Requirements and Analysis Document for Operation 5A (RAD)

Table of Contents

Version: 1.0

Date 2013-05-23

Author
Linus Hagvall, Vidar Eriksson, Martin Calleberg, Jonatan Magnusson

1 Introduction

We want to create a top-down shooter game for PC. The game's purpose is to survive as long as possible by collecting different resources and avoiding or killing enemies. When the player dies a score will be displayed and saved on a high score list.

The player will be able to navigate a 2D map with the keys W, A, S, and D and shoot with left mouse button. The crosshair will be controlled with the mouse.

1.1 Purpose of application

The purpose of the project is to create a game - Operation 5A. This game's purpose is to entertain the user.

1.2 General characteristics of application

The game will be a top-down shooter game with the objective to survive as long as possible. The player will have status bars to measure his food level, hit points and two integers representing amount of ammunition in the weapon and the amount of ammunition he carries. Enemies will try to approach and hit the player which will result in a drop in the player's hit points. The game is over when the player's hit points reaches zero. There will spawn items (weapons, ammunition, medical pack and food) and enemies at semi random time and places.

1.3 Scope of application

The functions we want the game to have:

- A world in which the player can move around in.
- Enemies that hunts the player.
- Functioning weapons (both range and melee).
- Health, ammunition, food, score indicators to keep track of the players stats.
- Items to pick up to help the player (food, medical pack, ammunition and weapons).
- A menu where the player can view and change simple settings, check highscore, save or load an old game and start a new game.

1.4 Objectives and success criteria of the project

We will consider the project done when we have all the points above (see 1.3) completed.

1.5 Definitions, acronyms and abbreviations

In this document we will use the following definitions, acronyms and abbreviations:

- GUI - graphical user interface
- JRE - Java runtime environment

2 Requirements

2.1 Functional requirements

1. Start a new game.
2. The player should be able to:
    a. move around,

    b drop items,
    c pick up new items,
    d hit enemies.
3 The player should be able to run out of food.
4 Enemies should spawn in the world and be able to chase and hurt the player.
5 The player should be able to die.
6 The game should keep a score of the current game.
7 End and/or exit a game.

2.2 Non-functional requirements

2.2.1 Usability

The game should be easy to understand for first time users, using a standard GUI which is often seen in similar games. Even without experience the game should be fairly easy to understand. See APPENDIX for GUI related images.

It should be easy to implement more languages than the default language (English). We will at least translate Swedish as well. There should be an user manual, however, this does not have to be translated to more languages than English.

2.2.2 Reliability

The game should not crash and is required to handle possible exception in a preferable way.

2.2.3 Performance

The game should not have any noticeable lag and should run at a minimum of 30 frames/second.

2.2.4 Supportability

The application must be implemented so that the GUI can easily be modified to suit other platforms, such as mobile devices or web applications.

There should be automated tests for all the classes in the analysis model. Tests for these classes, and the classes, should be kept in separate source folders.

2.2.5 Implementation

We will use Java to implement the code and use JRE 7.0 when writing the application. Therefore, to run the application as it is intended this or a newer version must be installed and functional on the user’s computer.

2.2.6 Packaging and installation

The program is packed as a zip archive. The zip archive contains the following:

- A file with the code (a standard runnable Java .jar file)
- All required resources.
- A README-file.

2.2.7 Legal

The development environments and plugins are all open source. We ourself have created all the images used in the GUI and ingame.

2.3 Application models

2.3.1 Use case model

See APPENDIX for UML diagram.

2.3.2 Use cases priority

1. Time passes
2. Move Player
3. Use weapon
4. Enemy hit by player
5. Player hit by enemy
6. Game over
7. Pick up item
8. Drop item
9. Player runs out of food
10. Player switch weapon
11. Start new game
12. Highscore
13. Change settings

2.3.3 Analysis model

See APPENDIX for analysis model.

2.3.4 User interface

We will have a menu where the user can, for example, load or start a new game. The GUI for these menus should follow the standard conventions regarding GUI:s. The GUI must take into account different screen sizes, but possibly not very small.

2.4 References

NA

APPENDIX

- Appendix 1 - Use cases UML
- Appendix 2 - Use cases first verision
- Appendix 3 - Use cases final verision
- Appendix 4 - GUI
- Appendix 5 - Analysis model