  
  
  
  
CS 319 - Object-Oriented Software Engineering  
Final Report  
  
  
Project Name: Survivor

Group 2-G

Metehan Kaya - 21401258

Mehmet Taha Çetin - 21400281

Ferhat Karaca - 21400921

Emre Yiğit Kuzhan – 21502056

Table of Contents

[1. Changes in the Implementation 3](#_Toc497515165)

[1.1. Change in Design Pattern 3](#_Toc497515166)

[2. Status of the Project 3](#_Toc497515170)

[3. User’s Guide 4](#_Toc497515171)

[3.1. System Requirements 4](#_Toc497515175)

[3.2. Installation 4](#_Toc497515176)

[3.3. Overview of the Game 4](#_Toc497515177)

[3.4. Game Entities Subsystem 5](#_Toc497515178)

[3.4.1 Game Objects 5](#_Toc497515179)

[3.4.2 Controls 5](#_Toc497515180)

[3.4.3 Game Screenshots 5](#_Toc497515181)

# **Changes in the Implementation**

### **Change in Design Pattern**

We intended to create design pattern through using three-tier architecture. Since we thought that the most efficient and suitable design pattern is three-tier, we started to design the project by using this architecture. However, we realized that model view controller design architecture is more efficient than three-tier design pattern and easy to implement it. The first layer of the three-tier architecture was presentation tier and it was responsible for the interface between the user and game. In our new architecture which is MVC (model-view-controller), view is responsible for the interface between user and game. The user interface of new architecture consist of scoreboard menu, help menu, credits menu, level menu, main menu and start game. The second layer of our new design is Game logic. It consist of three classes. These are user, map manager and sound manager. This layer of this project called as controller and it is responsible for the interaction between game and user. The last layer our MVC architecture of our project is components. This layer is responsible for the data of game component. Component layer consist of game objects which are animal, enemy, player, obstacle, objects classes. According to our design architecture last layer called as model. The last layer of our design manages the behavior of data application domain. MVC design pattern provide us to flexible implementation in the project. The main alteration of our implementation is that we decide to change our design architecture three-tier to MVC after the first iteration of the project.

# **Status of the Project**

We have successfully created the object of player. Player object can move left, right, up and down with acceleration. Also we created simple obstacles. Player should not crush these obstacles in order to stay alive. We handle the collisions for all the maps. There are different types of obstacles in each environment. For instance, we used birds as obstacle of atmosphere environment and also we used fishes as obstacle in the ocean map. There are 4 environments which are space, ocean, earth, and atmosphere in the Survivor game. Each environment has three levels according to the degree of difficulty one to five. We completed all these environments and levels. Each environment has different movement conditions. Since there is no gravity in the space map, it’s difficult to control the player the space environment. We also add the teleport as a new feature to the project. Player can teleport from one point to another. There are two specific location to realize this purpose. We used one of point as an entrance of teleport. The other point used as exit of teleport. Also, we finished main menu of the survivor game completely. There are 4 option in the main menu. We completed all of these options which are credits, helps, high scores and environments (play game). Other thing that we handle is the scoreboard of the game. We keep the score of player in the all environments using this scoreboard in the local disk of computer.

# **3. User’s Guide**



## **System Requirements**

Survivor is a Java based game. Therefore, Java run environment should be installed to the computer in order to play the game

**Minimum System Requirements:**

- Windows XP

- Pentium2 233 MHz CPU or higher.

- 256 MB of RAM or higher

- Screen resolution: 1080x1920

**Recommended System Requirements:**

- Windows 10

- Intel i5 2 GHz CPU or higher

- 1 GB of RAM or higher.

- Screen Resolution: 1080x1920

## **Installation**

The only way that play the Survivor is compiling and running the code that we wrote with Java IDE

## **Overview of the Game**

If player chooses the play option from the main menu, the game starts. There are 4 different maps in the game. Each map has three levels according to the difficulty of the game. In each map we there are different type of obstacles and player object. Player should not crash these obstacles in all environments according to the rules of game. Player can move around the map using with the directions buttons of the keyboard. Also player object moves with acceleration according to the frequency of input that read from the user.

## **Game Entities Subsystem**

## **Game Objects**

**Player:** The main object of the game. User controls this object using through the directions buttons of the keyboard.

**Obstacles:** Different type of obstacles created in our project.

## **Controls**

Up button: move north

Down button: move south

Right button: move east

Left button: move west

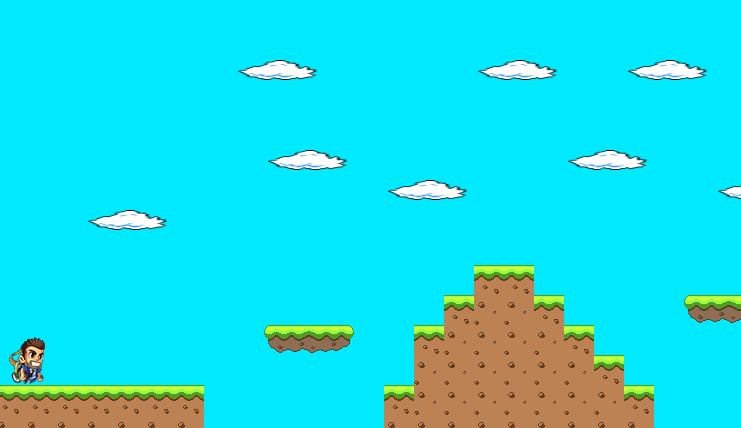
## **Game Screenshots**

****

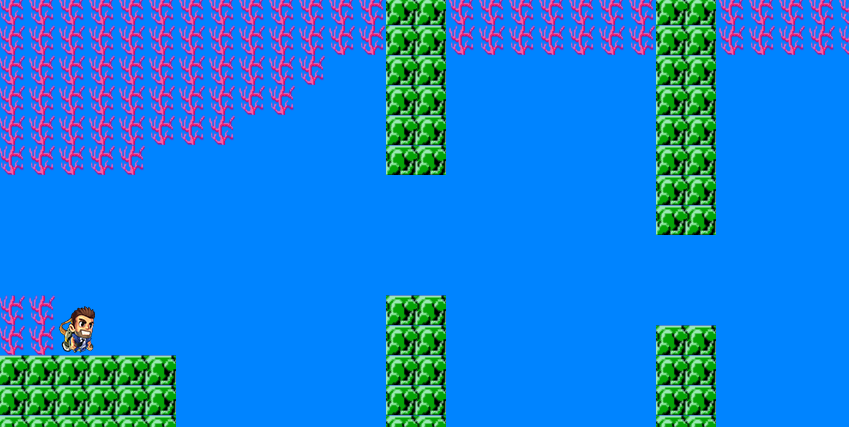
**Main Menu:** When user runs the game, this menu will appear. In this menu the user can choose play game, view help, view credits, and exit game.

**We have 4 different maps**

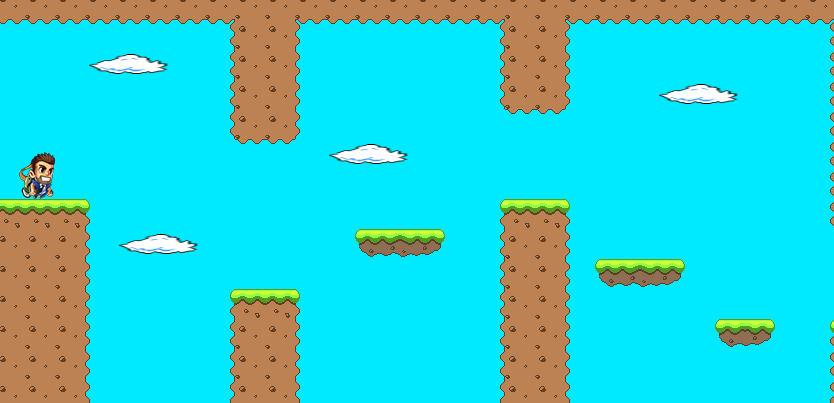
**Earth:**



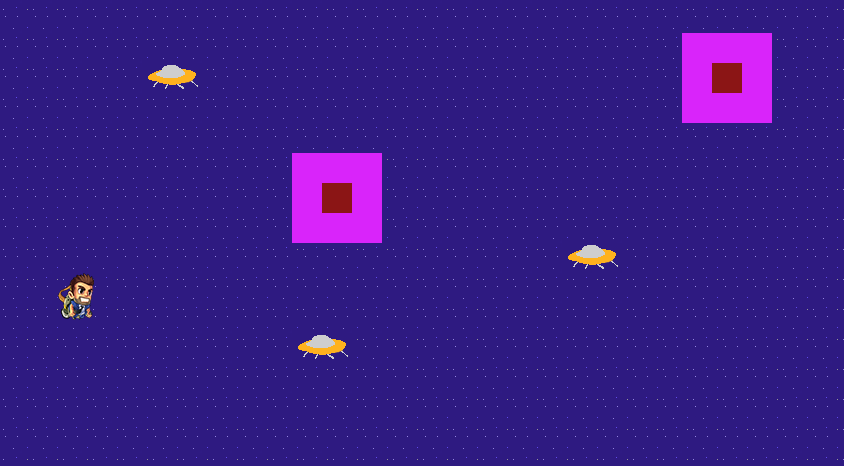
**Water:**

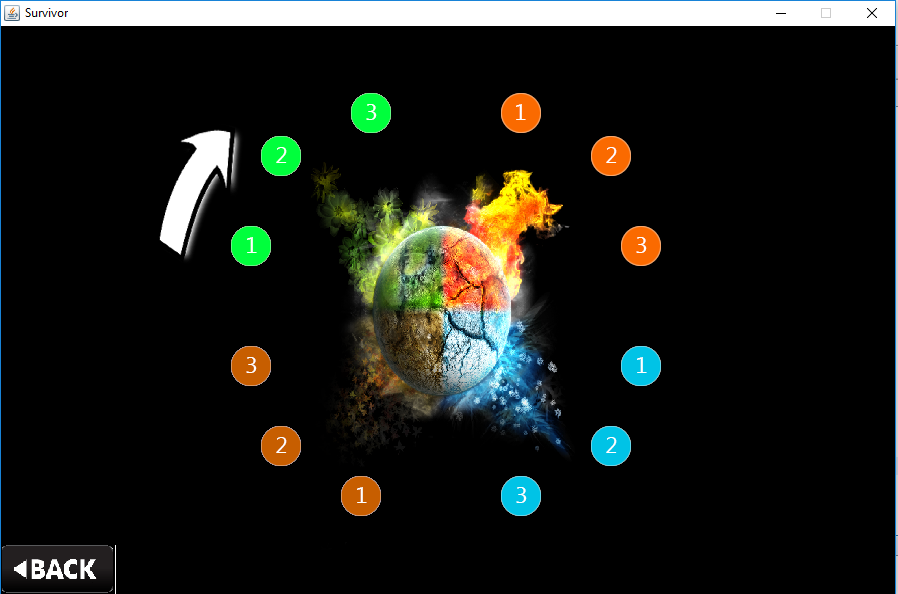


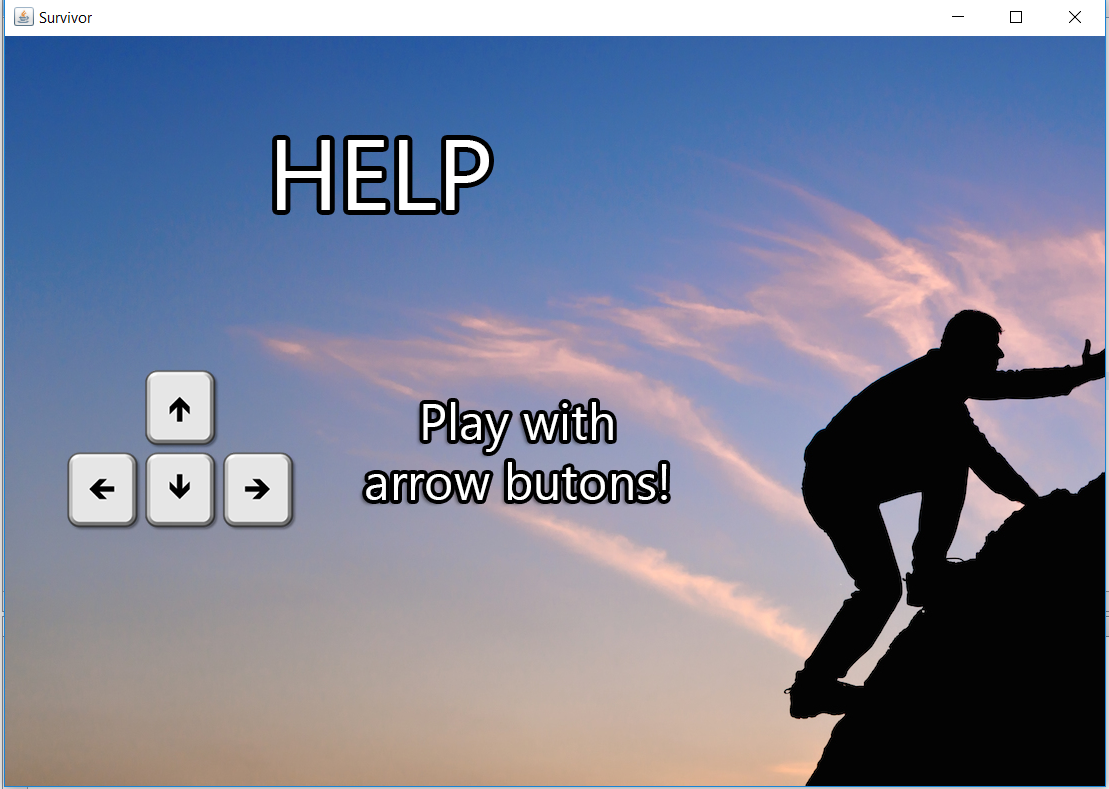
**Air:**



**Space:**



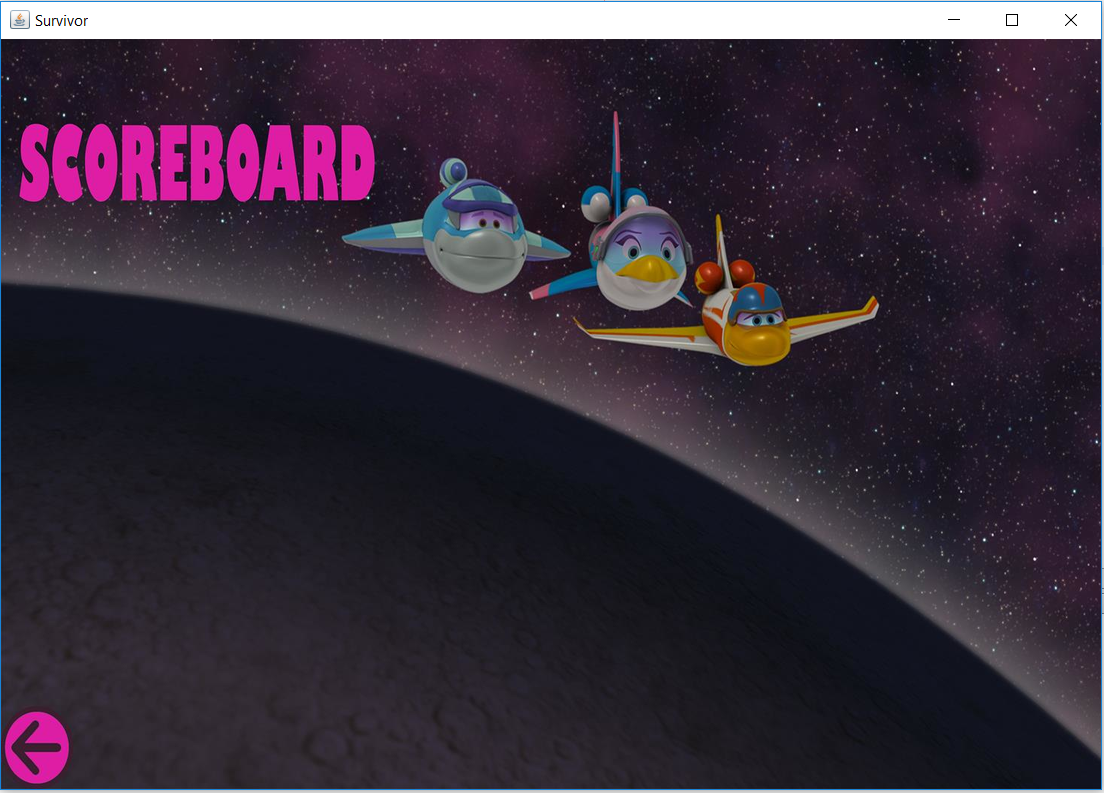
**Play Game:**  When player choose this option, game starts.

****

**View Help:** If the user chooses help, some useful contents about the game and instructions are shown

****

**View Credits:** If the player selects “Credits” button, names of project developers will be shown.

****

**High Scores:** View High Scores: When “High Scores” button is pressed, game will show the user scoreboard which contains top 10 scores with player names. Initially, the scoreboard has 10 player names and some smooth numbers as scores.