Requirements and Analysis Document for Group 8

Table of Contents

Version: 1

Date: 2013-03-20

Author: Julia Friberg, Emma Westman, Oskar Dahlberg, Jonathan Thunberg

This version overrides all previous versions.

1 Introduction

The following subsections gives you a short description of the project.

1.1 Purpose of application

The purpose of the application is to entertain those who play the game. The game is of the genre tower defence. It will be possible to build various kinds of towers and the game will have different levels depending on how well it is played. We have been inspired by similar games already existing on the internet and the basic rules are the same in our game as in those. Rules and abbreviations are explained later in the document.

1.2 General characteristics of application

The application is a desktop application and single player application.

The Tower Defence genre is well establish with these general rules:
Monsters follows a path from start to finish and during that time the player has to stop them. The player does this by placing towers along the path to shoot at the monsters and thereby stop them. If enough monsters manage to get to the finish line the player loses, otherwise the player wins and can continue to the next level.

Before the first wave is sent on to the gameboard the player has unlimited time to build towers and when the player is ready the game can be started and the first wave is sent.
There on out the player has the option to either send the waves earlier or let the time run out and the wave is sent by the system.

If a monster is killed the player is rewarded with resources as well as a higher score.

The player can upgrade and build new towers with the earned resources.

If the player where to sell a tower the returned resources would be lower than the cost of the tower.

1.3 Scope of application

The game is a single player game and the progress is saved between played levels. Therefore the player can not exit a level and continue where she left.
For each level there is a high score. Once a level is completed a new level is unlocked.

1.4 Objectives and success criteria of the project

1. It should be possible to play at least three levels with at least five waves for each level.

2. There should be at least four different towers available for the player to build. Each tower should have the ability to be upgraded more than one time.

3. There should be four different monsters.

1.5 Definitions, acronyms and abbreviations

- GUI, graphical user interface
- Java, platform independent programming language, must be Java 1.6 on mac OSX
- JRE, the Java Runtime Environment. Additional software needed to run an Java application, must be Java 1.6 on mac OSX.
- Wave, a group of monsters follows the path and if they enter the radius of a tower they get shot at and loses life. When the monsters loses enough life they die, but if they manage to get to the end of the path the player loses one life.
- Level, one level contains a number of waves. If all waves passes and the player still has at least one life left the level has been successfully completed.

- Resources, the amount of money the player has.

2 Requirements

In this section we specify all requirements

2.1 Functional requirements

1. Start a new game
2. Resume a game
   (a) Choose a level from the map
3. Towers
   (a) Build a tower
   (b) Sell a tower
   (c) Upgrade a tower
4. Wave
   (a) Send in monsters earlier
5. Exit the application

2.2 Non-functional requirements

To be able to run the game the hosting computer need to have Java 1.6 and Slick2d.

2.2.1 Usability

Instructions should not be necessary, the game itself should be easy to understand. The game will be tested by four or more different none-computer-professionals to verify the usability.
The game is in English because it is a global language and the game will be understood by a broader audience.

2.2.2 Reliability

NA

2.2.3 Performance

All actions the player does should have immediate response time.

2.2.4 Supportability

The application should be implemented so the GUI easily can be replaced with another or modified to suit another platforms like for example web or mobile apps.

There should be some test included to verify that the model works correctly.

2.2.5 Implementation

To run the application all hosts need a Java environment installed and configured and Slick2D must also be installed. To be able to run the application on mac OSX Java version 1.6 or lower is required to work with Slick2D.

2.2.6 Packaging and installation

Installation:
1. To launch the application the host needs to have Java Runtime Environment (1.6 or lower for mac computers) and Slick2D installed.
2. Launch the application by running the Main class that is found in the TowerDefence package.

2.2.7 Legal
There are legal issues regarding the smurf trademark and this is not covered here.

2.3 Application models

2.3.1 Use case model
See APPENDIX

2.3.2 Use cases priority

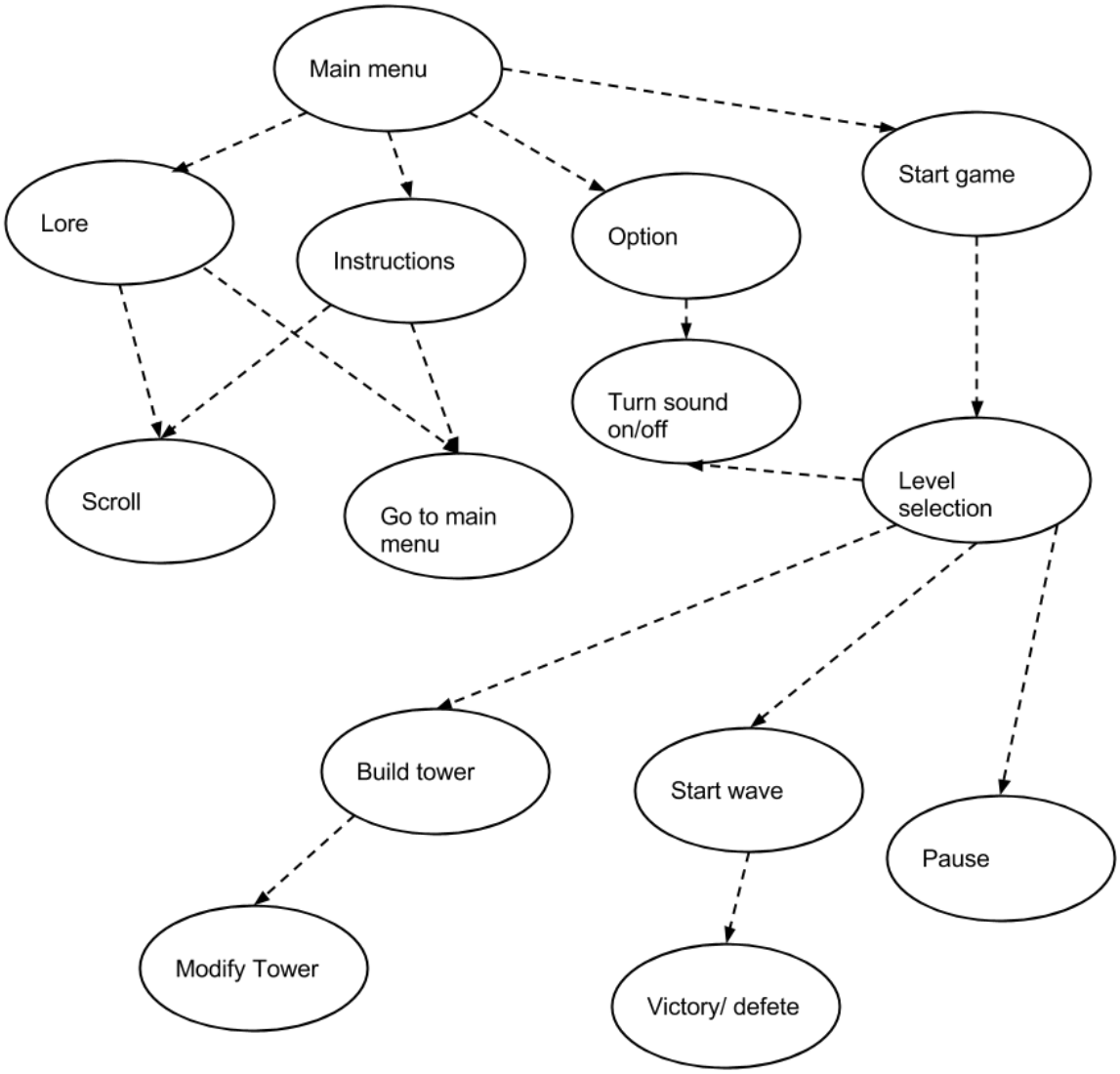
1. Wave
2. Build tower
3. Victory/defeat
4. Main menu
5. Map/ level selection
6. Modify tower
7. Pause
8. Turn on/off sound
9. Options
10. Go back to main menu
11. Scroll

2.3.3 Analysis model
For UML see APPENDIX

2.3.4 User interface
Minimum required size for screen: 800*600
The GUI will use fixed sizes and requires a minimum size of 800 x 600 pixels. The GUI is not able to resize or change the games theme.

2.4 References

APPENDIX
Use case
Overview



Use case texts

Use Case: Build tower
Summary: This is how the player builds a tower on the game board.
Priority: High

Extends:
Includes:
Participators: The player, the application

Normal flow of events:

	Player	System
1.	Select a square on the game board.	
2.		Show selected square with a circle around it containing the different choices of towers, if resources are too low towers are disabled.
3.	User selects a tower.	
4.		The selected tower is placed in the selected square.
5.		Updates resources.

Use Case: Pause
Summary: This is how the player starts and pauses the game.
Priority: Low
Extends:
Includes:
Participators: The player, the application
Normal flow of events:

	Player	System
1.	Clicks on pause button.	
2.		The game is paused and a pause screen with four options are shown. 1: “Continue” 2: “Main Menu” 3: “Restart level” 4: “Quit Level”
3.	Player selects “Resume”.	
4		The game is played again.

Alternative flow
Flow 3.1
Go to Main Menu

	Player	System
3.1.1	Player selects “Main Menu”.	
3.1.2		The main menu is shown.

Flow 3.2
Restart level

	Player	System
3.2.1	Player selects restart level.	
3.2.2		Restarts the current level.

Flow 3.3
Quit level

	Player	System
3.3.1	Player selects “Quit Level”.	
3.3.2		Shows the map (level selection screen).

Use Case: Turn on/off sound
Summary: This is how all of the sound is turned on/off.
Priority: Low
Extends:
Includes:
Participators: The player, the application
Normal flow of events:

	Player	System
1.	Clicks on sound Button.	
2.		Sound is turned off/on, symbol changes from on to off/ from off to on.

Use Case: Wave
Summary: This is how a new wave of monsters is sent on to the game board.
Priority: High
Extends:
Includes:
Participators: (The player), the application
Normal flow of events:

	Player	System
1.	Clicks on timer icon	
2.		Sends in new wave of monsters.
3.		Monsters follows the road.
4.		Monsters finish.

Alternative flow
Flow 1.1
Send in new wave when time is not up

	Player	System
1.1.1		Time is up.
1.1.2		Sends in new wave.

Flow 4.1
Monsters get hit by projectiles and dies.

	Player	System
4.1.1		Monster enters the range of a tower
4.1.2		Tower fires projectiles.
4.1.3		Monster get hit by projectile.
4.1.4		Monster dies.

4.1.5		Resources and points are added to the player.

Flow 4.1.3.
Monster get hit by projectile and makes it to the finish line.

	Player	System
4.1.3.1		Monster get hit by projectiles
4.1.3.2		Monster passes finish line.
4.1.3.3		Player loses life.

Use Case: Modify tower
Summary: This is how the player can upgrade an already existing tower or sell it.
Priority: Medium
Extends: Build tower
Includes:
Participators: The player, the application
Normal flow of events:

	Player	System
1.	Clicks on tower.	
2.		Show selected tower with two buttons under it one for the upgrade and one to sell the tower, if resources are too low upgrades are disabled. The towers shooting range is shown here as well.
3.	Selects upgrade.	
4.		Upgrades tower.
5.		Updates resources.

Alternative flow
Flow 3.1
Sell tower

	Player	System

3.1.1	Selects “Sell tower”.	
3.1.2		Removes tower from game board.
3.1.3		Adds resources to the player.

Use case: Main menu
Summary: Shows the Main menu
Priority: Medium
Extends:
Includes: Options
Participators: The player, the application
Normal flow of events:

	Player	System
1.	Clicks “Start Game”.	
2.		The map is shown with the progress made before exiting the last time.

Alternative flows:
Flow 1.1
Instructions

	Player	System
1.1.1	Clicks “Instructions”.	
1.1.2		Changes view to a instruction text.

Flow 1.2
Settings

	Player	System
1.2.1	Clicks “Options”.	
1.2.2		Options screen is shown (see use case “Options” for more information).

Flow 1.3
Lore

	Player	System
1.3.1	Clicks “Lore”.	
1.3.2		Lore screen is shown.

Use case: Map / level selection
Summary: Shows a map where the players progress can be seen and the player can choose the level he/she desires.
Priority: Medium
Extends:
Includes:
Participators: The player, the application
Normal flow of events:

	Player	System
1	Clicks on level button	
2.		Changes the state of the game and starts the level.

Alternativ flow:
Flow 1.1
Go back to main menu

	Player	System
1	Clicks on go back button	
2.		Shows the main menu.

Use Case: Victory/Defeat
Summary: When the level has ended a screen with score, the outcome and options of how to continue appears.
Priority: Medium

Extends:
Includes: Main menu
Participators: The player, the application
Normal flow of events:

	Player	System
1.		The level is over. Show end screen with three differer options: 1. ”Continue” 2. “Restart” 3. “Main menu”
2.	Chooses continue.	
3.		Shows map screen.

Alternative flows:
Flow 1.1
Restart

	Player	System
2.1.1	Clicks “restart”.	
1.1.2		Restarts the level.

Flow 1.2
Main menu

	Player	System
1.2.1	Clicks “Main menu”.	
1.2.2		Main Menu screen is shown.

Use Case: Options
Summary: The settings a player can make in the game.
Priority: Medium
Extends: Main menu
Includes: Turn on/ off sound
Participators: The player, the application
Normal flow of events:

	Player	System
1.	Uses sound slider.	
2.		Changes the sound of the background music or soun effects depending on which slider that been changed

Alternative flows:
Flow 1.1
Start over

	Player	System
1.1.1	Clicks “Start over” button.	
2.1.2		The progress and all high scores will be reset and tak the player to the Main menu.

Flow 1.2
Go back to Main menu

	Player	System
1.2.1	Clicks on back button.	
2.2.2		The main menu is shown.

Flow 1.3
Turn on/ off sound

Use Case: Scroll
Summary: How to scroll on the page on lore and instruction page.
Priority: Medium
Extends: Main menu
Includes:
Participators: The player, the application
Normal flow of events:

	Player	System

1	Clicks on down button.	
2		The text scrolls up on the page.

Alternative flows:
Flow 1.1
Scroll

	Player	System
1.1.1	Clicks on up button.	
2.1.2		The text scrolls down on the page.

Use Case: Go back to main menu
Summary: How to go back to main menu.
Priority: Medium
Extends: Main menu
Includes:
Participators: The player, the application
Normal flow of events:

	Player	System
1.1.1	Clicks on back button.	
2.1.2		The main menu is shown

GUI

Analysis model

Model UML-diagram