# TEST PLAN

## Test Plan Identifier:

* Enigmatic Maze Test Plan

## Introduction:

* The test plan will be utilized to ensure important features for the maze and riddles are implemented.
* Goals include ensuring the mazes are randomized, ensuring the proper riddle types appear throughout the maze, making sure the riddles are randomized based on the proper type selected, make an optimized program, and ensure user inputs work correctly.
* Constraints include not spending money and having a fully developed product before April 12th.

## References:

* List the related documents, with links to them if available, including the following:
  + Project Proposal <https://github.com/Wesasaurus/CSU-Senior-Project/blob/master/docs/Project%20Proposal.pdf>
  + Requirements Document <https://github.com/Wesasaurus/CSU-Senior-Project/blob/master/docs/Requirements.pdf>
  + WBS for Coding

## Test Items:

* Maze Game Program

## Features to be Tested:

* Functionality of the maze
* Functionality of the riddles
* Functionality of randomization tools
* Functionality of user input
* Functionality of riddle selection
* Functionality of parsing files for selection
* Functionality of Visuals
* Speed of program
* Fault-Tolerance of Answers
* Scalability
* Accessibility
* Ensuring proper access
* Automatic shutdown

## Features Not to Be Tested:

* Support will not be tested as this is a post release aspect of the program.
* Ensuring game isn’t offensive will not be tested as there are no culturally based riddles implemented at the moment & all the riddles selected are appropriate for all ages.
* Capacity will not be tested as it can be determined strictly by looking at the file size.

## Approach:

* The overall approach to testing will be through manual white box testing. I will be manually checking each input and if the program is doing the intended task. I will have specific outcomes that I will want to program to output and will manually check if it does so correctly. The test plan will likely be a phase test plan as I will test things one at a time at different phases.

## Item Pass/Fail Criteria:

* The criteria that will be used to determine whether each test item has passed or failed testing depends on each test, but overall will rely on if the program had the intended output. For example the pass/fail for the maze is if it generates a random solvable maze each time. Fails would include not having an exit or generating the same maze repeatably.

## Suspension Criteria and Resumption Requirements:

* Criteria to be used to suspend the testing activity include if the function has not begun development, program has passed its test, or testing requires another feature to be implemented to ensure appropriate results.
* Testing activities which must be redone when testing is resumed include the maze randomizer, the riddle randomizer, and answer validation.

## Test Deliverables:

* List test deliverables, and links to them if available, including the following:
  + Test Plan (this document itself)
  + Test Cases with Pass/Fail measurements

## Test Environment:

* Environment consists of a Windows 10 Home using Visual studio Code, Command prompt, and network access.

## Estimate:

* Testing for the parsing part of the program will begin the week of the 25th of February, randomization of the riddles as well as pairing riddles with the answers will be tested on the 4rd of March, Testing of the maze will begin March 18th, riddle integration will begin testing on the 25th of march, and then a comprehensive test to ensure everything is working together as intended will begin the 1st of April.

## Schedule:

## A screenshot of a computer program AI-generated content may be incorrect.

## Staffing and Training Needs:

* No staffing needs.

## Responsibilities:

* Wesley Cassel; Project manager, lead developer, lead tester, lead writer.
* Mr. O’Neill; Project Client

## Risks:

* Risks: Program gaining access to parts of computer it shouldn’t have access to, and Program crashing computer.
* Mitigation Plan: Do a plethora of testing to ensure the program only has access to the specified file, ensure that if the program takes too long to load it automatically ends.

## Assumptions and Dependencies:

* Assumptions: Program will be run on a Windows 10 Home Device, program will use .txt files for riddle documents, program runs assuming person knows how to run an exe.
* Dependencies: <iostream>, <string>, <fstream>, <ctime>, <list>, <cstdlib>, <conio.h>, <chrono>, <thread>, and <filesystem>.

## Approvals:

* Mr. O’Neill; Project Client

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_