

# Requirements and Analysis Document for Tower Defense Game

Hannes Svahn      Felix Nilsson      Oscar Forsyth  
Jon Emilsson      Simon Larsson

September 2020

## 1 Introduction

The application that is currently in development is a Tower Defense game. The game's purpose is for the player to defend himself through building towers which shoots at the incoming enemies. Placing different towers at different places is where the game involves some strategy from the player, and where some depth is found. The goal of the game is either to play until the player survives a number rounds. The game's target group is people already interested in games as a form of entertainment. Users who will benefit from this particular game are those who have an interest in gaming or perhaps fans of this particular genre of game.

### 1.1 Definitions, acronyms, and abbreviations

The following list contains shortened version of concepts that we use often.

- TD, Tower Defense. A particular genre of strategy where the player places towers to protect their base from attacking enemies along a path.
- OOP, Object-Oriented Programming. A programming paradigm characterized by the use of objects containing data and code.
- AOE, Area of Effect. A common term used to denominate if something affects a certain area instead of just a single thing. Frequently used with AOE damage, which refers to something damaging all things in an area rather than specific targets.

## 2 Requirements

### 2.1 User Stories

User stories are given the following format: *Story Name - ID*

#### **Buyable Towers - OP-8**

*Description:*

As a player I would like to buy towers because I would like to defeat the enemies in the game

*Confirmation*

Being able to select towers that the player want to place, as long as the player has enough money.

*Functional:*

- Can I choose what towers i want to buy?
- Can I see how much a specific tower costs?
- Can I see how much money i currently have?

*Completed:*

Yes.

#### **Placable Towers - OP-17**

*Description:*

As a player I want to place the towers I purchase because I want to apply my own startegy when it comes to completing the game

*Confirmation:*

- Being able to place selected towers on the grid where towers can be placed.
- Being able to place multiple towers on the grid for a varied strategy

*Functional:*

- Can I select where to place the selected tower?
- Can I place multiple towers?

*Completed:*

Yes.

#### **GUI theme - OP-35**

*Description*

As a gamer I want a GUI that follows the theme of the game, and blends in.

*Confirmation*

The GUI has a similar colour scheme

*Functional*

- Can I see the different aspects of the game, without the theme "getting in the way".

*Non-Functional*

- Do I get pulled out of the experience when i open a menu?

*Completed*

Yes

### **Un cluttered screen - OP-36**

#### *Description*

As a gamer I dont want the screen to be cluttered with unnecessary panels., such as a thin sidebar housing the turrets that are purchasable, a hidden settings menu that can be pulled up, and a closable detailed view of a specific.

#### *Confirmation*

Most panels should have a close function.

#### *Functional*

- Can I pull up a settings menu, and remove it? - Can i open a detailed view of a turret and close it? - Is the sidebar housing the purchasable turrets thin

#### *Non-Functional*

- Can I understand what I'm looking at in a moments notice? - Is the screen cluttered with non game-related GUI elements?

#### *Completed*

Yes

### **Theme of the Game - OP-10**

#### *Description*

As a player I would like an engaging theme to the game since it keeps me more engaged in the gameplay

#### *Confirmation*

- Having a theme in the game which the audience can be interested in.

#### *Non-Functional*

- Can I understand the theme of the game?

#### *Completed*

No

### **Enemy and Tower Sprites - OP- 55**

#### *Description*

As a player I want enemies and towers to have sprites that connect to the rest of the games, with a common theme.

#### *Confirmation*

- Having enemies look part of a theme. - Having towers look part of a theme

#### *Non-Functional*

- Does the enemies and towers fit in with the decided theme?

#### *Completed*

No

### **Menu theme and title - OP - 54** *Description*

As a player I want the menu to introduce me to what the game is about, theme-wise and gameplay- wise.

#### *Confirmation*

- The menu makes the theme clear and has a fitting title or other description

*Non-Functional*

- Does the menu have backgrounds and titles that engage me in the game? -  
Does the menu have backgrounds and titles that make me understand what the game is about?

*Completed*

Yes

### **Upgradable Towers - OP-7**

*Description*

As a player I would like to upgrade my towers because I like depth to a game

*Confirmation*

- Having different upgrades for the different towers

*Functional*

- Can I upgrade the different towers with different upgrades?

*Completed*

Yes

### **Different Towers - OP-15**

*Description*

As a player I would like there to be some difference between turrets so that buying and placing them involves some strategy

*Confirmation*

- Having different towers to choose from.

- Having tooltips that show what each tower does.

*Functional*

- Can I see what each tower does?

*Completed*

Yes

### **Different Damage Types - OP-11**

*Description*

As a gamer I want to have different types of damage to make the game more strategic

*Confirmation*

- Having different damage types

*Functional*

- Can I see what each tower has for damage type?

- Can I see each enemies resistances

*Completed*

Yes

### **Menu - OP-13**

*Description*

As a player I want a menu at the start of the application so I can choose what I want to do

*Confirmation*

- Having a menu with different settings that the player can choose from.

*Non-Functional*

- Can I understand the menu so that I can navigate to the location i desire?

*Completed*

Yes

### **Difficulty settings - OP-14**

*Description*

As a player I want different difficulty levels to make it more challenging

*Confirmation*

- Having a menu with different difficulty settings that the player can choose from.

*Functional*

- Can I understand the difference between the different difficulties?

*Completed*

Yes

### **High Score - OP-9**

*Description*

As a player I want to keep track of my high-score because as a player I like to improve at games I play.

*Confirmation*

- Have a high-score function where it saves your highest score and you can view them.

*Functional*

- Can I see the highest score?

*Completed*

No

### **Multiplayer - OP-32**

*Description*

As a player I want to play with my friends since it is fun to play with others

*Confirmation*

- Have a multiplayer function were two people can either play on the same map together or against one another

*Functional*

- Can I play with my friends?

*Completed*

No

## **2.2 Definition of Done**

For a user story to be considered done, there are multiple criteria that need to be completed.

- The code needs to be reviewed so that it follows OOP guidelines.
- Each class needs to have necessary tests so that everything works as expected.
- Code should be reviewed, tested and under version control.

## 2.3 User interface

Figure 1 represents the first sketch of the main UI section, drawn on a whiteboard. It features the key elements of a typical TD game: A grid style map, where enemies travel along a path, turning at right angles. To the right is a sidebar containing information like currency and health, towers available for purchase and a button to start a new round. Being a mockup, it lacks obvious features such as surrounding views, a settings or a menu, as well as how it should react to user input.

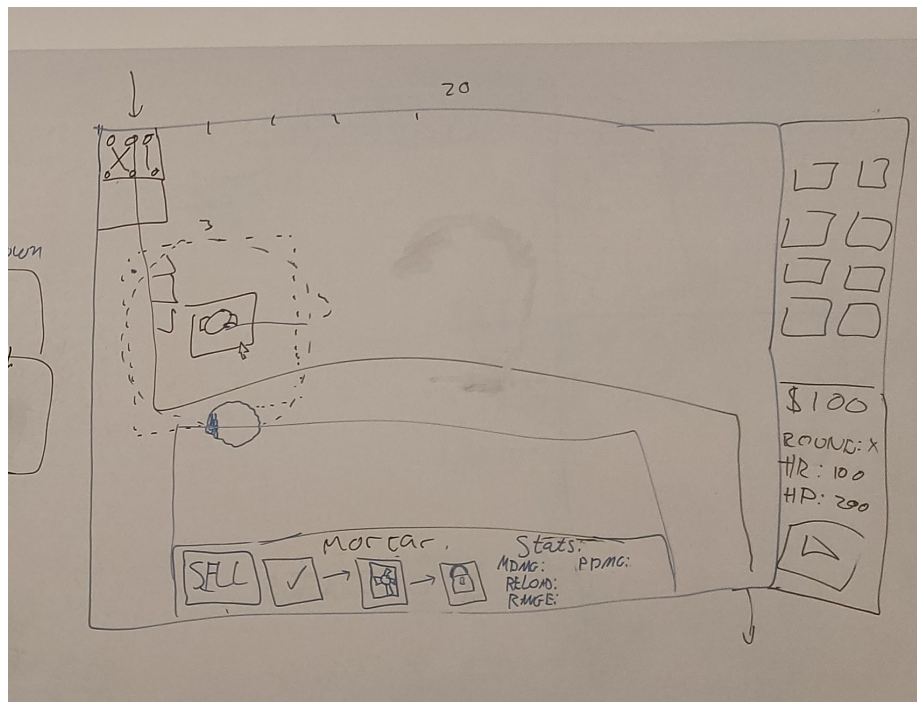


Figure 1: The first sketch of the UI

Figure 2 shows an early runnable version of the same main UI section as was drawn as a whiteboard mockup. It features many of the same elements mentioned, this time with temporary sprites and solid colours, which are planned to be replaced at a later stage. It is reached through a main menu as seen in

figure 3, displayed to the user when starting a new game. In the main game, the user can also reach a lightweight settings menu which allows for changes to the such as ingame volume or whether or not to show the grid on the map, this can be seen in figure 4.

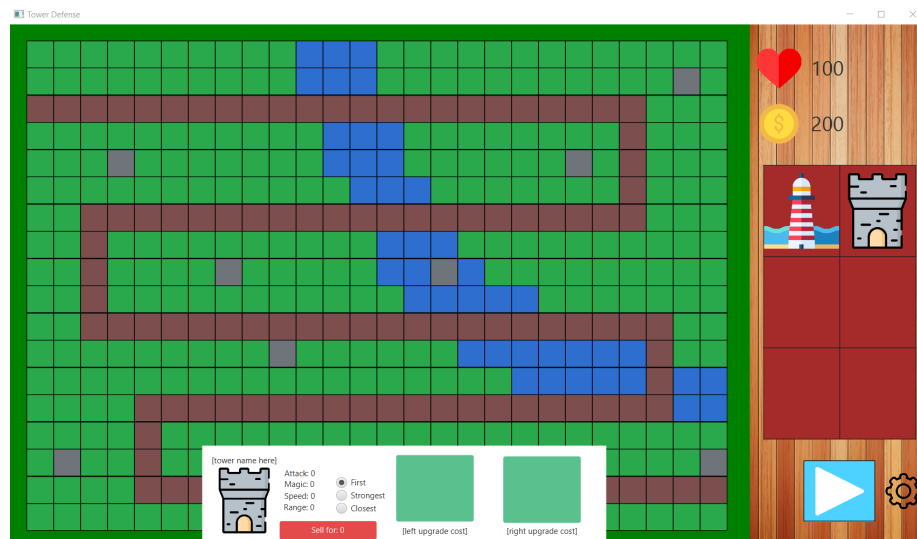


Figure 2: An early runnable UI

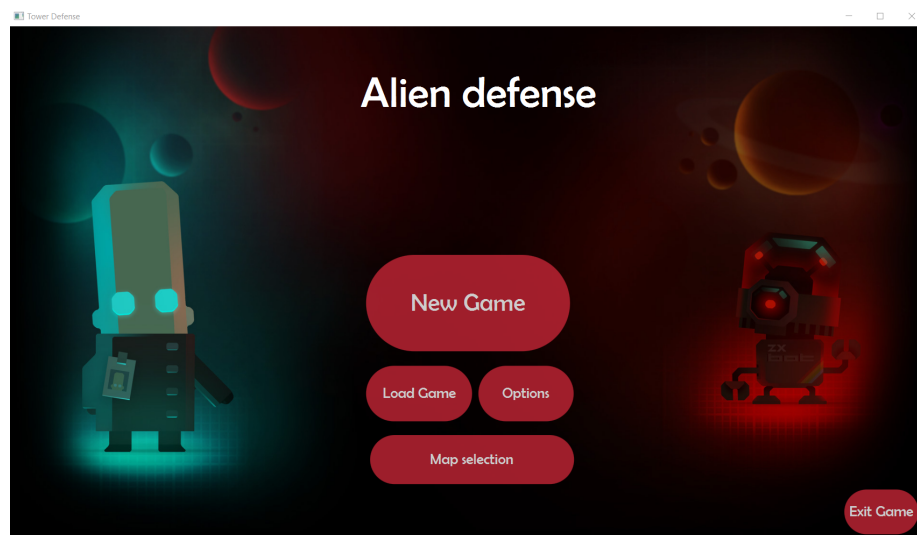


Figure 3: An early main menu

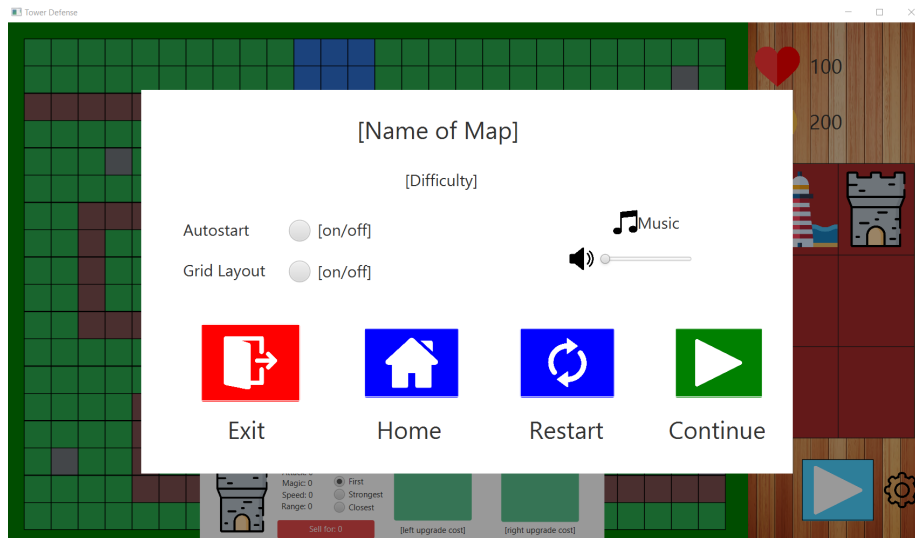


Figure 4: An early settings menu

From this main menu three views are currently accessible: A "New Game" button which takes us the main game, shown above. There is a submenu labeled "Map Selection" for accessing a list of maps, currently only featuring the one map shown earlier. This can be seen in figure 5. There is also an "Options" submenu, giving the player a choice of difficulty between easy, medium and hard. This can be seen in Figure 6.

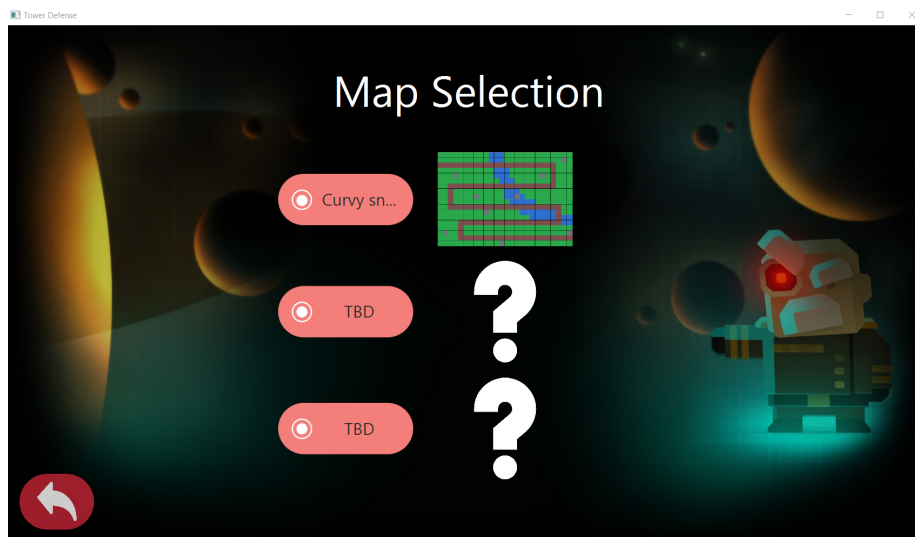


Figure 5: An early map selection



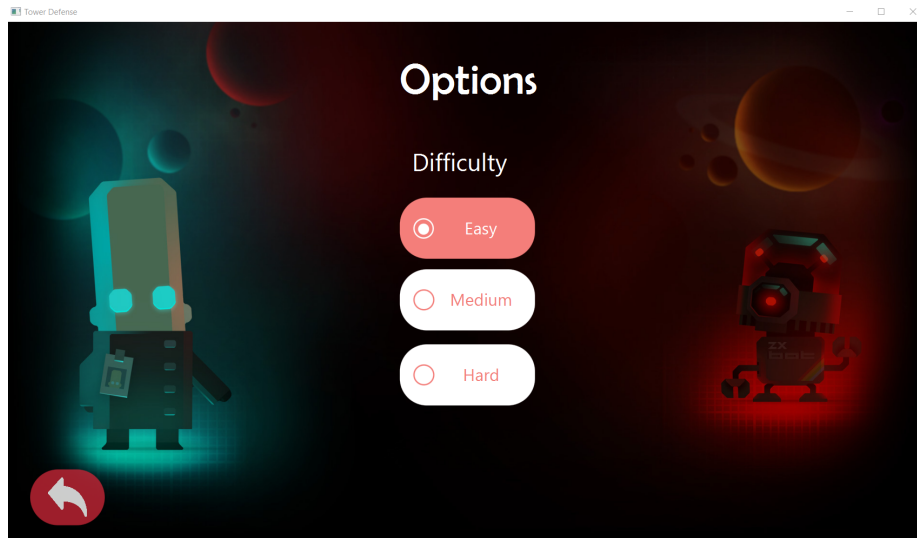


Figure 6: An early options menu

To clarify, the following figure is a representation of how the views related to one another. The lines represent that they are reachable through a click of some screen element.



Figure 7: An overview of the different views present in the game

### 3 Domain model

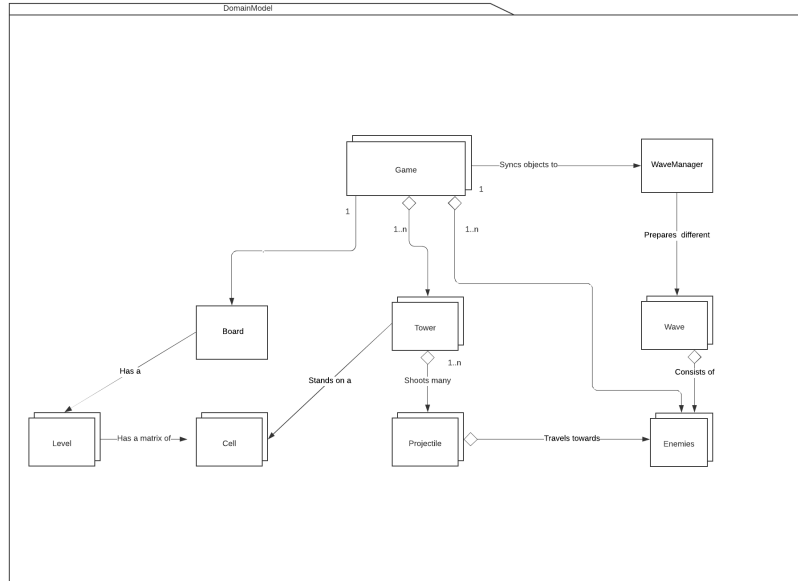


Figure 8: Domain model

#### 3.1 Class responsibilities

The following section contains the most important classes and their respective responsibilities. The different classes are grouped under the same category because they share similar traits and responsibilities.

##### 3.1.1 Game

Game's responsibility is to be the center of the model and make sure all the pieces fit together to a running game. It'll be the outside of the model which other parts of the program and the user can access. It will also be the largest part of the program as it handles most information.

##### 3.1.2 Tower

Tower is an interface that is the common denominator for all towers. The interface methods are realized in BaseTower and all the different Towers that have their own class. The different types of Towers, like MageTower and ArcherTower concretize the Towers with different shapes and abilities. Each of them

has a private BaseTower in the form of composition which gives structure to the more basic behaviours of a Tower. All towers are then created through their own factory class which inherits behaviour from a common interface. There is also a projectile class which handles the object that Towers shoot, therefore each Tower also has a type of projectile. This class can be further developed to allow things like AOE damage and other things that allow different strategies for different Towers.

### **3.1.3 Enemies**

Similar to Tower, Enemies apply a similar pattern by having an Enemy interface and a BaseEnemy. Where the classes with lower level of abstraction concretize the enemies. In the same way as Tower, all enemies are also created through factories, in a similar style as with Towers.

### **3.1.4 Wavemanager**

The purpose of wave and wavemanager is to provide Game with waves of enemies depending on the current round, difficulty and map.

### **3.1.5 Cell and Board**

Similar to how Enemy and Tower works, Cell is an interface with corresponding lower abstract classes, which separate behaviour. Most cells are used to put towers on but there are other cells such as obstacle cell which does not allow you to put towers on it and path cells which the enemies walk on. Similarly all these also have a BaseCell which gives basic behaviour that all cells have.

Board is the main user of cell as it creates different maps with the cells that as mentioned acts as the terrain in the game. The board also handles the general logic of the maps, so the enemies know in which direction they should move in each space.

### **3.1.6 Player**

Player is a basic class which purpose is to keep track of the highscores, player name and id.

## **4 References**

-Scene Builder for the map and toolbars -Maven (soon) -JUnit for tests -IntelliJ as IDE