CS 319 - Object-Oriented Software Engineering

Final Report

Sea Adventures

Group 2-G

Alper Kağan Kayalı

Büşra Oğuzoğlu

Kasymbek Tashbaev

Salih Zeki Okur

Table of content

1. Introduction

1.1. Implementation status of the first iteration

The implementation process started right after the Design report of the first iteration was finished. We decided to implement our project with IntelliJ IDEA. Each of us downloaded IntelliJ and connected to our main GitHub depository named 2.G-Sea Adventures-CS319 Project, which was created by Alper at the beginning of the course.

After we finished Design report of the first implementation we divided the tasks among each other such that Alper was going to implement User Interface and some control objects of Game Management: GameEngine, CollisionManager, Salih - the rest of the Game Management: SoundEngine, SettingManager, Kasymbek - Game Entities: GameObject, Submarine, Enemy, etc., and Busra was going to implement other classes that left: FileManager, SkillManager, etc. Until Project Iteration 1 Demo, we already have implemented the entity objects and some parts of the Game Management component.

1.2. Implementation status of the second iteration

Since we implemented half of the game until Project Iteration 1 Demo, we decided to add a new functional requirement - new game mode in the second iteration, it was going to be implemented by Alper. Furthermore, everyone continued to implement the part of the game that he or she was assigned for. We could finish all of our promised functional requirements and the additional functional requirement on time.

2. Design Changes during the implementation

2.1. New classes

In the design report, we did not have Experience class because originally experience was going to be a numeric attribute of Submarine class and will not be displayed on the screen. However, during the implementation, we decided to make a new class Experience that has its image attributes that are displayed to the player, so now the player can see the current level and experience of the submarine.

2.2. Removed classes

In the design report, we had CooldownManager class that will control cooldown times of the skills. However, during the implementation, in order to reduce the system's complexity, we removed CooldownManager class and added its functions to the Skill class.

2.3. Improved classes

According to the design report, Skill class was going to extend GameObject. However, during the implementation, we noticed that Skill class uses only a few functions of the GameObject, so we decided to use Delegation design pattern in Skill class, so now Skill has an instance of GameObject, instead of extending it.

During the implementation, we added some constant attributes to the most of the game entity classes, so making changes in that classes was easier.

There are minor changes in most of the classes, such as some new secondary functions in some classes, changes in the number or type of parameters of the function, new attributes.

3. Lessons Learnt

The most important things that we got from this project are experiences of doing software engineering, working with the team and communication with the clients. During this project, we felt what it's like to be a software engineer. The first, we learn how to decide what kind of product we need to create by asking each other what he or she wants to see in the game and according to our preferences we chose the features that our game will have. After our decision on the product, we learned how to plan our implementation of that product. In general, we learned how to apply our knowledge of computer science in solving a real-life problem.

In this project, we also experienced working in a team with ones who we did not know before. This helped us to work in a more professional manner. We have set meeting times and everyone tried to adjust their program to have these meetings every week. We learned that this helps the project be done in a more easy way. We learned how to write the reports in the way that the one who is not familiar with your code could easily understand it by reading the reports. We also learned some non-academic skills that are needed to the engineers, such as following the deadlines, discipline,

Furthermore, we experienced communicating with the clients, by getting feedback from the instructor and from the feedback that we get we improved our reports and implementation.

4. User's Guide

4.1. System Requirements

The game is able to operate on any basic computer with a Java Runtime Environment and an operating system, so only thing required to run Sea Adventures on the computer is to have Java Virtual Machine and Java SDK. As hardware aspect, a basic keyboard and mouse is required to play the game.

4.2. User's Install Guide

Firstly, if the user has no Java Virtual Machine and Java SDK installed in the computer, he needs to download it from: https://java.com/en/download and install it

User should download the game's GitHub folder from https://github.com/alperkagankayali/2G.SeaAdventures and "open /src/main.java" directory to launch the game. In addition, the user can learn about the development process of the game reading reports in "/doc" directory.

4.3. How to use

a. How to Start

There are two modes, Scenario and Survival. In scenario mode, the player should complete three levels, in each level he has to reach the end of the level and kill the boss. In survival mode, there is no end of the level, enemies appear until the player loses the game, the goal is to get more scores. The player can start the game mone that he wants by clicking Play Scenario Game/Play Survival Mode in the main menu.

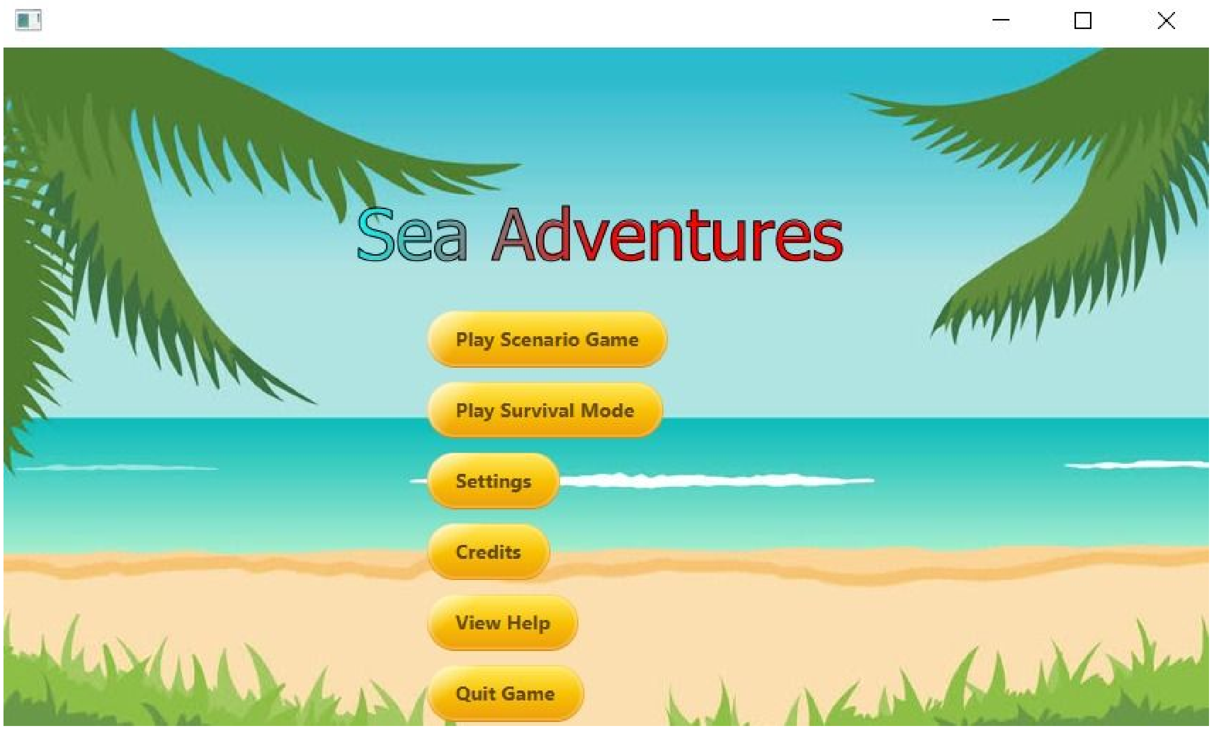


Figure 1: Screenshot of Main Menu

b. How to Play

Player controls submarine which can move up, right, down, left, shoot and use skill pressing the corresponding keys. The submarine has a level, 3 skills, health, and energy. It has to shoot the enemies and get experience and score by killing them. When he has enough experience, he will level up and restore health and energy to full, unlock new skills, update weapon. There are power-ups which restore energy or health when picked up by submarine. Submarine loses health when collides with enemy or bullet, and when the health is 0, the game is lost. The player can view help by clicking View Help button in the main or pause menu.

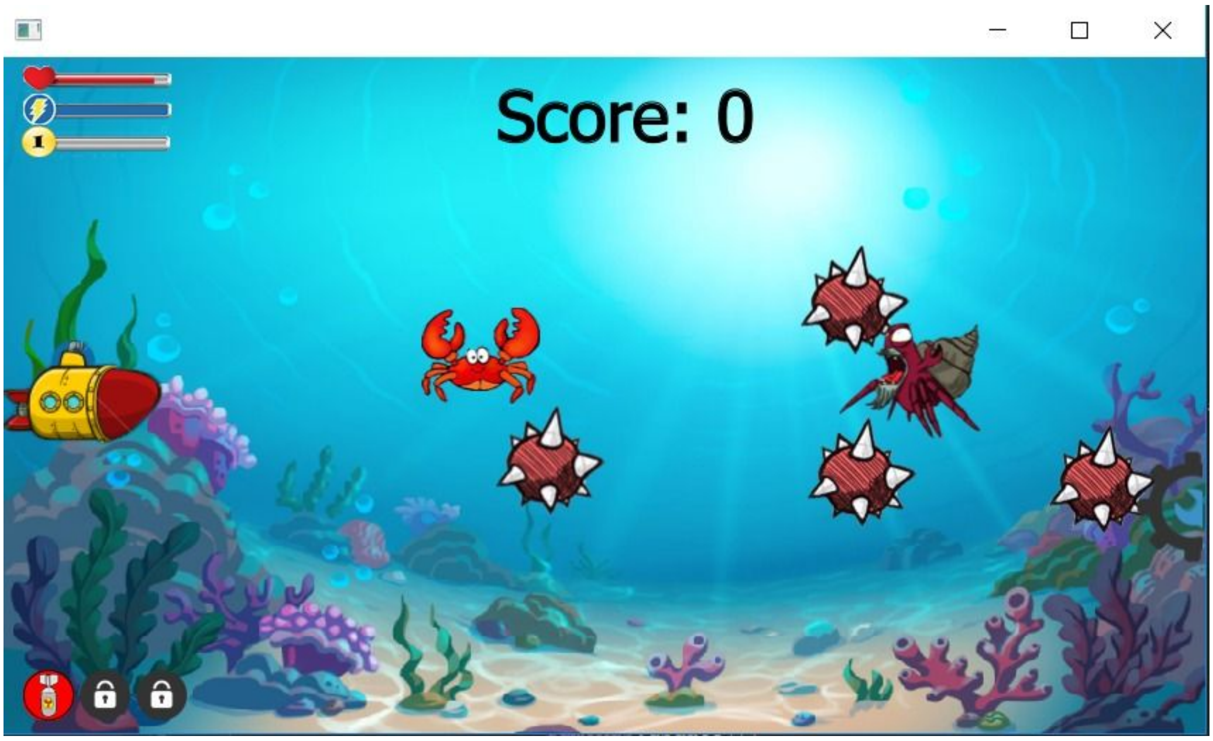
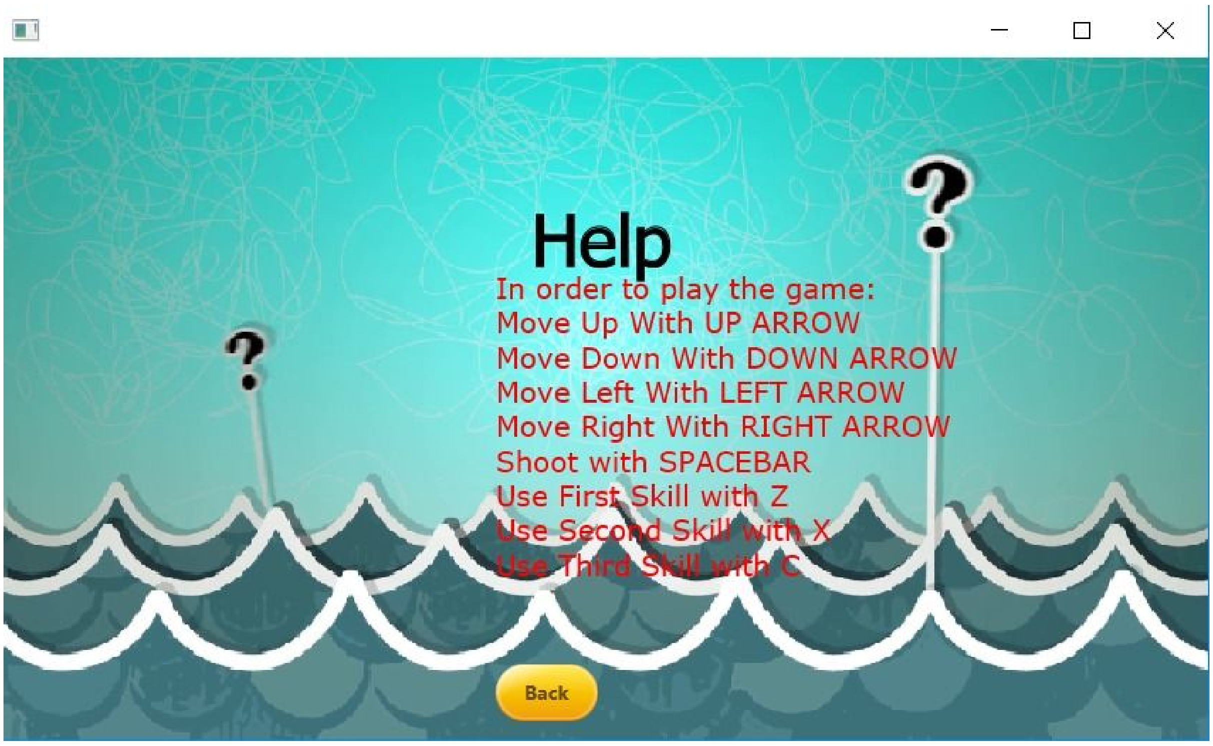


Figure 2: Screenshot of the Gameplay



Help
In order to play the game:
Move Up With UP ARROW
Move Down With DOWN ARROW
Move Left With LEFT ARROW
Move Right With RIGHT ARROW
Shoot with SPACEBAR
Use First Skill with Z
Use Second Skill with X
Use Third Skill with C

Figure 3: Screenshot of the View Help

c. Highscores

Player gets a score for killed enemies and picked up power-ups. He can learn the high scores list by opening View High Score on the main menu. In the list shown up to top 10 highest scores.

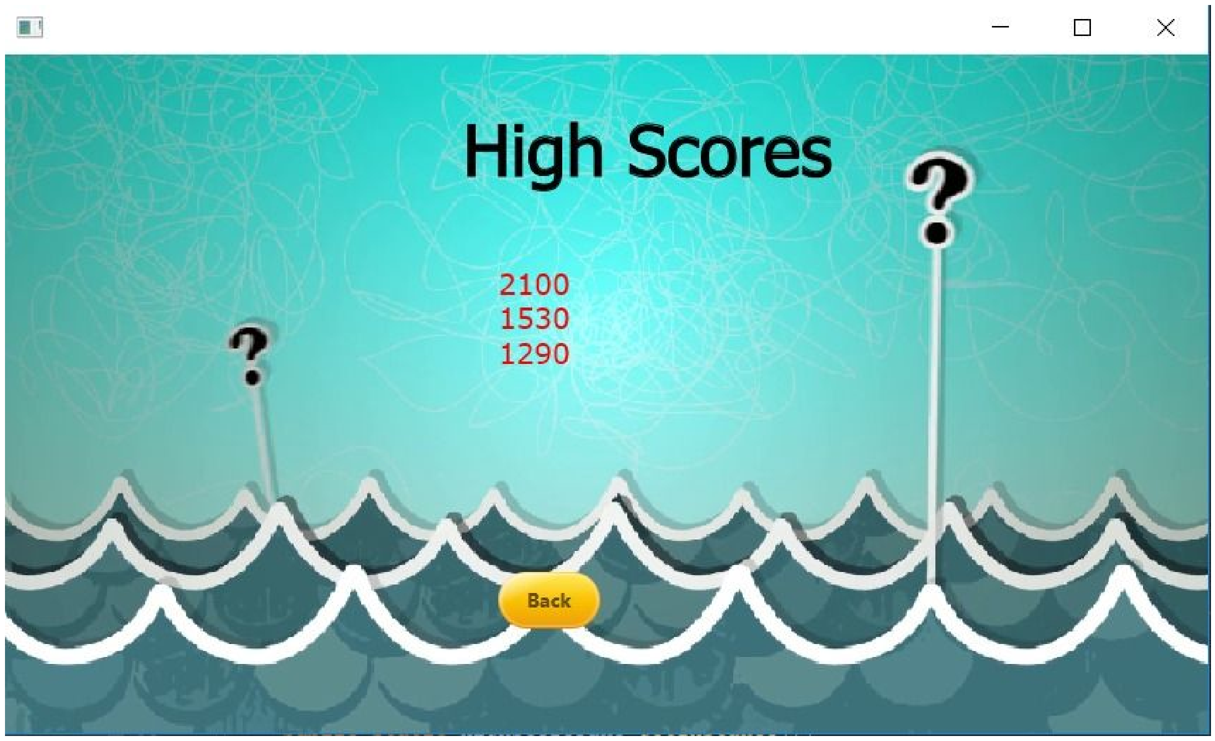


Figure 4: Screenshot of Highscores Screen
(In the screenshot there are only 3 scores because the game at that time was played only 3 times)

d. Settings

The player can open Settings Screen in main and pause menus. In the setting screen, he can change music or the volume of sounds and music. Also, the player can change controls in the settings screen.

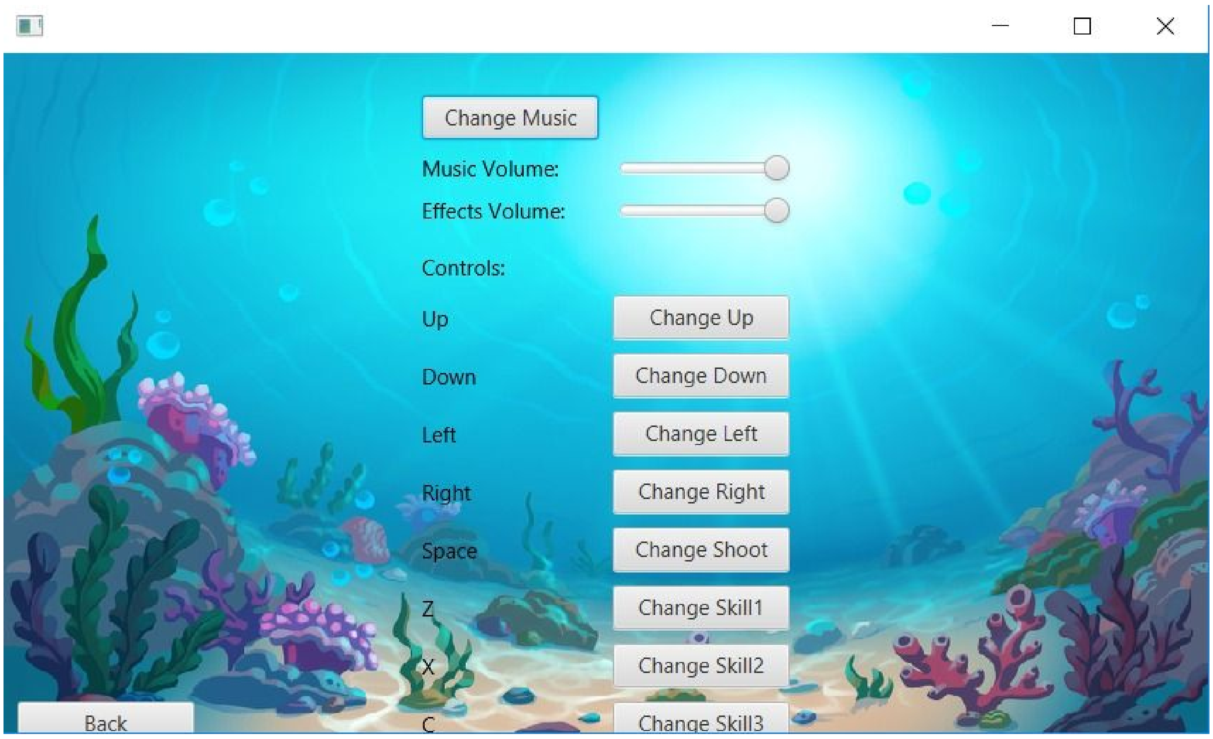
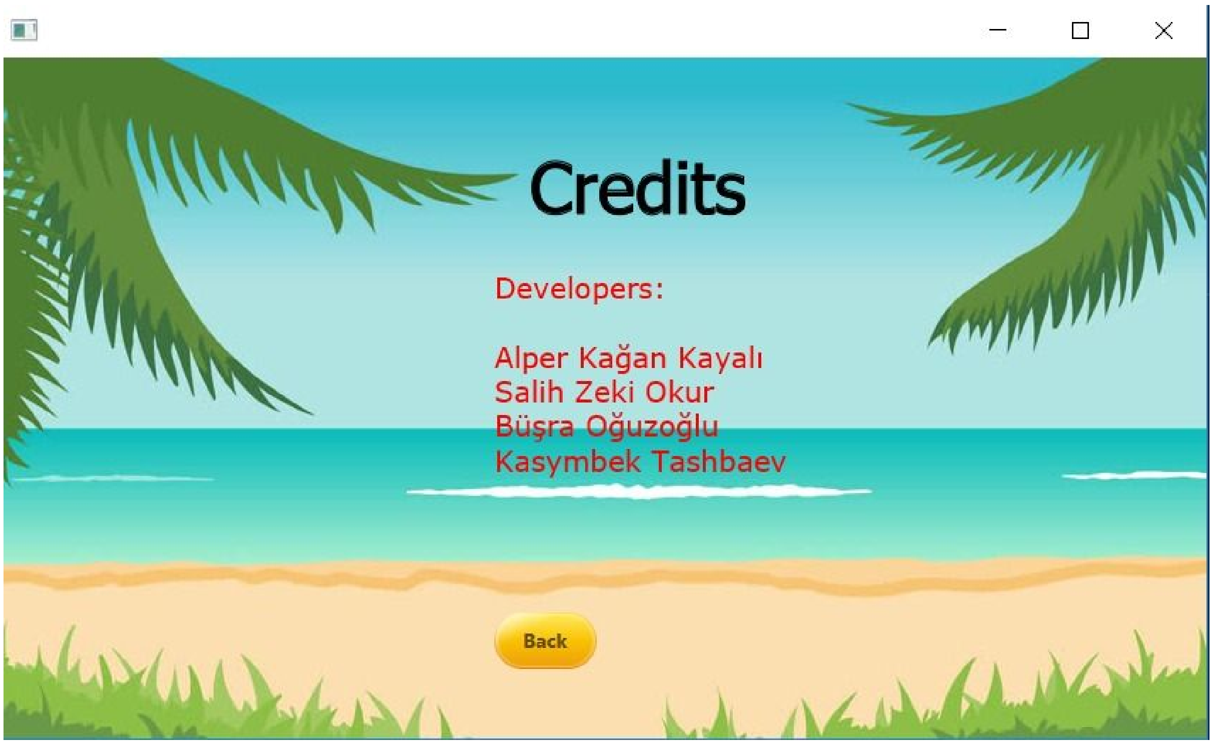


Figure 5: Screenshot of the Settings Screen

e. Credits

Player can view the information about the authors of the game by opening Credits Screen in the main menu.


Credits

Developers:

Alper Kağan Kayalı
Salih Zeki Okur
Büşra Oğuzoğlu
Kasymbek Tashbaev

Figure 6: Screenshot of the Credits Screen