Requirements and Analysis Document for Project Five (RAD)

Table of Contents

Introduction

1.1 Purpose of application

The purpose of the project is to build an application for a simple, real time 3D-rendered, multi-player game, which could be placed in a “real time tactics”-genre. The application will be easy to use and no expertise should be needed in order to play.

1.2 General characteristics of application

The initial application will be a multi-player game for two or more players. It will be using the jMonkeyEngine supporting all desktop platforms (Windows/Mac/Linux).

The game consists of an ocean-like terrain where each player controls a single unit which he can move across the battlefield. A round ends when a player has eliminated all other player controlled units, which is done by depleting the opponents hit points to zero using projectiles or collisions.

A game ends when a player reaches a given amount of won rounds.

1.3 Scope of application

The application does not include computer controlled units. It is designed to be played with and against other people on a single computer running Linux, Mac or Windows. The application will not save any particular information, between games but should be able to produce and present statistics of the last played game, such as winner.

1.4 Objectives and success criteria of the project

It should be possible to play, and finish, a game with two players using the default settings. The game should be able to start, players should be able to complete a number of rounds, until one player wins enough rounds by being the last man standing. During gameplay, the players must be able to pick up power ups, fire shots and collide with the opponent. After a finished game, the winner must be announced.

1.5 Definitions, acronyms and abbreviations

- **GUI** - Graphical User Interface
- **jMonkeyEngine** - Open Source Java Game Engine. jmonkeyengine.org
- **Desktop** - Stationary computer
- **Hit point** - Sort of a health point. When 0, player is dead.
- **Statistics** - Number of shots fired per player, accuracy, winner etc...
- **Unit** - For example tank, ship, spaceship, humvee etc
- **Powerups** - For example speed, health, extra damage etc

2.1 Functional Requirements

The player(s) should be able to:

Before the game:

- Start a game with settings from text file.

In the game:

- Move the unit forwards.
- Steer the unit left/right.
- Accelerate/slow down speed of unit.
- Fire projectiles.

- Get hurt.
- End game by losing.
- End game by winning.
- Collide with other units and damage it.
- Pick up item that provides unit with powerup.

2.2 Non-functional requirements

2.2.1 Usability

First-time users should be able to start a new game round without any problems. Any necessary information will be displayed on screen.

The game will only be in English since user does not have to be very familiar with the language to be able to play it anyway.

Since the game is very simple, no user manual will be needed. All information required will be available in the game.

2.2.2 Reliability

N/A (A game does not need to be reliable in this meaning)

2.2.3 Performance

The game will work with up to four players without any frame drops on a fairly modern computer using correct open GL drivers.

2.2.4 Supportability

The application will be implemented so that the the game and physics engine is easily replaceable to i.e support other platforms. The estimated time for and an operation like this will not exceed 1 man-month.

Automated tests for each model will be included to ensure that all functionality works. GUI will be tested and results recorded to be included in the final documentation.

2.2.5 Implementation

The application is implemented in Java environment to support Windows, Mac and Linux. The JRE is therefore required in order to run the game. Linux systems using the Mesa OpenGL driver may require additional libraries such as libtxc-dxtn to support DXT1-compression.

2.2.6 Packaging and installation

The whole package will contain:
- A README file documenting how to start and use the game
- An executable jar file to launch the game with
- Resources such as graphics, and other media used by the game.

Packaged into a zip archive.

2.2.7 Legal

There are no known legal issues regarding neither the issued frameworks nor the game title.

2.3.1 Use Case Model

See appendix.

2.3.2 Use Case Priority

1. StartRound
2. Navigate
3. Fire
4. EndRound
5. Collide
6. RunGame
7. EndGame
8. PickupItem
9. Exit
10. Pause

2.3.3 Domain Model

See appendix.

2.3.4 User Interface

The application will use a very minimalistic, fixed (non-resizable, non-themeable) GUI.
Application will run on fullscreen and everything will thus be scaled accordingly.
During game, GUI will be limited to the running screen with the display of health bars.