Requirements and Analysis Document for 0x2EE

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

date 2022-10-20

1 Introduction

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.

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 by killing enemies, before they kill the player.
- **Health Potions -** a potion that gives the player more health
- **Coins -** Currency of the game
- **Armor -** Reduces the enemies' damage, can be upgraded to reduce damage even more.
- **Weapon -** The player has a weapon which can be upgraded to deal more damage.
- **Enemy -** Entities that try to harm the player.

2 Requirements

2.1 User Stories

All user stories are implemented and have passed the tests

 # Game Window

Story Identifier <1>

Story name <Game Window>

Description

As a player 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

Requirements

- If I start the game can I see a window appear?
- Can I view the game?
- Should work for both Mac and Windows

 # Menu and Navigation

Story Identifier: <2>
Story Name: <M & N>

Description

As a player, I need <a menu with relevant information I can navigate through> so that <I ie. start the game, learn how to play, see highscores or quit the game>.

Confirmation

- Pressing play starts game
- Pressing quit quits game
- Pressing how to play brings you to that page

Requirements

- Is the navigation keyboard only?
- Does the menu have buttons which I can choose?

- Does the menu contain everything I need, ie. How to play, Start game, etc. ?
- Navigation is correct (the correct subpage is loaded when the corresponding button is chosen).

Pause Menu
Story Identifier: <3>
Story Name: <Pause Menu>

Description

As a player, 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
- Enemies do not move while paused.

Requirements

- A pause menu appears when the pause button is pressed (escape).
- Pause menu appears when pressing escape
- I have the option to navigate to the main menu.
- I have the option to restart the game.
- Controller notifies View and Model to pause

Game Map

Story Identifier: <4>
Story Name: <Game Map>

Description

As a player, 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 has terrain which can not be passed by the player.

Requirements

- The player can not move through obstacles.

- The player can not move outside the game map.
- The game map is large enough for the user.

Player Movement

Story Identifier: <5>
Story Name: <Player Movement>

Description

As a player, 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)

Requirements

- The player moves in all directions according to a conventional WASD mapping.
- Acceleration is used to make sure that speed is not static

Camera

Story Identifier: <6>
Story Name: <Camera>

Description

As a player, 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.

Requirements

- The camera does not show what’s outside the border.

- The camera zooms in without distorting the view.

Weapon

Story Identifier: <7>
Story Name: <Weapon>

Description

As a player, 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.

Requirements

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

Armor

Story Identifier: <8>
Story Name: <Armor>

Description

As a player, I need armor so that I can take reduced damage from enemies.

Confirmation

- The armor reduces damage taken from enemies.

Requirements

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

Enemies

Story Identifier: <9>
Story Name: <Enemies>

Description

As a player, I need enemies in the game which I can shoot with my weapon so that I am challenged in the game.

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.

Requirements

- If I collide with an enemy, does my health decrease?

Different types of enemies

Story Identifier: <10>
Story Name: <Enemy types>

Description

As a player, I need different types of enemies so that the game keeps my interest and doesn't become repetitive.

Confirmation

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

Requirements

- Different enemy types have different movement speed/health/damage.
- Enemies extend an abstract class used by all entities in the game

Ingame Shop

Story Identifier: <11>
Story Name: <Shop>

Description

As a player, 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.

Requirements

- 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.

Highscores

Story Identifier: <12>
Story Name: <Highscores>

Description

As a player, I need the possibility to show highscores and add new highscores so that I have a highscore to beat when I play the game and the possibility to remember my previous records.

Confirmation

- If a new highscore is reached, it is stored in the text file and viewable on the highscore page.

Requirements

- Previous high scores can be viewed through the menu.
- If I click on the Highscores menu button does the highscores page appear?
- High scores are stored in a text file to carry over between sessions.
- Highscores are loaded in descending order.

How To Play Menu

Story Identifier: <13>
Story Name: <H2P>

Description

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

how to play the game>.

Confirmation

- 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.

Requirements

- Can I easily navigate to the “How to play” page from the start menu?
- The language is well adapted to the player
- The instructions contain enough information such that a new player can easily start playing the game without difficulties understanding controllers etc
- The “how to play” description is easy to understand.

 # Health potions and coins

Story Identifier: <14>
Story Name: <Coins and Potions>

Description

As a player, I need drops from enemies, health potions and coins so that I can feel rewarded when killing an enemy.

Confirmation

- Enemies Sometimes drop coins or potions when you kill them.

Requirements

- I have a chance to receive a coin or a health potion when killing enemies.
- The spawn-location should either be random or an average passable position based on enemy positions.
- The drops have a set chance to drop when killed, i.e. 30%.

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.

6. The Program can be run with maven.

2.3 User interface

Here follows early sketches of the game.

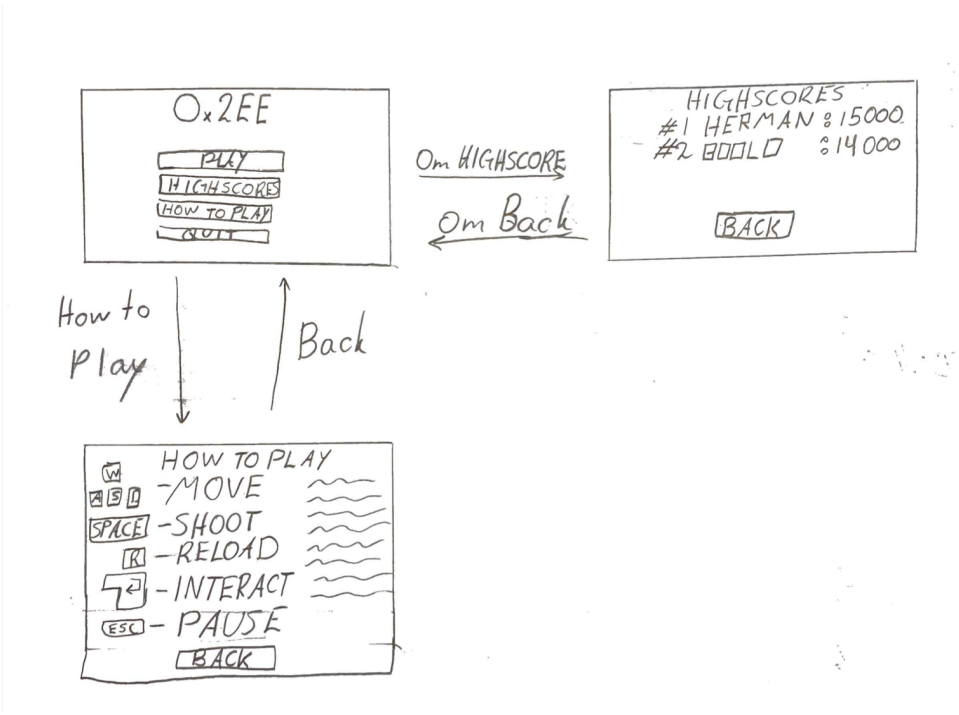

Figure 1: Main menu, Highscores & How to play

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.

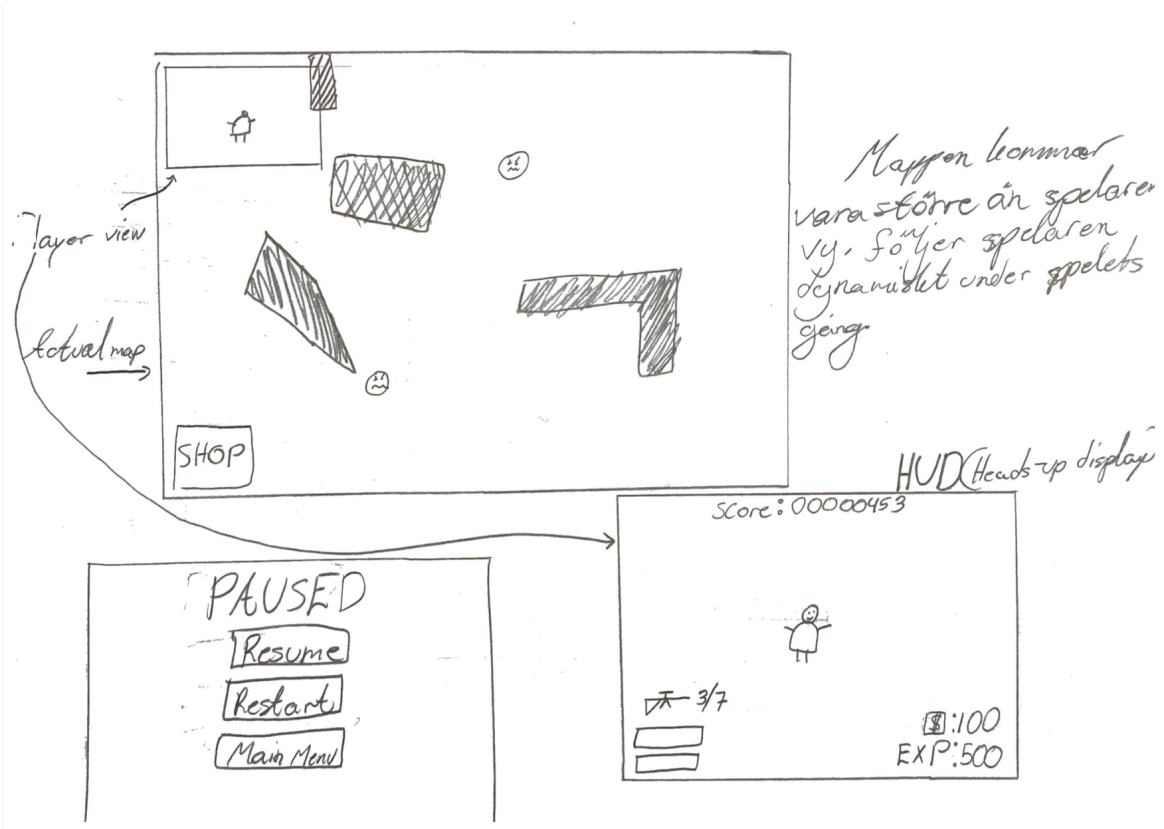

Figure 2: In game & Pause menu

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.

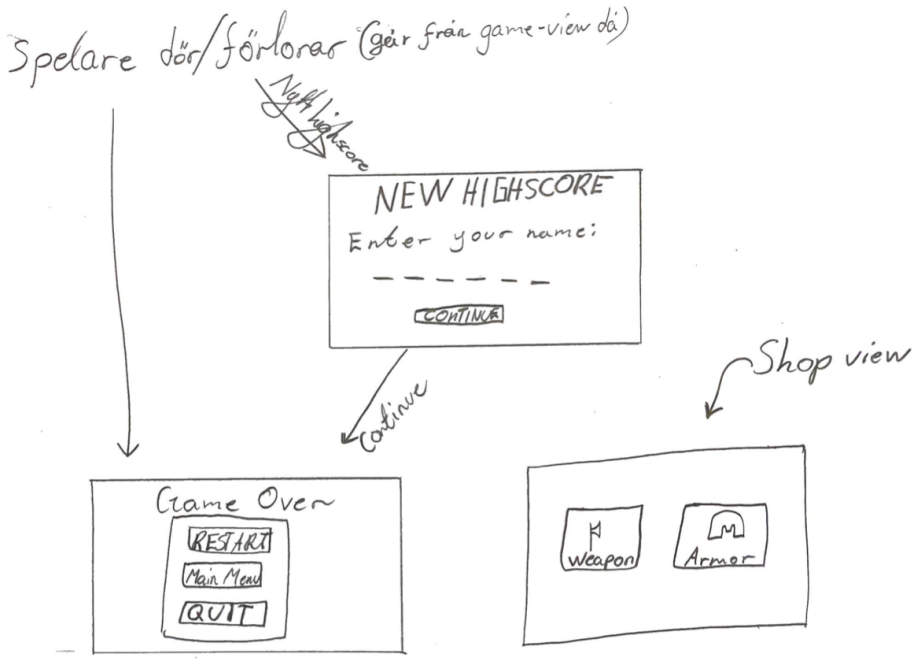

Figure 3: New Highscore, Shop, End

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 and 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.

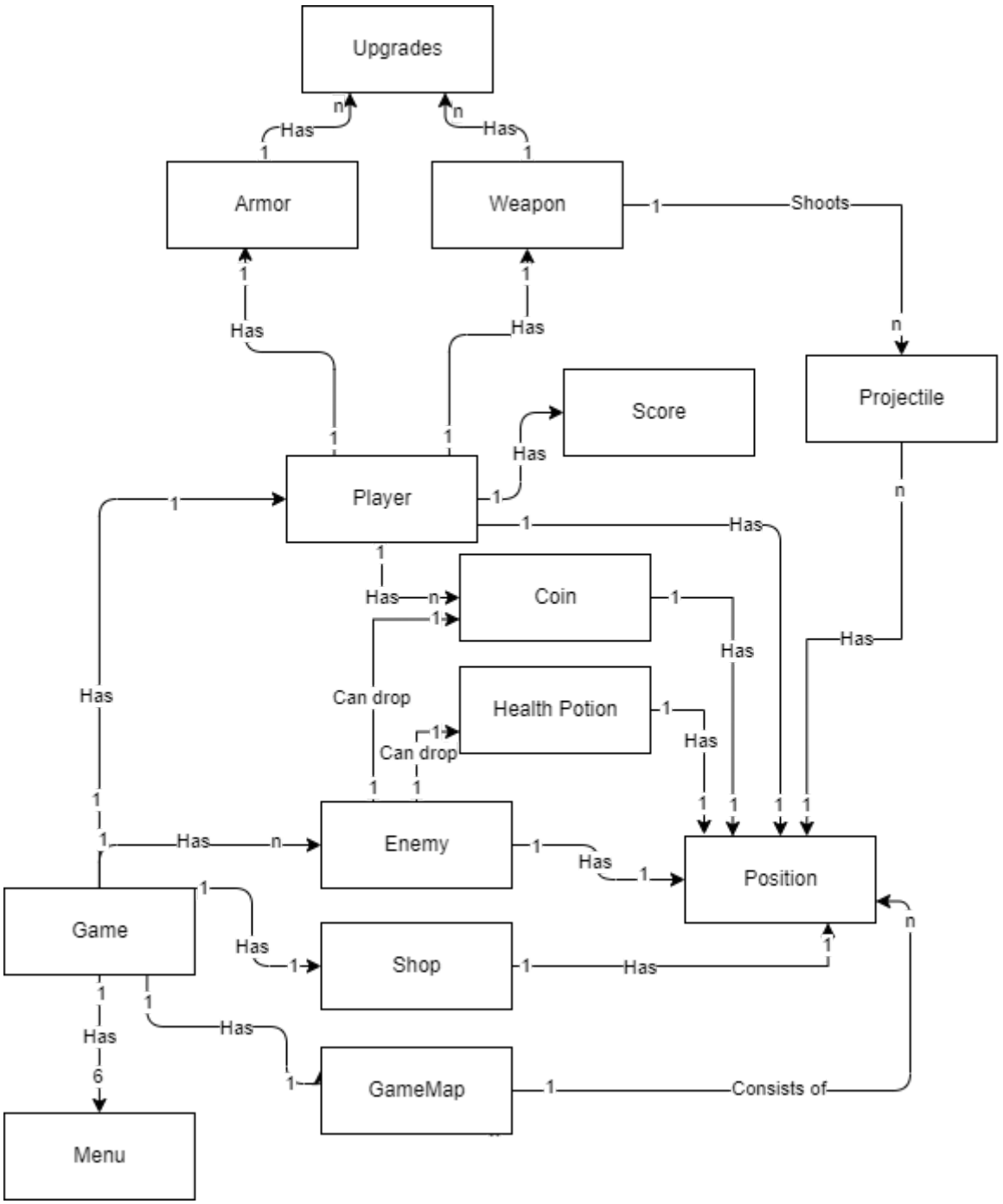

Figure 4: The domain model describing the project.

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.

Player - This class handles all information related to the actual player of the game. It's 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 regarding its own 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 . It is 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 shop is responsible for handling transactions between the player and upgradable items.

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

Position - Coordinates on the map an object is placed at.

Score - Amount of points the player has gathered throughout the game.

Potion - Add heal to the player's health when the player intersects with the object of this class.

Coins - Updates the player's balance when interacting with an object of this class.

4 References

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