Requirements and Analysis Document for 0x2EE

Grupp 1
Arthur Alexandersson, Gustav Gille, Herman Norén,
Kasper Ljunggren, Rickard Leksell

date (Yes please)
version (3.8.1 MVN)

1 Introduction

****First draft****

0x2EE is inspired by a game created by Sean Cooper named Boxhead[1] where the goal was to acquire the largest score, which is done by killing zombies, before the player dies by damage taken from them.

The goal of the application of 0x2EE is a pastime amusement game while the stakeholders in the group's project are people who want to pass the time with a nostalgic game and those who want to see how object-oriented games are made through analysis and sprint documentation.

The group's game differs in multiple factors from its inspiration such as more implementation as in a shop where the player's weapon and armor can be upgraded, different artstyle and different enemies. The goal of the game is still the same as the original, to acquire the largest score possible before dying.

//

(Give some background and explain the purpose of this application. Describe the functionality of the application. Describe the stakeholders of the project, highlight who will benefit from/use this particular application.)

1.1 Definitions, acronyms, and abbreviations

Create a word list to avoid confusion and give a definition of every abbreviation you use in the document.

- Boxhead:
  - a game created by Sean Cooper [1] where the player tries to acquire the

largest amount of points before dying.

- Score:
  - The measurement of how many enemies the player has destroyed.
- Highscore:
  - The highest score scored from a game.
-

2 Requirements

2.1 User Stories

# Game Window

Story Identifier <1>

Story name <Game Window>

## Description

As a user I need to be able to view the game on my screen so that I can play the game.

## Confirmation

- A window appears when I start the program

### Functional

- If I start the game can I see a window appear?
- Can I view the game?

### Non-Functional

- Does it work on different OS?

# Start Menu Navigation

Story Identifier: <2>
Story Name: <SMN>

## Description

As a <user>, I need <navigation in the main menu> so that <I can view different pages containing different information and the availability to start playing the game>.

## Confirmation

- I can navigate to different parts of the program through the main menu.

### Functional

- The navigation is keyboard only.
- If I have a button selected it is clearly highlighted.

### Non-functional

- MVC View handles navigation independently with input from controller

# Page With Highscores

Story Identifier: <3>
Story Name: <Highscores>

## Description

As a <user>, I need <the possibility to show highscores> so that <I have a goal and highscore to beat when I play the game>.

## Confirmation

- Highscore page which is updated whenever a new highscore is set.

### Functional

- Can I view previous highscores?
- If I click on the Highscores menu button does the highscores page appear?

### Non-functional

-

# How To Play Menu

Story Identifier: <4>
Story Name: <H2P>

## Description

As a <user>, I need <a “how to play” description> so that <I, as a new player, know how to play the game>.

## Confirmation

- The “how to play” description is easy to understand.
- It has all the information I need as a new player.
- The page contains a short description of the goal of the game.

- The page shows which buttons on the keyboard the player can use while playing the game.

### Functional

- Can I easily navigate to the “How to play” page from the start menu?

### Non-functional

- Is the language well adapted to the user?
- Does the instructions contain enough information such that a new player can easily start playing the game without difficulties understanding controllers etc?

# Start Menu Quit

Story Identifier: <5>
Story Name: <Quit>

## Description

As a <user>, I need <to be able to quit the game with ease from the main menu> so that <the game does not keep on running when I do not want to play>.

## Confirmation

- The menu contains a Quit button.
- The button, when chosen, exits the program.

### Functional

- Can I quit the game with ease by choosing the Quit button?

### Non-functional

# Start Menu Play

Story Identifier: <6>
Story Name: <Play>

## Description

As a <user>, I need <the option to choose Play to start playing the game> so that <I can play the game>.

## Confirmation

- There’s a button I can choose from the start menu which starts the game.

### Functional

- Can I start the game from the main menu?

### Non-functional

- The game can not be started without choosing the Play button.

# Pause Menu

Story Identifier: <7>
Story Name: <Pause>

## Description

As a <user>, I need <the option to pause the game during playtime> so that <I can take a break from the game or navigate away from the current session>.

## Confirmation

- The user can pause the game.
-

### Functional

- A pause menu appears when the pause button is pressed (escape).
- Can I press escape on my keyboard to pause the game?
- If I press escape, does a pause menu appear?
- Does the game pause when I pause?

### Non-functional

# Navigate to Main Menu From Pause Menu

Story Identifier: <8>
Story Name: <Pause To Main Menu>

## Description

As a <user>, I need <to be able to navigate to the start menu from the pause menu> so that <I can return to the main menu if I no longer want to play>.

## Confirmation

- The pause menu contains a button which takes the user to the main menu.

### Functional

- Can I choose a button which takes me to the main menu?
- When I press the main menu button does it take me to the main menu?

### Non-functional

- availability...
- security…

# Game Map

Story Identifier: <9>
Story Name: <Game Map>

## Description

As a <user>, I need <a game map with randomly generated obstacles> so that <my in-game character has somewhere to be and obstacles to avoid which are in new places every time I play so the game is kept interesting>.

## Confirmation

- The game map is large enough for the user.
- The game map has terrain which can not be passed by the player.

### Functional

- The player can not move through obstacles.
- The player can not move outside the game map.

### Non-functional

- availability...
- security…

# Player Movement

Story Identifier: <10>
Story Name: <Player Movement>

## Description

As a <user>, I need <to be able to move the player> so that <I can avoid enemies and experience different locations on the map>.

## Confirmation

- The movement is responsive, when the button is pressed I should get instant response.
- The movement follows button conventions (W,A,S,D)

### Functional

- Can I press ‘W’ to move up?
- Can I press ‘A’ to move left?
- Can I press ‘S’ to move down?
- Can I press ‘D’ to move right?

### Non-functional

# Camera

Story Identifier: <11>
Story Name: <Camera>

## Description

As a <user>, I need <the game camera to follow the player and be able to change to other focusable objects> so that <I can see the player, and be able to see when i.e. a boss spawns>.

## Confirmation

- The camera follows the player.
- The camera can change the object which it focuses on.
- The camera can zoom.

### Functional

- The camera does not show what’s outside the border.
- The camera zooms in without distorting the view.

### Non-functional

# Weapon

Story Identifier: <12>
Story Name: <Weapon>

## Description

As a <user>, I need <a weapon which shoots projectiles> so that <I can damage enemies>.

## Confirmation

- The weapon shoots projectiles which move on the map.
- The projectiles can not move through obstacles.

### Functional

- Can I press a button to shoot a projectile?
- Does the projectile damage enemies?

### Non-functional

- availability...
- security…

# Armor

Story Identifier: <13>
Story Name: <Armor>

## Description

As a <user>, I need <armor> so that <I can take reduced damage from enemies>.

## Confirmation

- The armor reduces damage taken from enemies.

### Functional

- Do I take reduced damage from enemies with my armor?

### Non-functional

- availability...
- security…

# Enemies

Story Identifier: <14>
Story Name: <Enemies>

## Description

As a <user>, I need <enemies in the game which I can shoot with my weapon> so that <I can receive score to compete for highscores and currency for upgrades>.

## Confirmation

- The enemies can be damaged and killed.
- The enemies spawn during the game.
- The enemies are chasing the player.
- The enemies can damage the player.

### Functional

- Can I kill enemies by shooting them?
- If I collide with an enemy, does my health decrease?

### Non-functional

- Graphics are “good”.

# Different types of enemies

Story Identifier: <15>
Story Name: <Enemy types>

## Description

As a <user>, I need <different types of enemies> so that <the game keeps my interest, the game should not get repetitive>.

## Confirmation

- Different types of enemies appear in the game.
- The different types of enemies have different graphics.

### Functional

- Different enemy types have different movement speed/health/damage.

### Non-functional

# Ingame Shop

Story Identifier: <16>
Story Name: <Shop>

## Description

As a <user>, I need <an in-game shop> so that <I can spend my currency and upgrade my weapon and armor to become stronger>.

## Confirmation

- The game has an in-game shop which can be used to advance damage and armor for the player.

### Functional

- When I open the shop I get the option to spend my currency to upgrade my player.
- If I choose to upgrade armor, my armor gets upgraded.
- If I choose to upgrade weapon, my weapon gets upgraded.

### Non-functional

# Health potions and coins have a chance to drop when an enemy is killed.

Story Identifier: <17>
Story Name: <Coins and Potions>

## Description

As a <user>, I need <drops from enemies, health potions and coins> so that <I can receive something else than score when killing enemies>.

## Confirmation

- Enemies have a chance to drop coins or health potions when killed.
- The drops have a set chance to drop when killed, i.e. 30%.

### Functional

- I have a chance to receive a coin or a health potion when killing enemies.

### Non-functional

- availability...
- security…

Use the template from the course website and list all user stories here. It is fine to have them in a spreadsheet (or other applications, such as Trello) at first, but they must end up here as well.

These user stories should describe what the user will be able to do. Write the user stories in the language of the customer, and give them a unique ID. List the user stories in order of priority.

You need to annotate an user story whether or not it is implemented. We need to know which user stories are implemented, such that we can check this during the oral presentation.

USER STORY MALL
# User story

Story Identifier: <id>
Story Name: <short name>

## Description

As a <role>, I need <requirement or feature> so that <goal / value / reason>.

(check INVEST guidance: https://en.wikipedia.org/wiki/INVEST_(mnemonic))

## Confirmation

List all acceptance criteria; you should be able to test/confirm these.

### Functional

- Can I ...
- If I click ...

### Non-functional

- availability...
- security…

2.2 Definition of Done

1. The code shall follow OOP-principles and design patterns.

2. The code adheres to the MVC-design pattern.

3. All methods/classes are commented with JavaDoc.

4. The classes implemented in the Model module are fully tested using JUnit tests.

5. All code is merged to the master branch without negative consequences.

2.3 User interface

Here follows early sketches of the game.

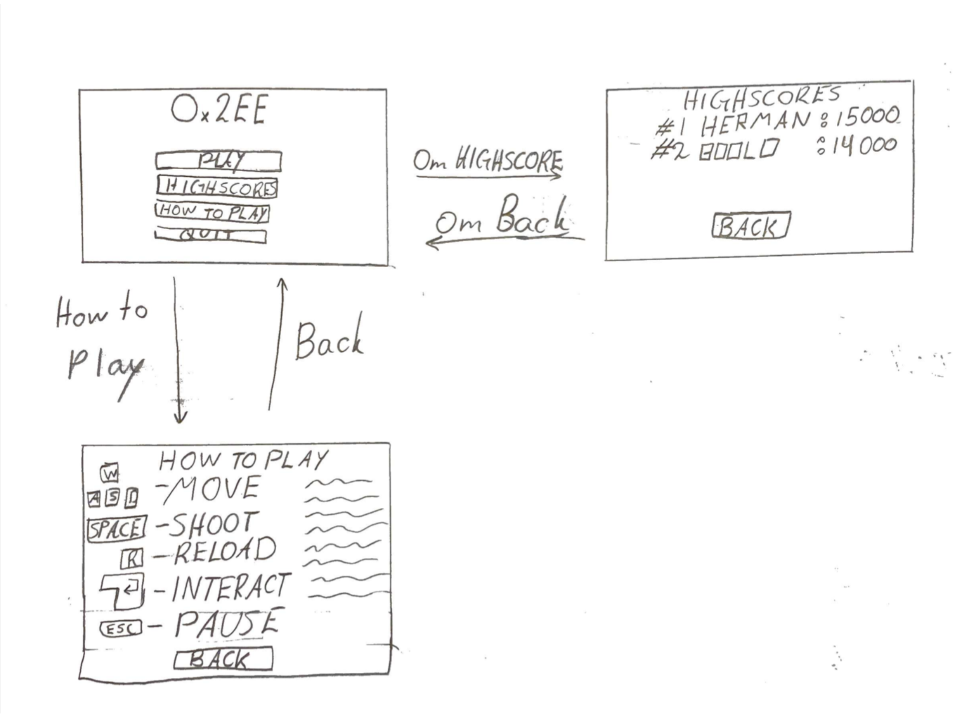

Figure 1: Main menu, Highscores & How to play

The main menu is shown at the top-left in Figure 1. Here you can choose what you want to do, and apart from playing the actual game, you get the possibility to show your highscores and instructions on how to play the game. In the highscoreview (top-right Figure 1) you can see names and scores for the top 5 best saved scores players have received. These are also presented in different colors to clarify each placement. In the “How to play”-view (bottom-left Figure 1) all controls for the game are shown together with a text describing how the game works.

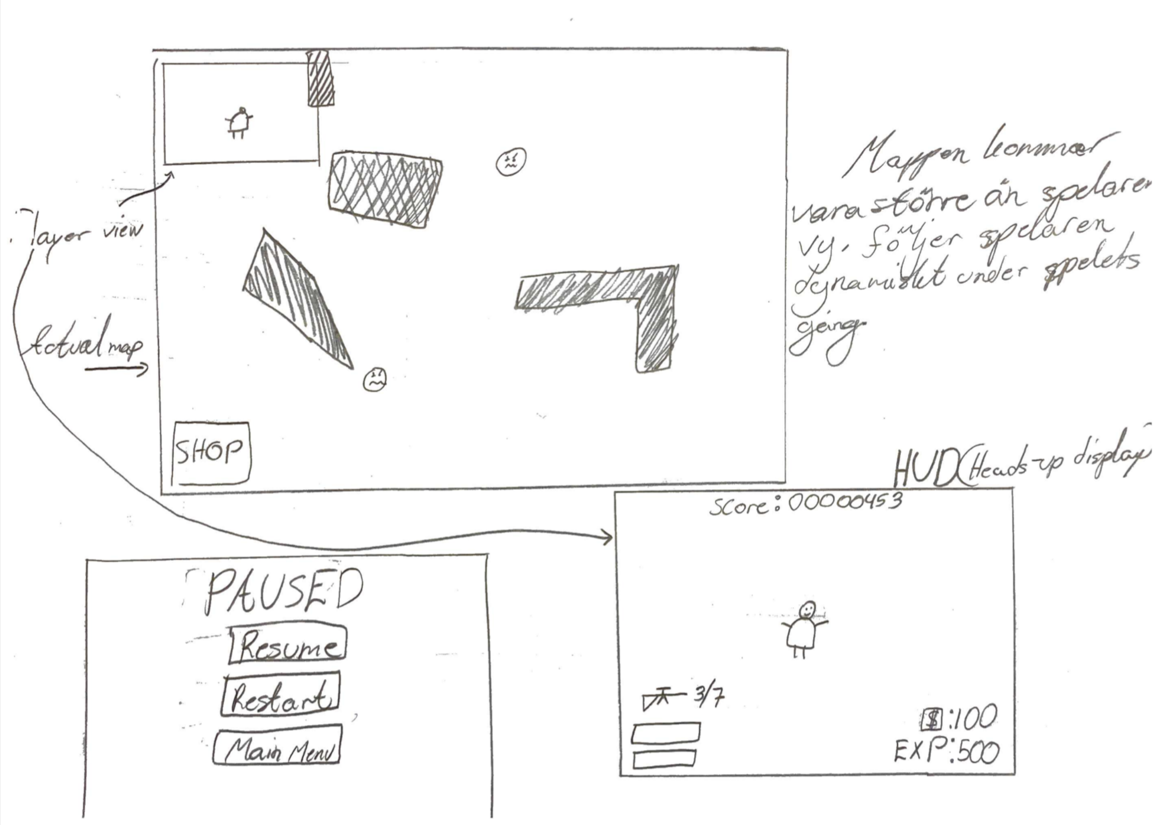

Figure 2: In game & Pause menu

In the top sketch in Figure 2 you can see how the actual game works. The game consists of a map bigger than the actual square that the player sees, and instead there will be a “camera” following the player around the map. There will also be obstacles positioned around the map which neither players or enemies will be able to pass through. The game is a “infinity-game” where enemies continuously spawn and attack until the player no longer is alive. On the map, there will also be a shop where the player can go to upgrade their weapon and armor.

In the bottom sketch to the right in Figure 2, you can see how the game will actually look for the user, whereas the character is placed somewhat in the middle of the screen. During the game, a HUD is also visible which updates the user on certain information, for example their health, score and money.

If you choose to pause the game, you will end up on the view to the bottom-left in Figure 2. Here, the user can choose to navigate back to the main-menu, restart the game or resume the current round.

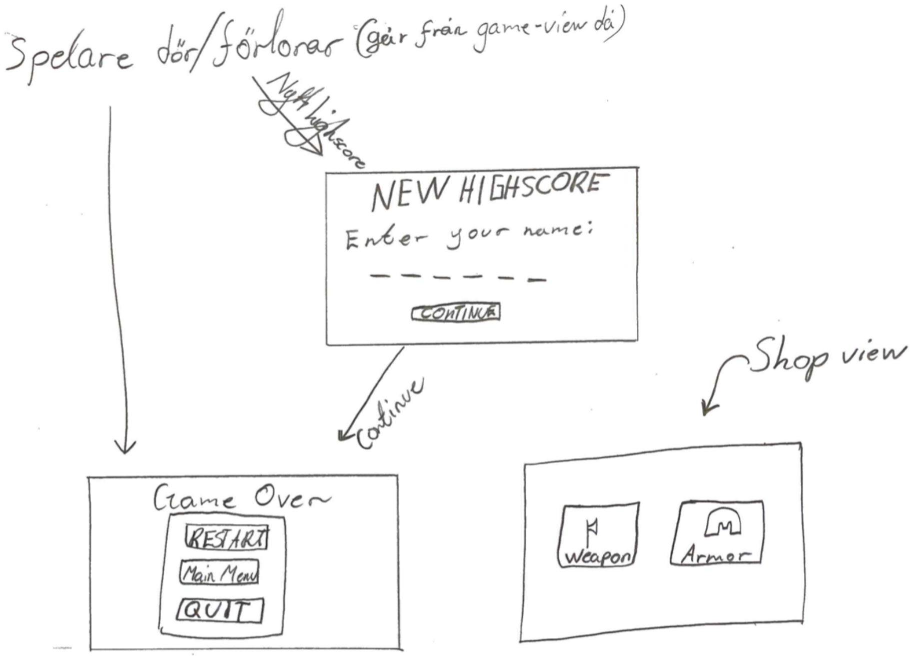

Figure 3: New Highscore, Shop, End

When the game is over, you will be directed to one of two views. If you reach a highscore, you will firstly come to the view shown at the top in Figure 3. Here you will write your name, which will then be saved in a file along with your current new score. When it has been sabed you will be directed to the “Game over”-view, shown at the bottom left in Figure 3. Here you can choose to play again, go back to the main menu och close the game. If you don't reach a highscore, you will instead be directed directly to this view.

On the bottom-right in Figure 3 you can see a sketch of the shop. This one will, among other things, consist of 2 buttons where you can choose to upgrade your weapon or armor.

3 Domain model

Below follows an overview of the project’s structure represented in a UML-diagram, see figure 4.

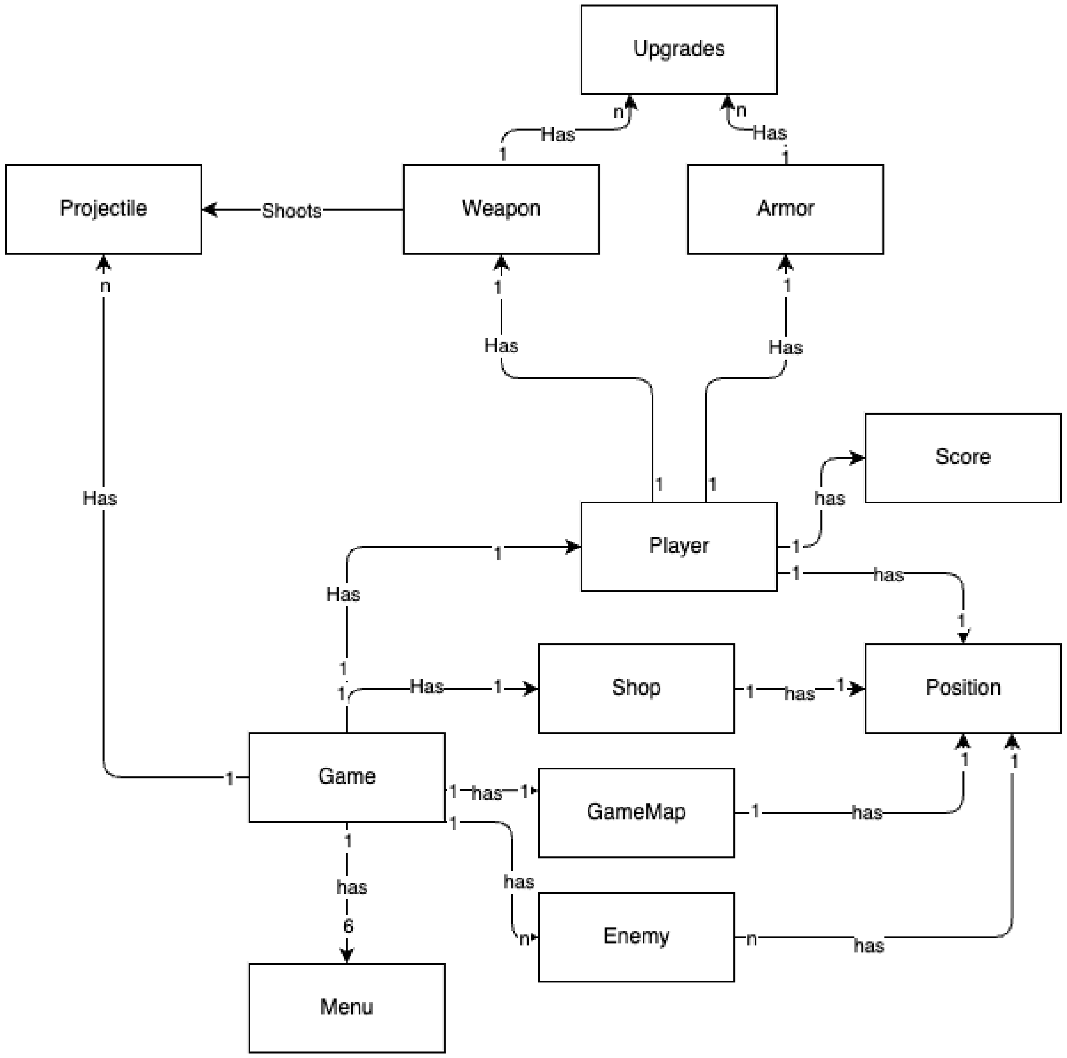

Figure 4: The domain model describing the project.

3.1 Class responsibilities

Explanation of responsibilities of classes in diagram.

Game - Connects all parts of the model together and stores important information about the current game status. (facade)

Player - This class handles all information related to the actual player of the game. Its responsible for storing and updating data like the players health, score & money. It also handles the connection between the player and its weapon and armor.

Enemy - Handles the movement logic for entities other than the player. Makes enemies move towards a certain goal, in this case the player. Every enemy is also responsible for storing and updating information about its health.

Weapon - Handles weapon logic, by storing information about a weapons current “status” i.e its level, how much ammo it has, damage, reloading and upgrading logic . Its also responsible for shooting.

Armor - Handles armor logic, by storing information about an armor's current “status” i.e its level and damage reduction, and also upgrading logic.

GameMap - Its responsibility is to generate a random map with obstacles and store information about the current tiles on the map.

Shop - The responsibility of this class is to handle transactions with the players money and upgradable items.

Projectile - Handles the movement of projectiles/bullets in the game.

Position/Score - (INGEN KLASS?)

4 References

[1] Wiki. 2022. *Boxhead (series)*. [online] Available at: https://boxhead.fandom.com/wiki/Boxhead_(series) [Accessed 3 October 2022].