Bilkent University

**Object Oriented Design**

IMPRISONMENT

Analysis Report

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

Progress Report

February 20,2016

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. **Introduction…………………………………………………………...3**
2. **Proposed System……………………………………………………...4**

**2.1 Overview…………………………………………………………...4**

**2.1.1 Features……………………………………………………...4**

**2.1.1.1 Dot……………………………………………………….……….4**

**2.1.1.2 Monsters………………………………………………………….5**

**2.1.1.3 Bonuses…………………………………………………………...5**

**2.1.1.4 Wall……………………………………………………………....5**

**2.2 Functional Requiremens………………………………………….6**

**2.2.1 Play Game……………………….…………………………...6**

**2.2.2 Settings……………………………………………………….7**

**2.2.3 Help…………………………………………………………..7**

**2.2.4 Exit…………………………………………………………...7**

**2.3 Non-functional Requirements.……………………………………8**

**2.3.1 Game Performance………………………………….……….8**

**2.3.2 Graphical Smoothness………………………………….……8**

**2.3.3 User-Friendly Interface……………………………………...8**

**2.4 Constraints...……………………………………………………….9**

**2.5 Use-Case Model…………………………………………………...9**

**2.5.1 Use-Case Descriptions………………………………………10**

**Use-Case#1………………..……………………………………………...10**

**Use-Case#2……..…………………………………………...……………11**

**Use-Case#3…………………………..………………...…………………12**

**Use-Case#4……………………………………………..………………...13**

**Use-Case#5……………………………………………………...……......14**

**Table of Figures**

**Figure 1: …………………………………………………………………….9**

**Figure 2: ……………………………………………………………………11**

**Figure 3: ……………………………………………………………………12**

**Figure 4: ……………………………………………………………………13**

# **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.

In the game, the user will be able to control a dot and the goal of the user is to complete the map by drawing quadratic shapes until the game is won. However, the monsters in the map makes it easier said than done. If monsters hit the dot or the line that is drawn by the dot the user will lose 1 life span.

Although ”Imprisonment“ is an inspired game, it contains many different features from its inspiration such as new bonuses, new maps and new sound effects. In addition the game will be designed as a desktop application that can be controlled by mouse and keyboard.

(<https://www.youtube.com/watch?v=vlYYXHICUis>)

# **Proposed System**

The monsters will randomly move around the screen and the player, a dot, will try to fill the screen from edges by drawing squares. While it is trying to it, if any of monsters hits the player, the game is lost.

# 2.1 Overview

In the game, while the masters who tries to catch the dot and causes to lose the game, a dot which is kind of trapped in a frames is going to try the fill the screen by drawing quadratic shapes and at the same time by running from monsters. Size of the frame will get smaller with each quadratic shapes including rectangles and squares and in the limited space within the monsters will give a challenging game to the users. In order to win the game the size of the frame must decrease %80 so basically if the user can keep continue to play until %20 of the map without losing he/she will win the game. In addition the user will have 3 lives to be successful at the game.

# 2.1.1 Features

## 2.1.1.1 Dot

Dot is the feature which is in the con troll of the user while playing the game. The colour of the dots changes according to the difficulty of the levels regarding as in the easy part the dot will have the colour of blue, in the medium part the dot will have the colour of green and in the hard part the dot will have the colour of red. The direction of dots will be limited in the edge of the frame in addition it will have only the ability to move left, right, up and down.

## 2.1.1.2 Monsters

Monsters will randomly move around in order to eliminate the user from the game and its move limits when it touches to wall. Additionally if the monster hits the dot or the line which is drawing by the dot, the user will lose one of its lives and the line will be removed from the frame. In each level the speed of the monster will change according to the difficulty of the level.

## 2.1.1.3 Bonuses

There are 3 types of bonuses which can be collected by the dot. a.Lives: In that bonus the dot will have extra lives to continue the game without losing the game.

b.Freeze time: When this bonus is collected the monsters will freeze for 3 seconds and the dot will have chance to play the game without any threats

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

## 2.1.1.4 Wall

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

# 2.2 Functional Requirements

* The implementation of the program will be done in Java Language because of the compatibility issues.
* The user will use the keyboard and the usage of the mouse will not be included in the game except selecting the options in the main page such as play game, settings.
* The user will be allowed to pause and then resume the game.

**2.2.1 Play Game**

The player tries to fill the screen up to %80 without being caught by monsters that are randomly moves in unfilled screen area. Player must not die three times in order not to lose the game. While there is monsters and a player, there will be random bonuses in the game so that the player can boost his/her dot up and makes easier to win. Also the bonuses will be in the unfilled area of the screen and when the player covers the bonus into the filled area, the bonus is collected.

When the player finishes a hardness level, he/she can select others such as easy, medium and hard. With the more hardness, the faster monsters and slower dot make the player’s job harder. This increases the eye and hand coordination while offers much more fun.

**2.2.2 Settings**

Some features of the game’s control is given to user. User will be able to change the some settings of the game and these changes are as follows

* Change the background image
* Turn down and turn up the music

**2.2.3 Help**

Although ”Imprisonment“ is user friendly and easy to understand and play by the user the game will include an help ingredient that shows the overview of the game and answers of the questions that might be frequently asked as a text document.

**2.2.4 Exit**

While the program is running the player may desire to not pause the game but quit the game and this operation is done by pressing the exit button. The user may not be satisfied with his/her current success such as user might have only 1 life span while only the %10 percent of the map is accomplished so in that case the user might want to exit the game and start a new one.

# 2.3 Non-functional Requirements

## *2.3.1* ***Game Performance***

The performance of the imprisonment is the high priority for overall design. The game will have a short response time with minimum delay for the user to play. Additionally the inspiration of the game is based on the old games, thus the system requirements will be tried to keep minimum.

## *2.3.2* ***Graphical Smoothness***

Although the game bases on the old games, the graphical interface the proper animations and smooth graphics are aimed in order to make the game more exciting and giving pleasure from the interfaces to the users and this will be accomplished by trying to keep the response time of the graphics as low as possible.

## *2.3.3* ***User-Friendly Interface***

User friendly interface is an essential attribute to make the game attractive for the users. The users should be able to play and understand the game easily and they ought to feel comfortable by facing with minimum difficulty. Because of the reasons which is specified above the benefits which is given to user such as bonuses are maintained to the users.

# 2.4 Constraints

# The game will be implemented in Java and will be able to played in all operating systems that support Java Language

* The graphs of the game will be smooth

## 2. 5 Use-Case Model

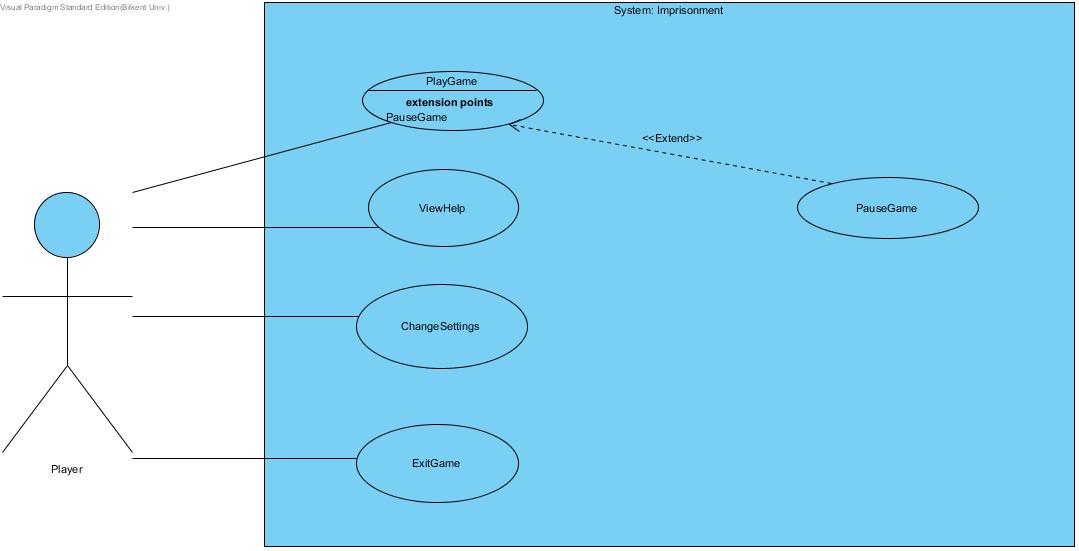


Figure 1: Illustrates the Imprisonment’s use case model

## 2.5.1. Use Case Descriptions

Use Case #1

**Use case name:** PlayGame

**Participating actors:** Player

**Entry condition:** The game must be open and it will be on main menu.

**Exit condition:**

• Player covers the map’s at least %80.

• Player loses lives level.

• Player chooses to exit to main menu.

**Main Flow of Events:**

1. The game is opened through executable.

2. Player selects the game difficulty.(Easy-medium-hard)

3. Player wins or loses.

4. System displays the time has passed of the player.

5. The exit music starts to play.

6. The game returns to the main menu.

**Alternative Flow of Event:**

• Player loses. (go step 4)

• Player exits the game before it finishes. This cause the progress will be lost.(go to step 6)

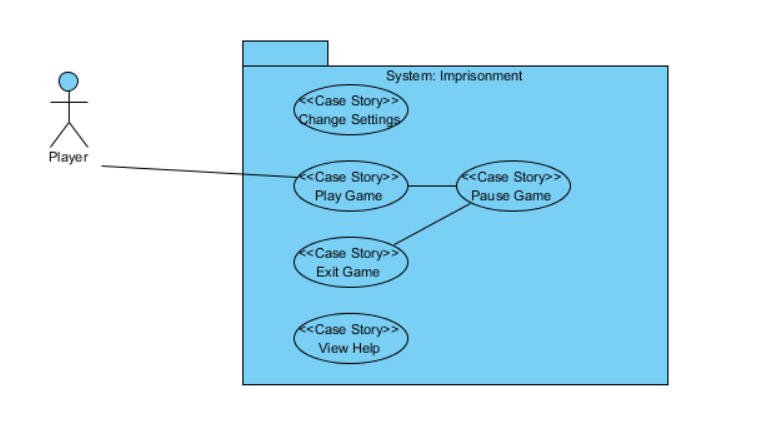


Figure 2

Use Case #2

**Use case name:** PauseGame

**Participating actors:** Player

**Entry condition:** A new game must have been started by the the player pressing the Play Game button.

**Exit condition:**

* Player chooses to exit to main menu
* Player decides to continue to play

**Main Flow of Events:**

1. Player chooses to play a new game
2. When he needs to take a break from the game, he clicks the pause game button.

**Alternative Flow of Event:**

* Player clicks the pause game button.
* Terminates the new game to go to main menu.

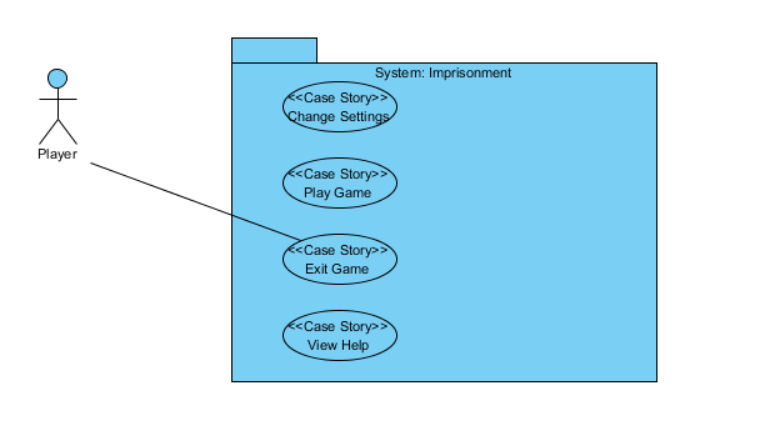


Figure 3

Use Case #3

**Use case name:** ViewHelp

**Participating actors:** Player

**Entry condition:** Player must be on the Main Menu and select “View Help”

**Exit condition:** Player selects back button to return the Main Menu

**Main Flow of Events:**

1. Player opens the View Help from the Main Menu
2. Player reads the texts which give instructions about how to play the game and its purpose
3. Player closes the View Help and returns back to Main Menu

**Alternative Flow of Event:**

* Player may not read the documents and return to Main Menu directly.

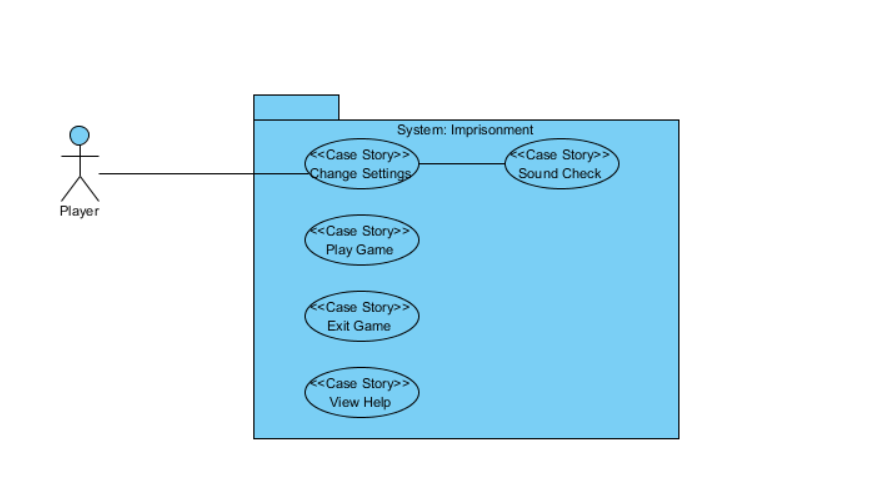


Figure 4

Use Case #4

**Use case name:** ChangeSettings

**Participating actors:** Player

**Entry condition:** Player must open executable and press Change Settings button.

**Exit condition:** Player closes the Change Settings button to go back to the Main Menu

**Main Flow of Events:**

1. Player opens the change settings menu
2. Player changes the Sound settings
3. Player closes the change settings menu

**Alternative Flow of Event:**

* Player may not change the sound settings and close change settings menu

Use Case #5

**Use case name:** ExitGame

**Participating actors:** Player

**Entry condition:** Program must be running

**Exit condition:** No exit condition

**Main Flow of Events:**

1. Player runs the executable
2. Player selects play new game
3. Player loses or exits the current game
4. Player quits the game by pressing exit game button

**Alternative Flow of Event:**

* Player opens the game
* Player closes the game by pressing exit game button