Version:

Date

Author

This version overrides all previous versions.

1 Introduction

This section gives a brief overview of the project.

1.1 Purpose of application

Ridiculous Rumble Royale is an attempt at making a fast-paced party game that is easy to pick up and play instantly. Combining the intense combat of classic games such as Super Smash Bro's and the accessibility of multiplayer games.

1.2 General characteristics of application

Ridiculous Rumble Royale is a fast-paced 2D multiplayer action-brawler which supports multiplayer over the internet. Four players fight in a closed arena until there is only one left, rounds are optional.

1.3 Scope of application

Initially it's not possible to play the game alone, however if there is time we may try and implement a computer controlled player (AI). Support for other devices than desktop can, but will not, be implemented. Character creation may be implemented at a later time.

1.4 Objectives and success criteria of the project

The goal of this project is to produce a fighting game of good quality, which supports multiplayer over the internet. Initially the game will only support four players with various attacks and combos, but the intention is for the game to support character creation as well as several different maps and AI for singleplayer games.

1.5 Definitions, acronyms and abbreviations

AI = Artificial intelligence: "computer controlled players"
Action brawler = A game focused on fighting
2D = Two dimensional. For example: "Super mario"

2 Requirements

2.1 Functional requirements

1. Join a game by providing an IP address and a port number.
2. Exit the application.
3. Choose a character.
4. Host a game.
5. Finish the game (last man standing wins).
6. Move the character.
7. Use attacks and abilities.
8. Start a game.

2.2 Non-functional requirements

2.2.1 Usability

The game is considered “usable” when you can play a game with at least one other participant online and win a match by beating your opponent.

2.2.2 Reliability

The application is considered reliable when the online multiplayer is stable under a decent internet connection, thus, not dropping connections.

2.2.3 Performance

THe application is to perform without any noticeable lag and stuttering on a mid level PC.

2.2.4 Supportability

Since the application is what it is, it’s really easy to support, however it will not need future support, since we do not have any dedicated servers. However, future implementations should be supported flawlessly.

2.2.5 Implementation

--

2.2.6 Packaging and installation

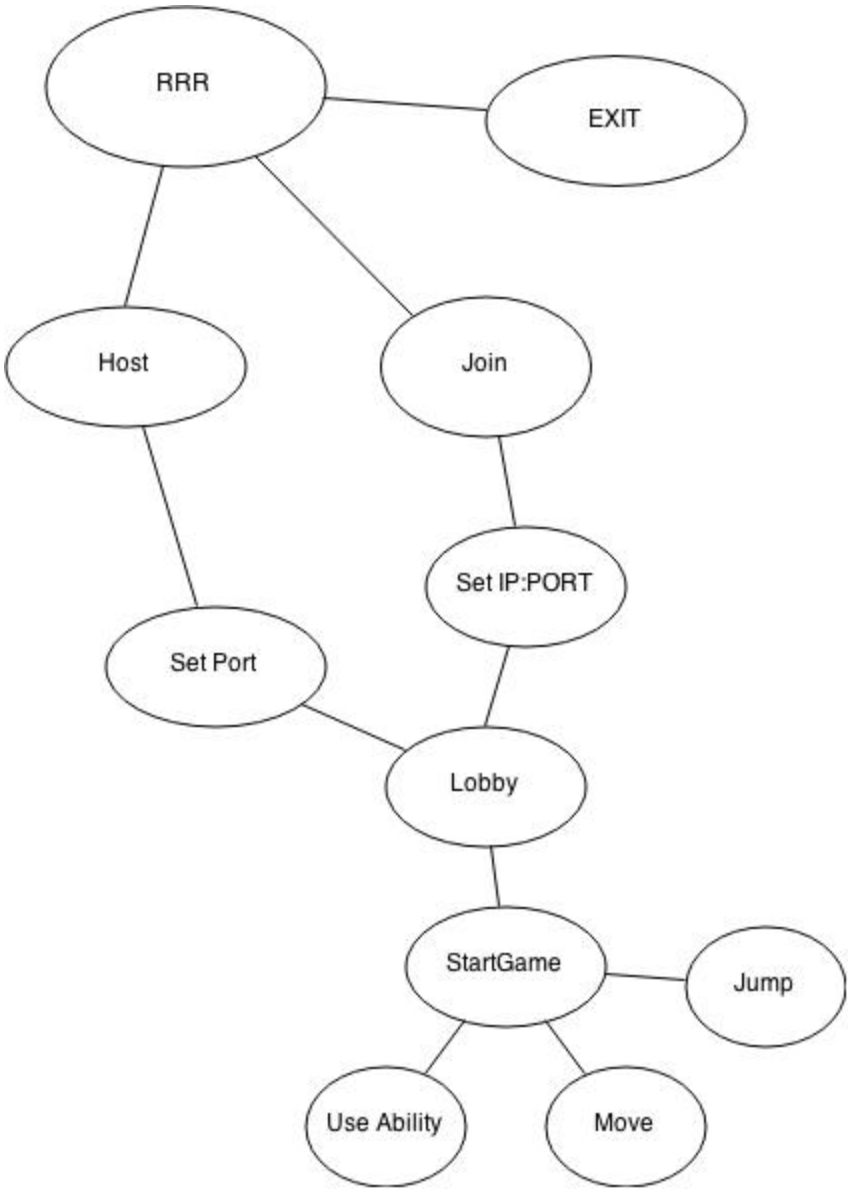
The installation will be as simple as downloading a file(jar) and running it.

2.2.7 Legal

Idk m8

2.3 Application models

2.3.1 Use case model

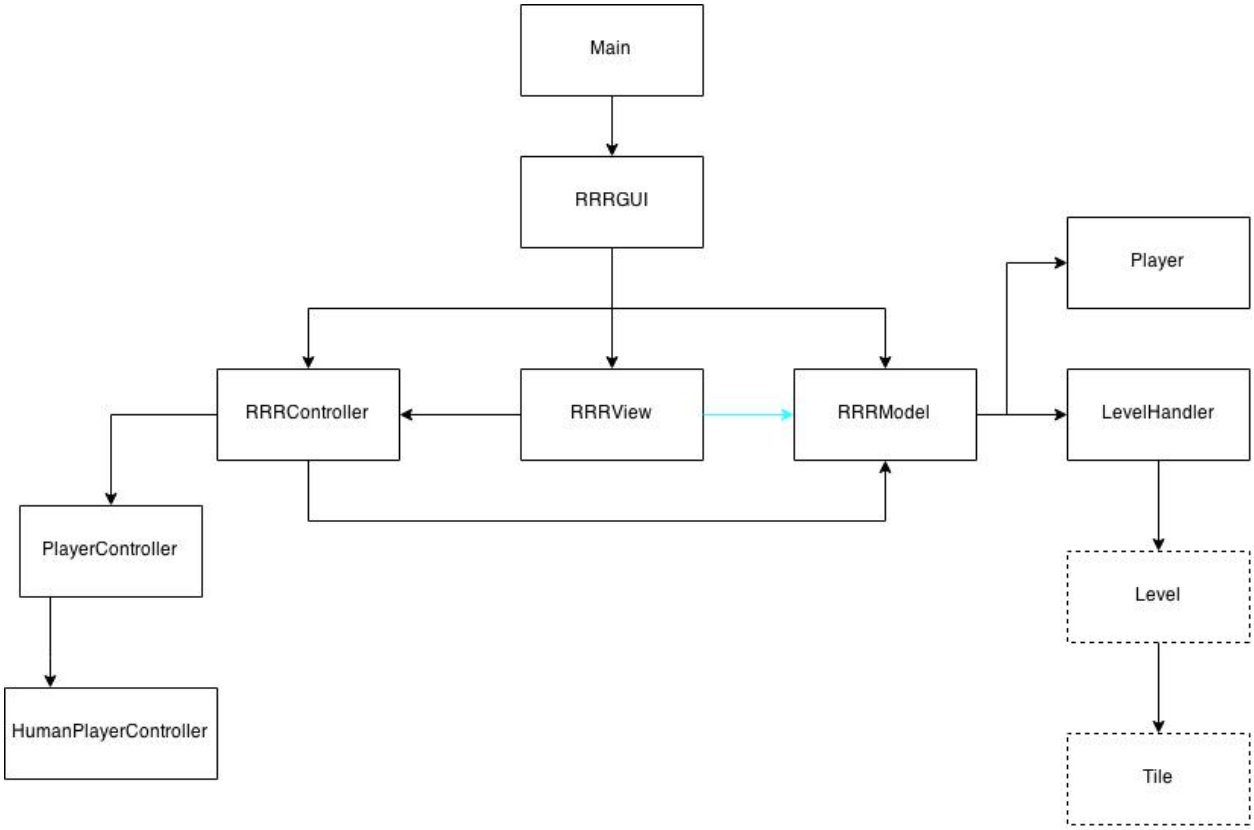


1. Exit button pressed at startmenu
2. Play button pressed at startmenu
3. Escape pressed during play state
4. W pressed during play state
5. S pressed during play state
6. A(/D) pressed during play state

2.3.2 Use cases priority

1. Exit button pressed at startmenu
2. Play button pressed at startmenu
3. A pressed during play state
4. W pressed during play state
5. S pressed during play state
6. Escape pressed during play state

2.3.3 Domain model



UML, possible some text.

2.3.4 User interface

The initial interface will only have two buttons. A play and an exit button. However, in future versions it may include a options button and a character editor.

2.4 References

APPENDIX

GUI

Domain model

Use case texts

c’