Object Oriented Software Engineering Project

Final Report

CS 319 Project: Saving Humanity

<u>Group 24</u>

Can Bayraktar
Erdem Karaosmanoğlu
Mert Sebahattin Kutluca
Oğuzhan Karakahya

Course Instructor: Uğur DOĞRUSÖZ

1. Introduction

Our project is a game named “Saving Humanity”. It is played from a top-down perspective and its maps are in form of maze. We have influenced from the old game called "Tank 1990" aka "Battle City". In this game the world has been invaded by aliens with their tanks and the only hope is you with your tank. Your mission is to kill all of the aliens and save the humanity. You will encounter different types of enemies and they will become stronger as you kill them. There are various powerups and bad powerups which are dropped by dead enemies. Also you can play it with your friends from the same keyboard. Once you finish all the levels, you will be the savior of the world.

2. Installations

Java SE Runtime Environment is needed to run this game properly. The setup of that can be found at: http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html

You can either run the executable to play the game or if you know how to code in Java, you can download the source code of the game and run it in an IDE like Eclipse, Netbeans, IntelliJ. You also need the data files which consists of sprites, .txt files and music files in a data folder which must be located next to the executable file.

3. User’s guide for the game

Once you open the executable file, you will see a menu with options:

3.1)Play SinglePlayer

Once you click this button, you will see a map with lots of different tiles, enemy tanks and your tank. The “golden” tank is your tank. You can control this tank using your keyboard. (Please refer to Help section for more information.) The other silver tanks are enemy tanks. You need to kill those enemy tanks to advance to the next level and win the game.

3.2)Play MultiPlayer

Once you click this button, you will see another screen where there is a choice box which you need to select how many players you will play the multiplayer mode. After that if you click “Play Deathmatch”, you will start a deathmatch mode where everyone is an enemy to each other. If you click “Play Team Deathmatch”, you will start a team deathmatch mode where there are two teams competing with each other. Players controlling Tank1 and Tank2 are going to be Team1, players controlling Tank3 and Tank4 are going to be Team2. “Back” button takes you back to the Main Menu.

3.3)View Highscore

This button will show you the top ten highscores achieved in the game.

3.4)Help

This button will get you a screen which shows you the controls of the tanks in both singleplayer and multiplayer. “Back” button takes you back to the Main Menu.

3.5)Exit

This button is to going to close the game.

4. Some Changes to the Design During Implementation Phase

1- First of all, there are some methods that the design documentation does not include. For our implementation to be short and readable, we decided to add some private methods to the project. Also, we have found some useless methods and removed them from our project. Also, there are excluded variables as well as some new variables added to the project.

2- We have decided to move the location of collision handler method from GameObject to GameEngine to simplify the process.

3- In the analysis and design report, we said that the game would run at 120 fps. However, we have decided to make the game fps independent. Therefore, the game does not rely on constant fps. As always, the game is running smoother with higher fps.

4-To make the game more user and eye friendly, we have decided to add a new Interface called Animation which applies to specific game objects like water tile.

5- We said that the level designs for all the levels would be different from each other, however because of reasons related with time we made all of the map designs same.

6-We said that, we would create a boolean array for InputManager to hold the pressed keys. During the implementation, we are faced with some complications and we have decided to go with a simpler but stronger implementation which uses keyPressed and keyReleased methods to solve the input problem.

7-We have also added a music to the background which is not explained in the design report.

5. Parts which are not implemented yet

Team Deathmatch implementation is not finished yet.

HighScore calculation and viewing highscore from the menu is not finished yet.

6. References

Sprites are taken from:
https://www.spriters-resource.com/nes/batcity/sheet/60016/

8-bit music is taken from:
https://www.youtube.com/watch?v=Xqdtu-Wyl-U