***SOFTWARE REQUIREMENT SPECIFICATION (SRS)***

***LABYRINTH***

*CREATED BY:*

**JOHN FOUAD**

**DANIEL MATHIEU**

**PATRICK BARRON**

**GABRIEL WOODSUM**

**CHRISTOPHER WONG**

**October 11, 2016, REV A**

1. Introduction

1.1 Purpose:

Labyrinth is designed to be a 3D, multiplayer maze game where players will have to avoid obstacles and enemies to get to the destination before the other players.

This document is meant to delineate the features of Labyrinth so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

1.2 Scope:

*In Scope:*

1. Competitive multiplayer desktop platform that is accessed through a web browser
2. Generate random maze configurations
3. Automated enemies to pursue users
4. Graphical User Interface that will display a 3D rendering of the playing field

*Out of Scope:*

1. Saving player’s state and progress after disconnection
2. No mobile platform or application development

1.5 Overview

The rest of this SRS is organized as the following: Section 2 gives an overall description of the software. This description will include the level of proficiency that is expected of the user, general constraints that go into the development of the software, along with any assumptions and dependencies that go along with the software. Section 3 gives functional requirements, listed by use cases.

2. Overall Description

2.1 Product Perspective

 Labyrinth is aimed towards an audience, which ranges from kids as young as 10 years old to adults in the online gaming community, including anyone who enjoys playing online games in their free time. It will allow for an enjoyable experience for users to take part in when they are seeking entertainment. Labyrinth will be user friendly, easy to play, and and should be reliable as long as the network connection holds.

     Labyrinth intends to be a stand-alone product that should not depend on other software. Our game will run on Unix and Windows. There are no memory constraints because we are not storing data. Labyrinths user interface will be a web browser (BUI).

2.2 Product Functions

|  |  |  |
| --- | --- | --- |
| Class of use cases | Use cases | Description of use case |
| Use case related game play | Create game | Creates a new game |
| Simulate Game Play | Simulate game play |
| Add Player | Adds player to the game session |
| End Game | Ends the game |
| Use case related to user controls | Join Game | Adds the user to the current game |
| Move Avatar | Moves the avatar in a direction |
| Exit Game | Removes the user from the game |
| Use case related to visual display | Create Maze | Generates a random maze configuration |
| Display Maze | Display the maze as a 3D image |
| Display Start Menu | Display Start Menu |
| Use case related to automated bot | Create bot | Create an automated bot |
| Delete bot | Delete automated bot |
| Move Bot | Bot automatically navigates the maze |

* 1. User Characteristics

1. The user should be familiar with controlling an avatar in the third person point of view.
2. The user should be familiar with the basic concept of navigating a maze.

2.4 Principal Actors

    The two principal actions in Labyrinth are the “player” or “user” and the “system”

* 1. General Constraints

1. Requires a constant internet connection
2. Internet bandwidth delays will be significantly noticeable
   1. Assumptions and Dependencies
3. Labyrinth is dependent on the availability of internet connection
4. Labyrinth would be available only on www.labyrinthgame.com. There is no other way to access this game.

3. Specific Requirements

3.1 Functional Requirements

We describe the functional requirements by giving various use cases.

*Use case related game play:*

**Use Case 1:** Create Game

*Primary Actor:* System

*Precondition*: User trying to join game

*Main Scenario:*

1. The application server shall check if there are any open game sessions currently running that are not at the full capacity of 4 players.
2. If there are no available games currently running on the server, the server shall generate a new random maze configuration and game session.
3. The server shall add the player to the game

*Alternate Scenario:*

1. An active game session is not at capacity of four players. The create game function is canceled.

**Use Case 2:** Add Player

*Primary Actor:* System

*Precondition*: A game session is available

*Main Scenario:*

1. The server shall  randomly position the user in a maze.
2. The server shall render the maze and player into a 3D representation.
3. The server shall send a response message to the user.
4. A graphical user interface shall be displayed for the user to navigate the maze.

**Use Case 3:** Simulate Game Play

*Primary Actor:* System

*Precondition*: Player is in a game session

*Main Scenario:*

1. The server shall restrict the user's movement to be within the confines of the maze and boundaries.
2. The server shall create and control 16 automated bots.
3. Once a player as reached the end of the maze, the server shall delete the game session

.**Use Case 4:** Delete Game

*Primary Actor:* System

*Precondition*: A player has won

*Main Scenario:*

.

1. The server shall notify the players that the game is over and whether or not they have won.

2. The server shall remove all of the players from the game session and return them to the start menu.

3. The server shall delete all automated bots.

4. The server shall delete the maze file.

*Use related to User Controls:*

**Use Case 5:** Join Game

*Primary Actor:* User

*Precondition*: User is at Start Menu

*Main Scenario:*

1. From the start menu displayed, the user shall be given the option to join a game
2. If the player decides to join a game, the server shall assign the user one of four possible avatars and a gamer tag.
3. The server shall add the player to the game session

**Use Case 6:** Move Avatar

*Primary Actor:* User

*Precondition*: User is in the game

*Main Scenario:*

1. User shall be given control of the avatar's motion through the use of the “w”, “a”, “s”, “d”, or arrow keys. These keys will be correlated to the north, south, east, and west directions.  However, the corresponding directions will change keys depending on the orientation of the player.

**Use Case 7:** Exit Game

*Primary Actor:* User

*Main Scenario:*

1. Any point in the game, the user shall have the ability to exit the game through the use of the “L” key and be returned to the start menu.

Use case related to visual display:

**Use Case 8:** Create Maze

*Primary Actor:* System

*Precondition:* A new game has been created

*Main Scenario:*

1. File shall be created using the 3D maze generating algorithm

      2. Input file shall be processed through the maze to display algorithm

**Use Case 9:** Display Maze

*Primary Actor:* System

*Precondition*: Maze has been created

*Main Scenario:*

     1. The output from the maze to display algorithm shall be piped to the GUI by the network connection

      2. The maze shall be displayed for gameplay

**Use Case 10:** Display Start Menu

*Primary Actor:* System

*Precondition*: Internet Connection

*Main Scenario:*

1. User opens web browser and sends a HTML request to server to access Labyrinth
2. Server shall respond with an HTML message that contains the start menu for Labyrinth
3. The display menu shall pop up when user connects to browser.

*Use case related to automated bot:*

**Use Case 11:** Create Bot

*Primary Actor:* System

*Precondition*: Created game

*Main Scenario:*

1. The server shall generate 16 enemy automated bots.
2. Bots shall be randomly places through the maze

**Use Case 12:** Move Bot

*Primary Actor:* System

    Precondition: Active gameplay

*Main Scenario:*

1. The bots shall move randomly throughout the maze.
2. When a bot is close proximity to a user, the bot shall move towards  the user’s avatar and the player shall be killed.

**Use Case 13: Delete Bot**

*Primary Actor:* System

*Precondition*: End of game

*Main Scenario:*

1. The servers shall delete all the bots at the end of the game.
   1. Performance Requirements
   2. Bandwidth speed should be fast enough
   3. Should run on a non-mobile web browser
   4. Should run on a non-mobile web browser
   5. 99% of responses should be within 100ms, except for downloading the map data
   6. Design Constraints
2. Security: The map layout and location of players should be secured from being viewed or manipulated by hackers.
3. Fault Tolerance: The system should be able to restart in case of a system crash or power failure.

3.4 External Interface Requirements

The majority of the screen will be dominated by a 3rd person point of view of the maze.  Only the immediate area to the player will be shown, which includes the walls and any enemies or fellow players.

4. Future Extensions

a. Functionality to attack obstacles in your way

b. Health bar that keeps track of your health

c. Change in maze configuration where the walls of the maze rotate to alter the direction the player is running

d. Beam of light down from the sky pointing at the destination for where the players should be headed