**Bilkent Univeristy**

# *Computer Engineering*

# *CS319 – Section 2*

# *Object Oriented Software Engineering Section TA: Gulden Olgun Supervisor: Bora Güng*ö*ren Implementation Report: Final Draft*

*DECEMBER 16th, 2017*

# *Bubble Popper A Project By Group 2-I*

***Serhat Hakkı Akdag***

***Orkun Alpar***

***Mustafa Mert Aşkaroğlu***

***Faaiz Ul Haque***

Implementation Report

# Implementation Approach

Initially we used our design report as a reference in creating the classes of our project. We had weekly meetings to construct an overview of our implementation and then began to code the classes individually. From our design report we continued with the MVC model.

We chose the Eclipse IDE to code on due to its ease and convenience in terms of usage. Since we worked separately on different classes we used EGit which is a plug-in for Eclipse that assists us to pull, push and commit files to our repository in Github easily. For the graphical user interface implementation, we are using Swing and AWT, which are Java libraries. After this we will continue to work on classes individually that have been divided among our group and will use the EGit to help keep the latest updated version amongst each group member.

For making menu interactions, we used CardLayout which allowed us to change between panels easily.

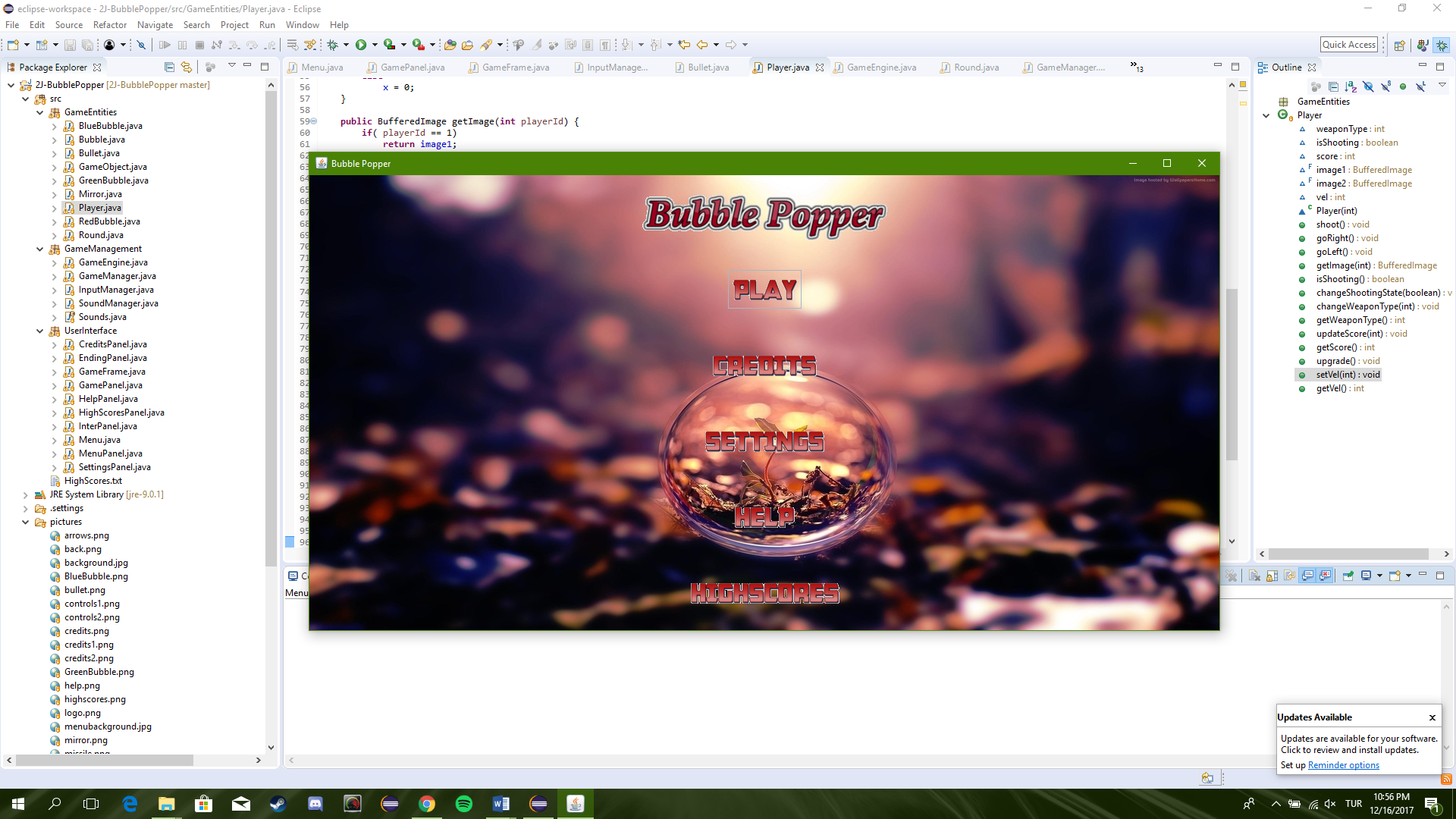
To implement collision handling methods in GameEngine, we used Java.awt.Rectangle. By doing so, we managed to handle collisions way easier and more reliable than the earlier implementations.

# Changes in Design

1. In our Analysis report we mentioned the usage of JavaFX for graphics. However, in our implementation we decided to use Java GUI libraries due to simplicity and ease of our application.
2. In our design report we put an InputManager class into GamManagement Subsystem. However, while implementing the game, we realized that it is better to use KeyBinding which was written in GamePanel. The reason for this was key listeners were losing their focuses while transitioning between panels. To fix this issue we used KeyBinding.
3. To show the messages and scores between rounds, we created a panel named InterPanel which informs players what score they have, what weapon will they use in the next round, and how many lives left.
4. In design report, there wasn’t something called menuPanel, because we implemented things without cardLayout and put buttons onto MainFrame which is Menu. However, after implementing CardLayout, we created menuPanel, which hold menu option buttons.
5. To use awt.Rectangle’s intersects methods, our game objects needed to hold width and height. Therefore these properties are added to gameObject class, which basically holds width and height of the game objects’ pictures’ dimensions.

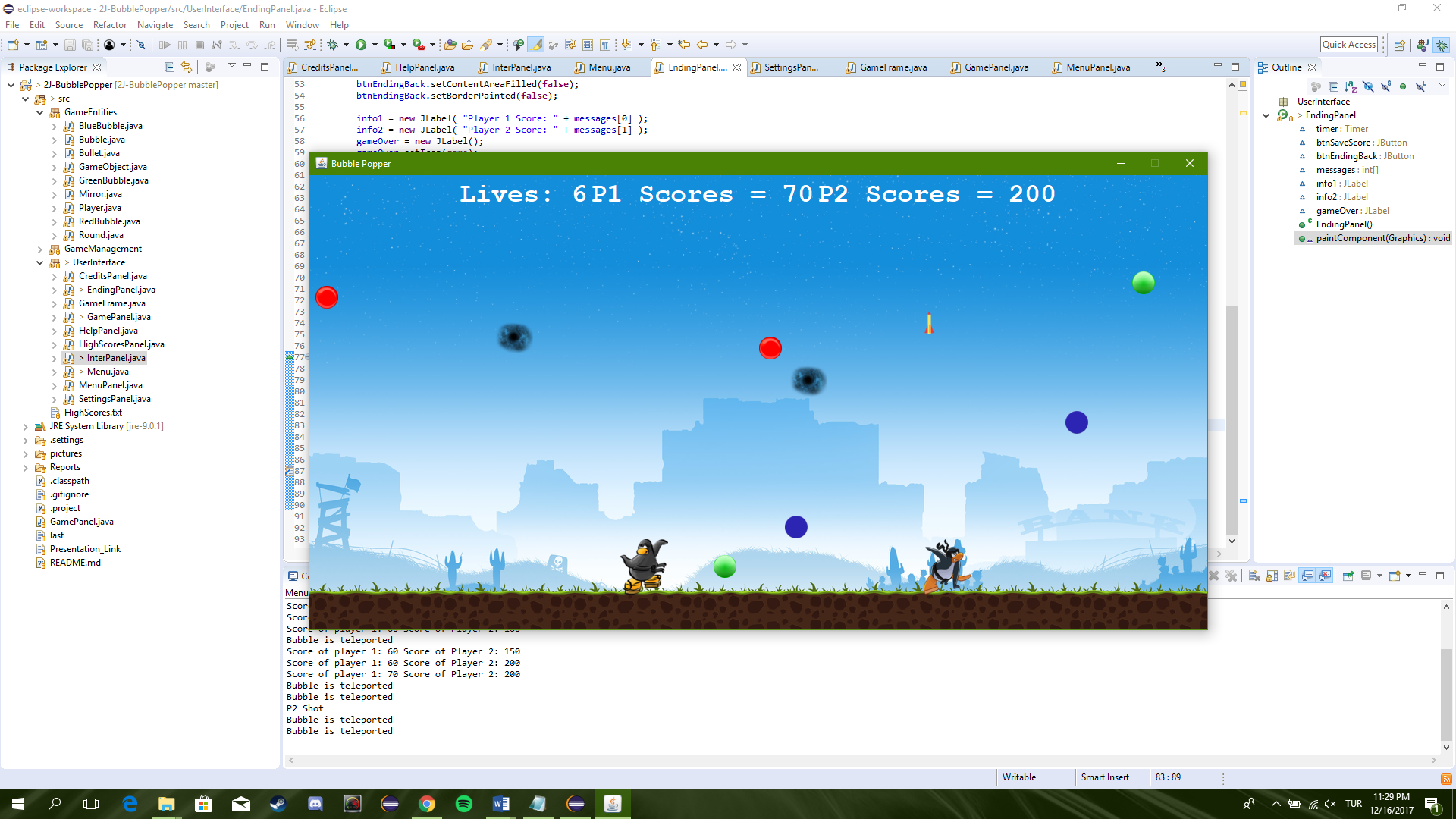
# User Navigational Guide

Menu contains usual interaction buttons.

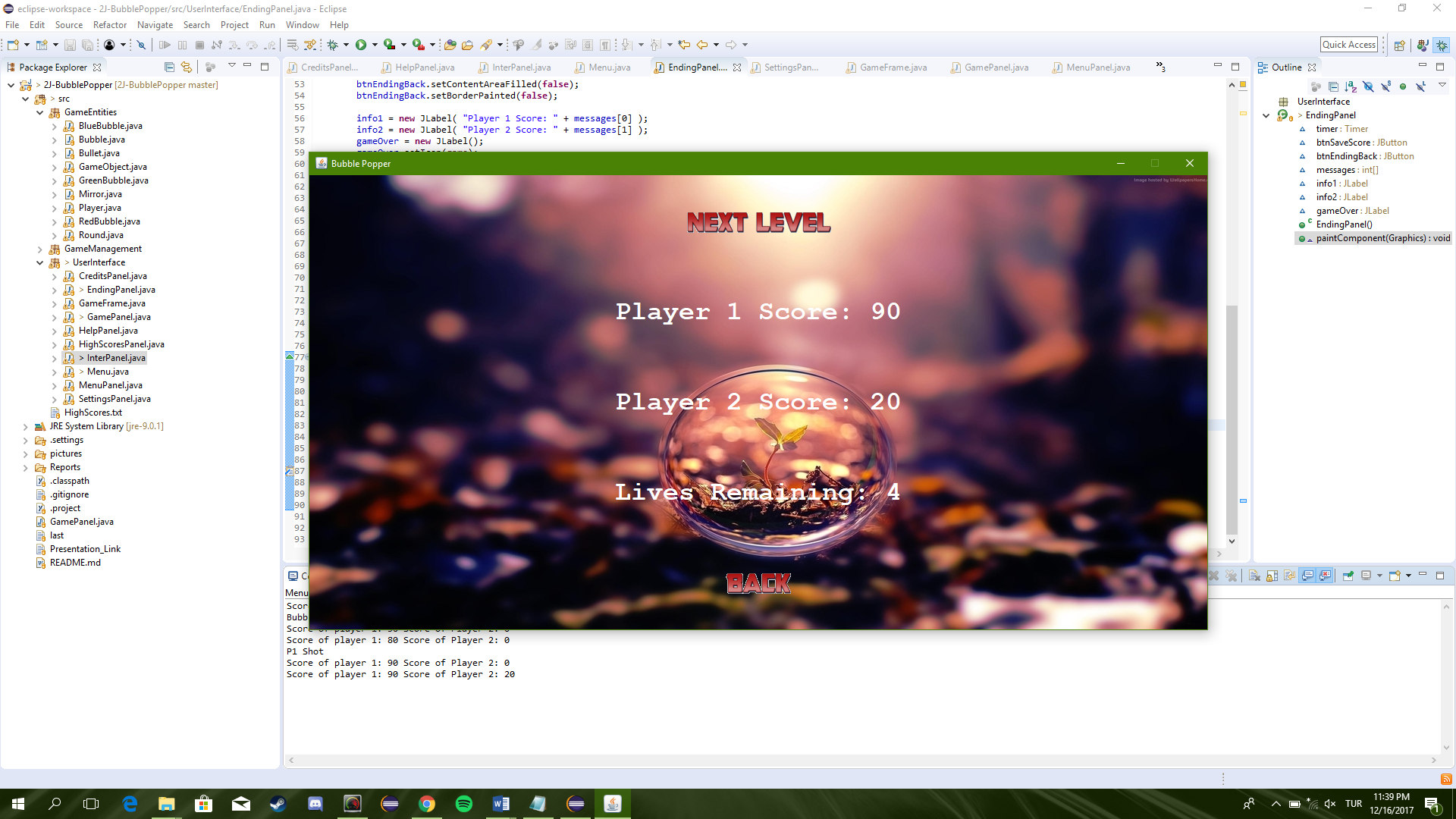


The panel shown below is used in actual playing of the game. According to round number bubbles and mirrors are created. In addition according to points of the players that they got in earlier rounds of game, their weapon type is changing making them easier to shoot bubbles. If a player has more than 100 points, he/she is informed between rounds and his/her weapon is upgraded.

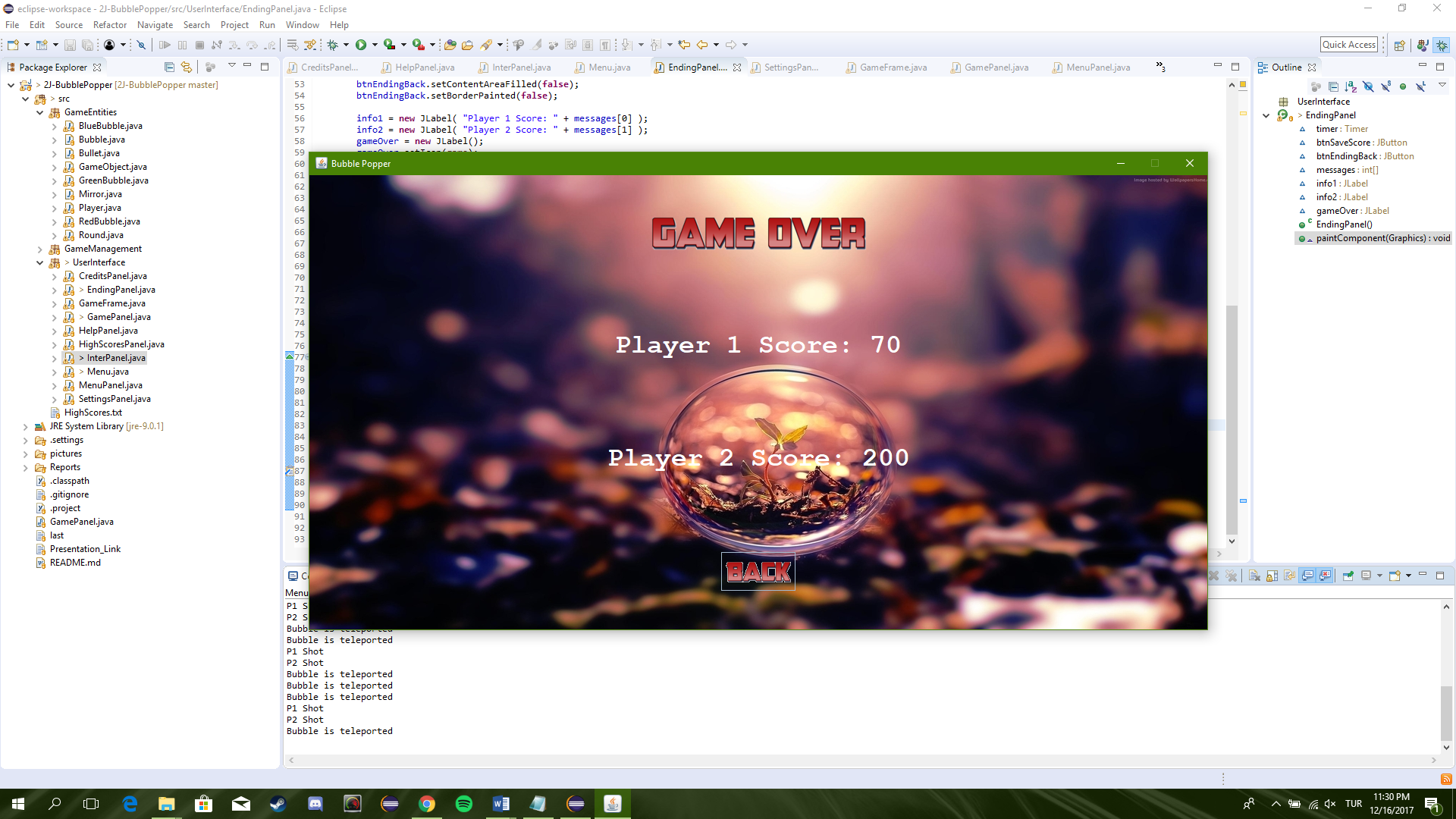
Mirrors are changed to black holes for making the game more fun. When bubble collides with a black hole, bubble is teleported to random location with random moving direction.



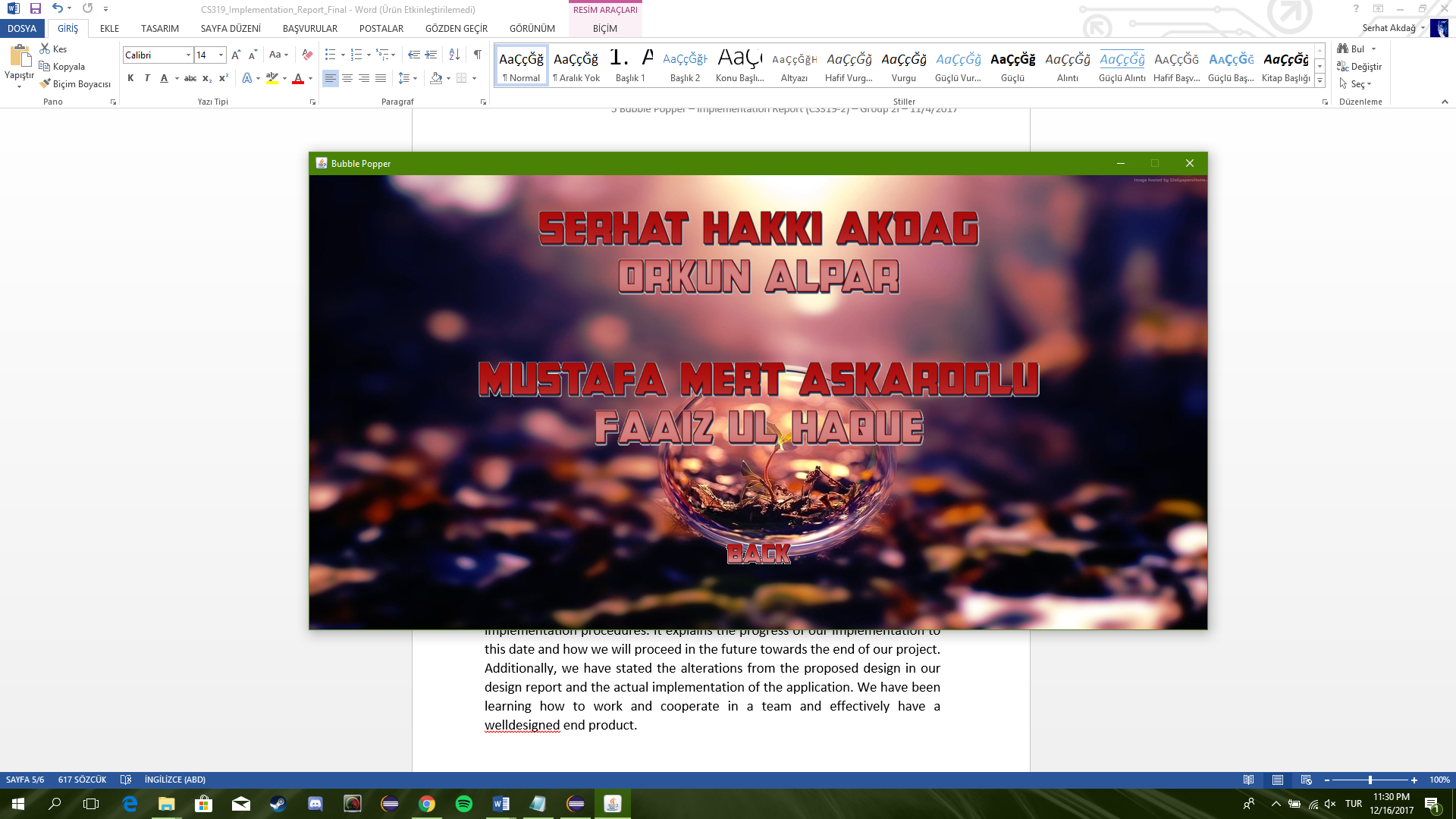
The panel below is prompted when players finish a round. From this panel players can view their current statistics including each player’s scores and their combined remaining lives. From here they can navigate through to the next level or return back to the main menu.



The panel below is prompted when the game is finish which occurs either when the players lose all their lives or complete the 10 rounds successfully. The panel shows the players final scores and also allows navigation back to the main menu. The scores will be updated to the highscore list if eligible.



This is the credits panel which shows the names of the contributors to this project.



The help panel displays the controls for each user according to keyboard.



# Conclusions

This report is intended to explain how the implementation worked out with respect to design and analysis reports. Additionally, we have stated the alterations from the proposed design in our design report and the actual implementation of the application. We have been learning how to work and cooperate in a team and effectively have a well-designed end product.