Bilkent University

Object Oriented Design

IMPRISONMENT

Final Report

İdil Ağabeyoğlu 21300761
Evren Ergen 21300781
Ismail Kerimov 21300355
Sinan Öndül 21200962

Final Report

April 30,2016

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfilment of the requirements of the Object Oriented Design course CS319/1.

Table of Contents

1. Exposed Documentation

1.1 Introduction

The game of “Imprisonment” is an action game that is inspired by “Volfied” which created by DOS game. DOS game can be considered as the ancestor of the video games. There is only one player which directs and commands dot in the game. The aim of character is to complete the map by escaping from the monsters and drawing quadratic shapes until the game is won. If monsters hit the dot or the line that is drawn by the dot the user will lose 1 life span. . The game map consists of walls, dot, lines drawn by dot and bonuses. After completing the required threshold, user can pass to the next level and the game consists of 3 levels.

1.1.1 System Requirements

The Standard Java Runtime Environment (JRE) is required in order in order to play Dangerous Maze game. The game can be installed and played by the CDs.

Minimum System Requirements:

- Windows 2000/XP
- 128 mb RAM
- Screen resolution: 800*600

Recommended System Requirements:

- Intel Core i5 or equivalent processor (Athlon/Duron/Celeron)
- 512 MB of RAM or higher
- Screen resolution: 1920*1080

1.1.2 Installation Information

Unzip the archive file which is named imprisonment.zip to a directory on your computer. To run the game you have two options:

1) Run by double clicking the imprisonment.jar file to execute the game. This is recommended.
2) If you have a little bit knowledge about java programming and using Java IDE as compile and run code in java. You can open in any Java IDE, compile code and run the main method. Then the console open and the game start.

1.2 Playing the Game

1.2.1 Game Overview

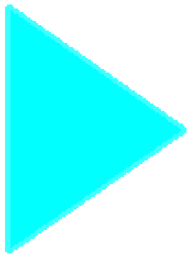
In the Imprisonment game, first the user may play game, exit or change settings as the features of the menu by mouse. Then when player starts to play the game, he/she controls dot by only keyboard and try to escape from monsters. Moreover, user will try to complete the map by drawing quadratic shapes until the game is won and goes to the next level. Game map is already defined before the main character gets into it. Additionally, if the dot collides monsters one of the life spans that the user has will be deducted and the user will have a chance to start from the where it dies as checkpoint. If the dot collides with a bonus according to the type of bonus, bonus will be activated and the user will benefit from it in order to win the game. There are 3 different types of bonuses regarding as freeze-time(the monsters will stop moving), slow-time(the speed of the monsters will be decreased) and destroyer(the user will choose one of the monsters and it will be destroyed)

There are 3 different levels in the game. In terms of difficulty, first one is easy, next one is medium and last one is advance. In every level, there is one big monster and 2,3,4 small monsters as follows according to the difficulty of the level.

The game has also default background which display during the game. Moreover, there are also three different backgrounds the user will be able to change from Settings.

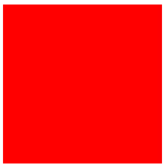
1.2.2. Game Objects

1.2.2.1 Dot

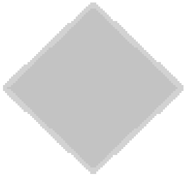


Dot is the feature which is in the control of the user while playing the game. The direction of dots is limited in the edge of the frame and it moves left, right, up and down by keyboard's direction keys.

1.2.2.2 Monsters



Big Monster



Small Monster

Monsters randomly move around in order to eliminate the user from the game and its move has some limits because of the wall(they are always inside of the frame and when they collide wall they start moving to the opposite direction before they hit the wall). Additionally if the monster hits the dot or the line which is drawn by the dot, the user loses one of its lives and the line is removed from the frame. In each level the speed of the monster changes according to the difficulty of the level.

1.2.2.3 Bonuses

There are 3 types of bonuses which can be collected by the dot.

a.Destroyer: In that bonus the user can remove one of the monsters by choosing them using keyboard.

b.Freeze time: When this bonus is collected the monsters freeze for 3 seconds and the dot plays the game without any threats.

c.Slow time: This bonus slowdowns the monsters for 5 seconds and with the decreasing speed of the monsters the user have chance to see the moves of the monsters more efficiently and so act accordingly.

1.2.2.4 Wall

The shape and the size of the wall is unstable regarding with the acts of the dot. It decreases in each quadratic shape that is drawn by the dot and work as a mirror in the aspect of reflecting the monsters and changing its directions in a straight line by horizontally or vertically.

1.3 Controller Settings

Player controls the character with keyboard’s direction keys.

1.4 Game Screenshots

1.4.1 Main Menu

Imprisonment has a main menu as the Figure 2.That menu is the first page of the game. It appears as soon as the game executed. Main menu panel contains Play Game, Settings and Quit buttons. These three buttons lead player to different pages.

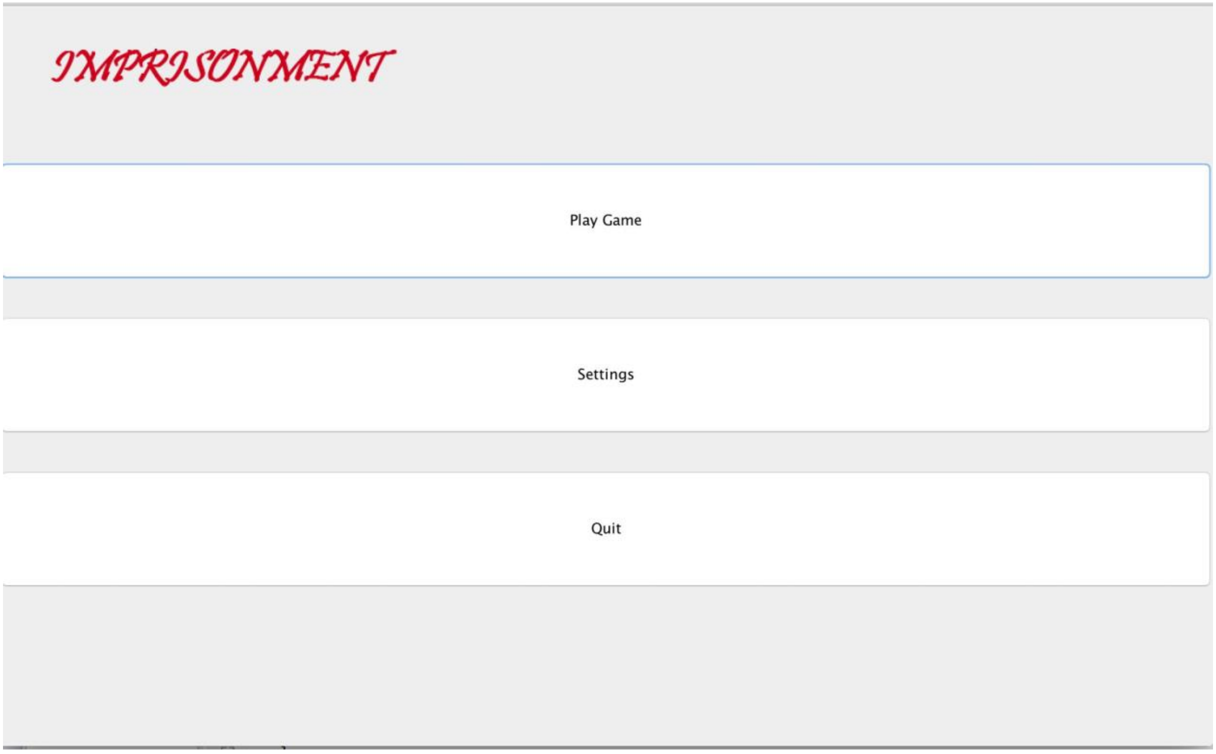


Figure 1

1.4.2 Play

This is the game panel Figure 2. When player clicks Play Game button at the main menu, that panel opens and the game starts.

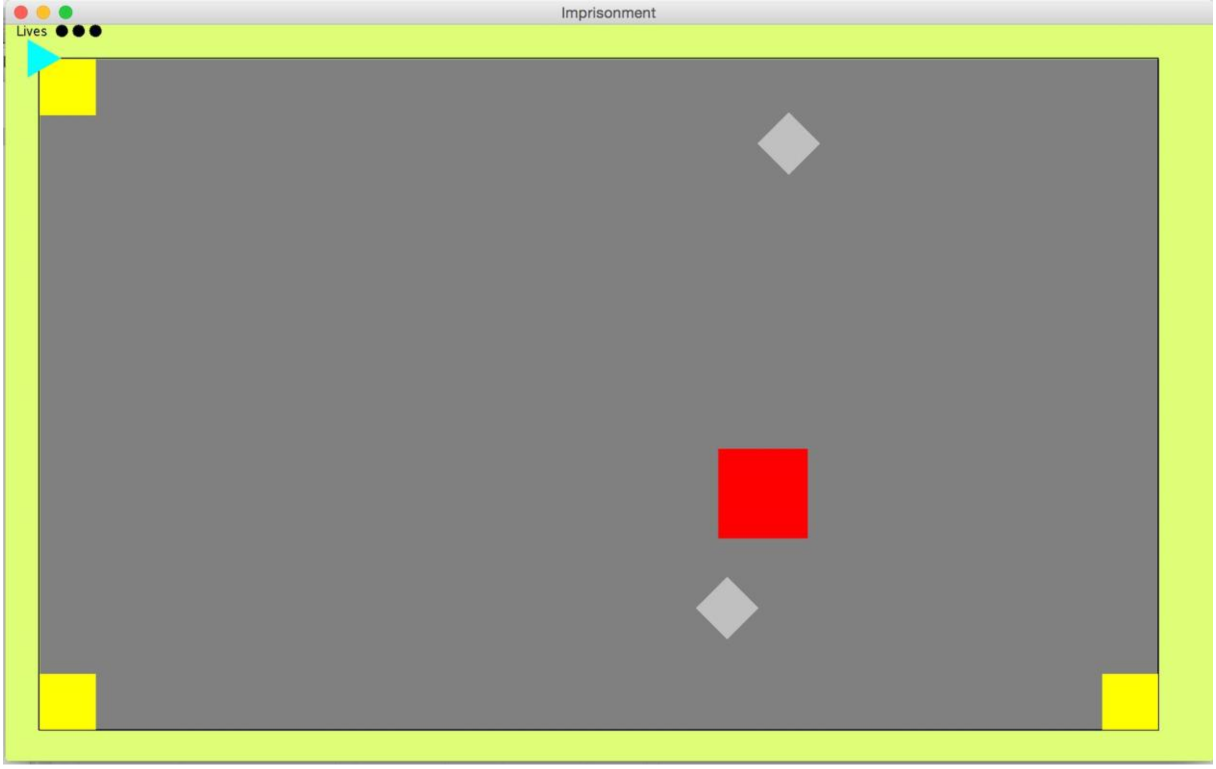


Figure 2

1.4.3 Settings

When the player changes the default settings of the game, he uses the panel in the Figure 3. Player can change name the background of the map.

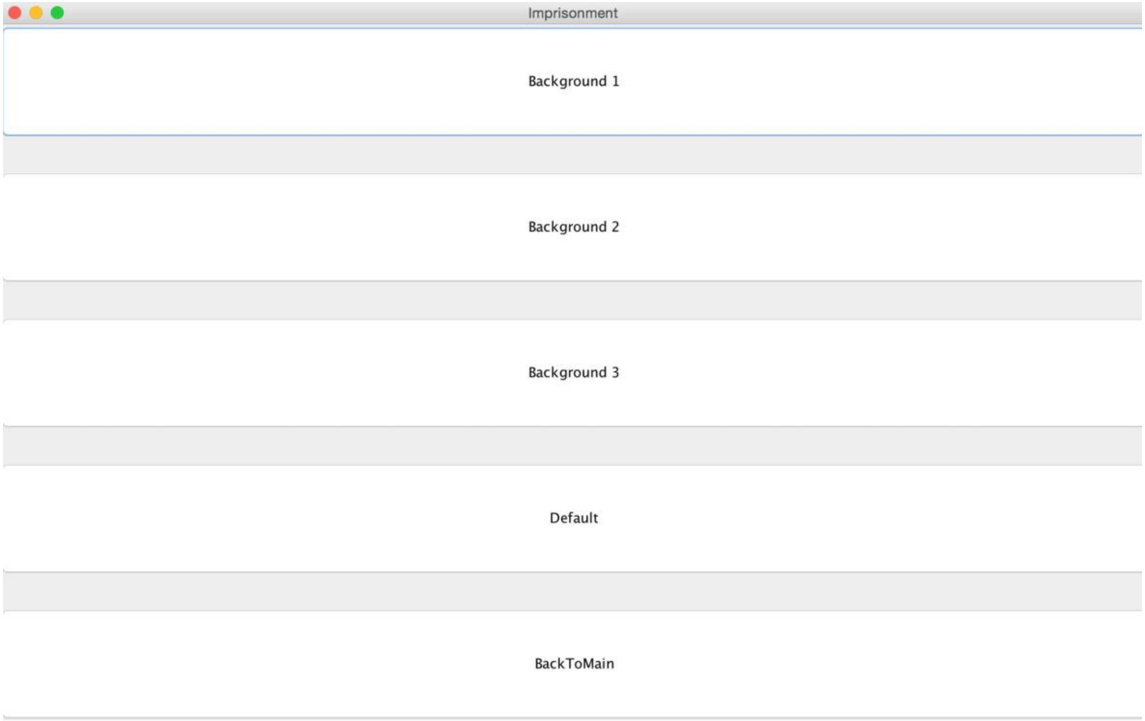


Figure 3

1.4.4 Exit

Exit button is the red button which is located at the upper right corner of the main panel. This button can be used to close the game directly. Player can also close the game with quit button which is located in the main menu.

2. Differences in Implementation

This report discusses how the implementation went and whether or not any major changes to the design had to be made due to complications during the implementation. The implementation of the game was more challenging than we expected. Although during the process of implementation our detailed analysis and designed report did our job much easier during implementation, in some parts they were not enough. Therefore such changes in the structure and content of the classes and features occurred.

2.1 Design Changes Not Covered in Analysis

Although the inspiration of our game "Imprisonment" is "Volfied", our aim was to enable the users a more developed game in a well-designed platform and with a more riendly user interface however since the time we had was limited and was not enough in order to fulfill what we discussed in analysis report. As an illustration; we did not generate help and pause feature for the users, changes for the colour of the dot according to the difficulty of the level and some bonus types such as lives and mapshrinker and pictures of the bonuses. Additionally, we did not implement sound for successfully in order to play in the background while the user plays the game, so the class SoundManager was not implemented.

2.2 Design Changes Covered in Analysis

Instead of creating a CollisionClass, we handled the collisions separately in each class and drawing the features such as updates on the wall made by dot, separately in for different class we created a MissingPolygon in order to fulfill such requirements.

Although in analysis report, SettingsMenu was the parent class of ScreenManager and SoundManager, because of we were not able to implement SoundManager we did not open a new class ScreenManager, we basically implemented the features of the ScreenManager in the SettingsMenu.

2.3 Design Improvements Not Covered in Analysis

Although we did not cover in the Analysis where will be the position of Dot will be after losing a life, in order to prevent the user to start the game in the same place exactly over and over, dot returns to the checkpoint where it was lastly located which is the safe spot in the edge of the wall. Additionally in order to game more challenging we made some features on the of the monsters. We created 2 types of monsters regarded as BigMonster and SmallMonster. There is always one big monster in each of the three levels however according to the difficulty the number small monsters increase as 2,3 and 4. Other than that we accomplished the general implementation of the code. We hope that, this program will be beneficial for people who enjoy playing DOS game. Furthermore, help, sound and high-scores will be the future extinctions.

3. Class Diagram

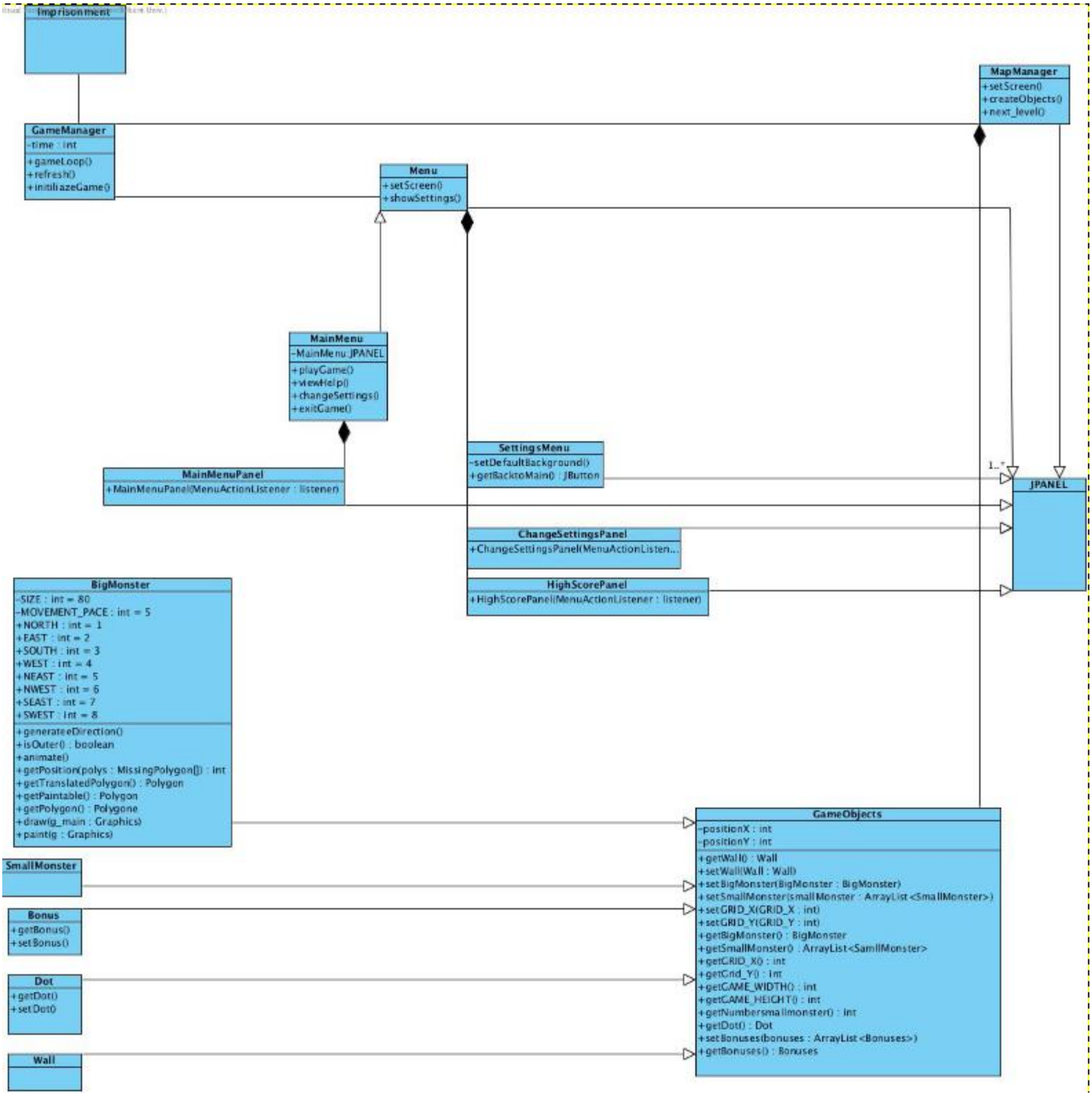


Figure 4

4. Use Case Model

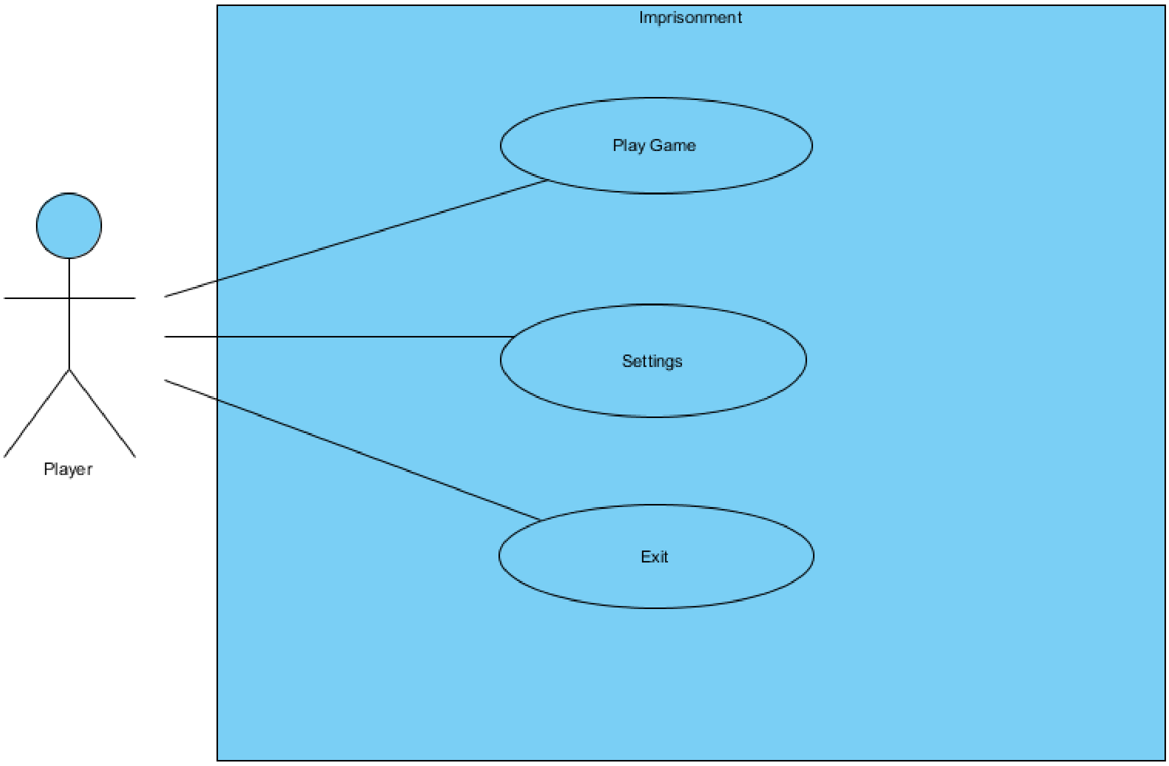


Figure 5