. During the testing the developer took notes to document certain behaviors from the tester.

After the tests were completed we gathered the data and analyzed it, looking at what the player liked, disliked and wanted different. We then re-iterate the game and performed changes based on the feedback.

As the testing progressed we started testing on mobile devices to see if our game’s assets are correctly sized and if text was readable on a wide array of devices.

We completed some design tests where we showed of different iterations of our concept art to receive feedback on what suits the game the most.

After a session of internal testing we discovered that we could lower the resolution of most of the assets. The way they looked with the current resolution adds to the overall size of the game. When playing the game on a mobile device you do not need as high resolution as if you were on a pc. The resolution was set to a point where you could zoom all the way into the map and all of the characters and towers were sharp and clean. By changing the resolution we conserved significant amounts of space for a mobile device.

We considered letting the player choose his own resolution when playing on a pc. Our main focus is to make the game functional and fun on mobile devices before moving to the pc. The reason for this is that we felt that it was smart to start out on the most basic platform and then port it over to something with more input options later. After the questionnaire part we had a long list over the most common phones. This gave us a clear idea over what devices we should optimize our game for.

We took some time to do some interviews to get an overall opinion on the tower defense genre as well as some useful feedback on the art of our game. We discovered through this process what people are looking for in these games and what they feel is less important. The biggest change we had to do based on the feedback of these interviews was to upscale the various units, towers and buildings in the game to a larger size, making them easier to see and interact with.

Most of the people we interviewed did not recognize the Vikings and thought they were animals or goblins. This was fixed by increasing the general size of the units. We did the same with buildings and the colored flags representing the player and the enemy. All of these changes made it easier for the player to know where to build and which tower he can build his towers on.

Figure 8, 9, 10, 11, 12, and 13 testing charts.

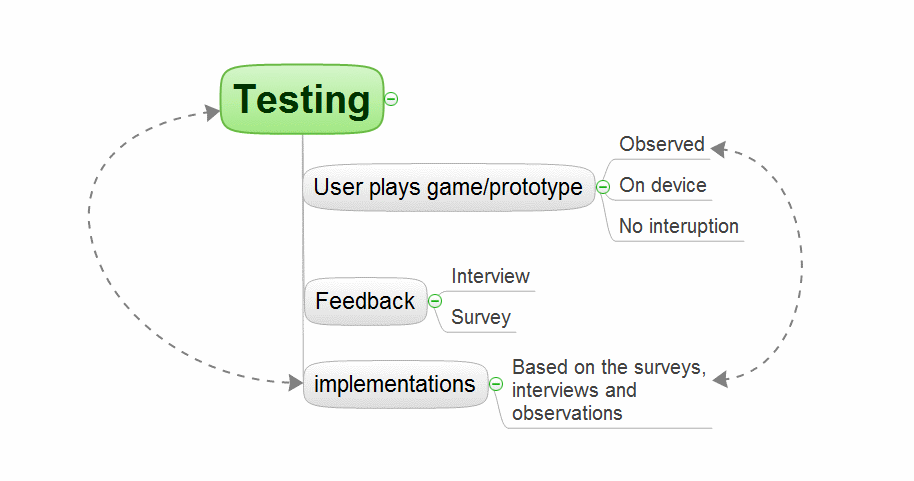


Figure 14 Testing Flowchart.

This flowchart represents the testing process from testing to implementation and how we received feedback. We started out with letting the player play our prototype on a mobile device. We observed without disrupting the players or helping them.

We then gave them a Survey for the earlier prototypes with questions about what they just played in order to give us an overview of their experience. In the later tests we sat down with the players and had more detailed interviews. We chose to do the interviews on the later tests so that we could get more detailed answers about certain aspects of the game.

After the process was completed we analyzed the results and implemented what seemed to be the most requested changes.

## 6.7 Testing Analysis

We decided to focus our first round of testing on simply showing our prototype to people and then have them answer a questionnaire. The questions were Tower Defense related and served to give us an idea of what phones people use as well as what games they prefer to play. The answers were quite useful and gave us information that we used during development of the game. Some of the answers were “funny”, but we knew that some would not take it seriously and because of this we tested some extra so that the joke answers would be phased out by the amount of answers.

An important question was about the mobile devices that people use. We found this to be important for later external testing, we used this information to find out what types of phones people had so that that we could work on making the game as compatible as possible.

Most of our testers were IT students, but we also have some students within other studies as well as people who are done with their studies and are now working. There was also a good mix between people who play games on a daily basis as well as people with little to no experience when it comes to games. This gave us a huge variety of answers.

Overall most of our testers had modern phones, but we still catered to the weakest phone on the list so that the game would be functional for most people. When it came to time spent on games such as the game we are creating, most of the testers said that they would spend up to two hours in one sitting. The average playing time for most mobile users is known to be from one to ten minutes in one sitting.

What seemed to be important for most of the testers was fast fluid gameplay with little to no interruptions during gameplay.

Considering sounds in mobile games, this seemed to be little to not at all important to some of the testers. This was something we knew as most people tend to play with their phones on silent, or playing external music while in the game. Even though sound may seem to be unimportant for most people, we decided not to neglect the sound environment for our game The sound itself was not our main focus until much later in the process, but we tried to make it sound good enough to make people actually turn their phones volume up to listen.

A large portion of the users said that the graphical style of the game was crucial to them actually playing the game. The style that most people seemed to like was a cartoony look as well as a clear distinction of what all of the different objects within the game are supposed to resemble. Micro transactions and ads are not popular, but some agreed that ads are fine as long as they are not shown constantly. Therefore, we looked into creating ways for micro transaction to be more cosmetic effects and not gameplay changing.

When it comes to historical settings it seems to be very varied what people wanted. Some said Vikings while other want medieval. Overall the difference between these two when looked at from above is minimal and we feel we can make both sides happy with our design.

One of the testers wanted an in progress save function so he can continue his single player game from the last played point. We find this to be an interesting idea and we tried to find a way to incorporate it into our game, but eventually it was cut. We did some more in depth testing to see what people feel about our graphical style as well as the pace of our game as often as possible to be able to modify it accordingly. A reason we decided to do the questions before the testing of the game itself was so that we could do some modification of our game based on the answers.

# 7.0 Multiplayer logic

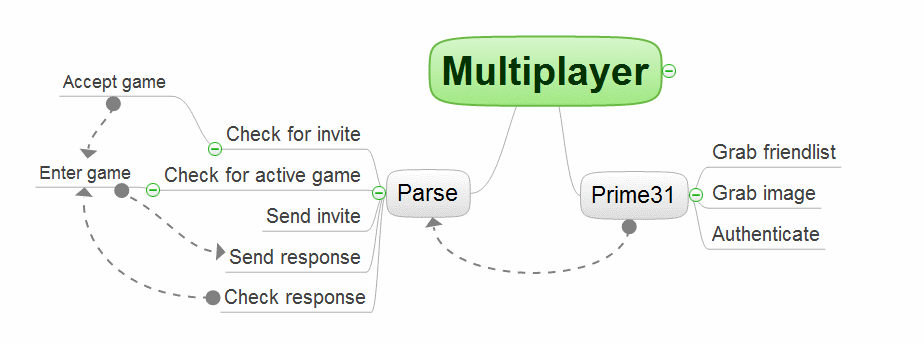


Figure 15 Multiplayer flowchart

# In order to limit the workload required for a functional multiplayer mode, we decided to use two extremely useful plugins for Unity to handle Facebook Integration (Prime31) and Database Management/Access (Parse). See section 5.2.7 and 5.2.8 for more details on the plugins themselves.

# As the user selects Multiplayer from the main menu, the app will automatically ask for Facebook permissions (if the user has the Facebook app installed) or prompt the user to log in to Facebook via a web-view.

# Once the user has been authenticated by Facebook, Parse creates a user (for first time users) which automatically links to the users Facebook ID. Then the app collects the users friends list and cross references it with the user data on our Parse server, and then displays all friends who also have the app installed and have used multiplayer.

# With the cross referenced friends list collected and profile images displayed, the user can choose a friend and invite them to a game (provided a game is not already in progress with said friend already), the app will then change the invite button to a "Pending..." label, and await acceptance of the invite from the second player. Once this has happened, the label will update to a "Play" button once the player either taps refresh or taps the second players profile image again.

# Once in the game, the player can recruit his units and build towers/buildings, before tapping the "End Turn" button in the bottom left corner. This will upload the actions the player has taken (units recruited, upgrades bought, towers built etc.) to our Parse server. Once both players have done so, the round will play itself out using the information uploaded to the Parse database from each user. Once all the units have run their course, the game either ends (if someone won) or progresses to the next round.

# 8.0 Sound design

It was important to us when developing a game that the sound was simple yet unique. We decided early on that we were going to create the sound ourselves using Audacity. When it comes to the sound of the units we created all of the sounds using our own voices. Through our testing process we realized that people don’t find sound that important for mobile games.

Based on this information we decided not to spend too much time developing the sound and instead try to keep it simple. We chose not to have any music while a round is playing so that the player would not be distracted from the already hectic game. A problem we discovered quickly when it comes to developing sounds like this is that even though we were using a high quality microphone, there were still sounds being picked up from any source around the room and outside the building. At first we found this to be a big issue considering that it rang through the audio takes, but because of Audacity’s ability to remove unwanted noise we were able to fix this.

9.0 Market Analysis

Looking at the current market today shows that within the tower defense category the most popular games are built up with colorful graphics and usually cartoony characters to appeal to a broader audience. The games themselves are built up to be playable with one finger in order to keep the interaction with the game as simple as possible.

We decided to adopt the cartoony and stylized type of design to appeal to a broader age group as well as both genders. We adapted the trend of using bright and vibrant colors in the game. This gives of a positive type of feedback to keep the player satisfied about the things he or she achieves in the game. We chose the cartoony graphical style because of it being the most popular style when it comes to games on mobile platforms. The cartoony style is also more appealing to a broader audience. Making the “violence” less violent by removing blood and anything that may decrease our user base was an important task. Another important part of designing the game is to keep the design modern and clean to give the user a more comfortable experience using the game.

By analyzing what is currently popular and used by bigger developers we will adapt our game to follow a similar style. Looking at these different developers we have noticed that the current trend is clean non sharp colors as well as simplicity.

Designing our game this way will allow users to quickly adapt and feel at home playing our game. We let our design be partially influenced by what is arguably the most popular mobile game within our genre currently, namely “Kingdom Rush”. We spent quite some time playing and analyzing the game and its rules in order to see what we felt made that game so much fun, but at the same time how we could improve on their design and gameplay choices.

Looking at different business models was also an extremely important part of our project because of the amount of different ways to monetize on a game. While the intent is to make money of our game we feel that it is important to not make the user feel like he or she has been cheated because that while turn the user base against us.

Our business model will be a version of the popular “freemium” model. This means that while the game itself will be free to download we will give the players a chance to buy simple upgrades or cosmetic skins to change the look of your game. By having a business model where the player can play through the game without feeling that he has to purchase items to complete the game is important to keep the player coming back for more. Looking at the current most played game on mobile platforms “Candy crush”.

The game itself is free and it is difficult, yet possible to play through without having to purchase any items, but in between levels you are given that option. It also does very well when it comes to integrating your friends into the game by Facebook sharing and the possibility to give friends more “life”. Their way of making it simple and quick to purchase help for the game at any time without making the player feel like he is being forced to purchase anything.

By doing this they are building up trust and at the same time they are making their users feel comfortable with using their product. Looking at a game that is quickly becoming the biggest mobile game of 2014 “Flappy bird “. The game is completely free and has no monetizing except for simple ads in the game. The game itself was released 2013, but is just now becoming known. The sheer popularity of the game has almost made sure that their next game is going to be noticed, so that they could in theory monetize well on their next game based on their Flappy Bird’s success.

It is incredibly important when you design a business model not only to look at what we consider good business models like the ones from Candy Crush or Angry birds, but at the same time look at companies that have notoriously unfair models. An example of this is the new Dungeon keeper game for the mobile platforms. The game is running what many consider to be a terrible freemium model where the game is constantly pushing its model to make you buy “gems” that will make everything build faster. This is a game that cannot be played efficiently without buying these gems because of the time it takes to do anything in the game. Some things take up to 24 hours to build unless you spend 8$ for the gems needed to do it instantly.

It is important that we analyze these kinds of models so that we can avoid making mistakes where our user base feels cheated by our model. If we can have a good model in between the terrible ones we will gain the trust from a user base that has grown tired of these kinds of business models. When it comes to tempting users into trying our game an important aspect is the icon that they will see when going to the different stores. Looking at current popular games there are certain trends that are being repeated.

The use of certain colors and fonts make the different logos seem more appealing to the eyes and we wanted to implement this in our game. When it comes to finding a name for our game the importance of something that people can instantly remember and the name itself should be connected to the gameplay. By doing this we let people know what our game is subconsciously. Attracting people’s interest using psychology is a common practice and it is important to use it subtly so as to not make the users feel cheated.

Another aspect that is important when it comes to making the players enjoy and come back for more is the sound aspect of such a game. Looking over games like “Candy Crush”, “Flappy Bird” and “Angry Birds” they all seem to follow the same theme. Even though their sounds a simple and minimalistic they are never sharp or unpleasant to listen to. They are very subtle, but at the same time they enhance the game experience by giving the player a good amount of positive feedback. Even though most players won’t think about the sound it does affect how long they will play the game without becoming tired.

## 9.1 Store guidelines

An important part of publishing a game on the different platforms currently available is to make sure that we don't break any of the rules set by the respective companies. We analyzed the App store, Windows store and the Google play store.

Even though the stores have most of the same guidelines [[18 app store][[16]](#footnote-15)](https://developer.apple.com/app-store/review/)

[[19 Google play][[17]](#footnote-16)](http://developer.android.com/distribute/googleplay/promote/brand.html)[20 Windows store][[18]](#footnote-17) there are some subtle differences. Because of the fact that our game is free and ads are important for the monetization of the game we need to be careful not to break their guidelines for ads or certain types of content.

Because of these guidelines we have decided not to have certain elements in our game.

Blood and any kind of violence is removed and because of apples guidelines. When it comes to simple to use GUI we created something innovative, colorful and fast to navigate. The guidelines also specify how we can place ads. It is important to not have ads in locations where they could be accidentally clicked by a player. An option to this is to only show ads when the player is in the menus. We were careful when it came to having shirtless Vikings considering that both stores are relatively strict when it comes to having too much skin showing. In the end we decided to keep one of the Vikings shirtless. The overhead view of the game makes it harder to see anything that might offend a player.

Overall there should not be too many issues with our game when it comes to the different policies, but we were prepared for the different things we might have to change.

# 10.0 Workflow analysis:

When designing a game for certain markets it is important to plan and detail a detailed risk analysis in case some things go wrong. Our plan was detailed to a point where we felt that we were prepared for things that could go wrong. The main problem we seemed to have, was absentees due to disease.

Health induced absence is a common problem in these types of projects and if not planned correctly it could halt the project. We had certain weeks were several people had to stay at home because of fever, but because of our risk plan we were able to keep working without too many problems.

Another common problem we faced was hardware failure. Even though all of our computers are modern we still faced problems with overheating and blue screen errors. One of these crashes was so severe that one of the laptops was crippled for two days. We were prepared and had backup tasks set for the person with the problems.

When it comes to the software we have been using it has for the most part been working well. We have had some issues when it comes to Unity crashing. The problem with Unity is that when working on larger projects with a lot of assets it may become somewhat unstable. This will cause it to crash if too many things are changed at once. We made sure to always have backups of our projects so that it would stop the work being done.

Most of our technical problems seem to come from using Github. Even though it was a useful program for saving files it is very unstable and we had to recreate the repository several times. The problem seems to lie in the fact that it becomes unstable when we had several build files in the same folder.

An important aspect of such a project is communication. Without having several different ways of reaching each other our project would have been halted. With such tools as Skype and Facebook we were able to have meetings and have an overview when it comes to what people were working on. Skype allowed us to keep the Syncrotec updated on our progress with frequent video meetings.

What took the most time seems to be the learning process of using a new tool such as Unity Even though we had skilled programmers it took them some time to grasp the ins and outs of Unity. This evened out during the third sprint and we managed to make back lost time.

Even with certain hiccups the workflow has been steady and consistent throughout the entire project.

# 11.0 Final product review

When we started the development, we had set ourselves some goals that we wanted to achieve. We planned and designed the game with the knowledge that some of the content would have to be cut due to time restraints.

The final product is close to the original idea that we started out with and contains most of the content that was originally planned. There were a few cuts due to time and balancing issues.

The biggest thing that was eventually cut was the Spy unit. The unit was supposed to be a way for players to see the other player base and plan his next moves based upon what the other player was building. The unit was also able to move unseen through the battlefield.

We cut the unit from the final version of the game because of it not making too much of a difference during a game as well as issues when it came to balancing it. In the end we could not decide if it was to be a unit or ability so we cut it from the current version. We also decided to cut one building from the game due to the same issues. The NSA building (working title) was to be the building you would have to build in order to be able to recruit the spy.

We did improve on the original idea as well with more detailed maps and a more fluid and balanced gameplay. Throughout the testing process we were able to make quick changes based on the feedback which in the end helped us develop a better game.

# 12.0 Future development

After the project delivery date at the 21st of May the product will be handed over to Syncrotec. The main objective at this point will be focused around marketing as well as a continuing development of new features in the game. These features will a based around micro-transactions as well as making the game profitable through other means (Ads, DLC).

By the 21st of May, the game is only optimized and tested for Android, specifically for the Samsung Galaxy S4 and Samsung Galaxy S3. We originally intended to do testing for iOS within the timeframe of the project, but the availability of Mac hardware and the fact that we received a Unity Pro license late in the process made that impractical.

There will also be a focus on getting the game out on the different stores so that it can be downloaded. How the company wishes to further develop the game is entirely up to them and may/may not include us. Future development also means the possibility to add more content to the game such as more buildings, units and levels.

It would also allow for even more work on design and further improving the code. Future development would also mean launching a simple marketing campaign for the game to increase the possible user base before release.

# 13.0 Conclusion

In conclusion, when working on a larger project like this correct method use and planning is crucial in developing a good end product. The first thing we did was to have a clear plan of what we wanted to develop. Through several iterations of testing and analysis we developed the game we had envisioned in the beginning. We started off with an idea that we were prepared to cut down content on if we felt it was necessary. By planning possible cuts from the beginning we felt confident on being able to develop a product the client would be happy with.

We spent a lot of time learning correct usage of methods such as SCRUM to maximize how much work we were able to do. We found the method to be useful when it comes knowing what is being worked on by who as well as giving us clear goals for each day.

Working closely with the client was also an important aspect of the process. We had to be able to make changes quickly based on the client’s needs and wishes. By using Skype for meetings we were able to keep in touch whenever we needed feedback on a certain aspect of the project.

Testing as much as possible is something that helped us shape the project into something a consumer would want. Without several iterations going through testing we would have had no idea on how the game feels for a regular player while still in the development phase.

The most difficult part of developing a game like this is the fact that we are working with an unknown IP in an overcrowded marked. Getting our game seen and played is a difficult process.

We decided against releasing the game on the different mobile stores at this moment so that we could have more time developing the game. Marketing is also something that is important before releasing an unknown IP. There has to be marketing done before the game is released in order to make sure that people know our game exists before it is released. This will increase our chances of having an adequately large consumer base.

When releasing a new mobile game the first week is crucial for its success. When released it will be placed on a new game list and is easily seen by everyone who goes to that store, but because of the flood of games that are released on a daily basis we wanted to make sure that it would be noticed in the list before other games were released. The end product was close to the original analysis with minor cuts due to time restraints, but we managed to keep all of the important aspects of the game intact.

14.0 Post Mortem

After developing this game there were things we had wished we had done differently. In the beginning we had several grand ideas on how we wanted the game to be, without taking enough time to discuss how we were to develop all of them.

This lead to several cuts in game content and slowed down the development process. Even though we were happy with the final product considering it was our first real game project, a lot of time could have been spent focused on less things, but instead try to develop them well. Over scoping is a common issue when it comes to development because of the fact that groups often plan their project after the idea that everything will go well always.

Even though the project in its entirety has gone well and we have not had any huge issues that haltered the project, there have been slowdowns due to late attendance and sickness. If we had more time to develop the game we would focus on adding more of the cut content back into the game. There would also be time spent on fixing some of the smaller bugs in the game and brushing up on the overall design. These things aside we have developed the game we intended with most of the original ideas implemented.

The end product is something we are proud of and we consider it an accomplishment considering all of our content and the complexity of the game code.

# 

# 15.0 Figure list

Figure 1 Scrum.................................................................8

Figure 2 Mindmap............................................................9

Figure 3 waterfall........................................................... 10

Figure 4 Gold balancing.......................................................... 16

Figure 5 Balancing……………………………………….17

Figure 6 Balance loop...................................................... 16

Figure 7: Game loop.........................................................19

Figure 8: Menu navigation................................................21

Figure 8, 9, 10, 11, 12, and 13 testing charts………………23 - 26

Figure 14 Testing Flowchart............................................ 26

Figure 15 Multiplayer flowchart................................... …28

16.0 Reference list

[http://nith.no](http://nith.no/)

<http://www.syncrotec.no/>

<http://no.wikipedia.org/wiki/Scrum>

<https://www.atlassian.com/software/jira>

[https://drive.google.com](https://drive.google.com/#my-drive)

[https://github.com](https://github.com/)

[https://www.tapasplay.com](https://www.tapasplay.com/)

<http://www.photoshop.com/>

<http://www.autodesk.no/products/autodesk-3ds-max/overview>

<https://unity3d.com/>

<http://www.visualstudio.com/>

<http://monodevelop.com/>

<https://www.facebook.com/>

<http://en.wikipedia.org/wiki/Flappy_Bird>

<http://www.candycrushsaga.com/>

<http://www.angrybirds.com/>

<http://www.ea.com/dungeon-keeper-and>

<http://www.kingdomrush.com/>

<https://developer.apple.com/app-store/review/>

<https://itunes.apple.com/no/genre/ios/id36?mt=8>

<http://msdn.microsoft.com/en-us/library/windows/apps/hh694083.aspx>

<https://play.google.com/store>

<https://developer.apple.com/app-store/review/>

<http://developer.android.com/distribute/googleplay/promote/brand.html>

<http://msdn.microsoft.com/en-us/library/windows/apps/hh694083.aspx>

<http://www.disney.no/>  
<http://www.skype.com/en/>  
<http://uxmatters.com/authors/archives/2012/06/ritch_macefield.php>  
<http://uxmatters.com/mt/archives/2012/06/ux-design-defined.php>  
<http://www.codewise.no/>   
<http://en.wikipedia.org/wiki/Three-click_rule>  
<http://www.conceptdraw.com/products/mind-map/>  
<http://www.getpaint.net/>   
<http://www.conceptdraw.com/>  
<https://parse.com/>  
<https://prime31.com/>  
Winston W. Royce (1970) *Managing the development of large software systems* Obtained from <http://en.wikipedia.org/wiki/Winston_W._Royce>

17 Attachments

Creating a building

GUI





This is an example of our first “Sketch” of the main menu. The colors and fonts are not yet complete, but it gives us an idea of what we want the graphical style to be like. The color of the title will be changed to give it a cartoony look. The grass and sky areas will have more detail.  We are also not done deciding where to place our Vikings. We want our design to be cartoony without giving our game a “childish” look. The final version of our main menu has more assets as well as redone text and backgrounds. Testing has shown that we had to edit the font to make it more readable.

### 17.1 Sprint Reviews

Sprint review for the first two weeks of our project. We spent most of the two first weeks of the project paintballing ideas between each other and trying to see what we actually could do with the time we were given. We drew up four to five “main” ideas that we talked about within the group. After this we had a meeting with the company we would be working for to see what they felt about our ideas and how we were to accomplish them. We gave ourselves one more week of brainstorming to see if we could keep the idea momentum going.

During this week we tried coming up with simple yet fun games that would be doable within our timeframe. We had to abandon some ideas because of the fact that they were either too similar to other famous games or that we felt that they would not be fun enough to last in the long run. There has been some discussion in this part on how we are to do the business model without losing the players interest. We did a in depth market analysis in order to take points from mobile games that have what is considered good models as well as the more known bad models.

At the end of the second week we had yet another meeting where we pitched the new ideas as well as showing off some prototypes we made to give a more visual representation of our ideas. We came to the conclusion that we were to make a tower defense game with a new twist.

We started off the next week by designing a basic proof of concept where we had some simple towers and moving enemies. The group dynamic was working well because of the fact that we all knew each other before we started.

We had assigned main tasks for everyone so that everyone had a main area of expertise. Because of this everyone knew what they were doing at each time. So far the daily stand ups have been useful for keeping everyone up to date with what is being done. We have set up a Jira account with an extensive backlog in order to keep everything under control. So far we feel that we are somewhat ahead of schedule when it comes to that game itself. The programmers understood Unity faster than we had though and this has led to a much more detailed second prototype than what we anticipated.

We are currently working hard on getting the game ready for our first official external testing. For this test we are planning to have a production quality map with the basic building and defending. We decided not to have every feature in this iteration.

At the end of iteration two we have gotten a lot of assets and back end coding done. The documentation has gotten quite a lot longer and detailed. When it comes to the testing we have decided to postpone it to the end of the next week so that we can make the prototype even better. As of now the things we will have to improve on is to get everybody to meet up at the correct time, but we are improving on this. Overall we are quite content with our progress on the iteration and we expect to do even better on the next iteration.

When it comes to the third iteration we have made small but important improvements to the code. Most of the work that has been done has been oriented around repairing broken code and doing basic testing. A big issue with this iteration is that we made some big mistakes when it comes to the game coding and we had to redo a lot.

We have improved on the aoe rock so that the smaller pieces of the rock also do damage to the enemy. We have begun on the code that lets units attack each other. When it comes to the artwork we have finished several important towers and buildings, they will have to be animated in the next sprint. We also finished a demo map that we showed off to different people at the school and received feedback on design.

We are currently trying out ways to add a save game function in our game. We decided to push the external testing back one week so that we can finalize the features we are currently working on.

In iteration four we made the biggest changes to the gameplay. The external testing this time was done on mobile phones. The testing focused on the graphical part of the game, as well as the new main menu. We felt that it was important to external testing on phones in this iteration. Testing our game on phones gave us valuable feedback on size and visibility. The testing showed that we need to make certain changes to our units, towers and GUI. The main changes we will have to make are the size of our units. Testers felt that they were a bit small and we will need to make them somewhat larger to compensate for a smaller screen. There will also have to be shadowing added to the towers and units to make them look less flat.

Overall we feel that we got a lot of work done in this sprint, and that we are closing in on finishing the back end portion of the project. This means that we soon can start polishing and mas develop maps for the game. We are also closing in on the biggest portion of the development, which is the network coding. We have given our programmers six weeks on this considering the size of this part.

In the fourth sprint we finished most of the GUI elements and we have also finished up most of the functionality. The back end of the project is getting closer to an end and we are currently preparing for next week’s network coding. The buildings and economy system is also close to completion and will be implemented in the next prototype of the game. We have made some changes to prefabs and scripts in order to make them more readable for later iterations.

Assets have also seen progress with the scout, barracks and market being completed and the blacksmith closing on completion. Balancing is completed on paper and will also be implemented soon. On the documentation side we have been putting together a test document for delivery.

The first week of the sprint saw some problems when it comes to sickness and we did not complete as many tasks as expected, but we made up for it by being more effective the last week.

### 17.2 Rules

Since the game is turn based (make towers/buildings/upgrade step, and unit attacking/ tower defending step), the game grants the player an amount of resources to spend for each building phase, this amount can be adjusted for each wave making it faster paced, or make it possible for the user to hurt the enemies amount.

There are restrictions on where you can build towers, this is in place to streamline the game in some form, and this rule also applies to the buildings in base. The main objective is to keep your town hall living after a set amount of waves.

You will lose the game when your town hall is destroyed by the invading force. When all waves have been defeated you will move on to the next level or if it is an online game you will move on to a new opponent or have the opportunity to have a rematch.

You will gain money by taking out your enemies as well as winning rounds. With the money you earn you can decide what you think you should spend them on from new towers to tower. As for the multiplayer aspect of the game we will have turns. This means that both players will plan their defense and offense and then they both have confirmed their moves the game will progress. There will always be a default group of enemies that will move out from both sides to not stop the game, but players can choose to send out certain enemies. The players will have to decide on if they want to focus on a heavy defense or a heavy offense. The last player with a town hall will win the game.

### 17.3 60 seconds of gameplay

The second wave of enemies is defeated and now the user has a new build phase. The user saw that the opponent used a lot of *heavy’s*. The user responds by upgrading the area of effect tower into an anti-heavy tower then selects jumpers from the barracks to counter the basic tower of the opponent. The user presses ready to start the next wave, the jumper moved swiftly past the opponent’s basic tower, and the anti-heavy upgrade did its job, if only the opponent didn’t build spies this would have been a good round.

### 17.4 Drawings:

This section is to show drawings that were used while creating the game.



Heavy unit: The heavy unit is a unit that focuses on power and health over speed and agility. He is a huge man with two shields to guard him from attacks.

.



Basic unit: The basic unit is the cheapest and first unit you will build before a round. This is a balanced unit meant to be mass produced.



Scout: The scout is a fast moving lightly armored unit. Its purpose is to be able to run fast trough the map in between the heavy’s. Its light armors makes it very vulnerable to towers and for him to be effective he should be accompanied with heavier units.



Town hall

The town hall is based on Viking long halls. Your main objective is to protect this at all cost.



Market

The market was based old town markets where you could buy more or less anything you might need. The colors were decided based on what we would consider less straining on the eyes.



Slowing tower

The slowing tower was to be a bell so that we could have people hitting it and causing tremors to slow enemies. The bell was colored yellow to not conflict with other objects in the scene.



Basic tower

The basic tower was designed as an old watchtower. We decided on this design to indicate that it is the simplest tower. The red flags are these to separate it from other assets you might see in the scene.



Example map. This is an example map that we used in the first couple of tests.

We decided on green hills inspired by Disney that was both colorful and detailed.



Barracks

The barracks in the game were based on older RTS games showing a training area outside a building. Colors were chosen based on what was the most comfortable on the eyes.



Lake

A basic lake with the edges. The edges around the lake are there to represent a drop and a change in terrain.



Blacksmith:

The blacksmith is used as a way to upgrade your unit’s life and movement speed.

Sketch for balancing loops

The five center black circles are the units, and the outermost black is the towers.



### 17.5 Visual themes

Our game will feature different environments for the player to advance through. We are currently looking at a forest like environment as well as a snowy winter landscape with ice cliffs and frozen lakes. The later levels will focus on a more grim burnt setting. This setting was designed to represent hell. We feel that these two environments in particular fit the theme we are going for. The contrast between the two different kinds of environments will give the player much needed variety so that they won't lose interest with the environments quickly. The themes also allow us to create some cosmetic assets to add variety to the environment. An important aspect of our design is to keep the games colorful even though they might be in environments that mostly consist of one color. We are working hard trying to add variety to these areas.

For our first external test we designed a frozen landscape for the player to test. Based on their feedback the design may be altered. An important part of the design is to make the map itself feel alive with cosmetic content.

This means that we will fill out the areas surrounding the path with trees, lakes and mountains, giving the world a more vibrant feel. It is also important that our cosmetic additions don’t distract the player or makes the player click on areas he is not supposed to. The artist has taken inspiration from other similar mobile games as well as Disney movies. The way the characters are being drawn gives them a unique hand drawn look which we feel is unique to the genre and will give our game a fresh look.

Main screen

        -Menu

                    -Graphic

                    -Sounds

        -Title

        -Background animation?

Back end

        -Music

        -Save function

        -Player to Player interaction

                    -Player to server

        -User to shop

                    -Micro transaction interface

                    -Client to server

Towers

Basic tower

        -Graphic

        -Projectile

                    -Sound

                    -Hit effect

        -Animation

                    -Attacking

                    -Idle

        -Upgrade tree menu

                    -Anti jumper

                               -Graphic

                               -Projectile

                                           -Sound

                                           -Hit effect

                    -Anti heavy

                               -Graphic

                               -Projectile

                                           -Sound

                                       -Hit effect

AoE tower

        -Graphic

        -Projectile

        -Animation

                    -Attacking

                    -Idle

        -Upgrade tree menu

                    -Anti scout

                               -Graphic

                               -Projectile

                                           -Sound

                                           -Hit effect

                    -Anti heavy

                               -Graphic

                               -Projectile

                                           -Sound

                                           -Hit effect

Slowing tower

        -Graphic

        -Projectile

                    -Sound

                    -Hit effect

        -Animation

                    -Attacking

                    -Idle

        -Upgrade tree menu

                    -Anti scout

                               -Graphic

                               -Projectile

                                           -Sound

                                           -Hit effect

                    -Anti jumper

                               -Graphic

                               -Projectile

                                           -Sound

                                           -Hit effect

Units

Scout  (Low health, Low damage, High speed)

        -Graphic

        -Animation

                    -Walking

                    -Hurt

                    -Doing damage

        -Sounds

                    -Getting hit

                    -Doing damage

Heavy (High health, Medium damage, Low speed)

-Graphic

        -Animation

                    -Walking

                    -Hurt

                    -Doing damage

        -Sounds

                    -Getting hit

                    -Doing damage

Jumper (Low-medium health, High damage, Jumps!)

-Graphic

        -Animation

                    -Walking

                    -Hurt

                    -Doing damage

        -Sounds

                    -Getting hit

                    -Doing damage

Spy (Stealth, High damage, Medium speed, Low health)

-Graphic

        -Animation

                    -Walking

                    -Hurt

                    -Doing damage

        -Sounds

                    -Getting hit

                    -Doing damage

Buildings

Town hall (used for health indication, and is the base structure of the game)

        -Graphic

                    -three stages

                               -Good

                               -Some damage

                               -Close to destruction

                    -Health indicator

        -Animation

                    -Spawning the basic unit

Barracks (Used to increase units produced to attack enemy, and uppgrades for the units)

        -Graphic

                    -Three stages of health

                               -Good

                               -Some damage

                               -Close to destruction

                    -Menu

                               -Hire units

                               -Upgrade units

Armory / Masonry (Used to upgrade towers, and the building in the base)

        -Graphic

                    -Three stages of health

                               -Good

                               -Some damage

                               -Close to destruction

                    -Menus

                               -Upgrade towers

                               -Upgrade buildings

Market (Used to affect the economy of the game)

        -Graphic

                    -Three stages of health

                               -Good

                               -Some damage

                               -Close to destruction

-Menu

-Financial “spells”

       -Icons

Tavern (Offensive upgrades to use against the enemy)

        -Graphic

                    -Three stages of health

                               -Good

                               -Some damage

                               -Close to destruction

        -Menu

        -Abilities

                   -Icons

NSA (Used for special units and intel abilities)

        -Graphic

                    -Three stages of health

                               -Good

                               -Some damage

                               -Close to destruction

        -Menu

                    -”Spells”

                               -Icons

                    -Hire unit

1. [↑](#endnote-ref-1)
2. http://www.codeproject.com/Articles/704720/SCRUM-explained [↑](#footnote-ref-1)
3. 1 http://en.wikipedia.org/wiki/Waterfall\_model

   2 http://uxmatters.com/authors/archives/2012/06/ritch\_macefield.php

   3 http://uxmatters.com/mt/archives/2012/06/ux-design-defined.php

   4 <http://www.conceptdraw.com/products/mind-map/> [↑](#footnote-ref-2)
4. http://www.photoshop.com/ [↑](#footnote-ref-3)
5. http://www.getpaint.net/ [↑](#footnote-ref-4)
6. http://www.conceptdraw.com/ [↑](#footnote-ref-5)
7. https://unity3d.com/ [↑](#footnote-ref-6)
8. https://github.com [↑](#footnote-ref-7)
9. https://drive.google.com [↑](#footnote-ref-8)
10. https://prime31.com/ [↑](#footnote-ref-9)
11. https://parse.com/ [↑](#footnote-ref-10)
12. http://audacity.sourceforge.net/?lang=nb [↑](#footnote-ref-11)
13. <http://uxmatters.com/authors/archives/2012/06/ritch_macefield.php> [↑](#footnote-ref-12)
14. <http://uxmatters.com/mt/archives/2012/06/ux-design-defined.php> [↑](#footnote-ref-13)
15. <http://gieger.com/site/research/The_3_Click_Rule.pdf> [↑](#footnote-ref-14)
16. https://developer.apple.com/app-store/review/ [↑](#footnote-ref-15)
17. http://developer.android.com/distribute/googleplay/promote/brand.html [↑](#footnote-ref-16)
18. http://msdn.microsoft.com/en-us/library/windows/apps/hh694083.aspx [↑](#footnote-ref-17)