1 Introduction:

Many people play tower defense games when they feel the need to challenge themselves in strategy. The game consists of enemies moving down a path, which you must prevent by putting down towers that will shoot down the enemies. This game is endless, and so the goal is to last as long as possible.

1.1 Purpose

This document will help anyone who reads this understand what this game consists of and what this game is about.

1.2 Scope

As a CS major in Stony Brook, this is one of the first major coding projects that I will have done on my own. I hope to be able to make more projects like this in the future, and so I hope I will learn to make good design choices from this experience.

1.3 Definitions, acronyms, and abbreviations

**The Path** - The path that is drawn on the map in the playing area. Enemies spawn at the start and go towards the end.

**Enemy -** An opposing unit. It goes down the path. Upon reaching the end, it decreases the amount of lives you have left.

**Tower -** A defensive unit that stays in place, doing damage to enemies that come into its range.

**GUI** – Graphical User Interface, visual controls like buttons inside a window in a software application that collectively allow the user to operate the program.

**Use Case Diagram** – A UML document format that specifies how a user will interact with a system. Note that these diagrams do not include technical details. Instead, they are fed as input into the design stage (stage after this one) where the appropriate software designs are constructed based in part on the Use Cases specified in the SRS.

1.4 References

**IEEE Std 830TM-1998 (R2009) –** IEEE Recommended Practice for Software Requirements Specification

1.5 Overview

This document will tell you how to play the game and explains how the game should function.

2. Overall Description

This Tower Defense game is an endless game of strategy. It is single player.

2.1 Game Description

The player is to prevent enemies from reaching the end of a path. The player will start off with a set amount of resources that will allow the player to place down towers (which costs resources to place) at the beginning of the game. Then, as the game progresses, the player will gain resources by defeating enemies and thus be able to build more towers. The enemies will become tougher to kill, thus increasing the challenge of the game. The score will the amount of time lasted. There is no save functionality and there’s no pause button, so each time the player plays, it will be a new game.

The player will start with a set amount of money at the start of the game. The player should use this money to build a few towers to defend the initial enemies and gain more money. This money should be used in preparation to defend against stronger enemies that will appear later in the game.

2.1.1 System Interface

This will be a standalone game. This game is able to load itself.

2.1.2 User Interface

The player will play the game entirely by using their mouse. The player can start or exit the game by clicking the corresponding buttons. While in the game, there is a side menu that the player can interact with to select a tower to place. The player can place down the tower where the mouse is by clicking.

2.1.2.1 Overview of Use Cases

Splash Screen

Play game

How to

Quit Game

How to Screen

How to play text

Gameplay Screen

Pick Towers

Place Towers

Start Round

Quit to Menu

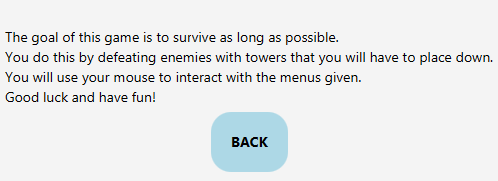
2.1.2.2 Play Game Use Case

|  |  |
| --- | --- |
| Use-Case: | Play Game |
| Primary Actor: | Player |
| Goal in Context: | Start new game |
| Preconditions | Application has been started |
| Trigger: | Start button has been pressed |
| Scenario: | 1. Player starts the application 2. Player sees the splash screen, which has an option to play game, an option to see more info, and the option to quit. 3. Player wants to play, and so presses the start button 4. Application proceeds to Gameplay Screen |
| Exceptions: | None |
| Priority: | Essential. Must be done |
| When available: | First benchmark |
| Open Issues | Styling |



2.1.2.3 How To Use Case

|  |  |
| --- | --- |
| Use-Case: | How to |
| Primary Actor: | Player |
| Goal in Context: | Explain to the player how to play |
| Preconditions | Application has been started |
| Trigger: | More info button has been pressed |
| Scenario: | 1. Player starts the application 2. Player sees the splash screen, which has an option to play game, an option to see more info, and the option to quit. 3. Player wants to learn, and so presses the more info button 4. Application proceeds to how to screen |
| Exceptions: | None |
| Priority: | Essential. Must be done |
| When available: | First benchmark |
| Open Issues | Styling |



2.1.2.4 Quit Use Case

|  |  |
| --- | --- |
| Use-Case: | How to |
| Primary Actor: | Player |
| Goal in Context: | Let the user quit the application |
| Preconditions | Application has been started |
| Trigger: | More info button has been pressed |
| Scenario: | 1. Player starts the application 2. Player sees the splash screen, which has an option to play game, an option to see more info, and the option to quit. 3. Player wants to quit, and so presses the quit button 4. Application quits itself |
| Exceptions: | None |
| Priority: | Essential. Must be done |
| When available: | First benchmark |
| Open Issues | Styling |



2.1.2.5 Pick Towers Use Case

|  |  |
| --- | --- |
| Use-Case: | Pick Towers |
| Primary Actor: | Player |
| Goal in Context: | The player selects a tower to place |
| Preconditions | Player has started the game |
| Trigger: | Start game has been pressed |
| Scenario: | 1. Game has started 2. The player uses the mouse to select a tower on the side panel |
| Exceptions: | None |
| Priority: | Essential. Must be done |
| When available: | First benchmark |
| Open Issues | Styling |

2.1.2.6 Place Towers Use Case

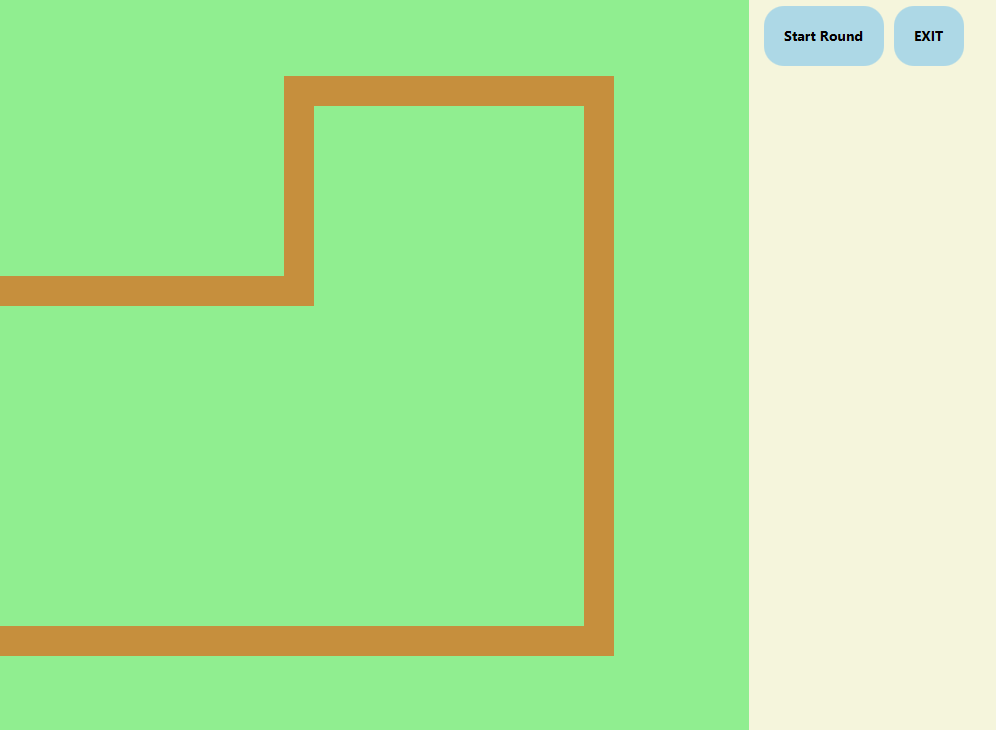
|  |  |
| --- | --- |
| Use-Case: | Start Round |
| Primary Actor: | Player |
| Goal in Context: | Starts the round, causing enemies to start spawning. |
| Preconditions | Player has clicked the play game button |
| Trigger: | Player has clicked start round button |
| Scenario: | 1. Player has clicked play game 2. Player has set up some towers using initially given resources. 3. Player starts the round by clicking the start round button. |
| Exceptions: | None |
| Priority: | Essential. Must be done |
| When available: | First benchmark |
| Open Issues | Styling |

2.1.2.7 Start Round Use Case

|  |  |
| --- | --- |
| Use-Case: | Pick Towers |
| Primary Actor: | Player |
| Goal in Context: | The player selects a tower to place |
| Preconditions | Game has been started |
| Trigger: | Start game has been pressed |
| Scenario: | 1. Game has started 2. The player uses the mouse to select a tower on the side panel |
| Exceptions: | None |
| Priority: | Essential. Must be done |
| When available: | First benchmark |
| Open Issues | Styling |

2.1.2.8 Quit to Menu Use Case

|  |  |
| --- | --- |
| Use-Case: | Quit to Menu |
| Primary Actor: | Player |
| Goal in Context: | Player wants to stop playing |
| Preconditions | Player has clicked play game button |
| Trigger: | Start game has been pressed |
| Scenario: | 1. Player has entered game 2. The player no longer wants to play the round, and presses quit to menu |
| Exceptions: | None |
| Priority: | Essential. Must be done |
| When available: | First benchmark |
| Open Issues | Styling |



2.1.3 Hardware Interfaces

This game is designed to be easily ported. The target platforms are those that can run Java programs, since the game is done in Java.

2.1.4 Software Interfaces

This game will be developed using the Java language and the JavaFX library

2.1.5 Communication Interfaces

This game is local. There is no networking here.

2.1.6 Memory Constraints

None.

2.1.7 Operations

The player’s goal is to complete the game.

2.1.8 Site Adaptation Requirements

None

2.2Product Functions

This game does not save in any way (yet). The game simply tells you a score at the end of it, so that you can remember and perhaps brag about it.

2.3.User Characteristics

This game should appeal to anyone who likes strategy and a bit of a challenge.

2.4 Constraints

The size of the window is not made to be resizeable.

2.5 Assumptions and dependencies

The size of the user’s screen is at least 750x500.

2.6 Apportioning of the Requirements

TBD

3 Specific Requirements

3.1 External Interfaces

The player will only use mouse input to start and play the game. Because of this, the GUI will only consist almost entirely of buttons.

3.2 Functions

As of right now, buttons will have a small visual scaling effect when the mouse hovers over it. Audio cues may be implemented if deemed necessary.

3.3 Performance Requirements

The primary performance concerns will be with rendering, since it is a real-time graphical application.

Note that rendering performance testing should be an important component of the development process.

3.4 Logical Database Requirements

None

3.5 Design Constraints

Efficiency is a concern, since there may possibly be a large number of objects on the screen at one time.

3.6 Software System Attributes

3.6.1 Reliability - The program should be carefully planned, constructed and tested such that it behaves flawlessly for the end user.

3.6.2 Availability - Customers may download and install the game application for free

3.6.3 Security - There’s nothing that needs to be secured here

3.6.4 Extensibility - It is important that this game is scalable

3.6.5 Portability - Should work on all platforms that support Java.

3.6.6 Maintainability - Update mechanisms will be addressed by future revisions

3.7 Organizing the Specific Requirements

Note that the game is simple enough that we need not worry about using an alternative arrangement of the content of this document

3.8 Additional Comments

The reason there are no pictures of the UI is because it is all still subject to change

4 Supporting Information

4.1 Table of Contents

1. Introduction
   1. Purpose
   2. Scope
   3. Definitions, acronyms, and abbreviations
   4. References
   5. Overview
2. Overall Description
   1. Game Description
   2. Product Functions
   3. User Characteristics
   4. Constraints
   5. Assumptions and dependencies
3. Specific Requirements
   1. External Interfaces
   2. Functions
   3. Performance Requirements
   4. Logical Database Requirements
   5. Design Constraints
   6. Software System Attributes
   7. Organizing the Specific Requirements
   8. Additional Comments
4. Supporting Information
   1. Table of Contents