

RAD

Project Cosmic Shooter

Team 30

Anton Andersson, Jonatan Arvidsson, Adrian Danielsson,
Linn Lindgren, Simon Bergström

Contents

1. Introduction.....	3
1.1 Project	
2. User Stories.....	4
2.1 Epic User Stories	
2.2 User Stories	
3. Domain Model.....	5
4. Figma.....	6
4.1 Menu	
4.2 Settings	
4.3 Cosmic Shooter	
4.4 Bottom Panel	
4.5 Pause	
4.6 Game Over	

1. Introduction

1.1 Project

The first draft all members agreed on in the group, of the object oriented assignment, is a game set in space inspired by the arcade classic Asteroids. Concisely, the game is called Cosmic Shooter, and it is an infinite game where the player, which is a spaceship, can move around freely on the frame in every direction, trying to avoid and eliminate enemies with its projectile guns. The most basic enemy also have projectile guns, and they will when entering a certain radius try to hunt down the player.

The player earns points to a total score every 30 seconds it has survived without dying from the enemies projectiles that damage the player if hit. The score also advances when an enemy is eliminated by the player, and the points earned differ depending on the enemies difficulty level. For example, except those basic enemies with guns, there will be an harmless enemy, called junk, which has no projectile guns, but just floats on the screen with a certain speed linearly.

To help the player from being eliminated, there will be power-ups floating on the board, free for the player to use. When flying on a power-up it will be collected in the players inventory, ready to be used whenever needed. This is a way for the player to survive longer during critical moments in the game.

Time will be the deciding factor when the game advances, meaning it successively becomes more difficult. When the player's health drops to zero, it is game over. The game is now over and the final score is shown along with the player's high score on the screen.

2. User Stories

2.1 Epic User Stories

As a member of today's society I would like any game I play to be fun and interesting!	As a user I would like a game that challenges me, so it keeps me in action and interested.
--	--

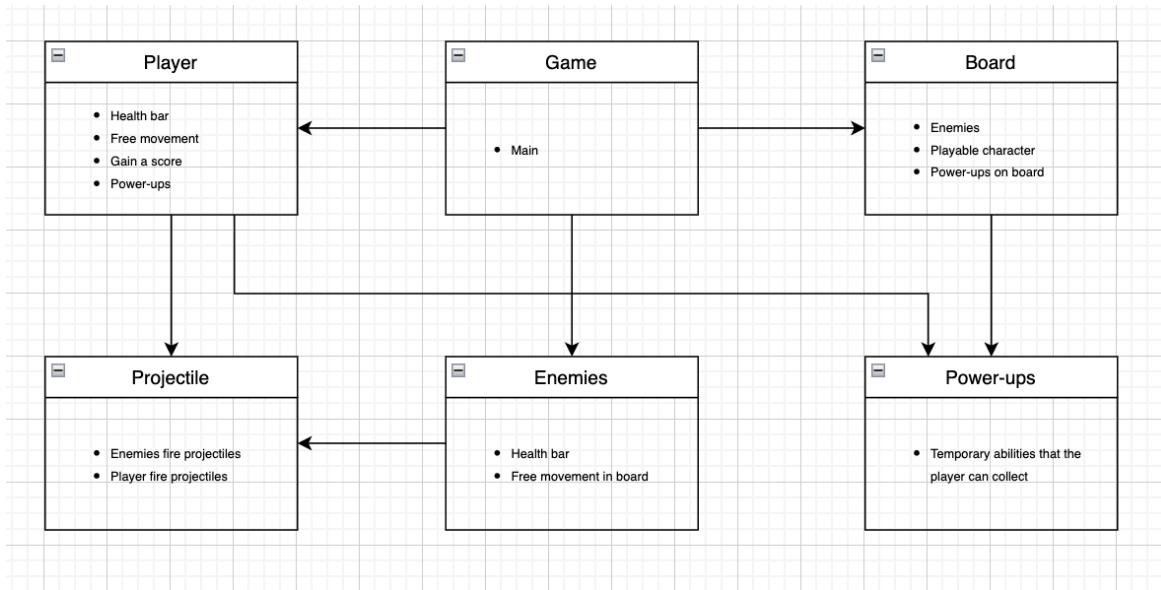
The project, Cosmic Shooter, implements these common wants by being both functionally fun and visually interesting. The game has variety to keep the user interested, the longer playtime without being eliminated unlocks new and more enemies, which also results in a greater challenge for the player. This keeps the user active and interested, and also the implemented score in Cosmic Shooter creates a want of trying again, so the player has a chance of getting a higher highscore. The power-ups are a tool to create a sense of strategy in the game, which will function as an alternative way of engaging the users, that also adds a unique and interesting touch to the game.

2.2 User Stories

As an industrial engineering student I want a clear UI so I don't have any misconceptions about what I can and can't do in the game!	As a boomer I want a clear UI to be able to understand the keybinds more easily.	As a web developing student I want more text in the UI to understand more clearly what things are on the bottom bar.
--	--	--

The UI of Cosmic Shooter consists of a board with a score up in the right corner, and a bar at the bottom which itself consists of the players health bar on the left side, and the four possible power-ups at the right. This is the current implementation after many changes. The design is clear and does not have unnecessary details that can confuse or mislead the user, also the power-ups slots are faded when empty, to make it more clear when they are not in use. At the top left corner above each power-up there is a small text that shows which keybind to use when activating the power-ups, to prevent the user from guessing how to activate each power-up.

3. Domain Model



The domain model does not show many details of how the project is structured, but it communicates the main idea of how Cosmic Shooter is supposed to work. Very briefly, the Game class contains main and is connected to the Player class, Enemies class and Board class. The board is connected to the Power-ups class because there are power-ups randomly on the board, as described in earlier paragraphs. The player class is also connected to the same Power-ups class because the player is the only one that can collect the items. Also, as described in earlier paragraphs, both player and enemies have access to projectiles that they can use to eliminate each other, and therefore their classes are as well connected to the Projectile class.

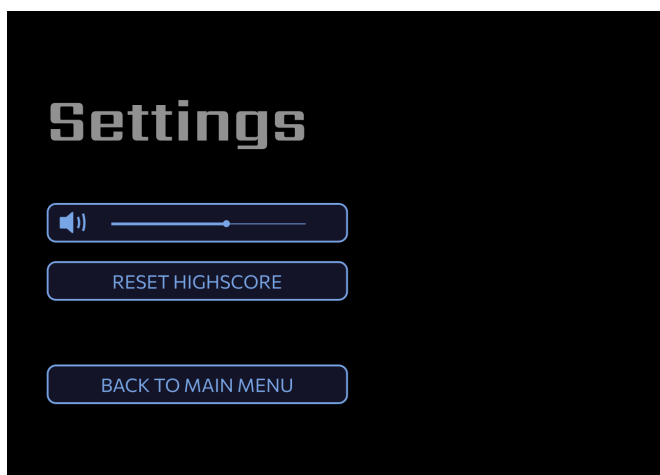
4. Figma

4.1 Menu



The start page is shown when starting the game. There are three options, start a new game, go to settings and quit the game. The user's highscore is also shown down below at the bottom right corner.

4.2 Settings



The settings are quite standard for a small-scaled game. The sound can be changed and it is also possible to reset the highscore.

4.3 Cosmic Shooter



The green spaceship is the player, whereas the grey spaceship is one of the basic enemies. The bolt is on the other hand the static harmless enemy called junk, which is more thoroughly described in the introduction.

Power-ups on the screen are surrounded by bubbles, where the given effect of the power-up is shown inside the bubble.

4.4 Bottom Panel



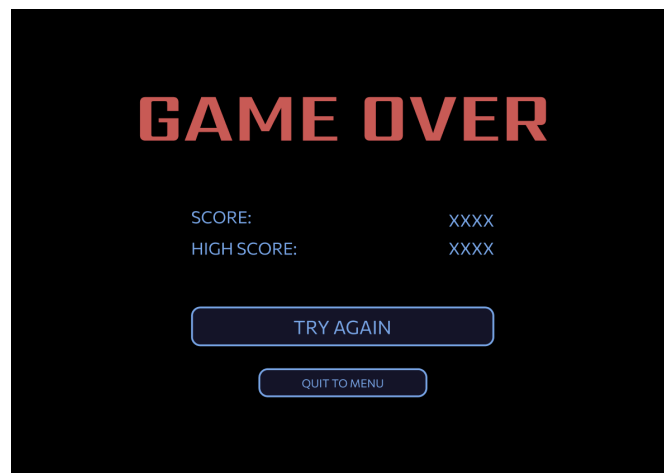
The player's health is shown at the left side, and on the right side the power-ups are shown. When a player collects a power-up, it will go from the faded state to the bright state, to clearly show which power-ups are in the players inventory and therefore possible to activate. More information about the design choices is found in the user stories paragraph.

4.5 Pause



The pause screen is a transparent overlapping screen with a text “pause” with two buttons with two possible choices, resume the game and quit.

4.6 Game Over



When the player's health bar drops to zero it is game over. On the game over screen, the users score is shown and their highscore. At this point, there's only two choices, try again or quit.