  
  
  
  
CS 319 - Object-Oriented Software Engineering  
Analysis Report  
RUNNING JON

Group  
ERKAN ÖNAL

NİHAT ATAY  
BARIŞ ARDIÇ

MERT KARA

Table of Contents

[1.Introduction 3](#_Toc464293915)

[2. Requirement Analysis 4](#_Toc464293916)

[2.1 Overview 4](#_Toc464293917)

[2.1.1 List of Characters 5](#_Toc435780107)

[2.1.2 List of Weapons 7](#_Toc435780108)

[2.1.3 List of Bonuses 9](#_Toc435780109)

[2.2 Functional Requirements 1](#_Toc464293918)1

[2.3 Non-functional Requirements 1](#_Toc464293919)2

[2.4 Constraints 1](#_Toc464293920)2

[2.5 Scenarios 1](#_Toc464293921)3

[2.5.1. Play Game 1](#_Toc464293922)3

[2.5.2 Pause Game 1](#_Toc464293923)4

[2.5.3 View High Scores 1](#_Toc464293924)5

[2.5.4 View Credits 1](#_Toc464293925)5

[2.5.5 View Help 1](#_Toc464293926)6

[2.6 Use – Case Model 17](#_Toc464293927)

# 1.Introduction

Running Jon is a desktop application action game. Its logic is similar to the games like Space Intruders. However, its scenario is different than space intruders. Our game’s scenario is based on the famous book and also TV series; Game of Thrones. We are planning to improve our games by increasing the quality of the graphics and game effects.

In Running Jon, there is a main character whose name is Jon Snow and the main purpose of the game is to save the Lady Sansa from the enemies. Jon Snow will have the capabilities of throwing blades to the enemies. He will be able to move left and right. He could improve his blade’s strong by dropping different items from the enemies that he killed. He got totally three lives. On the enemy side, there are different types of solider like ordinary soldiers, white walkers and king’s guardian. They also throw blades. When they hit you or if blades of them hit you, you lose one life point. They have different capabilities. Ordinary soldiers are weak compared to others. King’s guardian got huge armor and good attack power. White walkers attacks are decreasing your speed along with your lives. If you kill them, you get extra power-ups like increasing your attack power, throwing multiple blades to different directions and increasing your speed to move left or right.

The movements of the Jon Snow will be controlled by keyboard arrows and when user hits space button. Character will throw blades to enemies.

# 2. Requirement Analysis

## 2.1 Overview

The game is composed of three levels. In each level, the difficulty increases. After end of each level, there will be strong boss having fast moves and fast attacks. Main purpose here is that completing the game without losing all of your life points and saving Lady Sansa. To do that, user will be careful about running away from the blades of enemy and he should not collide with them. In addition, to increase the chance of saving Lady Sansa, user should collect the power-ups falling down from enemies because when game goes along, to take on strong enemies, character needs to be more powerful.

In the game, any time that user wants to pause, he can pause and he can continue later on.

In the game, there will be also high score side. The more enemies user kills the more points he will get and the more power-ups he collects the higher his point will be. Strong enemies’ points will be higher than the weak ones. Bosses have the highest points.

The backgrounds of each level is different than other one and the sounds of different enemies and bosses are different.

### 2.1.1 List of Characters



Figure 1: Jon Snow

The main subject of the game.



Figure 2: Solider One

Weak Solider.



Figure 3: Solider Two

This solider’s attacks are stronger than the solider one.



Figure 4: Boss number 1

This is the boss of level 1, Cercei Lannister.



Figure 5: Boss number 2

This is the boss of level 2, Tywin Lannister.



Figure 6: Final Boss

This is the final Boss, Ramsey Bolton.

### 2.1.2 List of Weapons

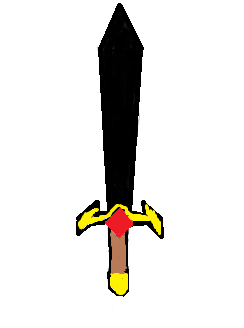


Figure 7: Black Sword

This is the weak sword that user throws.

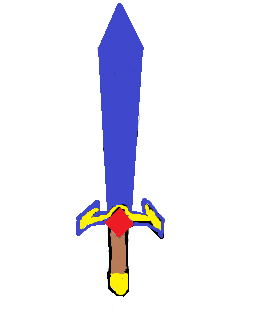


Figure 8: Blue Sword

This is the average strong sword that user throws.

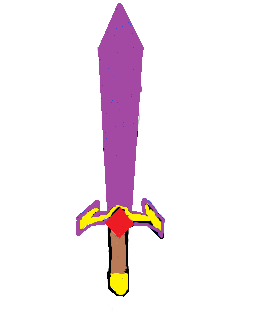


Figure 9: Purple Sword

This is the strongest sword that user throws.

### 2.1.3 List of Bonuses



Figure 10: Speed Up

This is speed increase item that can drop from enemies.



Figure 11: Life

This is +1 extra life item that can drop from enemies.



Figure 12: x2

This is double attack item that can drop from enemies.



Figure 13: x3

This is triple attack item that can drop from enemies.

## 

## 2.2 Functional Requirements

* User will be able to control Jon Snow by using keyboard arrows.
* User will able to throw blades to enemies to eliminate them.
* User will be able to access the help menu that consists of information about how to play the game, items and useful tips.
* The game shall have three different levels. The game is more challenging at higher levels.
* User can see the high scores of top 10 players.
* The user shall be allowed to pause the game and then resume.
* The player character can be buffed with power ups that drop from dead enemies.
* The program will end the game session if the player character is out of health points.
* Enemies will shoot the player character with projectiles when these projectiles collide with player character health points will be removed from the character.
* The game sound can be turned of or on by the user.

## 2.3 Non-functional Requirements

* Control mechanism of the game shall have short response time that allows the player to play with minimal delay.
* Concepts of the levels in the game are easy to understand and react to.
* The game shall offer numerous power-ups.
* Jon Snow object shall respond at most 1 millisecond.
* The game music shall be atmospheric.
* A level shall be harder than previous levels.

## 2.4 Constraints

* The game will be implemented in Java.
* The game will be played with 60 FPS.
* Draws in the game will be smooth.
* The game will run all operating systems that support Java.

## 2.5 Scenarios

### 2.5.1. Play Game

|  |
| --- |
| **Use Case Name**: Play Game  **Primary Actor**: Player  **Entry Condition**: Player hit the “Play Game” button in the Main Menu.  **Exit Condition**: Player selects “Return To Main Menu” from the “Pause Menu”, OR  Player has completed all three levels successfully, OR  Player has lost all three lives before the end of the game.  **Event Flow:**   * Player starts the game. * System creates game environment for player. * Player completes all three levels successfully. * If player has better score than players that are in high scores table, system asks player’s name to save high scores table. * System records player’s name and his/her score to high score table. * Player returns to Main Menu.   **Alternative Event Flow:**   * Player lots his/her 3 lives and game ends. If player has better score than players that are in high scores table, system asks player’s name to save high scores table. Player returns to Main Menu. * Player chooses to exit game before it finishes or before s/he loses all three lives. |

### 2.5.2 Pause Game

|  |
| --- |
| **Use Case Name**: Pause Game  **Primary Actor**: Player  **Entry Condition**: Player hits to “Esc” button.  **Exit Condition**: Player continues to play the game, OR  Player chooses to return to the main menu, OR  Player exits from game.  **Event Flow**:   * Player hits to “Esc” button during the game. * System displays “Pause Menu” * Player goes back to play game again.   **Alternative Event Flows:**   * Player chooses to return main menu. If player has better score than players that are in high scores table, system asks player’s name to save high scores table. Player returns to Main Menu. * Player chooses to exit from whole game. |

### 2.5.3 View High Scores

|  |
| --- |
| **Use Case Name**: View High Scores  **Primary Actor**: Player  **Entry Condition**: Player hits “View High Scores” button in the Main Menu.  **Exit Condition:** Player hits “Back to Main Menu” button in the View High Scores window.  **Event Flow**:   * Player selects the View High Scores in the Main Menu. * System reads highest scores from file and displays these scores on the screen. * Player sees highest scores in the game, then s/he returns to the Main Menu by pressing “Back to Main Menu” button in the View High Scores window. |

### 2.5.4 View Credits

|  |
| --- |
| **Use Case Name**: View Credits  **Primary Actor**: Player  **Entry Condition**: Player hits “View Credits” button in the Main Menu.  **Exit Condition**: Player hits “Back to Main Menu” button in the View Credits window.  **Event Flow:**   * Player selects “View Credits” in the Main Menu. * System displays names of contributors. * After seeing contributors’ names, player returns to the Main Menu. |

### 2.5.5 View Help

|  |
| --- |
| **Use Case Name**: View Help  **Primary Actor:** Player  **Entry Condition**: Player selects “View Help” in the Main Menu, OR  Player selects “View Help” in the Pause Game.  **Exit Condition**: Player returns back to Main Menu, OR  Player returns back to play game.  **Event Flow:**   * Player chooses to display help menu. * System displays information about game and some instructions about how to play game. * Player returns back to Main Menu. |

## 2.6 Use – Case Model

This section provides information about the main use case model of Running Jon.

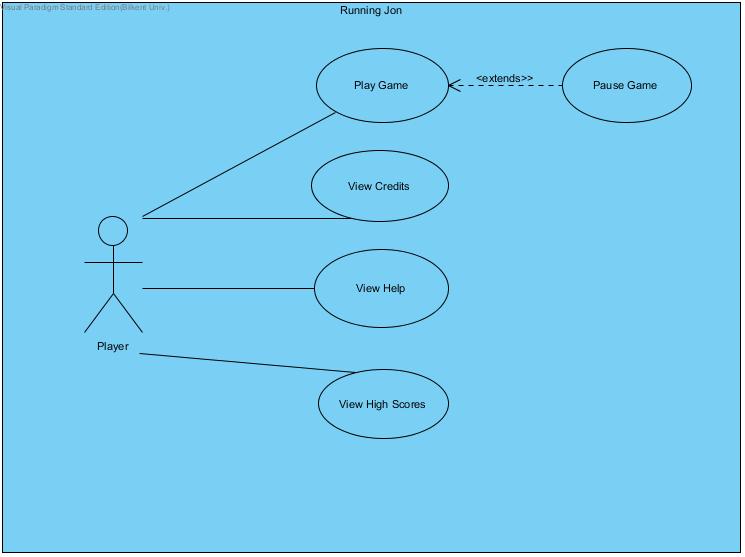


Figure: Use-case model of Running Jon