Maraya Burks
Lea Davis
Dr. Marshall
Software Engineering Project: Technical Documentation
November 13, 2020

Technical Documentation

This document contains a snapshot of the application's technical requirements.

Scope of Work:

This project is the implementation of a maze game on a web page. This maze will be accessible and free for the public. The maze doesn't have the options to choose difficulty level, instead the user can click the start button to take them to the standard maze game template. After clicking start, the user is given three minutes to get from the start point to the finish point. As in a typical maze, there are horizontal and vertical lines called borders that the user should not go through. The user is able to move freely through the maze. When the timer reaches zero, a message pops up on the screen with the option for the user to play again. This maze is intended for recreational purposes only, for the maze does **NOT** keep tab of users, score or frequency played.

Stakeholders:

- Lea (Project Manager and SWE)
- Maraya (Product Manager and SWE)
- Dr. Marshall (Program Manager)
- Public (Children, Teenagers, Adults and Seniors)

Enumerated Functional Requirements:

FUNCTIONAL REQ-*x*	**Priority Weight**	**Description**
FR01	High	The webpage shall include a landing page with team member name, introduction and picture.
FR02	High	The maze shall be 2 dimensional.
FR03	High	The maze should have a countdown timer present starting with three minutes.
FR04	High	The maze shall have a clear start and finish point.
FR05	High	The user shall have to start on the start point.

FR06	Medium	The maze shall display a status message to the user at the end of the game. Ex: Your game has ended and you won/lost!
FR07	Medium	The maze shall open up in a new page on the website.
FR08	Medium	The user shall be able to click a button to start maze.
FR09	Medium	The user shall be able to move freely throughout the maze.

Traceability Matrix

Trace ID	**Functional Requirement**	**Test Case**	**Priority**	**Status**
1	The maze should have a countdown timer present starting with three minutes (**FR03**).	There is a working timer present that counts down from three minutes (**TC01**).	High	Complete
2	The maze shall have a clear start and finish point (**FR04**).	There are start and finish points that are present and can be reached (**TC02**).	High	In Progress
3	The user shall have to start on the start point (**FR05**).	The user is at the start point when the game begins (**TC03**).	High	Complete
4	The user shall be able to move freely throughout the maze (**FR09**).	The user is able to move any way and direction they choose (**TC04**).	High	Complete

5	The maze shall display a status message to the user at the end of game (**FR06 - MODIFIED**).	The correct status message should display on the screen at the correct time (**TC05**).	Medium	In Progress
6	The user shall be able to click a button to start maze(**FR08**).	There is a working button that the user can click to start the maze (**TC06**).	Medium	In Progress
7	The maze shall open up in a new page on the website (**FR07**).	There is a working link that directs the user to the page where the maze is located (**TC07**).	Low	Complete

Design of Tests:

Test Cases

Test Case ID	Test Case
TC01	There is a working timer present that counts down from three minutes.
TC02	There are start and finish points that are present and can be reached.
TC03	The user is at the start point when the game begins.
TC04	The user is able to move any way and direction they choose.
TC05	The correct status message should display on the screen at the correct time.
TC06	There is a working button that the user can click to start the maze.
TC07	There is a working link that directs the user to the page where the maze is located.

Test Coverage

We will have a test coverage of 85%. We will be able to test our code from the start of the game to the finish. There have been extra tests implemented to ensure that we get the most out of our testing.

Integration Testing Strategy

In this section we have included how we are approaching our testing.

We will be testing as we go through our implementation of the code. This will ensure that we are not running into major errors at the conclusion of implementation. At the start of testing, we will work our way down the test case chart. Each test case that has been tested successfully will be marked with a check mark. This is how we will know what test cases still need to be tested. If a test case has been tested, but it isn't functioning properly, we will not check this test case off as there are still some things that need to be done. Lea will be testing the application, making note of the test, and sending it off to Maraya to troubleshoot (if the test case is not successful). Our team didn't take advantage of any testing software. We simply met frequently and kept tabs through a live document.
Note: We have reached a point in our project where we want everything to work as expected, but with time permitting we cannot expect our test cases without a high priority to be fully functional.

Additional Testing Plans

We have put additional testing plans in place to ensure that not only our functional requirements are tested.

Additional Testing #	Description
AT01	Non-functional requirement, **NF02,** will be needed to be tested. This requirement says that the application shall be available on our designated GitHub page. We will test the visibility of our application on the website.
AT02	Non-functional requirement, **NF03**, will be needed to be tested. This requirement says that the application will not crash when a user is playing the maze.We will test this by having test users test the maze in the live environment.
AT03	We will test the users ability to refresh the page and ensure the current page is displayed after refreshing.