Requirements and Analysis Document for SSHA (RAD)

Contents

Version 0.9
2013-05-26

Sebastian Bellevik, Adam Jilsén, Tomas Hasselquist, Rasmus Lorentzon

This version overrides all previous versions.

1 Introduction

This section gives a brief overview of the project.

1.1 Purpose of application

To create a multiplayer arena game where players can connect to a server and play against each other with classes and skills. The game is based on a game called Warlock which is a fan-made modification of the game Warcraft 3. The player will have to use skills to either kill the other players or lure them into traps.

1.2 General characteristics of application

It is an arena multiplayer game where the player can move around within the confined space, after selecting a class and skills, and try to kill the other players before they get killed in the game themselves. The game is round-based and each round ends when all but one player are dead.

The map is quite small to make the rounds fairly quick and it is also filled with different traps and various things that can kill the player. After each round the player can upgrade or buy skills and items to help with the next round.

There are several ways to kill the other players. Such ways can be to push them into traps, use skills at them or try to outplay them to lure them into hurting themselves by traps or missplaced skills.

The skills range from various different types such as passive effects that affect the player who used it as well as skills that the player uses on the map in the hope of hitting someone. Some skills affects the player who used such as teleporting the player to another location.

A GUI will be used to display the map, the players, skill being used and the different menu modes such as the shop where the player can upgrade skills. After bying skills the player can put them in a skillbar that holds all the skills the player can use.

1.3 Scope of application

It will contain a stand-alone java based game which can connect to a server. Both client and server will be developed simultaneously to be integrated together. The player may play alone against computer controlled players but the choice to play against other player with the help of a server-client system is possible. No data is saved between sessions so when the game is closed the player will have to start over the next time the game is started.

1.4 Objectives and success criteria of the project

1. Full functionality for the player to move around a map and use skills as intended. All the use-case will have to work.
2. Possibility to connect to a server and play against each other with basic functionality.

1.5 Definitions, acronyms and abbreviations

- GUI, graphical user interface.
- Java, platform independent programming language.
- JRE, the java Run time Environment. Additional software which is necessary to run a Java application.
- Host, the computer the game is run on.
- Round, a breaking point where all players but one is dead. All players are sent back to get ready for another round when this happens.
- Gold (for players), the resource which the player uses to buy and upgrade skills and items.
- Arena, the map where the players can move around and use skills on.

2 Requirements

In this section we specify all requirements

2.1 Functional requirements

Use-cases:

- Possibility to start the game
- Movement of a player
- Attack from a player
- Taking damage and dying as a player
- Using skills on the map
- Getting to the end of the round and start a new one
- Buy something from the store
- Choosing classes
- Connecting to the server

2.2 Non-functional requirements

2.2.1 Usability

Users should be able to understand the basic functionality of the game within a short time period. The user should however have some understanding of games and how to navigate in a most basic way in a game that is controlled with both mouse and keyboard.
The communication between the game and the user must be very clear to shorten the learning period of the user. This is done by using clear menu functionality and descriptions on everything that may be difficult to understand.

2.2.2 Reliability

NA

2.2.3 Performance

The actions the player does in the game should not exceed 1 second with response time in worst case. In the menu the player should not exceed 4 seconds response time due to connecting to a server may take time.

2.2.4 Supportability

The game will implement a server-client system that will be easily modified to add or remove what will be sent and applied to players.

The classes are implemented to make is easier to add new skills and classes.

2.2.5 Implementation

Platform independence is an aim for this game. The library Slick2D does however not support Mac OS when using Java 7. To fulfill that purpose the application will use a Java environment which means that all hosts must have the JRE installed and configured. The application needs to be installed on all hosts where it will run. The game can not be run on Mac OS right now though.

2.2.6 Packaging and installation

The application will contain the following in a zip archive:

- A standard Java jar-file with the application code
- All resources that are needed to run the application.

2.2.7 Legal

There are legal issues regarding skill images taken from various different games, all of which are trade marked. This is not covered here as the game is not meant to be released to the public.
The images are taken from the following companies:

- gPotato
- Arena Net
- Bioware
- Blizzard
- Bethesda
- Riot Games
- Kabam

2.3 Application models

2.3.1 Use case model

See appendix for UML diagram.

2.3.2 Use cases priority

1. Possibility to start the game
2. Movement of a player
3. Attack from a player
4. Taking damage and dying as a player
5. Using skills on the map
6. Getting to the end of the round and start a new one
7. Buy something from the store
8. Choosing classes
9. Possibility for several users to play together using a server-client system

2.3.3 Analysis model

See Appendix

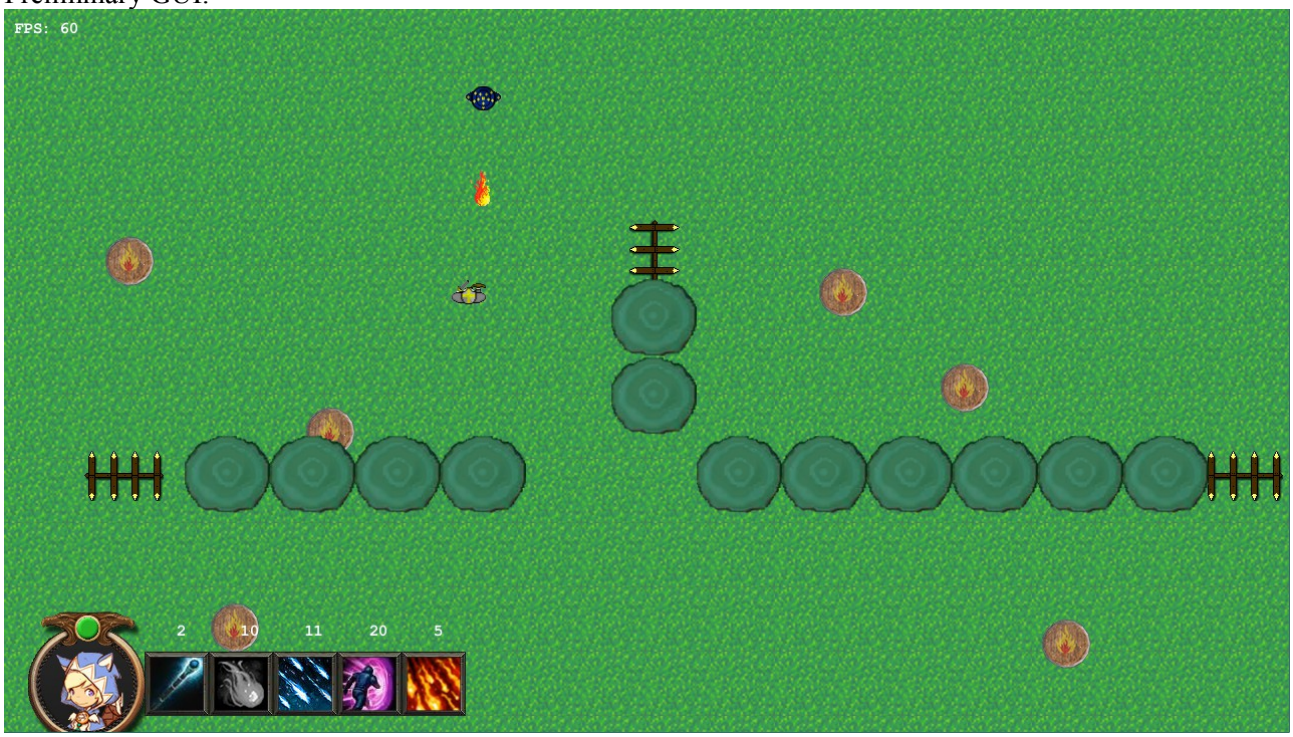
2.3.4 User interface

The application will use a GUI that people familiar with arena games will feel comfortable with. The GUI follows standard conventions by putting the necessary information for the player in the places on the screen the user is familiar with. Currently the game can only be run with 1280x720 pixels.

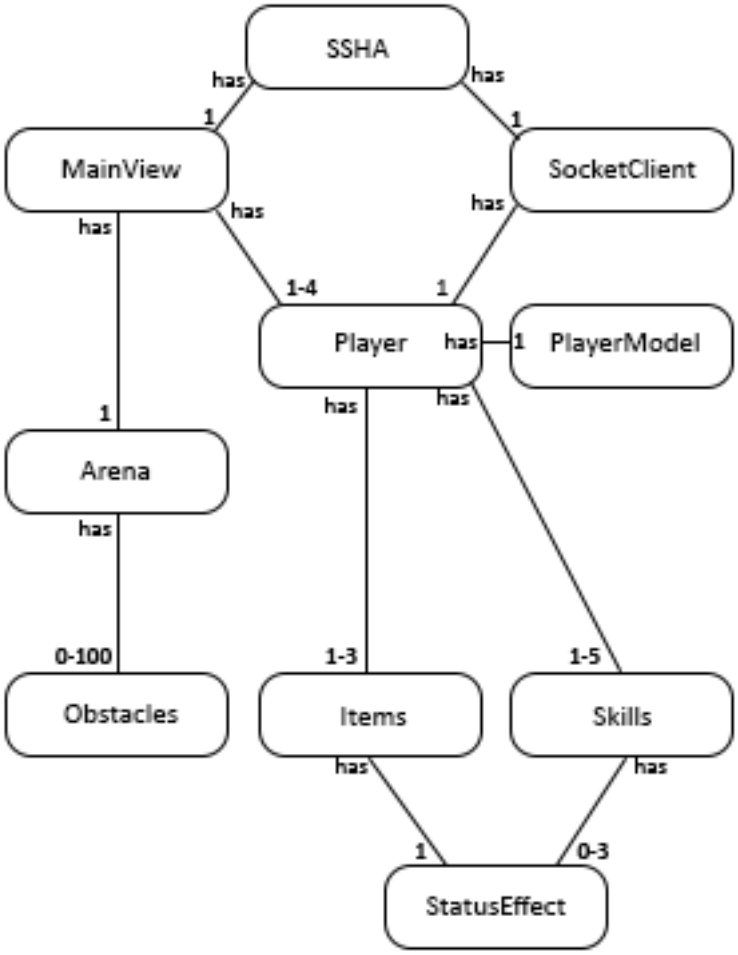
2.4 References

3 APPENDIX

GUI
Preliminary GUI.



Analysis model



Use cases
Overview