

CS 320 Course Project Final Report

for

**Maze Web Application**

Prepared by

Group Name: *Team 19*

|  |  |  |
| --- | --- | --- |
| **Patric Collins** | **11617373** | **patric.collins@wsu.edu** |
| **Matthew Norvell** | **11690711** | **matthew.norvell@wsu.edu** |
| **Andrew Keyes** | **11595911** | **andrew.keyes@wsu.edu** |
| **Daniyal Abbas** | **11714220** | **daniyal.abbas@wsu.edu** |

|  |  |
| --- | --- |
|  |  |
| Date: | **2020-12-16** |
|  |  |

Content

Contents ii

1 Introduction 1

1.1 Project Overview 1

1.2 Definitions, Acronyms and Abbreviations 1

1.3 References and Acknowledgments 1

2 Design 2

2.1 System Modeling 2

2.2 Interface Design 2

3 Implementation **6**

3.1 Development Environment 6

3.2 Task Distribution 6

3.3 Challenges 6

4 Testing 7

4.1 Testing Plan 7

4.2 Tests for Functional Requirements 7

4.3 Tests for Non-functional Requirements 8

4.4 Hardware and Software Requirements 8

5 Analysis 9

6 Conclusion 10

Appendix A - Group Log 11

# Introduction

This project is the final project for CS 320. Our group of four people has created a website that meets the guidelines for the project by handling and manipulating data.

## Project Overview

Our project is centered around a web application that allows a user to play through randomly generated mazes. These mazes are then accessible by others through a global leaderboard. Section 2 outlines use cases for the website. Section 3 outlines the class structure of the website. Section 4 outlines both state and sequence diagrams for aspects of the website. Appendix A is a record of all meeting information for the project group.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| DB | Database |
| Score | The time taken by a user to complete a maze |
| Seed | A number used by the website to generate a maze. |
| Username | A unique identifier for a user. |

## References and Acknowledgments

We did not need to cite any sources for our project.

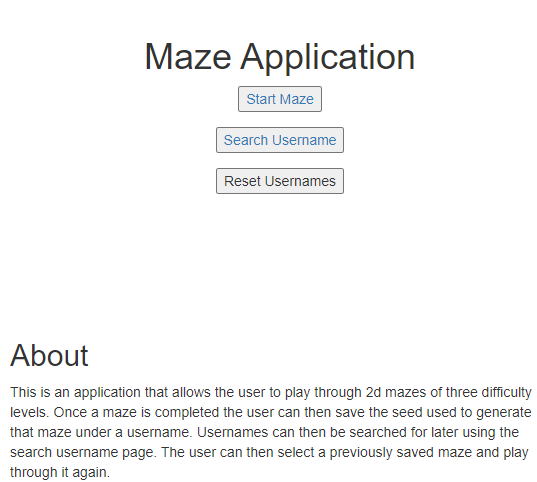
# Design

## System Modeling

Our Implementation strictly follows the design document (milestone 2).

## Interface Design

### Landing Page



### Difficulty Selection Page

### Maze Page

### Post Maze Page

### Username Search Page

# Implementation

## Development Environment

Daniyal: Visual Code for coding, Firefox for website testing

Andrew: IntelliJ IDEA for coding, Google Chrome for website testing

Matthew: Atom for coding, Firefox for website testing

Patric: N/A

Github for version control. HTML, CSS, and JavaScript for project functionality.

## Task Distribution

Daniyal: Timer implementation and difficulty select page

Andrew: Username database storage and search, main page and formatting

Matthew: Maze generation and display

Patric: N/A

## Challenges

We had two major implementation issues. First was the setup of a backend server for a database. We decided to use local storage in the browser for data instead of a whole backend server. Second, one of our teammates, Patric, did not participate in the later stages of the project after contacting him and the professor. There was an additional workload on the remaining members, although we did finish without much pressure.

# Testing

## Testing Plan

**Functional Requirementsz**

|  |  |  |
| --- | --- | --- |
| Maze generator | Maze is generated | 2020-12-07 |
| - | Maze is playable | 2020-12-14 |
| Username submission | Fully functional storage | 2020-12-14 |
| Username search | Fully functional search | 2020-12-14 |

**Non-functional Requirements**

|  |  |  |
| --- | --- | --- |
| Maze generation time | Maze should take less than 5 seconds to generate | 2020-12-07 |

**Software Requirements**

|  |  |  |
| --- | --- | --- |
| Browser testing | Works on all major HTML5 browsers | 2020-12-16 |

## Tests for Functional Requirements

**Maze generator**

To check if the maze was generated correctly, 20 different random seeds were used to generate mazes that are displayed to the screen. These mazes were then checked for continuity from start to end to make sure there were traversable.

*2020-12-07 Success*

To check if the maze was fully playable, we used 5 random seeds to generate mazes. All directions of player movement were tested as well as all of the bounds of the maze. The player sprite was then directed to the end of the maze to check for completion.

*2020-12-14 Success*

**Username Submission**

To check if usernames and maze times could be searched successfully, we completed 5 mazes and stored three under “A” and two under “B”. We then checked the localStorage to see if the usernames and associated values were stored correctly.

*2020-12-14 Success*

**Username Search**

To check if usernames and maze times could be searched successfully, we completed 5 mazes and stored three under “A” and two under “B”. We then searched both names and made sure their associated values were correct. We then selected a maze to complete again and made sure it generated the same maze as the first run.

2020-12-14 Failure (maze seeds did not work in generator)

2020-12-16 Success

## Tests for Non-functional Requirements

**Maze generation time**

Multiple random mazes of the largest size were generated to see how long it would take to generate. These large mazes must generate in 5 seconds or less.

2020-12-07 Failure (*hard mazes took 15 seconds to generate)*

2020-12-08 Success

## Hardware and Software Requirements

**Browser Testing**

We ran all previous use cases on Chrome, Firefox and Edge to confirm their use across all major HTML5 compatible browsers.

*2020-12-16 Success*

# Analysis

Hours worked:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Milestone 1 | Milestone 2 | Milestone 3 | Total Hours |
| Daniyal | 5 | 5 | 5 | 15 |
| Andrew | 5 | 5 | 12 | 22 |
| Matthew | 10 | 5 | 20 | 35 |
| Patric | 5 | N/A | N/A | 5 |

Milestone 1 took a considerable amount of effort as we had to synthesize and elaborate an idea for our project. Our final SRS document turned out to be 13 pages long.

Milestone 2 took the least amount of effort of the three. The diagrams were easy to create as they stemmed directly from how we spoke about approaching our problems in our meetings.

Milestone 3 took the most amount of time out of the three milestones as it contained the bulk of the programming work for the project. The final documents and SRS document revisions took place at the very end.

# Conclusion

This project was many of our teammates first experience designing and developing code in a group environment. With this comes a few challenges that are not present when working alone. When working on a project by yourself, your vision and design are the only ones that matter, and you do not need to relay that to or rely on someone else to have that same vision. This project helped us learn how compile many different viewpoints and ideas on how something should function or be implemented. To do this, it was very important that we collaborated on how the different parts that we each designed would fit together. We used what we learned about design document to facilitate this process. Once we had one cohesive plan on what to implement, we needed to figure out how to properly divide the work, develop, and then fit those pieces back together into a working application. This was done using GitHub. Before this project, none of us had any real experience using git. So, we all had to learn together about how to properly branch, commit, and merge our work. Most of our group had not worked very much with html or Javascript before, so this project has given us a lot of experience with the production of front-end web design. This project has taught us quite a few things about the software development process and will hopefully prove valuable in future careers as software engineers.

Appendix A - Group Log

Our group usually met at least once a week throughout the semester.

|  |  |
| --- | --- |
| Date & Time | Subject |
| *2020-10-05 (3:00 - 3:45)* | *Brainstorming and deliberation of the project idea. Many ideas were proposed, but a vote favored the maze idea. We fleshed out aspects of the maze and all functionality it had.* |
| *2020-10-19 (3:30 - 4:15)* | *Delegation of tasks for the SRS document. Each member was given a section to complete: Section 1 - Matthew, 2 - Patric, 3 - Andrew, 4 - Daniyal. The SRS Github repo was created for us to commit to.* |
| *2020-10-26 (3:00 - 3:15)* | *A quick check-in between all members to determine progress and any challenges. Section 2 was completed and work on the other sections had begun.* |
| *2020-11-02 (3:00 - 4:00)* | *We discussed what we had each written and disputed any differences of opinion. We still had some things to complete, which were added by the due date.* |
| *2020-11-09 (6:30-6:45)* | *We created the main project repo and discussed implementation details.* |
| *2020-11-16 (3:00 - 4:00)* | *Delegation of tasks for the software design document. Each member was given a section to complete: Section 1 & 4 - Matthew, 2 - Andrew, 3 - Patric & Daniyal.* |
| *2020-11-30 (3:00 - 3:30)* | *We discussed possible implementation paths and set out exactly what we wanted the website to look like.* |
| *2020-12-07 (3:00 - 4:15)* | *We delegated tasks to certain members in the group and created a to-do list that broke down what was needed to complete the project into manageable steps.* |
| *2020-12-15*  *(10:20 – 10:30)* | *We went over what happened in our demo and discussed what tasks remained for all of us to complete.* |