Requirements and Analysis Document for Project “Programming United Network Game”

Version: 1.0

Date: 2016-03-24

Author: Martin Sixtensson

Contents:

1 Introduction

1.1 Purpose of application

The purpose of project “Programming United Network Game” is to learn more about how multiplayer games and networks work while creating a multiplayer game.

1.2 General characteristics of application

The application will be a PC application consisting of one server and one client portion. The application will have a simple GUI for things done outside of the actual game. The game itself will consist of a 2D gamespace. A player can either join or host a game. Several people on different computers can join a game before it starts. The application will score the player depending on how well the player does according to the game. The application will end or restart itself when the players lose according to the game rules or the application itself is canceled. There is no situation a player can win, a score is only given when the players have lost.

1.3 Scope of application

“Programming United Network Game” is game in which players must work together to defend a house from hordes of zombies or enemies controlled by the computer. The game takes place in a 2D gamespace to which four players in total can connect and play a game. The game starts when all players are ready and the enemies will start appearing. The players must then aim and shoot at the enemies to destroy them. The wave is ended when all enemies of a wave are destroyed. The game itself, however, only ends when all players are killed by the enemies. The enemies kill the players by getting close to them and attacking them. The players are scored depending on how long they survived and how many enemies they killed.

1.4 Objectives and success criteria of the project

The following objectives determine the success of the project:

- The development of a working client and server for the game
- The design of a working GUI to host and/or join a game
- The design of a 2D gamespace
- The creation of a controllable player
- The creation of enemies controlled by the computer

1.5 Definitions, acronyms and abbreviations

NPC Non-player character. Game characters not controlled by the players.

AI Artificial Intelligence. The functions that control the enemies.

Host The person or computer who hosts the game. Hosting in this instance means having the server running on the selected computer.

Client A client is the part of the program that connects and talks to the server. A player could be considered a client.

Server Controls most of the of the game's different states. Players can create a server to host their own game or a player can join an existing server to play.

Frame Rate How fast the client updates what it sees on the screen. I.e if the the frame rate is 30 frames per second, the client will redraw the position of the enemies and players thirty time per second.

Sprite The skin of the moving enemy or player.

Enemy Controlled in this case by the computer. The goal in this particular game is to destroy/remove enemies.

Map The board or space where the player and enemies can move around.

Tile A map is built from tiles. There are different tiles;

- Wall tile; where players cannot move.
- Floor tile; where players are free to move around.
- Spawner tile; where zombies are spawned/created.

Rate of fire How fast bullets are created from a certain gun.

2 Requirements

2.1 Functional requirements

The player(s) should be able to;

1. Host a new game.
2. Join a new or ongoing game.
3. Move around with the character on the gameboard.
4. Shoot their weapon.

Create a list of high level functions here (from the use cases).

2.2 Non-functional requirements

Possible NA (not applicable).

2.2.1 Usability
The application should be easy to use. A normal user should be able to use the application with some to little computer knowledge.
There should be an english tutorial for both setting up a game and how to play the game.

2.2.2 Reliability
It is very important that the application is reliable. There should not be regular disconnects from the host/server. A game with all four players should be able to play the game reliably for at least 30 minutes.

2.2.3 Performance
The game should run at a stable framerate and at least over 30 frames per second. The delay between the different clients and server should be kept as low as possible.

2.2.4 Supportability
The application only needs to be able to run on a Windows Computer. The application has to be usable on a screen with a resolution of 800x600 or more.

2.2.5 Implementation
The application will use the Java environment. All hosts and clients must have JRE installed and configured.

2.2.6 Packaging and installation
We plan to package the game in a jar file. We will supply a readme file with instructions on how to start the application and how to join or host a game within the application.

2.2.7 Legal
NA

2.3 Application models

2.3.1 Use case model
See appendix for use cases.

2.3.2 Use cases priority

1. Host game
2. Join game
3. Move
4. Shoot
5. Buy weapon

2.3.3 Domain model

See appendix.

2.3.4 User interface

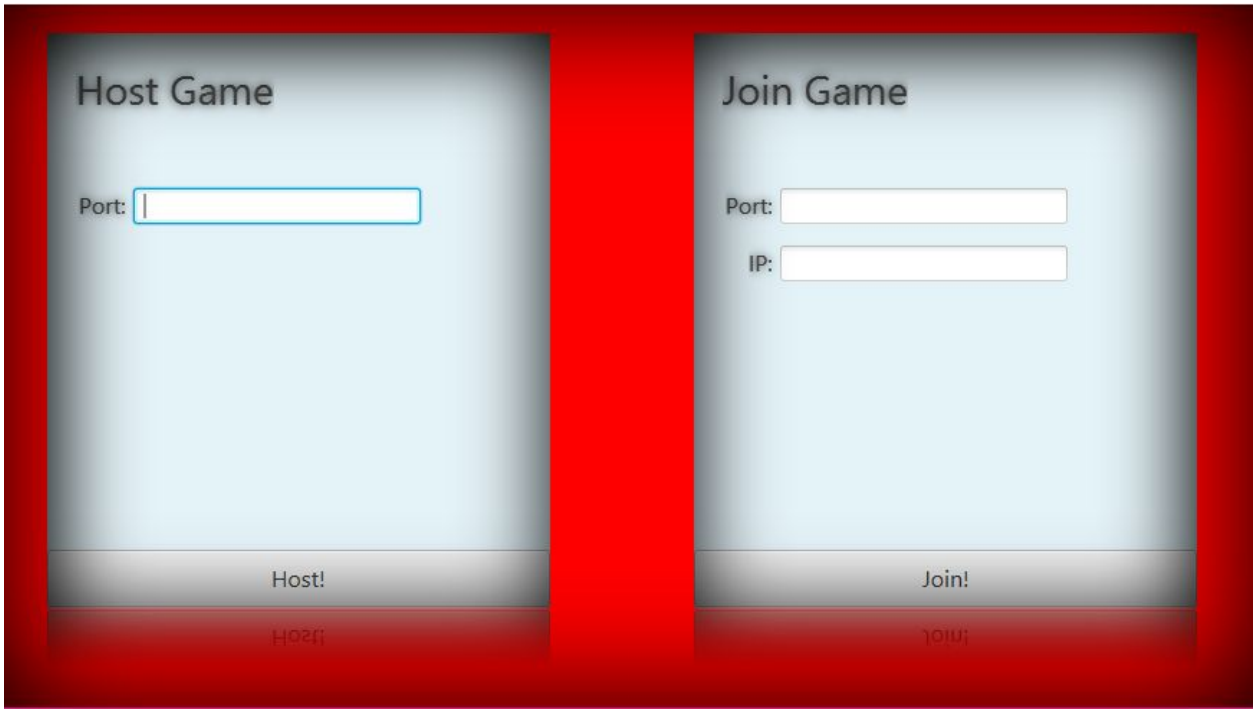
Application will use separate GUI to launch into game. See appendix for screens.

Text to motivate a picture.

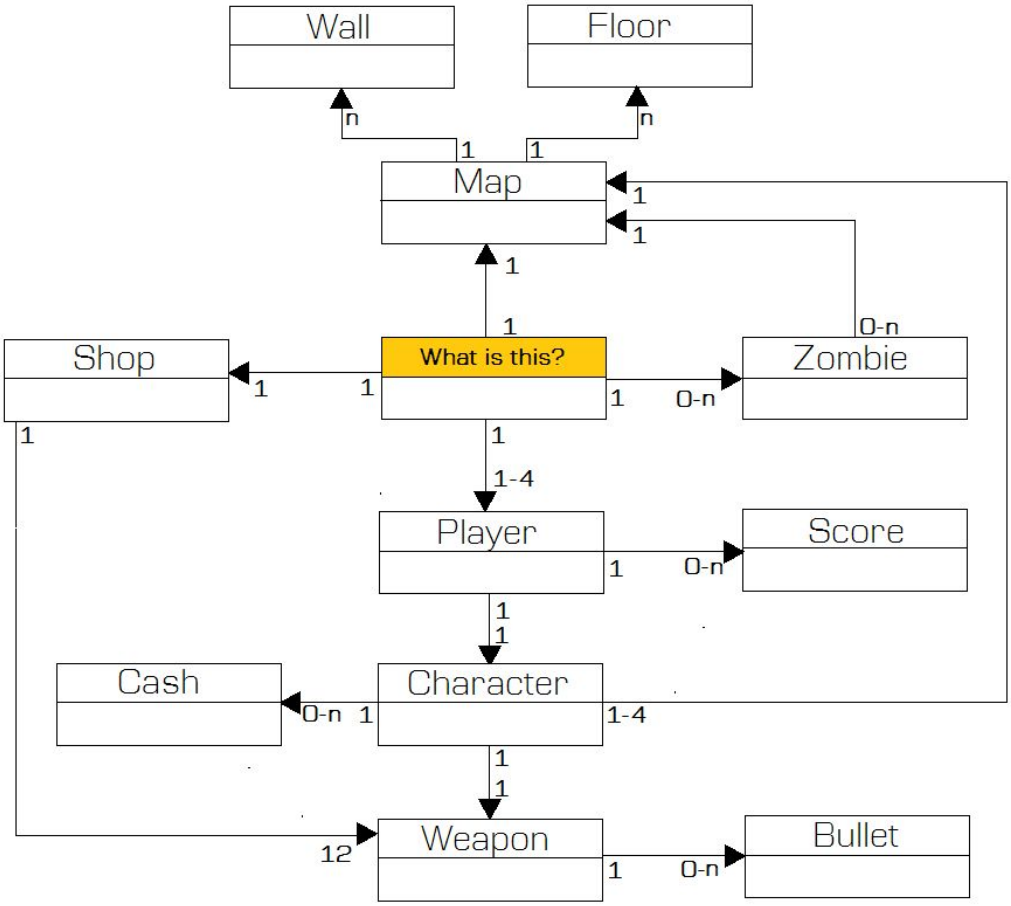
2.4 References

APPENDIX

GUI Screen



Domain model



Use case texts

See separate documents.