ShroomTowerDefence

Requirements and Analysis Document

Contents

Version 1.0

Date May 16, 2012

Authors Emil Edholm, Emil Johansson, Johan Andersson, Johan Gustafsson

This version overrides all previous versions.

1 Introduction

This section gives a brief overview of the project.

1.1 Purpose of application

This project aims to create a variant of a tower defense game. A tower defense game is a game where you have a base or fort and you must defend it from incoming waves of enemies. You do this by building weapons to survive each attack which increases in both intensity and the number of enemies.

1.2 General characteristics of application

The application will be a desktop, standalone, singleplayer application with a graphical user interface for the Windows/Mac/Linux platforms.

The application will be played with a mix of turn based wave releases and realtime enemy/player movement. The player will choose when to activate next wave and will have an unlimited amount of time to build/upgrade his or hers defense. The game will end according to the set rules or when canceled by the player. When the player loses the game a score screen will present him or her with how high score he managed to get calculated from how many enemies the player managed to kill.

1.3 Scope of application

The game will include a pseudo computer player. The computer player will ensure that each level will have different enemies and different amount of enemies. When a player's base gets destroyed the score can be saved in a high score list. The game will also feature a player character which can be moved by the player using the arrow keys. The player character will act as a weapon against the enemy wave.

Enemies can have effects applied to them by using a special kind of towers.

1.4 Objectives and success criteria of the project

A game capable of the doing the criteria stated above. A player should be able to start a new game, build his or hers defences and initialize at least one wave of enemies.

1.5 Definitions, acronyms and abbreviations

Effect this modifies the behaviour of the enemy, such as slowing or stun effect.

2 Requirements

In this section we specify all requirements of the game

2.1 Functional requirements

- Start the game
- Player can build towers (the objects that defends the player)
- Each wave can be started/initialized by the player when ready
- The player should be able to move his or hers character
- End game when user quits or when player dies. (A specified number of enemies has breached the player's defense)
- Exit game

2.2 Non-functional requirements

A map editor may be implemented. This is used to create new maps/levels to the game, but is not needed for gameplay.

2.2.1 Usability

Usability is high priority. Normal users should be able to play the game with a very short learning period.

The application must communicate the state of the game in a very clear fashion. Tests with at least four different non-computer-professional should be performed to verify the usability. Test results should be part of the final documentation. There should be an English user manual, how to play the game.

2.2.2 Reliability

N/A

2.2.3 Performance

Any actions initiated by the player should not exceed a 1 second response time in worst case.

2.2.4 Supportability

The application must be implemented so that the GUI is easily modifiable to suit other platforms (Web, Mobile Apps, Pads, e.t.c.). The estimated time to adapt the GUI to a web based application should not exceed 160 work hours.

The implementation should prepare for the dividing of the application into a client/server architecture for net based games. It should be easy to partitioning the application into a client-server architecture. A time estimation for this should be included.

There should be automated test verifying all use cases. Code related to the GUI could be tested manually. GUI test should be recorded and included in the final documentation

2.2.5 Implementation

To achieve platform independence the application will use the Java environment. All hosts must have the JRE ¿= 1.7 installed and configured. The application needs to be installed on all hosts where it will run (possibly downloaded).

2.2.6 Packaging and installation

The application will be delivered as an zip-archive containing:

1. A file for the application code (a standard Java jar-file).
2. All needed resources, images, icons, sound, e.t.c.
3. Start programs (scripts) to start the game on the different platforms.
4. A README-file documenting installation and start of application.

2.2.7 Legal

No legal issues foreseen.

2.3 Application models

This section describes the model of the game, map editor and visualizes the use cases.

2.3.1 Use case model

See appendix A

2.3.2 Use cases priority

1. StartWave
2. BuildTower
3. Sell
4. Upgrade
5. PauseGame
6. Move (Player character)
7. Highscore
8. Other use cases not included

2.3.3 Domain model

See appendix B for the game model and appendix C for the map editor domain model.

2.3.4 User interface

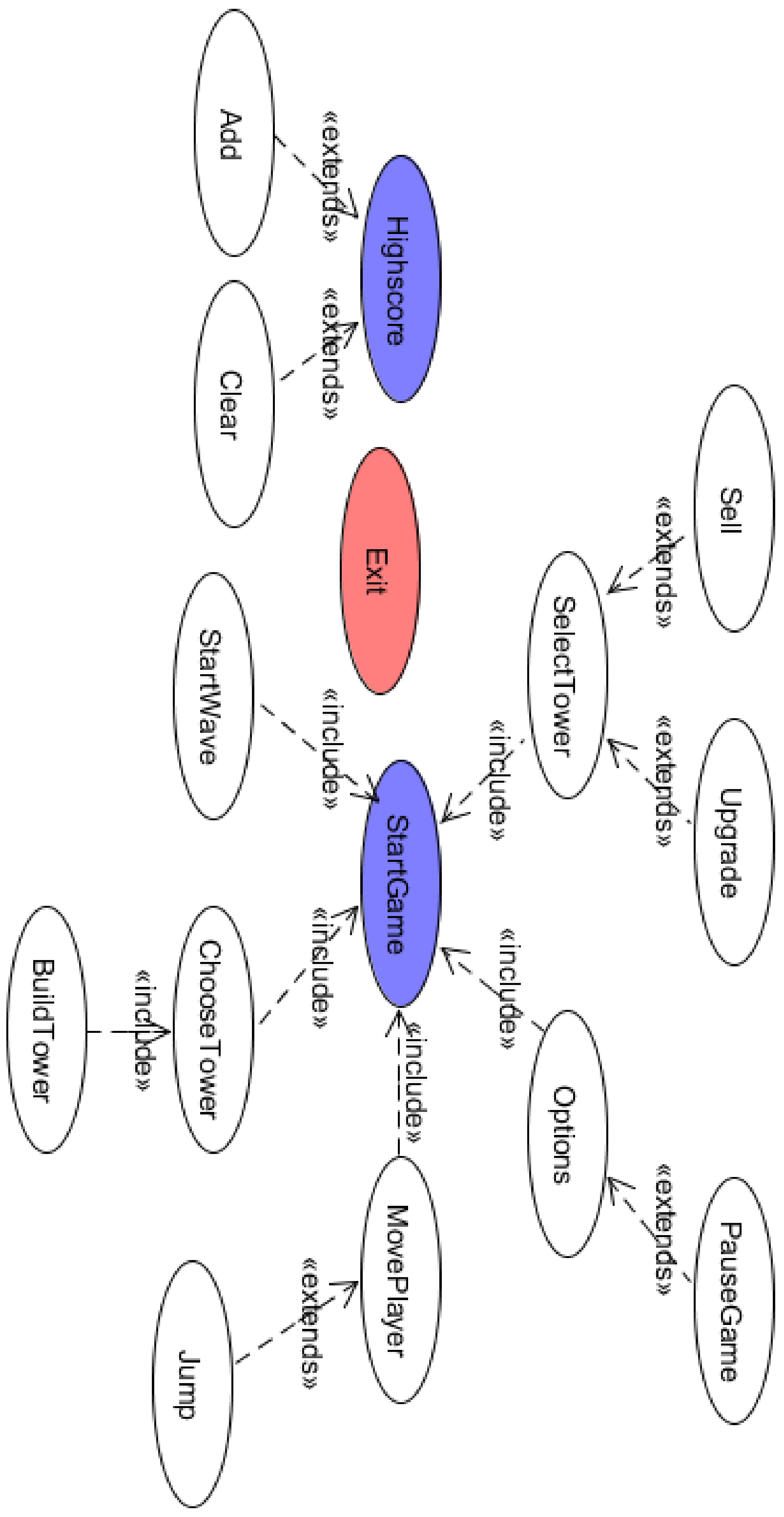
The GUI will have a fixed size of 1024x720

Text to motivate a picture goes here.

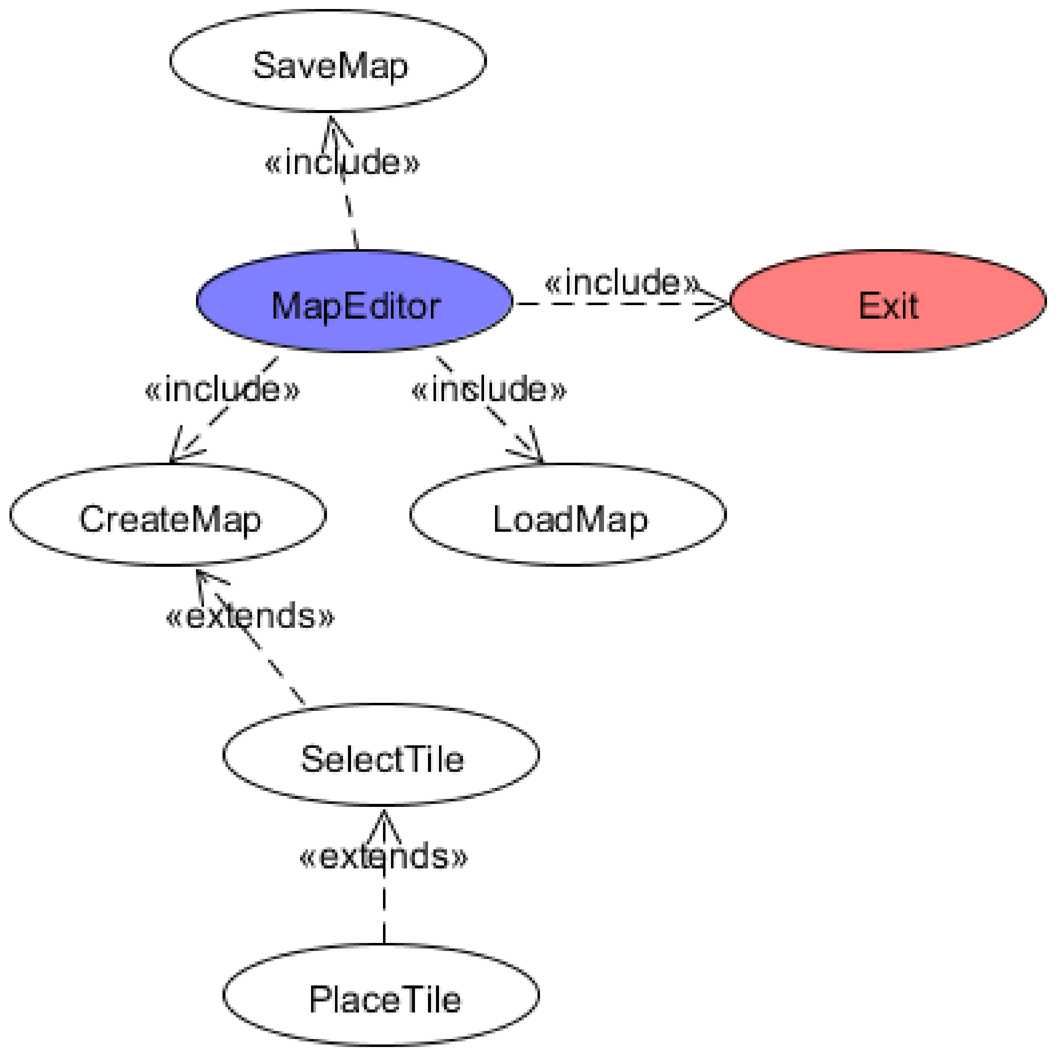
2.4 References

Tower defence `http://en.wikipedia.org/wiki/Tower_defense`

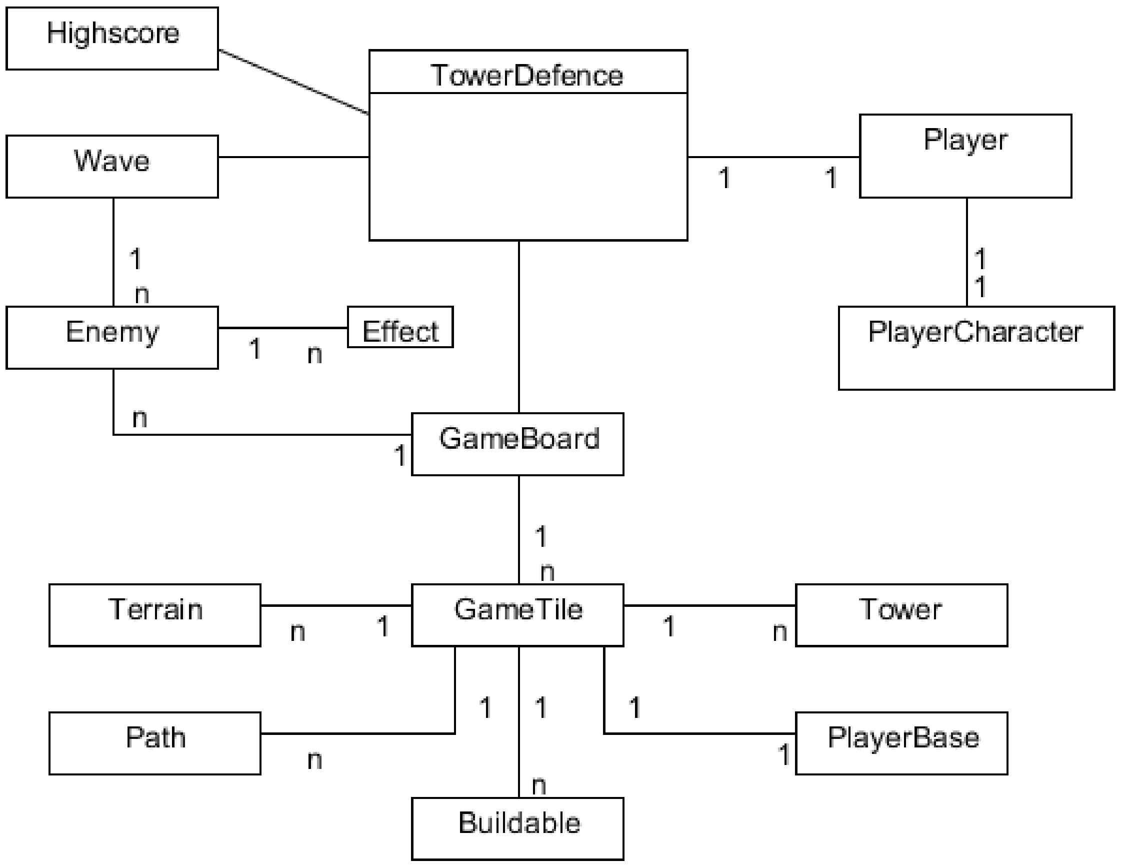
A Game use case model



B Map editor use case model



C Domain model


Highscore
TowerDefence
Wave
Player
1
1
1
n
1
1
Enemy
Effect
1
n
PlayerCharacter
n
GameBoard
1
1
n
Terrain
GameTile
Tower
n
1
1
n
1
1
1
Path
PlayerBase
n
1
n
Buildable