Bilkent University

Department of Computer Engineering

Object-Oriented Software Engineering Term Project

CS319 Term Project: Hurdle Run

Analysis Report

Rabia Ecem Afacan, Güneş Batmaz, Haluk İncidelen, Selim Mıdıkoğlu

Supervisor: Prof. Dr. Uğur Doğrusöz

**Progress Report**

March 6, 2016**Contents**

[1. Introduction 1](#_Toc445060104)

[2. Current System 3](#_Toc445060105)

[3. Proposed System 3](#_Toc445060106)

[ Functional Requirements 3](#_Toc445060107)

[3.1.1. Play Game 3](#_Toc445060108)

[3.1.2. Change Settings 3](#_Toc445060109)

[3.1.3. Display Score 4](#_Toc445060110)

[3.1.4. Pause Game 4](#_Toc445060111)

[3.1.5. View Help 4](#_Toc445060112)

[ Non-Functional Requirements 4](#_Toc445060113)

[3.1.6. Game Performance 4](#_Toc445060114)

[3.1.7. Graphical Display 5](#_Toc445060115)

[3.1.8. User Interface 5](#_Toc445060116)

[ Pseudo Requirements 5](#_Toc445060117)

[4. System Models 6](#_Toc445060118)

[4.1.1. Scenarios 6](#_Toc445060119)

[4.1.2. Use-Case Model 1](#_Toc445060120)

[4.1.3. Object and Class Model 5](#_Toc445060121)

[4.1.4. Dynamic Models 5](#_Toc445060122)

[4.1.5. User Interface and Navigational Path 5](#_Toc445060123)

[5. Glossary 17](#_Toc445060124)

[6. Important decisions in overall Analysis 17](#_Toc445060125)

[7. Conclusion 17](#_Toc445060126)

[8. References 18](#_Toc445060127)

Analysis Report

CS319 Term Project: Hurdle Run

# Introduction

As the members of this group, we would like to introduce you the Hurdle Run. We decided to implement a game, and settled down in the idea of Hurdle Run as we would like to combine the simplicity in usage with the perfection and pure joy, just like we remember the games that we played in our childhood.

As it can be easily observed from its name, The Hurdle Run will be about running sprinter(s). It will include two different modes to play the game, single-player and multi-player. In single-player mode, the player will have opportunity to practice on one’s own. The game will be limited by the mistakes that player does. In multi-player mode, two players will be accepted, and they will play the game from one computer. They will run to win and when they make mistake, their velocity will decrease. Therefore, the one who does less mistakes will win.

This game will be desktop application compatible for Microsoft Windows, and will be controlled by the keyboard.

In this Analysis Report, we will make detailed analysis of this game, and attempt to create a well analysed game.

**2. Overview**

Hurdle Race is a android base game which has two different play modes. The enviroment of a game is olimpic running track with audiences on the background. Two play modes have different screens for running. In a single mode there is only one runway, however in a multiplayer mode number of runway is two for double player.

**2.1. Single**

Player Mode Hundle Race is a game in which player try to finish a game as soon as possible without hit the any obstacle. In a single player mode control of the sprinter is provided by a space key from the keyboard. The aim of the game is get a high score which is calculated by the time of run and number of obstacles that hit from the sprinter. At the end of the game score is displayed on the last screen of a game.

**2.2. Multiplayer**

Mode Like the single player one, in the multiplayer mode purpose of a game is finish the run as soon as possible without hit any obstacle. However there is a competition in this mode, thus before starting the game player can select the number of games for a deal. At the end of the deal scores of each runner and winner of the deal is displayed on the last screen. In multiplayer mode control of two sprinters are provided by space and up keys from the keyboard.

**2.3.Gameplay**

Playing state of the will work like we told in multiplayer and single mode parts. We will keep the score at the end of every single race than will calculate it according to the time the runner finishes the race and the number of the hurdles he hits. We will show the user the score. Additionally user will have chance to restart in the middle of the race or user will have chance to stop the game with continue option.

# Current System

We are inspired a horse hurdle game on internet. At that game there were only single player and the aim was jump the hurdle as much as possible. We thought that we may make a runners hurdle game. We can add multiplayer mode so that 2 player can play on the same PC.

The game that we are inspired can be found in: <http://www.agame.com/game/horse-jumping>

# Proposed System

In this section, the needed requirements are listed.

## Functional Requirements

### Play Game

Hurdle Race is a basic type of a java based game. The game has two different play modes. First one is single player with one sprinter, and the second mode is multiple players with two sprinters. In the single player mode, the aim of game is finish the run as soon as possible by hurdling different obstacles. Water and fences are types of obstacles which cause the losing time and points.

In the multiple-player mode two sprinters try to finish the run first. Likely single player mode different obstacles, water and fences, cause the losing time and points. Before starting the deal player have chance to choice number of plays. Hurdle Race is a good exercise to increase the eye hand coordination.

### Change Settings

     In the opening screen there are two choices to select the player mode:

-    Single Player

-    Multi Player

   Player is also able to turn off or turn on the sound of game while playing.

### Display Score

     At the end of each play in single mode, player can see the score he/she get. In multiple player mode winner of a deal, number of games gained from each player and scores get from each player can be seen in the screen after finish the deal.

### Pause Game

The game can be paused during gameplay and Player can continue game from where he paused. When game is paused, closing application will cause any game progress to be lost.

### View Help

Player is able to get any information about the game as an instructive explanation which includes the following:

-    Rules and aim of the game.

-    Difference between modes.

-    Player controls.

-    Sound settings.

   The purpose of this document is to help players to learn game without spending time while trying to find rules and controls by themselves, and increase the fun they have.

## Non-Functional Requirements

### Game Performance

The main aim of the game is to have high performance. Both graphical and playing process should be fast enough with high performance. We will have dynamic displays like dropping hurdles when runner crushes them or flush of the water when runner could not jump over the water hurdle. One of the displays will be the runners, since there will be two game mode solo and double player choice, for the single mode we will display the screen with one runway and for double player option we will have display two runways next to each other. Additionally, we will show runners’ in motion. We will keep these properties low as much as we can.

### Graphical Display

To display the game flowingly, we have two main objects which will be dynamic objects in motion runners and hurdles. The graphics of these two objects need to be smooth, fluent and should be displayed without any deficiency. So achieving high performance on graphical displaying is one of our significant objectives.

### User Interface

To make game easier to direct we will include easy and understandable user interface in our game for the users. Hurdle run is a basic game with basic instructions like restart, play, so we are aiming to make really easy and efficient interface for the users.

## Pseudo Requirements

In this project, the pseudo requirements can be specified as follows:

* This project will be implemented in Java.

# System Models

### Scenarios

#### Execute Game

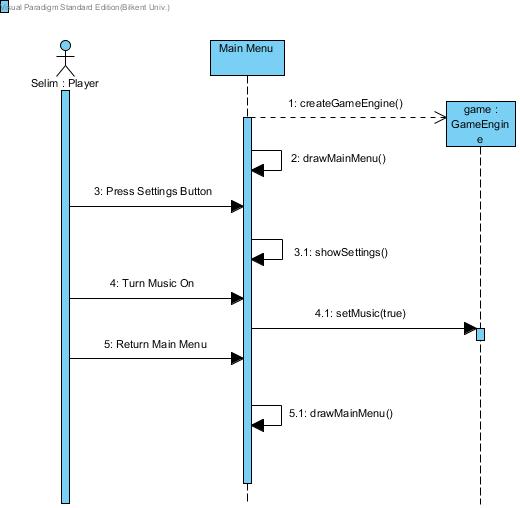
**Scenario:** Selim double clicks the game icon in the desktop and main menu comes up. At the main menu Selim clicks the "Settings" button. Because Selim likes listening to music, he turns the background music on. Then, he returns main menu and clicks "Single Player Game" to play the game. 

Figure 1 : Sequence Diagram of Execute Game

**Description:**

At the main menu, user may choose "Single Player Game", "Multiple Player Game", "Settings", "Exit Game" buttons. In this scenario, the user has chosen "Settings" and "Single Player Game" buttons.

#### Play Single Player

**Scenario:** Haluk clicks "Single Player Game" button and user interface of the game comes up. After 3 second countdown time is up, the runner starts to run and points counter of the game starts. Haluk steps over the hurdles by pressing spacebar. And he gets extra points. At some points of the runway there are puddles as a kind of hurdle. Some of them Haluk jumps early and get attached the puddle. So, one of his lives is gone. At the beginning of the game he had 8 lives but now he only has 1 last life. He is very careful now. After a while he jumps late at a hurdle and he lost his last life. the "The End" screen comes up, he got 750 points.

**Description:** The game starts with 3 second countdown. This provides the user time to get ready. Score counter does not count like time counter. While the user steps over the hurdles over and over without stumble to it, the score counter counts faster. There are two types of hurdles. One of them is regular hurdles like sport of athletics has and one of them is waterhole which is called puddle. The game has no database support. After the user learn the score point, the score is deleted. There is no way to view High Scores.

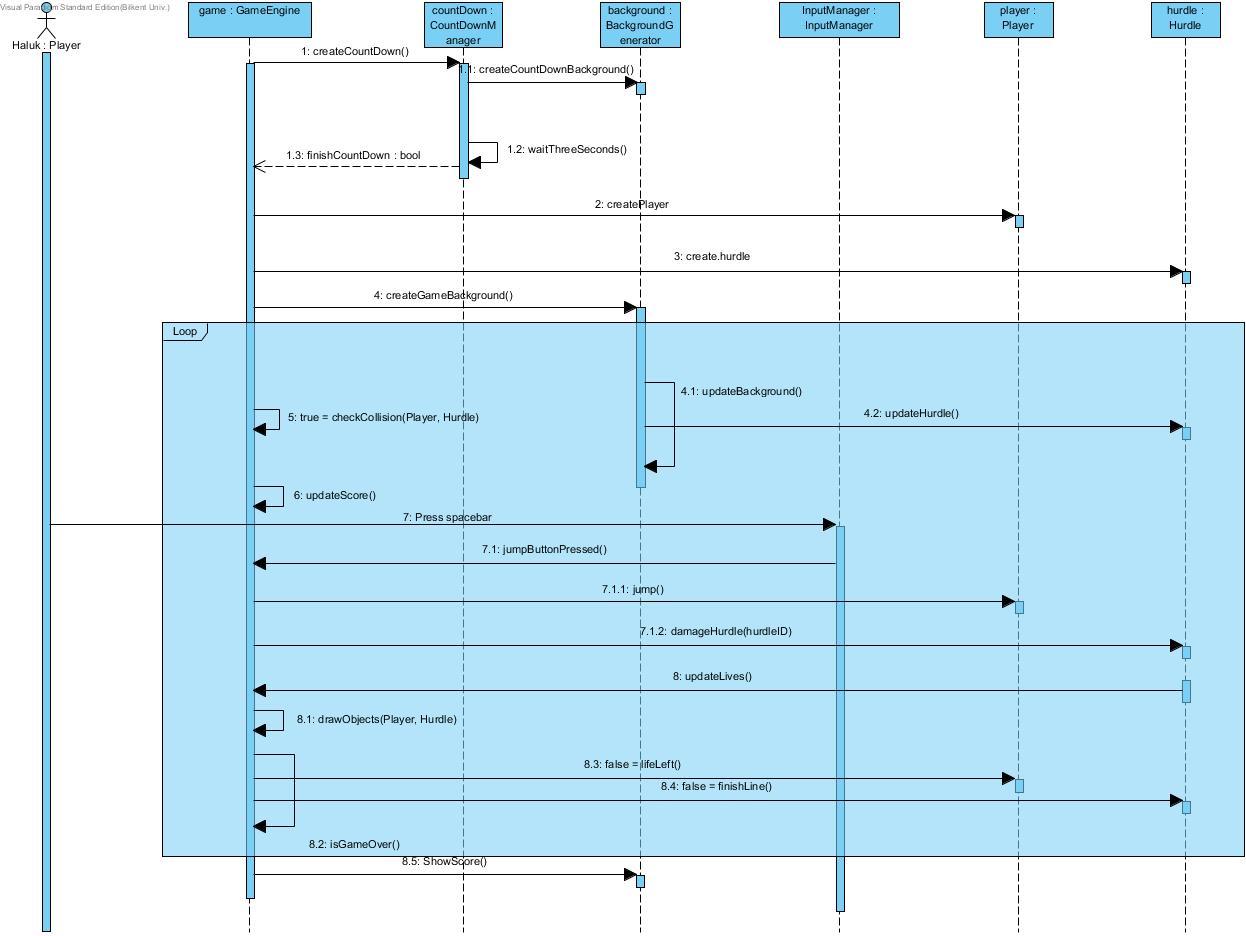


Figure 2 :Sequence Diagram of Play Single Player Game

#### Play Multiple Player Game

**Scenario:** Güneş and Ecem click "Multiple Player Game" button and user interface of the game comes up. After 3 second countdown, the competitive race starts. While Güneş jumps the hurdles by pressing spacebar, Ecem uses up arrow key to jump. While the runners are running side by side and a hurdle is coming, Güneş pretends to make a early jump and Ecem is fooled from this fake and she actually makes and early jump. So this puts Güneş a few miles ahead at the race. After this, Ecem decides to concentrate on the race and she waits a mistake of her rival. By the way Güneş is confident. She is so sure of winning the game that she does not even look at screen. This cause a late jump of Güneş and they are now side by side. While Güneş tries to understand what happened, she makes a second late jump and Ecem gets ahead. The finish line appears and the race is over with the victory of Ecem.

**Description:** There are only 2 player can play on the same PC at the multiple player mode of the game. The users use two keyboard key to jump. In this case, these are spacebar and up arrow key. At the beginning of the game the runners are at the middle of the screen and run side by side. If one of them makes mistake, she is seem a bit behind of the middle of the screen; and the other one is seem a bit ahead of the middle of the screen. Therefore the midpoint of the runners is fixed at the middle of the screen. There are fixed number of hurdles in the race and after all of them are passed, finish line appears. The one pass the finish line earlier wins.

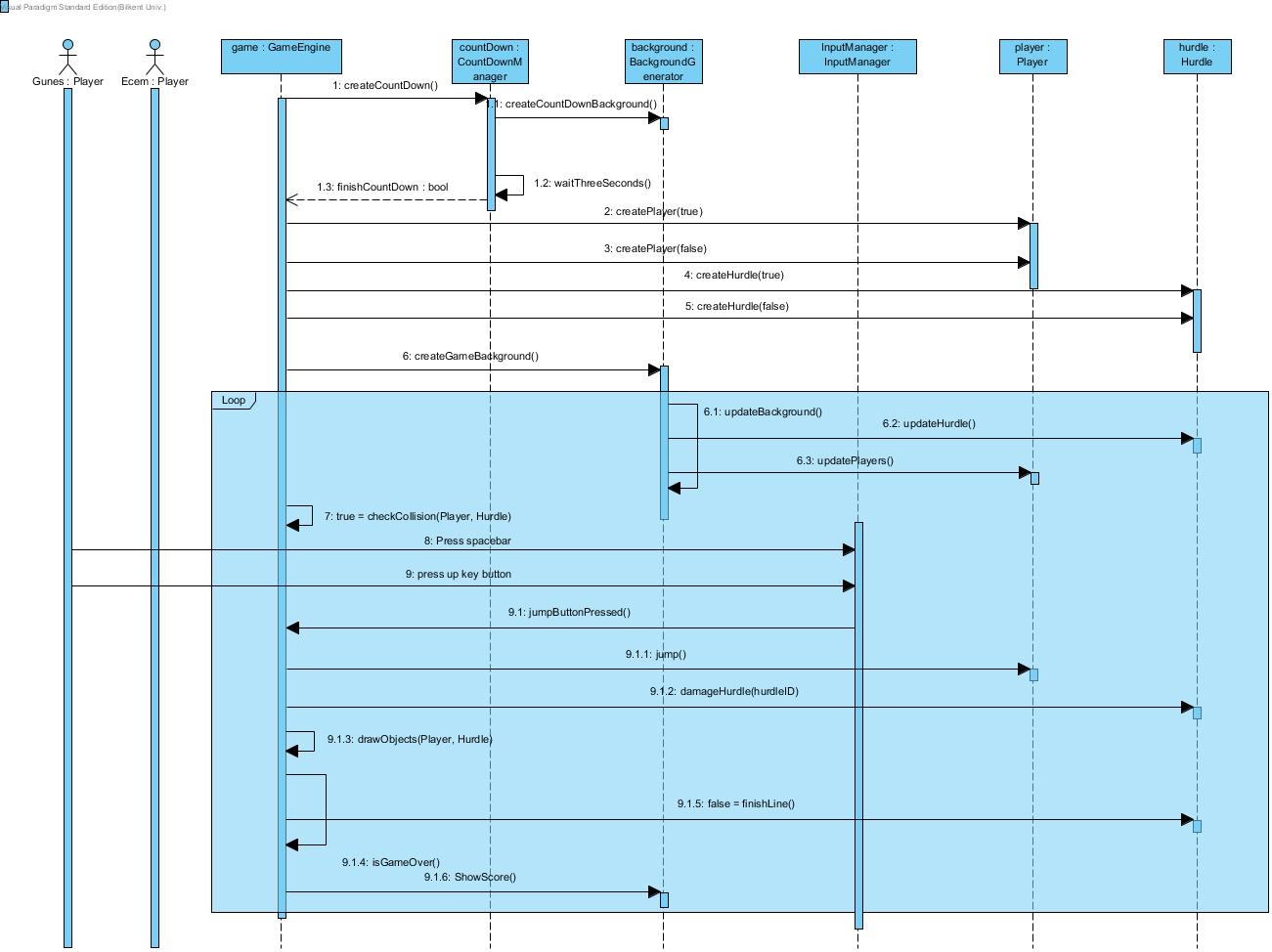


Figure 3 : Sequence Diagram of Play Multiple Player Game

### Use-Case Model

In this section, you can found use-case model of Hurdle Run followed by detailed use case explanations.

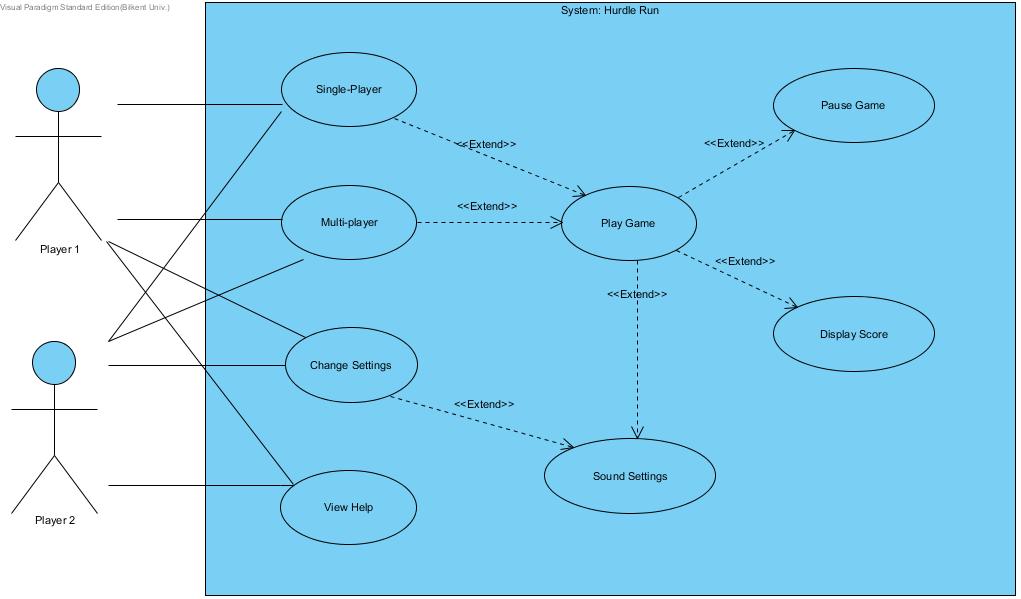


Figure 4

#### Use Case Descriptions

#### Use Case #1

#### Use Case Name: Execute Game

**Primary/Participating Actor:** Player 1 or Player 2

**Stakeholders and Interests:**  
-Player-1 wants to play the game.

-Player-1 has installed the game.

**Pre-conditions:** Game is installed.  
**Post-condition:** -  
  
**Entry Condition:** Player 1 double clicks the game icon in the desktop.

**Exit Condition:** When the Player 1 clicks the exit button on top-right.

**Main Flow of Events:**

1. The user (Player 1) double clicks the icon.
2. The system starts.
3. The main menu occurs in the screen.

**Alternative Flow of Event:**

* The user may enter the game in application menu.
* The user may exit from the application by the “Exit” option.

#### Use Case #2 Use Case Name: Single Player Play

**Primary/Participating Actor:** Player 1

**Stakeholders and Interests:**  
-Player 1 wants to play the game.

-System displays the Single-Player Game Play screen.

-Player-1 selects the “Single-Player Play” button.

**Pre-conditions:** The game is installed and opened (executed).  
**Post-condition:** -  
  
**Entry Condition:** The Player 1 needs to select “Single-Player” in the main menu.

**Exit Condition:** When the Player 1 pushes the “Go back to main menu.” in Display Score Screen.

**Main Flow of Events:**

1. Player 1 clicks "Single Player Game" button.
2. Interface of the game comes up.
3. After 3 seconds the competitive race starts.
4. The hurdles are starts to come.
5. Player 1 tries to avoid the hurdles as much as possible
6. If one of them fails to avoid, s/he will lose some of his velocity, and one of his lives.
7. If Player 1loose all of hers/his lives the game ends.
8. The Display Score Screen occurs in the screen.

**Alternative Flow of Event:**

* Players may pause the game. When the game is paused, the Players positions and their velocities are saved and when the game starts, it will continue where it was.

#### Use Case #3

#### Use Case Name: Multi Player Game Play

**Primary/Participating Actor:** Player 1 and Player 2

**Stakeholders and Interests:**  
-Player-1 wants to play the game.

-System displays the Multi-Player Game Play screen.

**Pre-conditions:** Player-1 or Player 2 selects the “Multi-Player Play” button.  
**Post-condition:** System needs to show Display Score Screen.  
  
**Entry Condition:** The Player 1 needs to select “Multi-Player” in the main menu.

**Exit Condition:** When the Player 1 pushes the “Go back to main menu.” in Display Score Screen.

**Main Flow of Events:**

1. Player 1 or Player 2 click "Multiple Player Game" button.
2. Interface of the game comes up.
3. After 3 seconds the competitive race starts.
4. The hurdles are starts to come.
5. Both players try to avoid the hurdles.
6. If one of them fails to avoid, s/he will lose some of his velocity.
7. The one who reaches the finish line, s/he wins.
8. The game ends.

**Alternative Flow of Event:**

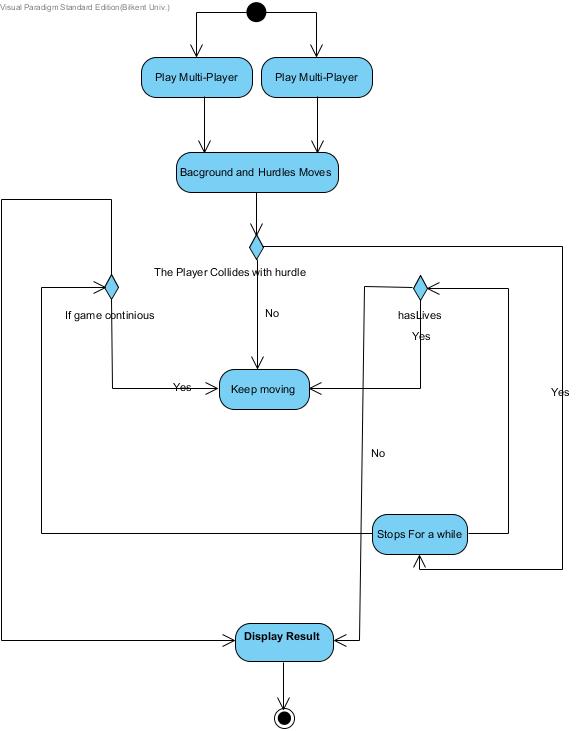
* Players may pause the game. When the game is paused, the Players positions and their velocities are saved and when the game starts, it will continue where it was.

### Object and Class Model

### 

Our operating class will be Game Engine class. This class will direct the game like with its methods like matching if runner hits hurdle or not, screening the hurdle crash or jumping of the runner exc. This class will take two object, player and hurdle. Three subclasses will be used for interactions in the game like input manager which determines button inputs. The other one is background manager which works for updating players, hurdles and background itself. The third one is countdown manager which is going to be used for displaying countdown of three before the running period starts. In Main Menu part we will create Game Engine object, we will display the menu and the setting.

### Dynamic Models



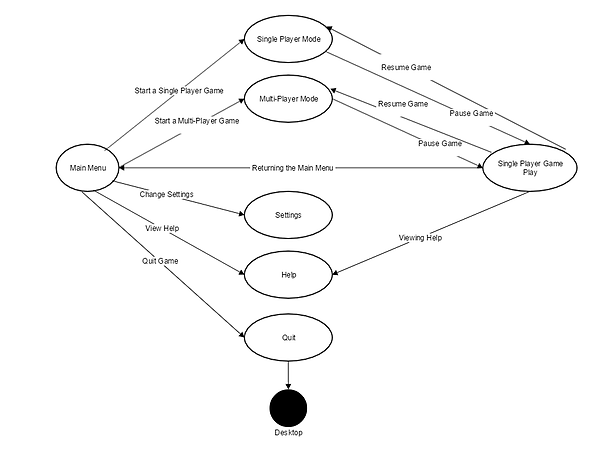
In Hurdle Race, there are two different play game options, Play Single-Player and Play Multi-Player. In either case, in other words when user choses one of the play modes, the game flows, in almost in the same way.

Basically, in single-player mode game starts when the user chooses the “Play Single-Player” and the background and hurdles starts to move to create an image that the runners run. The game engine keeps checking if there is a collision between the objects of the runner and the hurdles. As long as there is no collision, nothing changes. In the case of collision, the runner stops for a while, then, if the player has lives, s/he keeps playing. When s/he has no lives, game engine displays the result screen.

In the second case, in other words when “Play Multi-Player”, only the second condition in the flow changes, the rest remains the same. That condition becomes “if the game continues”, in other words, it will check if the hurdle run is completed or not since in the multi-player mode, there is a limited number of hurdles. After this check point, we remaining parts will also be the same.

### User Interface and Navigational Path

**Navigational Path**



**Main Menu**

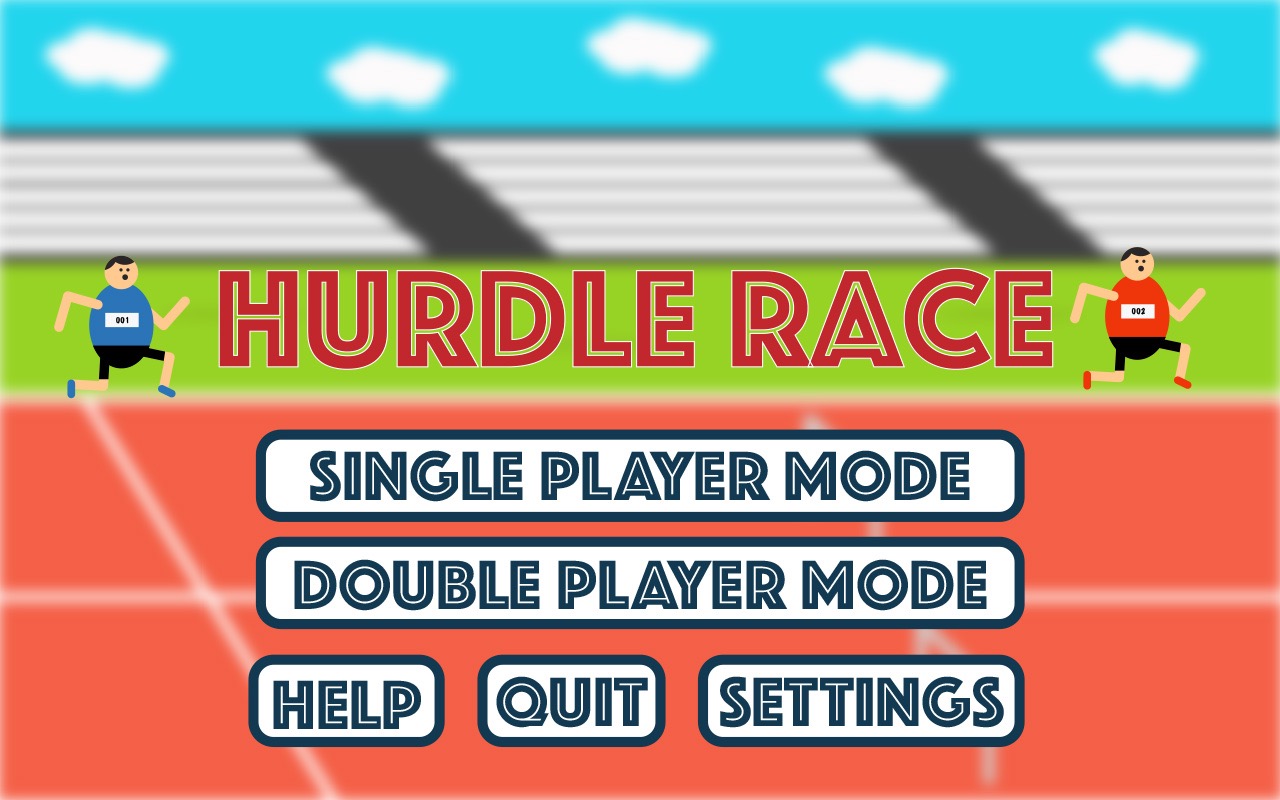
When the application starts to run, main screen is displayed. There is four options to select by player which are “Single Player Mode”, “Double Player Mode”, “Help” and “Settings. (See figure 2)

Figure 5 : Main Menu

Main Menu

**Single Player Menu**

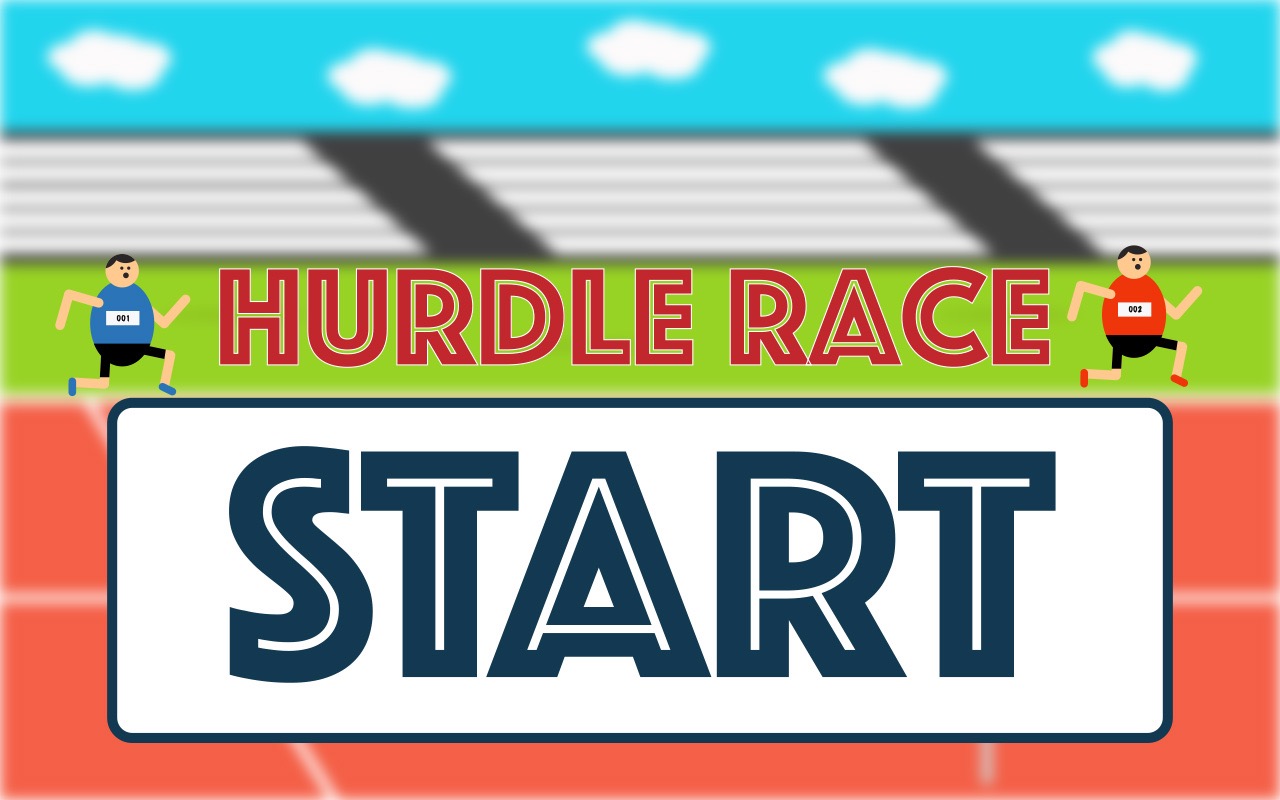
* When user selects the “Single Player Mode” option start screen will be displayed. There is only one selection in this screen which is “Start”. (See figure 3)

Figure 6 : Start Screen

**Play Screen**

After push the button “Start” play screen for single player mode will be displayed. (See Figure 4)

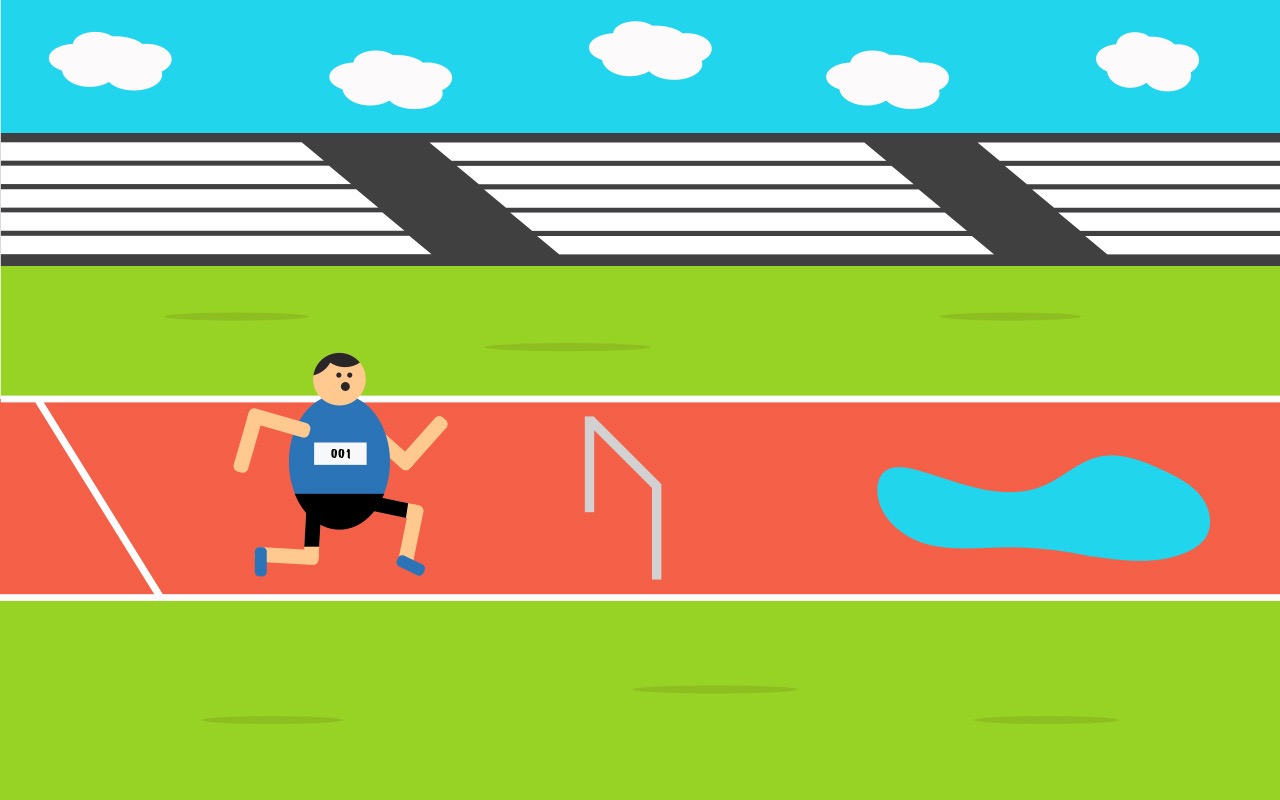


Figure 7 : Play Screen For Single Player Mode

**Finish Screen**

When user finish the run score of game will be displayed on the screen. There is option to return to the main menu on this screen. (See Figure 5)

Figure 8 : Finish Screen For Single Player Mode

**Double Player Menu**

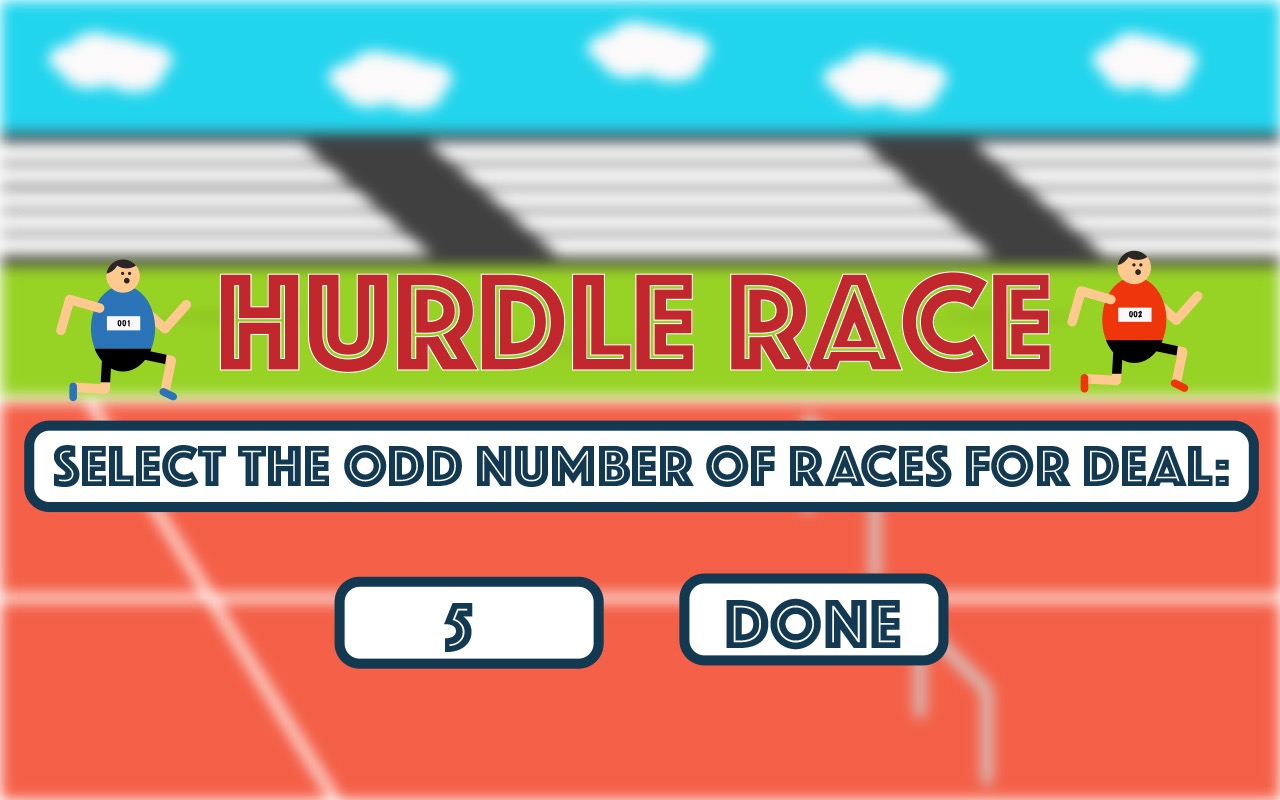
When user select the “Double Player Mode” option, screen for selection of number of games for deal will be displayed. Only odd numbers are allowed for this selection. (See figure 6)

Figure 9 : Selection for Number of Games for Deal

Figure 5.1.1.4 Selection for Number of Games for Deal

After select the number of games “Start” screen will be displayed. (See Figure 3)

**Double Player Play Screen**

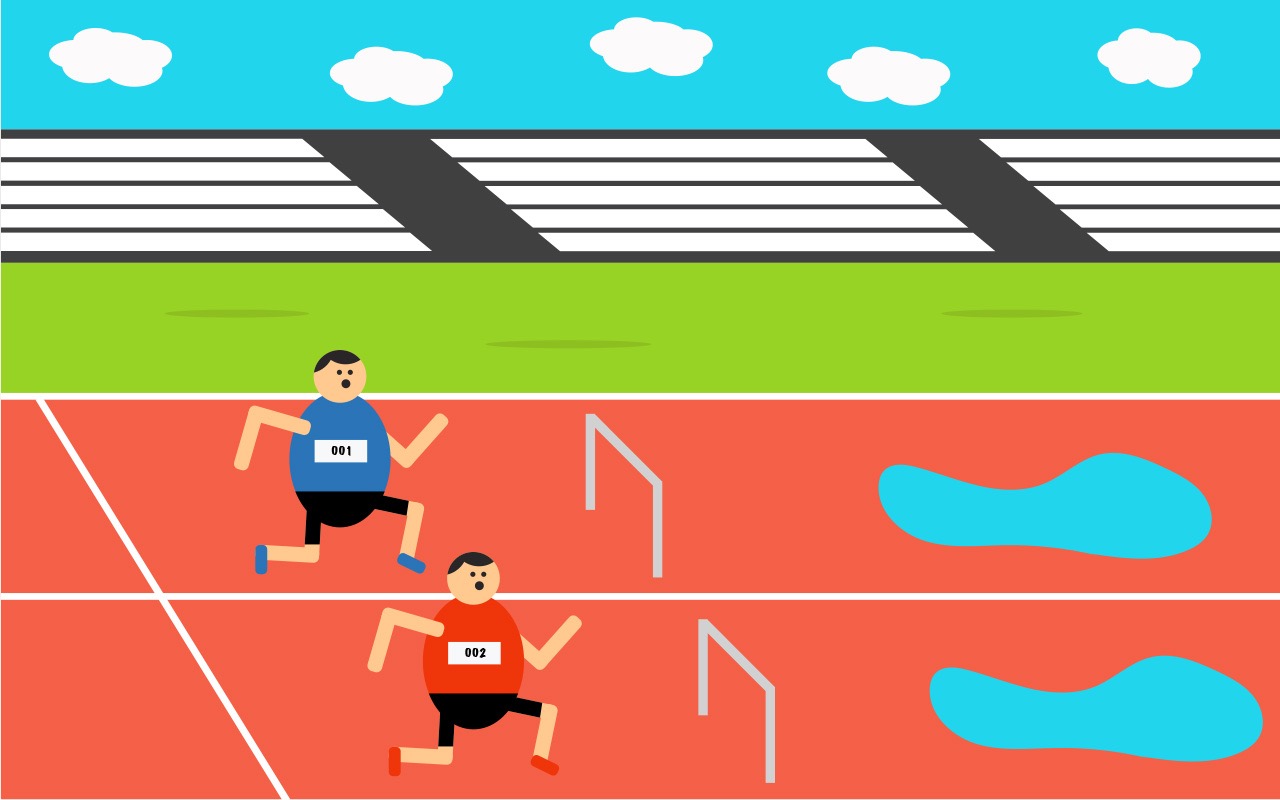
When user push the button “Start” play screen for double player mode will be displayed. (See Figure 7)

Figure 10 : Play Screen For Double Player Mode

Figure 5.1.1.5 Play Screen For Double Player Mode

**Finish Screen**

When each race finish scores of sprinters will be displayed on the screen. There are two options on this screen: return to the main menu or continue to deal. (See Figure 5.1.1.6)



Figure 11: Score Screen for Double Player Mode

**Finish of Deal Screen**

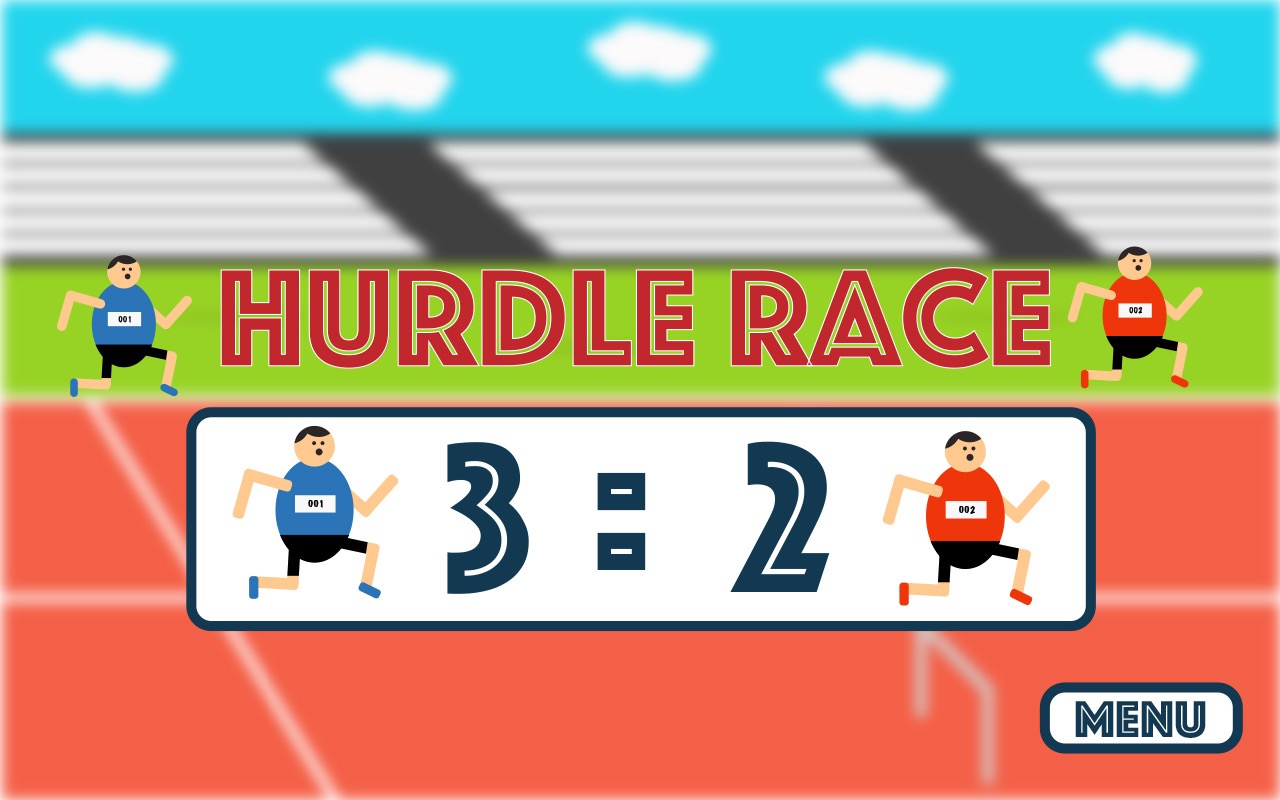
When all the races finish for a deal winner screen will be displayed. There is an option to return main on this screen. (See Figure 9)

Figure 12: Winner Screen

**Help Menu**

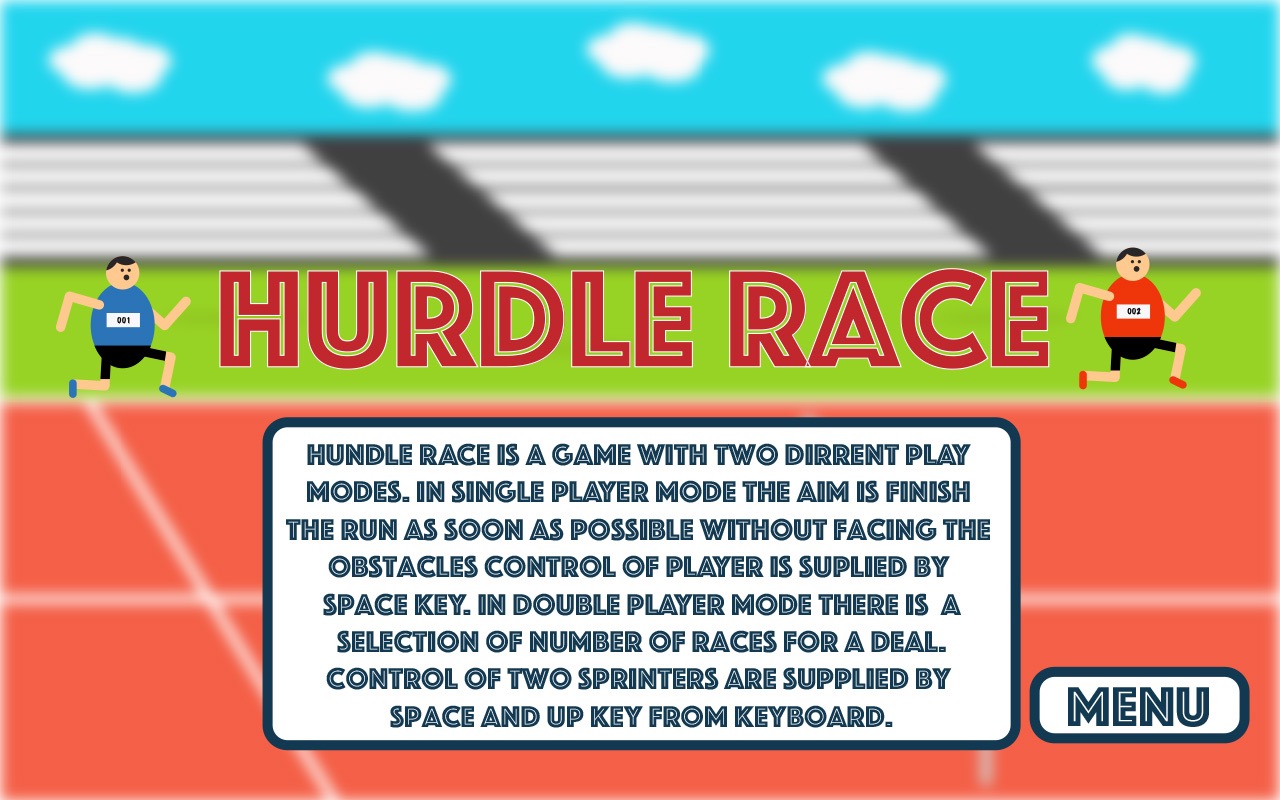
“Help Screen” will be displayed on the screen when user select the third option of main menu or press “esc” while play continuing and select the help option on this pause menu. (See Figure 10)

Figure 13 : Help Screen

**Pause Menu**

While playing the game if user press the “esc” key from keyboard “Pause Screen” will be displayed. There are three choices in this screen which are “Resume Game”, “Help” and “Menu”. (See Figure 11)



Figure 14 : Pause Screen

**Settings Menu**

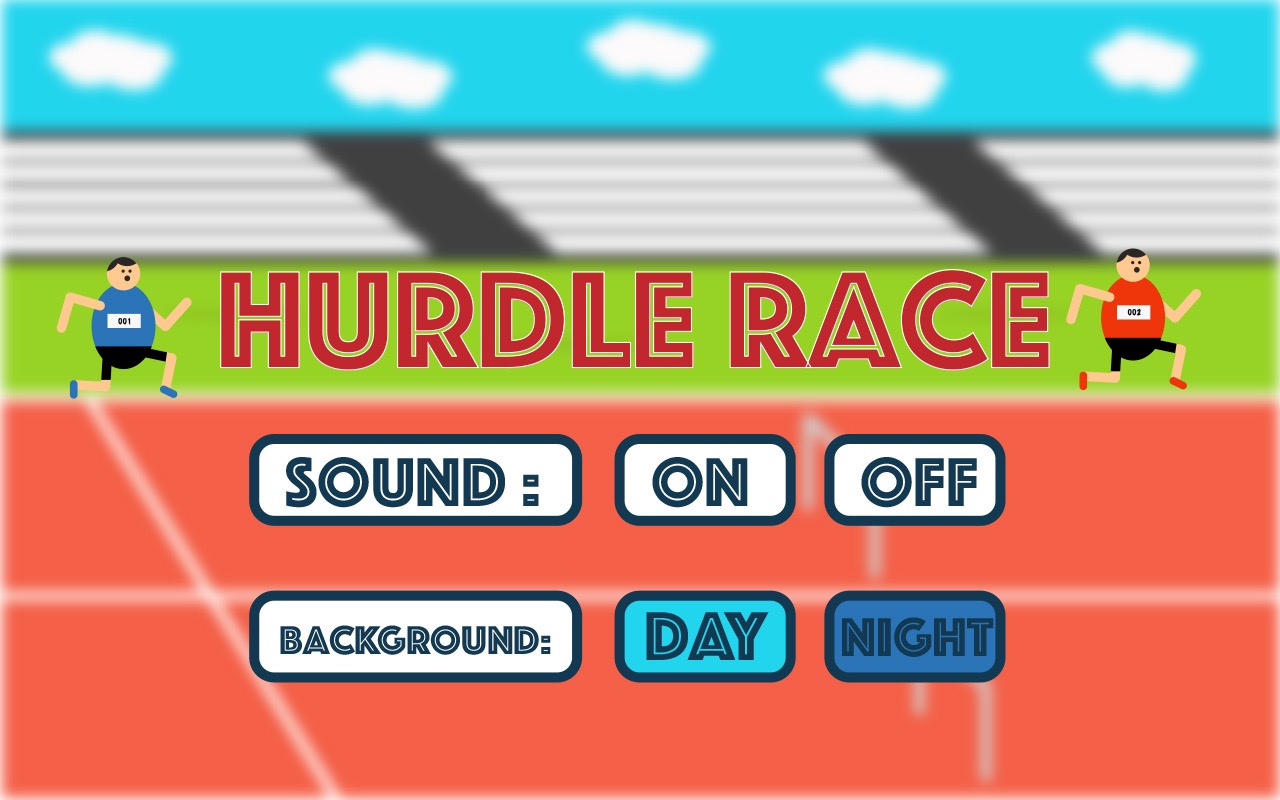
When user select the “Settings” option in main menu, “Settings” screen will be displayed. (See Figure 12)

Figure 15 : Settings Screen

# Important decisions in overall Analysis

• The most crucial base class of our system is operate class. In the hierarcial order operate class will be directing the other classes by extension or with the help of the other inheritance futures. To gain more beneficial system, operate class is to key element to provide increased complexity and simplified system organization.

• The game objects are another part of well organized futures in terms of inheritance. We created GameObject class which we will inherit our other objects from.

# Conclusion

• Through this report paper, we provided an analysis report about a game, “HurdleRun”. As a first step, we are given requirement specification. As a second step, we declared our system model in more detailed.

• Examining the user requirements was the first thing we did. Than we specified all the requirement and futures that user can perform on game. We specified all the functional and nonfunctional requriements. Since we know determining the requirements is the most significant part, we want to fulfil all the requirements and futures that user needs to have.

• System Model includes four parts Use case Models, Dynamic Models, Class Models and User Interface.

• As we decide on our use case , we thought more about what are the user requirements to play or what are the actions we are depending on.

• In our dynamic model we used state diagrams, sequence diagrams and activity diagrams. We used sequence diagram to show actions between users and the system.

# References

1. Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.