



Bilkent University

Cs 319

Object-Oriented Software Engineering Project Final Report (First Draft)

Section 2 / Group 2K

Ege Darçın

Alptuğ Albayrak

Özgün Özkan

Furkan Erdem

Instructor: Bora Güngören

Table Of Contents

1) Introduction

2) Installation

3) GamePlay

3.1) Main Menu

3.2) Player Select Menu

3.3) Settings Menu

3.4) Credits Menu

3.5) HowToPlay Menu

3.6) In Game Screen

4) Non Implemented Specifications and Changes

1.Introduction

Headball is a 2D arcade game. The game will basically start with the main menu which includes play, settings, credits, how to play and exit sections. Before starting the game, user is going to have an option to choose his/her favorite character to play. Main object of the game is to score against the other player and to make higher score than the other player when the given time is finished. When the game starts, there will be two goals on each sides and the football players will begin the game in front of their goals. A ball will appear in the middle of the field and each player is going to try to hit the ball to score. In order to make the game more enjoyable, there will be different modes (normal and the random mode) and powerups. There will be different power ups such as fast movement and fireball which will be able to be used by the player when the user triggers the related powerup via a key binded to it. The current score is always going to be displayed and updated on the provided scoreboard.

2. Installation

Java Runtime Environment has to be installed on the computer to run the game. By clicking the .jar extended file game will basically run and using jre, will give us opportunity to run the game on various operating systems.

3.GamePlay

Menus:

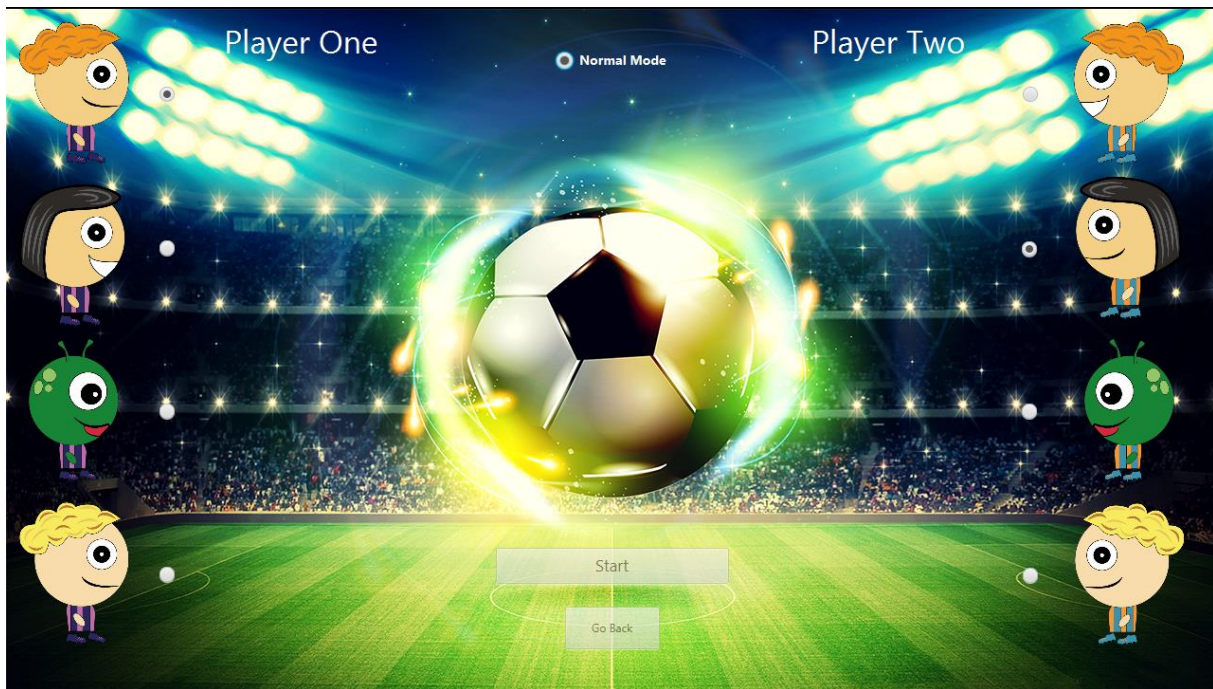
3.1 Main Menu:



Main menu consists of five buttons:

- **Play:** Opens the player select menu.
- **Settings:** User can change sound settings from this menu.
- **Credits:** Shows the names of the people who worked on the game.
- **How to Play:** Shows key bindings.
- **Exit:** Exits the game.

3.2 Player Select Menu:



Player select menu consists of two buttons and three radio buttons group.

- **Player One Select:** First user can select his character from these 4 different avatars. Default is the first avatar. This player's goal will be on the left side.
- **Player Two Select:** Second user can select his character from these 4 different avatars. Default is the second avatar. This player's goal will be on the right side.
- **Game Mode Select:** Players can select different game modes from here. Default is "Normal". Note: current implementation of the game only supports "Normal Mode".
- **Start:** Starts the game with the selected avatars and mode.
- **Go Back:** Returns to the previous menu (main menu).

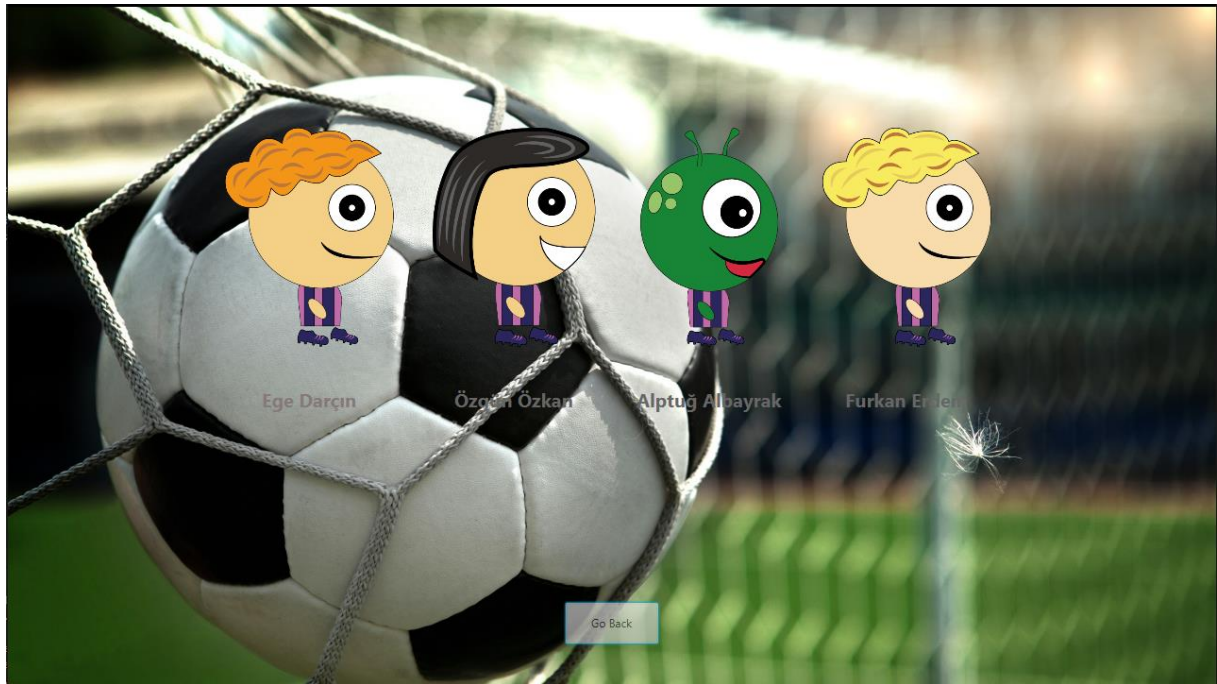
3.3 Settings Menu:



Settings menu consist of one checkbox and one button:

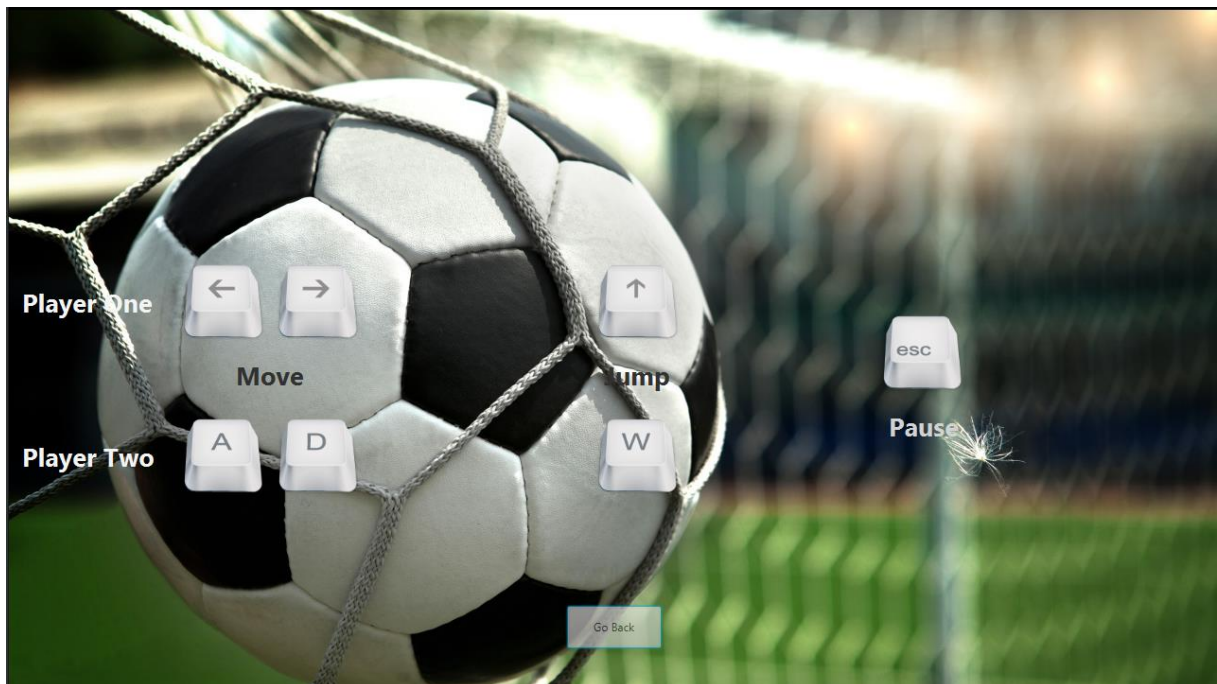
- **Sound:** User can turn on/off the sound of the game. On by default.
- **Go Back:** Returns to the previous menu (main menu).

3.4 Credits Menu:



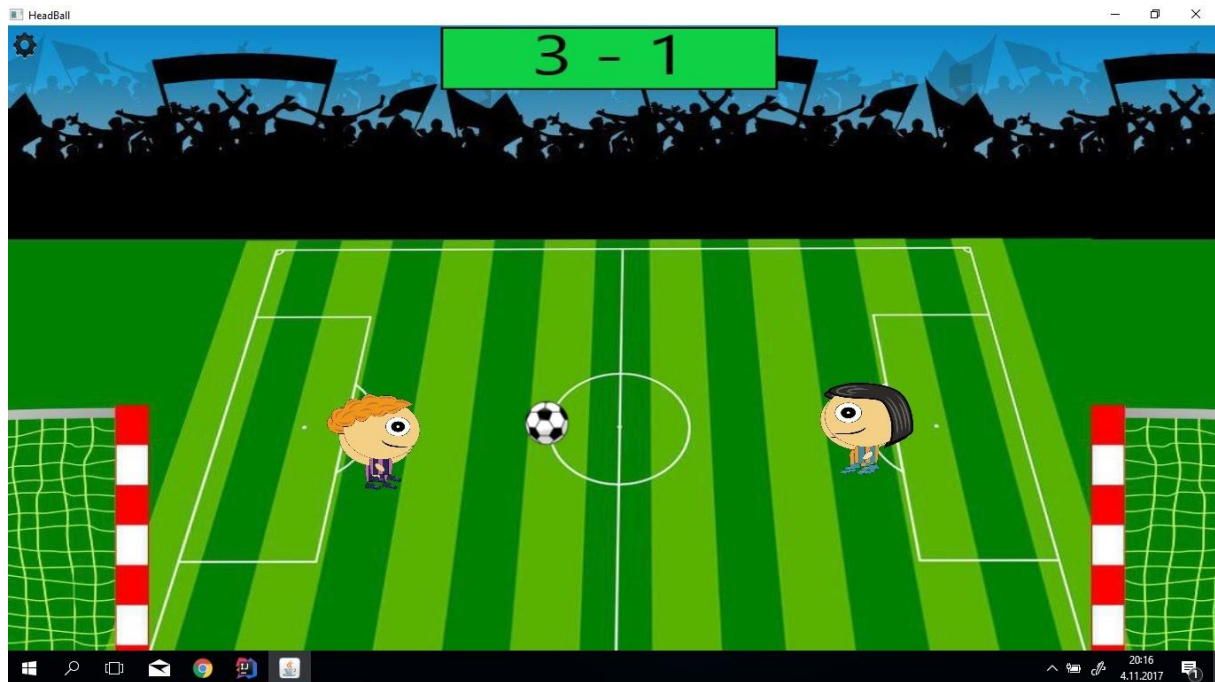
Credits menu consist of group member names and a “Go Back” button to return to the previous menu.

3.5 How to Play Menu:



How to play menu consist of key bindings of the game and a “Go Back” button to return to the previous menu.

3.6 In Game Screen:



In the game, players will use their headballers to score goals. With arrow keys and w,a and d, players command their headballers to score goals.

To the moment, no powerups are implemented and frame paint rate issue is at hand, but they are soon to be implemented/fixed.

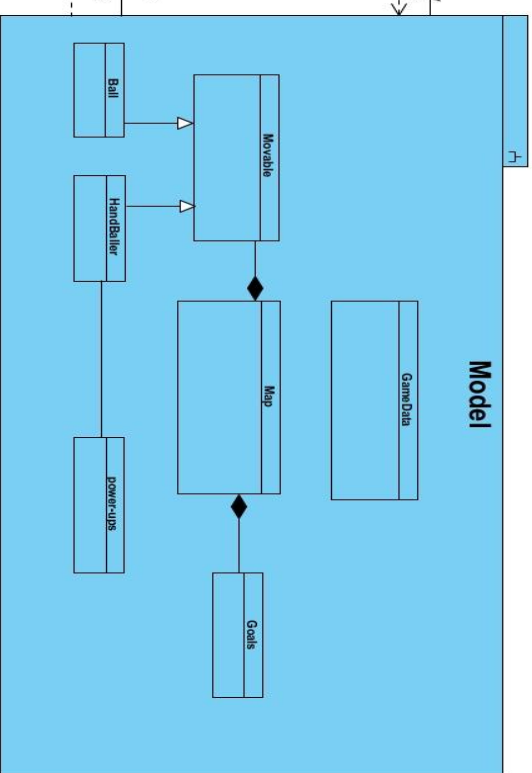
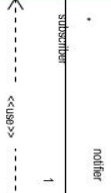
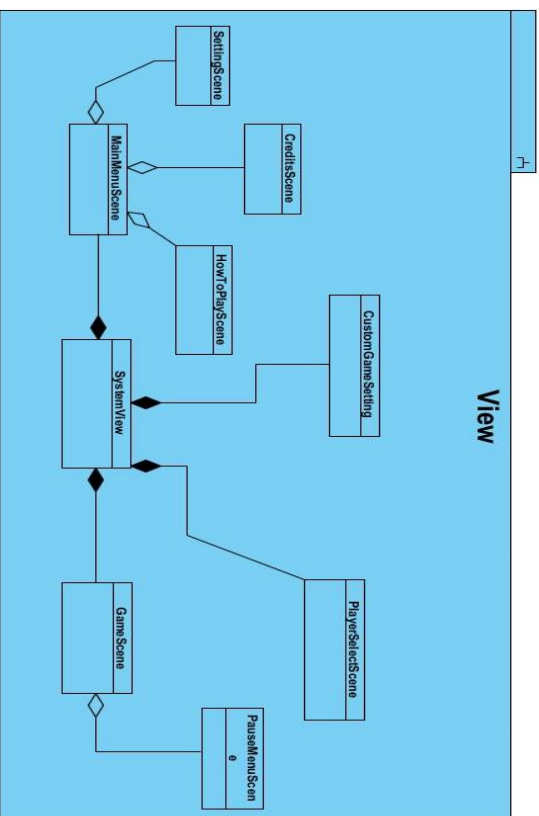
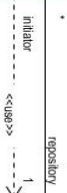
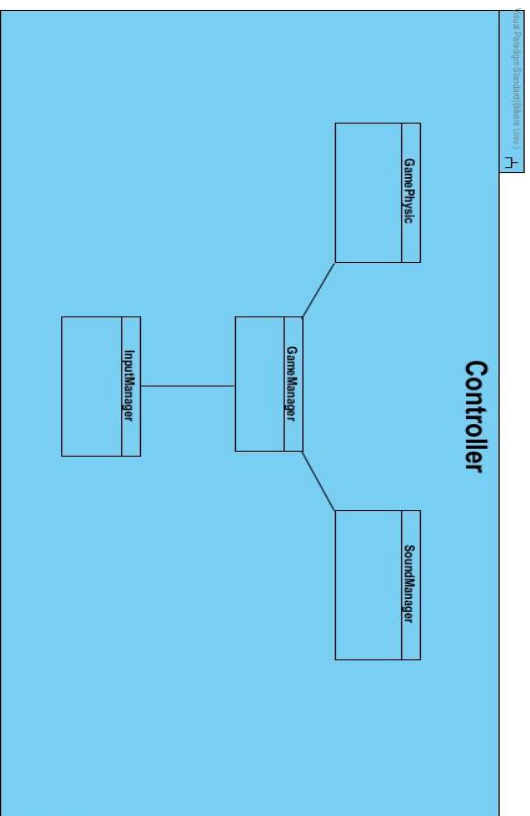
If players press escape or click the icon on the top left of the screen, options menu will pop up and bring out the pause menu.

4) Non Implemented Specifications and Changes

Changes: In terms of our implementation, we decide to use javafx GUI library to build our interfaces instead of swing library. Additionally, there were some changes to our UML

diagram. We renamed our View classes by changing part of “panel” to “scene”. We omitted “ImageManager” control class. We changed also structure and order of menus during process of initialization of the game. Naturally class diagram and use case are changed a bit. We reduce number of classes in model by removing “HowToPlay” and “Credits” and we created a new class “GameData”. This class holds all information related Game, such as help, score, credit and time information. We changed also relation between “Map” and “power-ups” classes. We don’t anymore define power-up as a map component. This class just manipulates physical attribution of ‘movable’ map component.

Non Implemented: We haven’t yet provided our games with the feature of power-ups and Custom game mode which allows user to choose several kinds of ball, stadium, and physical effect.





Controller

