

Requirements and Analysis Document for ShroomDefence

Table of Contents

Version: 0.1

Date: march 12, 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 some kind of base and you must defend it from incoming hordes of enemies. You do this by building obstacles and/or 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 in realtime with a countdown between each wave where the player will have a set amount of time to build/upgrade his or hers defense. The player will have the ability to cancel the countdown to begin the wave before the timer has reached zero. The game will end according to the set rules or when canceled by the player. When the player loses or ends the game a score screen will present him or her how many waves he managed to beat. i.e. the score of the game.

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 will be saved in a high score list.

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

None used.

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.
- 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

Possible NA (not applicable).

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

NA

2.2.3 Performance

Any actions initiated by the player should not exceed a 2 sec response time in worst case.

2.2.4 Supportability

2.2.5 Implementation

2.2.6 Packaging and installation

2.2.7 Legal

2.3 Application models

2.3.1 Use case model

See appendix.

2.3.2 Use cases priority

A list goes here

2.3.3 Domain model

See appendix.

2.3.4 User interface

Text to motivate a picture goes here.

2.4 References

No references used.

APPENDIX

GUI

Domain model

Use case texts