Enigmatic Maze Game

April 2025

Wesley Cassel

Bachelor’s in Applied Computing with a Cybersecurity Concentration

Mr. O’Neill

Statement of Purpose

There are a plethora of maze games within the world. Imagine if we could expand upon solving said mazes and make a more interesting and difficult game out of it? In an ideal world solving mazes would be an engaging game utilized for strengthening the brain, but generally they are seen as easy, as you can see the whole maze and mentally map possible avenues before committing to a direction. This leads to them mostly just being used as a child’s game on the kid’s menu. This is a problem as these games could be used as a fun challenge to stimulate the brain, but they are so simple they are delegated to something to pass the time while waiting for food. This is an important problem to solve because if we managed to make maze games more challenging and interesting it could be utilized as a brain boosting game to help child development and elderly people. Mazes have been shown to help with sharpening memory, improving physical involvement such as hand-eye coordination, and helping with patience (Why Solving Puzzles, 2021). A solution to making mazes more impactful would be to add riddles throughout the maze to solve. Riddles would not only provide fun information throughout solving the maze, but it would add a nice difficulty curve for those who want a more challenging experience. This is evident through the kind of thinking riddles require compared to mazes. Riddles require less straight forward thinking and require the solver to utilize more of their knowledge of language, experience, and other mental activities to solve it (Marcel, 2009). They also improve cognitive functions relating to critical thinking, problem solving, and memory skills (Health Benefits). Compare this to mazes which require more visual and motor skills to solve such as hand-eye coordination (Why Solving Puzzles, 2021). Combining them together would increase the required brain usage and result in a more stimulating activity. Overall the simplicity of mazes makes them extremely easy to solve, but a simple addition of riddles throughout the maze can enhance the experience and create a better brain exercise for the user.

Research & Background

I will be expected to make files containing different types of riddles, and then code will be made that lets the user choose different riddle types. Depending on the riddle type, a random riddle in the file will be chosen. Then code will be made so that the user can solve the riddle by entering the answer in. Code will be added to give the user multiple attempts per riddle. After which code for a working maze will be constructed. Upon setting up the framework for a working maze, both the maze and the riddles will be combined, so that throughout the maze random riddles will take place. User will navigate through it by typing WASD for up left down right.

The game will present itself on the command prompt utilizing characters for walls and empty space. X = exit, empty space acts as traversable points, W represents the player, and # represents the walls. Small example maze shown below:

#W####

# #

## ####

# ####

## x

######

Expected Deliverables: First Code Drop (Getting riddle types set up), Second Code Drop (the maze framework working), and Final Code Drop (Working maze with random riddles).

Program Language, Software, & Hardware

The program will be written in C++ and be run on a Windows 10 device. The program will be created using visual studio code and require at least 208 kb of storage. The program will be played utilizing the terminal window.

Project Requirements

Requirement ID: 1

Type: Functional

Description: The basic functionality of the project should include a working randomized maze selector that, upon traversal, has the user answer a myriad of riddles based on the type setting they have chosen.  
Rationale: The requirement is justified as without the basic functionality of the project, it as a whole it would fail and be incomplete.

Fit Criterion: A measurable output to see if the requirement is met would be to run the game and see if a maze pops up, and to see if while completing a maze a riddle appears.  
Priority: 1

Dependencies:

Requirement ID: 2

Type: Look

Description: The product must have a clean and clear look.  
Rationale: If the maze looks disjointed and incomprehensible then the user will have an even more difficult time traversing the maze; therefore, having a clear and understandable design for the project is necessary.

Fit Criterion: A measurable way to test if this is met would be to have a sample of the target audience for this product test out the product for the first time, and then receive feedback.  
Priority: 2

Dependencies:

Requirement ID: 3

Type: Ease of Use

Description: Project should be fairly simple to understand and use. Users should be able to type out or select settings and/or movements in the maze.   
Rationale: This requirement is justifiable as if the project is overly complex in how it works it could confuse users causing frustration and removing desire to play the game.

Fit Criterion: A measurable way to test if this is met would be to have a sample of the target audience for this product to test out the product for the first time, and then receive feedback.  
Priority: 4

Dependencies: 2, 4

Requirement ID: 4

Type: Personalization

Description: Let users decide the types of riddles they want to solve.  
Rationale: This requirement is justifiable as it provides a form of difficulty and personalization for the user. It allows the user to decide the kind of riddles they want to solve and gives a sense of difficulty if they are choosing a type they are unfamiliar with.

Fit Criterion: You can test for this by having the user select a riddle type and then once a riddle pops up make sure it fits with the typing selected.  
Priority: 3

Dependencies:

Requirement ID: 5

Type: Learning

Description: Make sure the user is learning throughout their use of the program.  
Rationale: The project has riddles that could help provide learning experiences for the user and the project as a whole could help as a brain exercise for users.

Fit Criterion: You could measure this by seeing how many attempts a user takes to solve a riddle.  
Priority: 5

Dependencies: 1, 4

Requirement ID: 6

Type: Speed and Latency

Description: The program will respond to input at a timely manner.  
Rationale: The project will be gaining a lot of inputs throughout the maze and so every input should have a fast response time to ensure rounds don’t take forever.

Fit Criterion: Measurable by how long an input takes to take effect.  
Priority: 6

Dependencies: 1

Requirement ID: 7

Type: Understandability and Politeness

Description: The instructions and riddles should be understandable in their format and polite in their wording regardless of if you win or lose.

Rationale: It is important that the maze game is enjoyable and making instructions unclear or being rude to players could push away players.

Fit Criterion: A measurable way to test if this is met would be to have a sample of the target audience for this product to test out the product for the first time, and then receive feedback.  
Priority: 15

Dependencies:

Requirement ID: 8

Type: Precision or Accuracy

Description: The project should be accurate in its answers.

Rationale: The project rationally should have accurate answers.

Fit Criterion: Could research answers for riddles.  
Priority: 7

Dependencies: 5, 1

Requirement ID: 9

Type: Robustness or Fault-Tolerance

Description: Inputs should have a level of precision to be accounted for.

Rationale: The project should properly account for answers and not consider single letters or misspellings as answers.

Fit Criterion: Check to see if the answer to a riddle and the input given are similar.  
Priority: 8

Dependencies: 1, 8

Requirement ID: 10

Type: Scalability or Extensibility

Description: The program will be capable of taking on more riddles and be able to use user created documents for riddle creation.

Rationale: Making the program be able to obtain an assortment of riddles increases its scalability and lets it become bigger.

Fit Criterion: Test by making multiple riddle input documents and seeing if they work.  
Priority: 9

Dependencies: 1

Requirement ID: 11

Type: Longevity

Description: The game will have a long life and continue to be played after release.

Rationale: The maze should still be getting played or new downloads after release.

Fit Criterion: Normally a way to test this would be to have the published site keep track of how many people download the game; however, I’m unsure how to do that.  
Priority: 13

Dependencies:

Requirement ID: 12

Type: Maintainability and Support

Description: After release the maze should still be getting support and bugfix updates

Rationale: Upon release, any bugs found after release should be patched as soon as possible.

Fit Criterion: Check if any bugs appear after release and fix them.  
Priority: 10

Dependencies:

Requirement ID: 13

Type: Access

Description: The app will have some access to computer files but only within the folder for the app. It will have read permissions and will use said permissions to generate riddles.

Rationale: The app needs some permissions to function and generate riddles.

Fit Criterion: Ensure that the app isn’t pulling data from other files  
Priority: 11

Dependencies: 1, 10

Requirement ID: 14

Type: Integrity

Description: The app will not make changes to sensitive data.

Rationale: The app should only modify a small spot on the computer and shouldn’t mess with sensitive data.

Fit Criterion: Ensure that the app isn’t modifying data from other files  
Priority: 12

Dependencies:

Requirement ID: 15

Type: Convenience

Description: Program should be convenient to operate and simple to understand.

Rationale: The app layout should be comfortable to use.

Fit Criterion: Have testers give feedback on if UI and app layout was easy and comfortable.  
Priority: 14

Dependencies: 2

Requirement ID: 16

Type: Accessibility

Description: The problem will be easily accessible and easy to use.

Rationale: Seeing as some of the consumers for this game would be elderly, the game must be easy to control so that as many people as possible may use it.

Fit Criterion: Get a couple of elderly testers to see if the game is accessible for them.  
Priority: 16

Dependencies:

Requirement ID: 17

Type: Capacity

Description: The game will take up a small amount of storage capacity.

Rationale: Anyone regardless of computer storage size should be able to run the program, and so, the game must be properly condensed so as to not take up a large storage capacity.

Fit Criterion: Measure the storage the game takes up and compare it to other similar games.  
Priority: 17

Dependencies:

Requirement ID: 18

Type: Privacy

Description: Ensure the privacy of the user is maintained and not compromised.

Rationale: Game should only have access to specific files and shouldn’t compromise the privacy of other files.

Fit Criterion: Check to ensure only files pertaining to the game are interacted with  
Priority: 18

Dependencies: 1, 3, 14

Requirement ID: 19

Type: Immunity

Description: Ensure the program is immune to bugs and cannot be tampered with in a way that breaks the system.

Rationale: The game must be as close to bug free as possible, as bugs cause issues for consumers.

Fit Criterion: Rigorous testing and checks must be done to ensure everything is functional.  
Priority: 19

Dependencies: 1

Requirement ID: 20

Type: Cultural

Description: The product will not be offensive.

Rationale: Program should be able to be appreciated by multiple cultures.

Fit Criterion: Ensure the riddles are not offensive by making sure each riddle is fact checked and that it does not mention any controversial topics.  
Priority: 20

Dependencies: 1

Project Description & Explanation

The project will be a 2D maze game that will also have riddles to solve throughout the maze. The game will have multiple riddle types to solve such as scientific riddles, trivia riddles, sports riddles, etc (Fig 1. Riddle Selection). The program will be written in C++ and will require the following packages:

<iostream> for input and output

<string> for explaining instructions

<fstream> for reading the riddles from another document

<filesystem> for reading riddles from another document

<ctime> for helping set up randomness of riddles

<list> setting up the maze

<cstdlib> setting up randomization

<conio.h> allowing for instant user input

<chrono> setup wait time between correct answers

<thread> setup wait time between correct answers

The program will then select a random maze from a generated collection of mazes and output the maze (Fig 2. Maze Generation). You will be able to traverse the maze, and while doing so a riddle of the selected type will pop up (Fig 3. Riddle generation). Upon getting the answer wrong it will remind you of your remaining attempts (Fig 4. Incorrect Answer), and upon getting the answer right it will inform you that you were correct (Fig 5. Correct Answer). Finally Upon completing the game it will disclose that you failed the game (Fig 6. Game Over), or that you were victorious in the maze and transform the X for exit into a V for victory (Fig 7. Victory). Repository Link: <https://github.com/Wesasaurus/CSU-Senior-Project>

A black screen with white text

AI-generated content may be incorrect.

(Fig 1. Riddle Selection)

A black background with white hashtags

AI-generated content may be incorrect.

(Fig 2. Maze Generation)

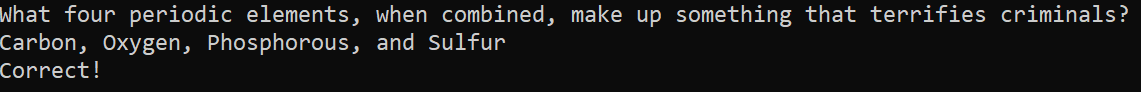


(Fig 3. Riddle generation)

A black screen with white text

AI-generated content may be incorrect.

(Fig 4. Incorrect Answer)



(Fig 5. Correct Answer)

A screen shot of a computer

AI-generated content may be incorrect.

(Fig 6. Game Over)

A black background with white hashtags

AI-generated content may be incorrect.

(Fig 7. Victory)

# 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 (Inside Project Proposal)

## 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
* 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: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Testing Results

Test Case #1 A screenshot of a document

AI-generated content may be incorrect.

Photo evidence

A computer screen shot of a computer program

AI-generated content may be incorrect.

Test Case #2 A screenshot of a test results

AI-generated content may be incorrect.

Photo evidence

A black screen with white text

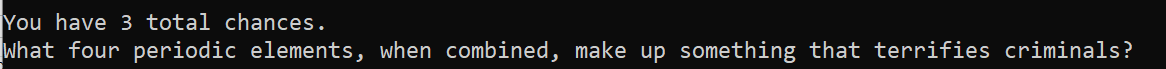
AI-generated content may be incorrect.

Test Case #3 A screenshot of a test

AI-generated content may be incorrect.

Photo evidence

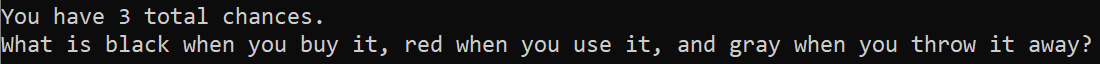
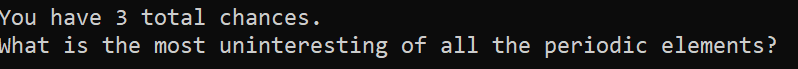
A black background with white hashtags

AI-generated content may be incorrect. 

Test Case #4A screenshot of a test

AI-generated content may be incorrect.

Photo evidence A black background with white text

AI-generated content may be incorrect.A black background with white text

AI-generated content may be incorrect.

Test Case #5A screenshot of a test results

AI-generated content may be incorrect.

Photo evidenceA black background with white text

AI-generated content may be incorrect.

Test Case #6A screenshot of a test results

AI-generated content may be incorrect.

Photo evidenceA black screen with white text

AI-generated content may be incorrect.

Test Case #7A screenshot of a computer

AI-generated content may be incorrect.

Photo evidenceA screenshot of a computer screen

AI-generated content may be incorrect.

Test Case #8A screenshot of a project

AI-generated content may be incorrect.

Photo evidence

A black background with white hashtags

AI-generated content may be incorrect. A black background with white hashtags

AI-generated content may be incorrect. A black background with white and blue squares

AI-generated content may be incorrect.

Test Case #9A screenshot of a document

AI-generated content may be incorrect.

Photo evidence

A black background with white and blue squares

AI-generated content may be incorrect. A black background with white and blue symbols

AI-generated content may be incorrect. A black background with white hashtags

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

AI-generated content may be incorrect.

Test Case #10A screenshot of a computer

AI-generated content may be incorrect.

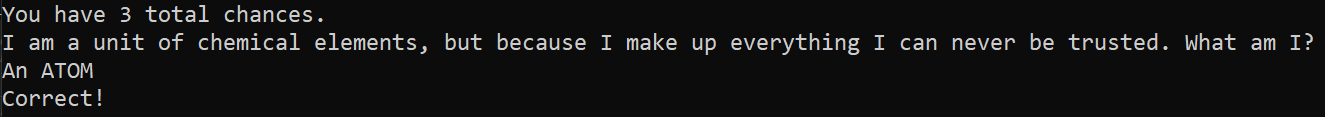
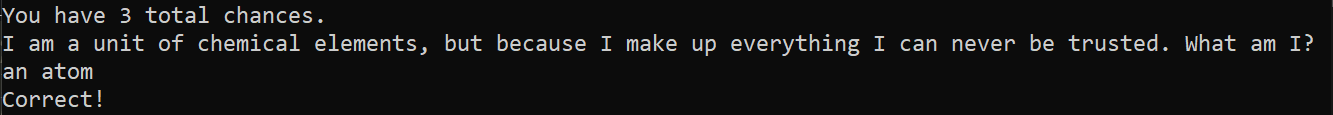
Photo evidence

A screenshot of a computer screen

AI-generated content may be incorrect.

Test Case #11A screenshot of a computer

AI-generated content may be incorrect.

Photo evidence

Test Case #12A screenshot of a computer

AI-generated content may be incorrect.

Photo evidence A screenshot of a computer error

AI-generated content may be incorrect.

Challenges Overcome

There are a multitude of challenges that I needed to overcome to complete this project. One notable example was the development of the riddle option display. During development one of the major issues to arise was that the riddle type display would keep showcasing documents that were not .txt files resulting in files such as mazegame.exe showing as riddle types to be selected. A temporary solution was made for a time to exclude any file with maze in its title; however, that would not stay as it was important to fix the feature. Eventually a solution was found that checked the filetype before displaying the options. Another challenge during development was the development of the maze and its systems. Getting traversal operational was a very hefty task and even once it was completed it was very choppy in its movement, resulting in some much needed optimization to smoothen the experience. This was done through clearing out the screen per display of the maze instead of simply using new lines to jump to the next version of the maze.

Future Enhancements

Multiple future enhancements could be made to better the project at hand. For example more riddles could be provided to further enhance the experience. This can be done very easily through simply adding more .txt files with riddles. Different difficulty riddles could be added for users to choose from. This could be done by adding riddle types with easy medium and hard in the title and making the file contain riddles based on the difficulty. A maze generation formula could be made to get better randomized mazes instead of utilizing the handmade mazes instead. This would be done by making an algorithm that generates mazes with constants such as 1 exit and 1 entrance & would also require vigorous testing to ensure it works properly. It would also need code to find the entrance and place the player in that spot.

Defense Presentation Slides

A maze with text overlay

AI-generated content may be incorrect.

A blue and white logo

AI-generated content may be incorrect.

A two blue and black logos

AI-generated content may be incorrect.

A close-up of a computer

AI-generated content may be incorrect.

A black and white page with white text

AI-generated content may be incorrect. A maze with a few wooden pieces

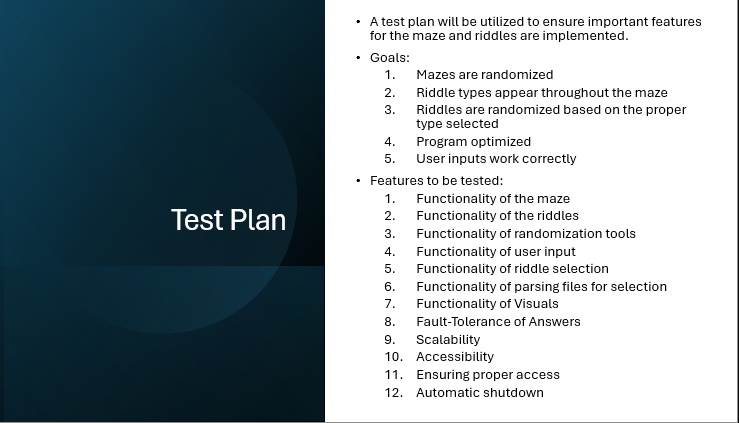
AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A maze with a maze and a maze with a maze and a maze with a maze and a maze with a maze and a maze with a maze and a maze with a maze and a maze with

AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a test

AI-generated content may be incorrect. A person standing in front of a maze

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A close-up of a puzzle

AI-generated content may be incorrect.

Works Cited

Health Benefits of Solving Riddles – DoveMed. (n.d.) [www.dovemed.com/healthy-living/wellness-center/health-benefits-solving-riddles](http://www.dovemed.com/healthy-living/wellness-center/health-benefits-solving-riddles)

Marcel, D. (2009). Puzzles and the brain. Psychology Today. <http://www.psychologytoday.com/us/blog/brain-workout/200904/puzzles-and-the-brain>

Why solving puzzles and Mazes helps with a growth mindset. Kai’s Education. (2021). <https://kaiseducation.com/why-solving-puzzles-and-mazes-helps-with-a-growth-mindset/>