Software Requirements Specification

Tower Defense

Group 11

|  |  |  |
| --- | --- | --- |
| Date | Changes | Version |
| 2/03/2017 | Initial draft | 1.0 |
| 2/10/2017 | First Revision | 1.1 |

Table of Contents

1. Introduction
   1. Purpose
   2. Scope
   3. Common terms
   4. Overview
2. Overall Description
   1. Product perspective
   2. Product functions
   3. User characteristics
   4. Constraints
   5. Assumptions and dependencies
3. Specific requirements
   1. External interface requirements
   2. Functions
   3. Performance requirements
   4. Logical database requirements
   5. Design constraints
   6. Software system attributes
4. Appendix

Phase 1 map mockup

1. ***Introduction***
   1. **Purpose**
      1. The purpose of this document is to describe in high level language what the functionality of a Tower Defense application.
   2. **Scope**
      1. The Tower Defence application will be a single player tower defence game. The user will be able to place towers that will attack enemies and attempt to complete all of the levels before running out of lives in order to win the game.
   3. **Common terms**
      1. Attack power (AP) - The power related to the attack of the towers.
      2. Health points (HP) - The amount of points required to eliminate the enemy.
      3. Java Runtime Environment (JRE) - Downloadable package of libraries that allow for the local execution of Java based code.
      4. MySQL - Database management language.
      5. Tower Defense Game (TDG) - The name of the application described in this document.
   4. **Overview**
      1. The contents of this document includes overall description, specifications and requirements of the TDG. The document is used to give the reader a basic idea of the functionality and design aspects the application.
2. ***Overall Description***
   1. **Product Perspective**
      1. System Interface
         1. The TDG will be constructed using Java. Java will require a virtual machine in order to run the application.
      2. User Interface
         1. The TDG will be a computer based game. This means that in order to run the game all necessary peripheral devices will be necessary. This includes, at the very least, monitor and mouse.
      3. Hardware Interface
         1. As this is a software based application, there will not be any hardware interfaces used.
      4. Software Interface
         1. The TDG will be located on a webpage, embedded with JavaScript.
      5. Communication Interface
         1. The TDG will be embedded on the internet, so an internet connection will be required to run the TDG.
      6. Memory
         1. The TDG will be running on a virtual machine and drawing from a graphics package that will utilize memory. The application will not be drawing an overwhelming amount of objects and will not require excessive memory.
      7. Site Adaptation Requirements
         1. The user must have the newest version of JRE in order to run the application.
   2. **Product functions** 
      1. The function of the TDG is to serve as a fun and entertaining game to play.
   3. **User Characteristics** 
      1. This TDG is a simple version of tower defender. This game will not require users to have much acknowledge to play this game, therefore any users with experience about tower defender should find it easy to play. There will be some tutorial on simple functionality to get started. This system will contain simple images and pictures, so there will be no age requirement to play.
   4. **Constraints** 
      1. The TDG will have a defined screen resolution 3,888 x 1,722 pixels, but the application will be able to be adjusted by the user. The game model and images will retain their respected resolution in order to maintain the continuity of the images.
   5. **Assumptions and Dependencies**
      1. The user will be required to have an internet connection. Also, see 2.1.7.
3. ***Specifications***
   1. External requirements
      1. The game does not rely on any external interfaces.
   2. Functions
      1. Increment 1
         1. Start Menu
            1. Starting the TDG will load the start menu. For increment 1, the start menu will provide two options, Play and Exit. Play will start the game and Exit will exit the application.
         2. Layout
            1. Starting the TDG will load the game layout. A mockup is shown in section 4.1. Paths and tower location indicators will be added, but will vary with the difficulty of the game. The user will be able to start the next level, place towers, or exit the game. Money and lives remaining will also be available for the user to view.
         3. Money
            1. The user will start the TDG with a set amount of money such that a tower can be purchased and placed. For every tower placed, the tower price will be subtracted from the money total. If the total were to be less than zero, the tower placement will be refused. Each enemy defeated will contribute a set amount of money to the total.
         4. Lives
            1. The user will start the TDG with a set amount of lives. If an enemy reaches the end of the map without being defeated, a life will be deducted from the total. If the total lives reaches zero, the game will end and return to the start menu.
         5. Start Level
            1. When the user starts a level, the set number of enemies for the level will begin to move through the path in the sequence and timing specific to the level.
         6. Exit
            1. If the user chooses to exit the game, the TDG will end and the application will close.
         7. Place Towers
            1. The user will be able to specify to place a tower. A tower may be placed as long as the user meets the monetary requirements as specified in section 3.2.1.3.1. Towers may only be placed in certain locations along the route such that the tower range will reach the route.
         8. Towers
            1. Towers will be characterized by a graphic, AP, range, attack speed, tower name, and cost.

Graphic

The graphic will be unique compared to the the graphic for the enemies.

AP

Each tower will have a set attack power indicating the amount of damage a tower will do to an enemy per attack.

Range

Each tower will have a range within which they can attack. If an enemy moves into that range, the tower will be able to attack the enemy.

Attack Speed

The attack speed of a tower specifies how long the delay between attacks is to last.

Tower Name

Each tower will be given a specific name to help identify individual towers on the map.

Cost

Each tower will be given a specific cost for which they can be bought. The user must meet monetary requirements as specified in section 3.2.1.3.1 in order to purchase new towers.

* + - 1. Enemies
         1. Enemies will be characterized by a graphic, speed, HP, and name.

Graphic

Enemies will have a graphic unique to them compared to towers.

Speed

Each enemy will have a speed with which they will move through the route.

HP

Each enemy will have an HP value that will decrease each time a tower hits the enemy. If the enemy’s HP reaches a value less than or equal to zero, the enemy will be removed from the route and a monetary reward will be added to the user’s total.

Enemy Name

Each enemy will be given a specific name to help identify specific enemies on the route.

* + 1. Increment 2
       1. Start Menu
          1. Three separate gameplay difficulty options will be added to the start menu, specifying easy, normal, or hard.
       2. Difficulty
          1. Each difficulty will have a set number of levels, differing to classify the levels as named. Easy will have easier waves of enemies, normal will be slightly more difficult than easy, and hard the most difficult of the three. The difficulty will be scaled via a percentage value attributing to all value with the exception of lives, which will remain constant at each difficulty.
       3. Towers
          1. New towers with upgradeable AP and unique name and graphic will be added.

Upgradeable AP

Users will be able to upgrade the AP of the towers they have placed on a map for a set amount of money. Their ability to upgrade the towers is dependent upon the monetary requirements specified in section 3.2.1.3.1.

* + - 1. Enemies
         1. New enemies will be added with greater hitpoints in order to make levels more difficult to complete. These enemies will have a unique graphic and name.
      2. Score
         1. Every enemy that is removed from the path will attribute to a total score. This is similar to the monetary requirements described in 3.2.1.9.1.3.1, but will only be used for the Leaderboards.
         2. Each enemy that shares type will attribute the same value towards the total score.
      3. Leaderboard
         1. Upon completion or failure of the game, or selection via the start menu, the system will allow users to view a list of other users who have completed the game.
         2. The Leaderboard should display 3 different 5 entry lists for each difficulty level described in section 3.2.2.2. Each list should display names sorted by the last 5 users to complete the difficulty.
         3. Users that have completed the game will be able to record their name to a database, along with the difficulty they chose and the date they are playing on, if they so desire. The system should then be able to query that information to a database that will be used to display it according to section 3.2.2.5.2.
  1. **Performance**
     1. The game should be designed so there is no noticeable lag or limitations when run on an adequate system.
  2. **Logical Database Requirements** 
     1. Leaderboard implementation, explained in section 3.2.2.5, will be handled via an externally maintained MySQL database. The database will be accessible to the game and scores will be added into the database as needed. Maintenance of the database will be handled separately, independent of the game interface.
  3. **Design Constraints**
     1. The time given to us only allows for a very basic model. This model is considered basic as far as the system will function, but any further development would take a considerable amount of time. The current design approach is to get the underlying functionality correct before any further iterations take place.
  4. **Software System Attributes**
     1. Reliability
        1. The system should maintain standard, error-free operation across all different users that would run it and across an indefinite amount of playthroughs.
        2. Errors should be resolved with a restart of the game.
        3. Errors in the Leaderboard database should be handled externally, though MySQL contains error-preventing measures internally.
        4. The user interface will be simple, the simplicity will decrease the strain placed on the game engine.
     2. Availability
        1. The system should maintain standard operation at anytime users could access it.
        2. This availability should remain independent of the status of the Leaderboard database. Should access to or operation of the database be unavailable, the system should maintain normal functionality, excluding the Leaderboard system.
        3. The application will be developed using Java, thus any supporting system will be able to run the TDG.
     3. Security
        1. Access to and manipulation of the Leaderboard database should be controlled by only the system itself. Users should not have direct access to any entries in the database unless provided by the system via readouts.
        2. In accordance with Assumptions and Dependencies in section 2.5, users should have the latest JRE version installed in order to access the game. Any user using an older JRE version may be susceptible to security threats stemming from outdated JRE support.
     4. Maintainability
        1. Additional features past Increment 1 and 2 can be added and changed simply by adding access to the functions through either the in-game Start menu or the game interface itself.
        2. The Leaderboard database can be maintained externally via MySQL commands by users with necessary credentials. Therefore, the Leaderboard functionality should be maintained independently of the game.
     5. Portability
        1. The game will be JRE-based, making it compatible with any operating system. Therefore, any personal computer will be able to run the application given it has an internet connection. The portability will be determined by internet availability.

1. ***Appendix***
   1. Map Layout Mockup

*(Picture caption)* ***4.1*** *Phase 1 map mockup (3,888 x 1,722 pixels)*