Bilkent University

Department of Computer

Object Oriented Software Engineering Project

RoM: Redeemers of the Monarchy

Analysis Report

Efe Ulas Akay Seyitoglu, Erkam Berker Senol, Izel Gurbuz, Nashiha Ahmed

Supervisor: Supervisor Name
Course Instructor: Ugur Dogrusoz

Progress
October 15,2016

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Object Oriented Software Engineering Project, CS319

Contents

Analysis Report

RoM: Redeemers of the Monarchy

1 Introduction

Our project is a "Tower Defense" type game. It will be a Java desktop application. Tower Defense is a strategy game. In Tower Defense games, players aim to defend their territories or possessions from enemies by destroying them before they reach the endpoint.

The report aims to give the reader a feel of what our game is and what to expect from it. Specifically, this analysis report gives an overview of the game, functional requirements, non-functional requirements, pseudo-functional requirements, and use-case scenarios and model.

2 Overview

Our game, “Redeemers of the Monarchy,” has only one level in which the player is required to survive all the waves of monsters in order to win the game. The player must do this without letting the life units (2000) of the castle at the endpoint to go zero. Each monster has distinct damage points, described in later sections. The player can destroy the monsters by placing defensive towers at predefined empty areas. These areas are near a path. The monsters can only walk on this path.

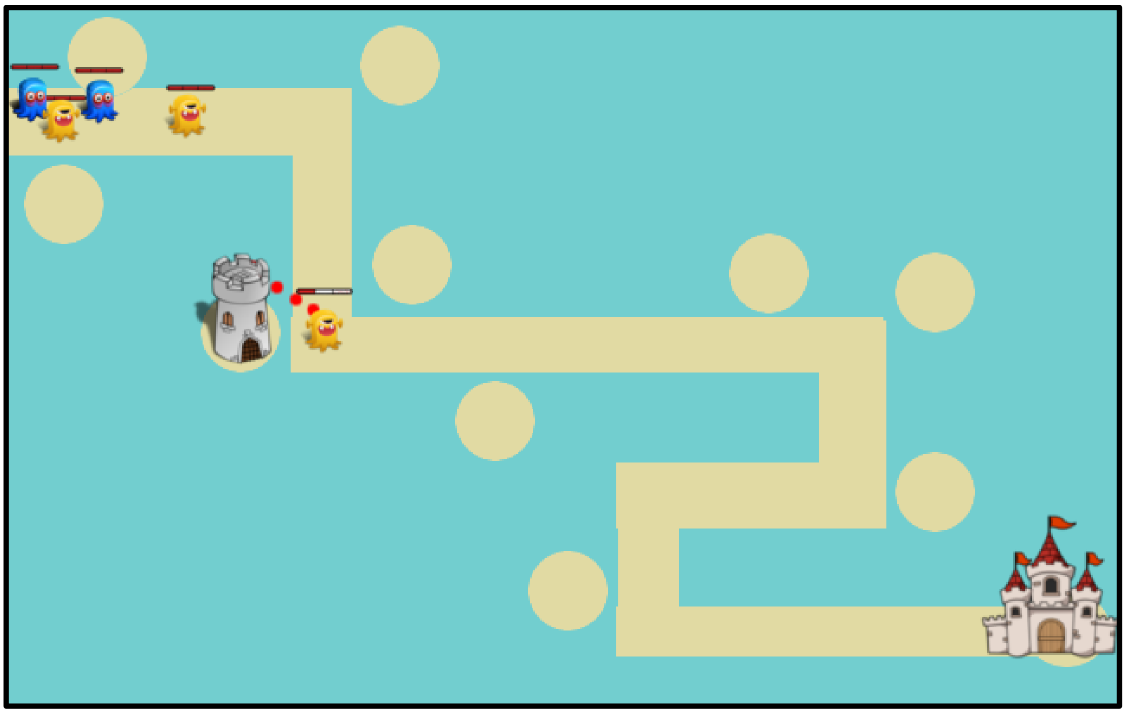


Figure 1-Path of the monsters.

The player is given 300 coins in the beginning of the game. These coins can be used to purchase towers or upgrade them. There will be four types of defense towers with different characteristics (Fire Range, Fire Damage, Number of bullets thrown/second, Time required to set the tower up). Upgrades of the towers will boost all the characteristics of the tower. The player will earn coins as defense towers kill monsters (5 coins per monster). As the game progresses the number of monsters in a wave will increase. There will be 9 waves in total.

The player can pause the game at any time. When game is paused, the player will have the following options: resume game, change settings, view tutorial, and quit game.

The game will have three difficulty levels (easy, medium, and hard.) Depending on difficulty level, certain conditions (number of monsters in a wave, high score of the player) will be applied.

The player will be able to play the game with a mouse and keyboard shortcuts. Additionally there will be a scrollbar to provide the player with a better vision of the game.

At the end of each game, whether completed successfully or not, the player will be given an option to save his/her high score. The highscore of the player will be determined by this formula (the number of seconds survived + remaining money)* Difficulty level (1.0 for easy, 2.0 for normal and 3.0 for hard).

2.1 Initial start of the game:

Initially the player will have 300 coins to purchase and upgrade defense towers. After 30 seconds first wave of monsters will be generated.

2.2 Monsters

Each monster will have a bar shown on top of their head that indicates their remaining health. A monster can only be killed if its health is zero. Monsters will be randomly generated but each of them will have different characteristics. The number of generated monsters will depend on which wave the game is currently in. 5 coins will be won by the player for each destroyed monster

2.2.1 Monster Types

There will be three different monster types each having different capabilities.

Speedy Monsters
Figure 2-Speedy monster [2].

Speedy monsters will be high in speed but they will cause less damage compared to the other monsters.
Health = 100 units
Speed = 200 units
Damage = 100 units

Damaging Monsters
Figure 3-Damaging Monster [3].

These monsters will cause high damage but they will not be as fast as other monsters.
Health = 100 units
Speed = 100 units
Damage = 200 units

Speedy & Damaging Monsters

Figure 4-Speedy & Damaging monster [4].

These monsters will cause high damage but they will not be as fast as other monsters.
Health = 100 units
Speed = 100 units
Damage = 200 units

2.3 Wave of monsters

As the game progresses, the number of monsters generated will also increase. Duration of every wave in this game is a minute. The monsters will be generated randomly according to the difficulty level chosen by the player. Also if the difficulty is easy, then there will be 2 monsters less in each wave and if the difficulty is chosen as hard, then there will be 4 monsters more in each wave.
The details of the number of monsters every wave every minute are bulleted below for medium difficulty:

- At first minute (first wave) , 5 monsters will be generated every 20 seconds.
- At second minute (second wave), 7 monsters will be generated every 20 seconds
- At the third minute (third wave), 10 monsters will be generated every 20 seconds.
- At the fourth minute (fourth wave), 10 monsters will be generated every 15 seconds.
- At the fifth minute (fifth wave), 12 monsters will be generated every 15 seconds.
- At the sixth minute (sixth wave), 14 monsters will be generated every 12 seconds.
- At the seventh minute (seventh wave) 20 monsters will be generated every 12 seconds.
- At the eighth minute (eighth wave) 25 monsters will be generated every 10 seconds.
- At the ninth minute (ninth wave) 50 monsters will be generated every 10 seconds.

If the player can prevent the castle from being overrun by the attacks of the monsters for ten minutes than the game will be won.

2.4 Towers

The player will be given a chance to pick which tower will be placed from the mini-menu at the right side of the screen. The player will be able to drag and drop the tower at the appropriate places (Empty spots near the paths of the map but the road). Each tower will require some time to be set up.

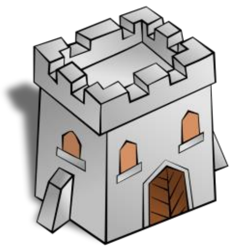
2.4.1 Types of towers

There will be four types of towers. Each tower will have a certain boosted characteristic. The different features of the towers include: Fire Range, Fire Damage, Number of bullets thrown/second, Time required to set the tower up. Each tower will have different graphical appearances and symbols. Thus, the player will be able to see which tower he/she is generating.

Range Tower

Figure 5- Range Tower [5].

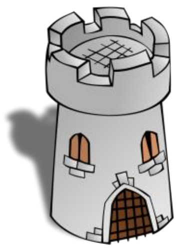
This tower will be able to attack monsters from further ranges. It will have normal capabilities in its other features.
Range = 200 units
Damage = 25 units
Setup time = 30 seconds
Cost = 50 coins
Frequency = 1 bullet in every 3 seconds



Damage Tower

Figure 6-Damage Tower [6].

This tower will be able to attack monsters with higher damage. It will have normal capabilities in its other features.
Range = 100 units
Damage = 50 units
Setup time = 30 seconds
Cost = 50 coins
Frequency = 1 bullet in every 3 seconds



Automatic Tower

Figure 7-Automatic Tower [7].

This tower will be able to attack monsters with more bullets/second. It will have normal capabilities in its other features.
Range = 100 units
Damage = 10 units
Setup time = 30 seconds
Cost = 50 coins
Frequency = 1 bullet in every 1 seconds

Easily-Generated Tower

Figure 8-Easily-Generated Tower [8].

This tower will be set up quicker compared to the other towers. It will have normal capabilities in its other features.
Range = 100 units
Damage = 10 units
Setup time = 5 seconds
Cost = 30 coins
Frequency = 1 bullet in every 3 seconds

2.4.2 Upgrading the Towers

The upgrading of the towers will boost all the characteristics of a given tower. The player can upgrade a tower at most four times.

2.4.3 Cost of Buying & Upgrading Towers

- **Buying a new tower:**
  Range Tower: 200 Coins
  Damage Tower: 200 Coins
  Automatic Tower: 200 Coins
  Easily-Generated Tower: 150 Coins.

- **Upgrading a Tower:**
  The cost of upgrade will not differ by the type of the tower but it will differ in how many times a tower is upgraded.

- First Upgrade: 50 Coins-damage of the tower will increase by %10
- Second Upgrade: 100 Coins-damage of the tower will increase by %30 and number of bullets thrown/second will increase by %20
- Third Upgrade: 200 Coins-damage of the tower will increase by %60, the range of the tower will increase by %20 and the number of bullets thrown/second will increase by %40
- Fourth Upgrade: 400 Coins-damage of the tower will increase by %80, the range of the tower will increase by %50 and the number of bullets thrown/second will increase by %60

2.5 Castle

Figure 9- Castle [9].

In this game, defending the castle will be the objective of the player. Castle will have a bar on top of it that shows its remaining health. As monsters get into the castle, the remaining health will be reduced by the amount of damage the monster possesses. Castle will initially have 2000 units of health.

3 Functional Requirements

3.1 Play Game

Player will be directed to the game "Redeemers of the Monarchy."

3.2 Pause Game

When the game is paused, everything stays frozen until the player resumes the game. Player is presented with options (resume game, change settings, view tutorial, and quit.)

3.3 Resume Game

The player will be redirected to the game that is paused.

3.4 Settings

The player will be able to:

- Change the difficulty of the game, there will be three options as easy, medium and hard. If the player does not modify the difficulty, it will be medium by default.
- Set music volume and sound effects volume.

3.5 View Tutorial

The player can get information about:

- Rules and explanation of the game
- Keyboard shortcuts
- Information about monster types, tower types and tower power ups.

3.6 View High Scores

The game will ask for the name of the player after the game finishes. High score table will include 10 highest points with the player name.

3.7 Credits

The player will see the contact information of the developers of the program.

3.8 Quit Game

The player exits from the game.

4 Non-functional Requirements

4.1 Response time

When the player wants to build a tower, there should not be any delay to improve game experience.

4.2 Smoothness

The graphics of the game will be smooth to make the game more playable because there are lots of movements included in monsters' actions and towers' attack consecutively.

4.3 Audio

The audio of the game will be changed to a fighting music when the monsters are on the attack, and the default music will be a pleasing audio. Also there will be a particular sound when a monster dies.

4.4 Extendibility

Extendibility is an important concept in software world. The game will be extendible for us to be able to add new upgrade styles or tower types or monster types etc. for the newer versions of the game.

5 Pseudo Requirements

The game will be implemented in Java. It will be a desktop application. Adobe Photoshop cs6 and Gimp v2 will be used for some graphical objects.

6 System Models

6.1 Scenarios

Use case name: Play
Participating actors: Player
Entry condition: Player has already opened the application but has not started a new game.
Exit condition:
-Player has won the game
-Player lost the game
-Player paused the game
-Player quits the game
Main flow of events:

1. Player starts the game
2. The system creates the game
3. Player plays the game. System acts according to how player plays.
4. Player wins the game with score above tenth highest score
5. Player's score is displayed.
6. Player enters name
7. System stores name and score
8. System changes high score table to include name and score of player. System displays this
9. Player quits the game

Alternative flow of events:

Alternative 1:
1. Player starts the game
2. The system creates the game
3. Player plays the game. System acts according to how player plays.
4. Player loses game
5. Player quits game

Alternative 2:
1. Player starts the game
2. The system creates the game
3. Player plays the game. System acts according to how player plays.
4. Player pauses the game where player changes settings
5. Player resumes game

The rest of the flow of events continue according to steps 3-8 in Main flow of events or 3-4 in Alternative 1

Alternative 3:
1. Player starts the game
2. The system creates the game
3. Player plays the game. System acts according to how player plays.
4. Player pauses the game
5. Player quits the game

Use case name: ChangeSettings
Participating actors: Player
Entry condition: Player has opened the application. Player has not started the game and will change settings before start of game. Alternative is that player is has paused game while playing.
Exit condition: Player starts or resumes playing game.
Main flow of events:
1. System displays settings
2. Player views settings
3. Player changes settings
4. The system saves settings and alters game accordingly
5. Player resumes or starts game

Alternative flow of events:
1. System displays settings
2. Player views settings
3. Player resumes or starts game without changing settings

Use case name: ViewTutorial
Participating actors: Player
Entry condition: Player has already opened the application but has not started playing the game. Alternative is that player has paused the game while playing.
Exit condition: Player starts or resumes playing the game
Main flow of events:
1. System displays tutorial
2. Player views tutorial
3. Player resumes or starts game

Alternative flow of events: --

Use case name: ViewHighScore
Participating actors: Player

Entry condition: Player has already opened the application but has not started playing. Alternative is player wins the game with a high score above the stored tenth high score.
Exit condition: Player starts playing the game. Alternative is player quits game after being placed in high score table.
Main flow of events:

1. System displays high score
2. Player views high score
3. Player starts game

Alternative flow of events:
Alternative 1:

1. Player plays game
2. Player wins game with high score above the stored tenth high score
3. System displays score and asks player name
4. System stores player name and score and modifies high score table accordingly
5. System displays high score table
6. Player views high score table
7. Player quits game

Alternative 2:

1. Player plays game
2. Player ends game with either win with high score lower than tenth high score or loses game.
3. System displays score and asks player name.
4. System displays high score table
5. Player views high score table
6. Player quits game

Use case name: ViewCredits
Participating actors: Player
Entry condition: Player has already opened the application but has not started playing the game.
Exit condition: Player starts playing the game
Main flow of events:

1. System displays credits
2. Player views credits
3. Player resumes to start the game

Alternative flow of events: --

Use case name: Pause
Participating actors: Player
Entry condition: Player has already opened the application. Player is playing the game.
Exit condition: Player resumes game. Alternative is player quits
Main flow of events:

1. Player plays the game
2. Player chooses to pause game
3. System displays options during pause game
4. Player chooses options. Options' flow of events are according to above use cases
5. Player resumes game*

Alternative flow of events: If quit option is chosen, player does not resume game. Player quits game.

Use case name: Quit
Participating actors: Player

Entry condition: Player has already opened the application. Player has not started the game. Alternatives are player pauses the game or player finishes the game with win or loss.
Exit condition: Player chooses to quit. Player exits application.
Main flow of events:

6. System gives player quit option before playing game
7. Player chooses quit.
8. System closes application

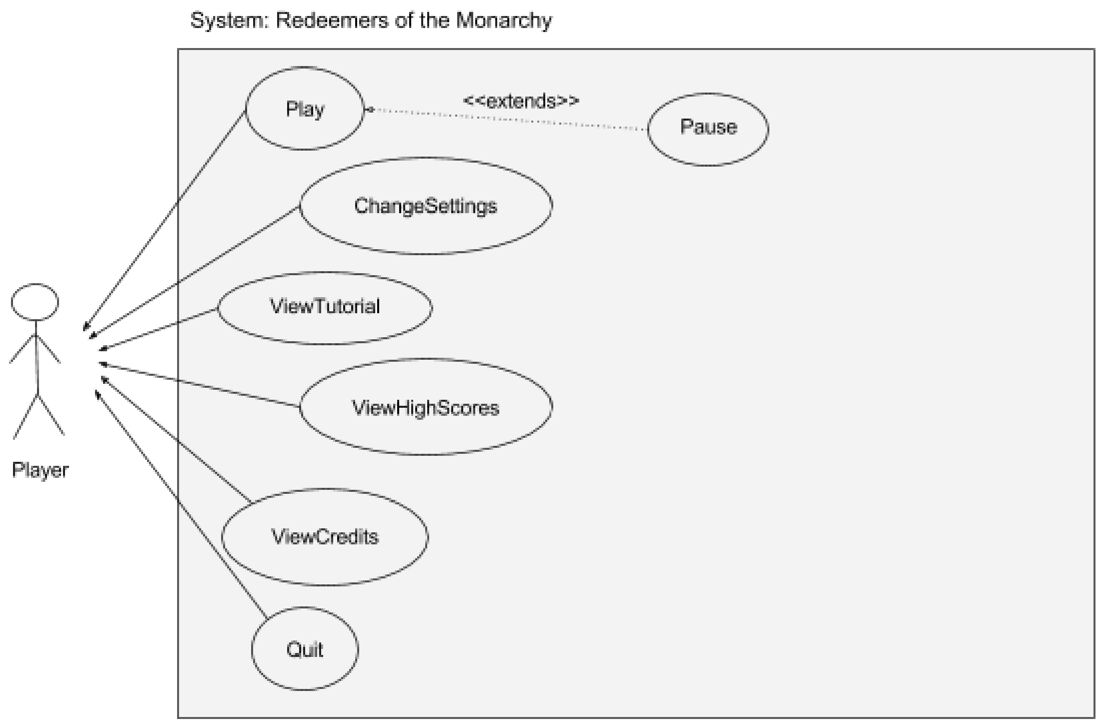
Alternative flow of events:
Alternative 1:

1. Player plays game. Player pauses game
2. Player chooses quit
3. System closes application

Alternative 2:

1. Player plays game. Player finishes game.
2. Player chooses quit.
3. System closes application.

6.2 Use-Case Model



7 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.

[2] Slimy Blue Monster, Viewed 10 October 2016, <http://iconbug.com/detail/icon/2063/slimy-blue-monster/>.

[3] Yellow Cyclops Monster, Viewed 10 October 2016, <http://iconbug.com/detail/icon/2048/yellow-cyclops-monster/>.

[4] Green Monster With Eye Patch, Viewed 10 October 2016, <http://iconbug.com/detail/icon/2059/green-monster-with-eye-patch/>.

[5] Tower Clipart, Viewed 10 October 2016, <http://www.clipartpanda.com/categories/tower-clipart>.

[6] Tower Clipart, Viewed 10 October 2016, <http://www.clipartpanda.com/categories/tower-clipart>.

[7] Tower Clipart, Viewed 10 October 2016, <http://www.clipartpanda.com/categories/tower-clipart>.

[8] Tower Clipart, Viewed 10 October 2016, <http://www.clipartpanda.com/categories/tower-clipart>.

[9] Cartoon Castle Clip Art, Viewed 10 October 2016, <http://www.clipartlord.com/category/structures-clip-art/castle-clip-art/ >.