**Maze Solving Application**

**A maze solving application that is able to visualise a run-time process, given user parameters.**

Table of Contents

[ANALYSIS 2](#_Toc146275604)

[Problem statement 2](#_Toc146275605)

[Research outline 2](#_Toc146275606)

[Further research 2](#_Toc146275607)

[Identify end user 2](#_Toc146275608)

[Background 2](#_Toc146275609)

[Objectives 2](#_Toc146275610)

[SMART objectives 3](#_Toc146275611)

[Modelling (diagrams) 3](#_Toc146275612)

[Prototyping and critical path 3](#_Toc146275613)

[DESIGN 3](#_Toc146275614)

[structure / hierarchy chart 3](#_Toc146275615)

[System flowchart 3](#_Toc146275616)

[DFD 3](#_Toc146275617)

[Class diagram 3](#_Toc146275618)

[Other diagrams 3](#_Toc146275619)

[Algorithms (pseudocode/flowcharts) 3](#_Toc146275620)

[Data structures 3](#_Toc146275621)

[File structure and organisation 3](#_Toc146275622)

[Database design 3](#_Toc146275623)

[HCI 3](#_Toc146275624)

[Hardware selection/design 3](#_Toc146275625)

[TECHNICAL SOLUTION 3](#_Toc146275626)

# ANALYSIS

## Problem statement

For my NEA, I would like to demonstrate an application of computer science that can be grasped and be appreciated by both students and high-level computer scientists.

I believe that to achieve this demonstration, I would need to choose an application that can be understood by anyone, as such: The idea of a maze makes complete sense. They represent a perfect blend of logic, creativity, and challenge. Maze generation can be described as a problem that require both algorithmic thinking and artistic design. Since mazes provide an excellent platform for this, they allow me to experiment with various algorithms and design elements to create something that is both visually appealing and artistically complex.

My curiosity on how mazes operate, and “certain tricks” such as “following the right wall of a maze” in order to find the exit, led me on a long search on how mazes can be solved. I believe what made me truly appreciate why I wanted to do this project is that the mathematical approach within the project is complex enough for A Level / University based projects. Since I will have to use things such as how 2D / 3D vectors, the implementation of a 2D array and how to convert each cell within in order to create a FIFO queue in order to go generate a maze

I believe that to ‘future-proof’ the scope of the software, I will designate the project as a “research task”, since there is not a specific end-user but rather a demonstration of computational skill. However, I believe I will

## Research outline

<https://www.mazegenerator.net/>

<https://keesiemeijer.github.io/maze-generator/>

<https://codebox.net/pages/maze-generator/online>

Links shown above are examples of maze generators. These examples incorporate varying features that I would like to “plagiarise”, for example, all the above generators have a parameter page where I can change how the maze is generated. I believe incorporating these features can both make the program enriched with many features that differentiate from other programs. Furthermore, as mentioned, the idea of a parameter page can be improved upon where I can create an online database, where users can upload or download mazes that can be “plugged into the program” to create or present different results.

## Further research

## Identify end user

Students / Teachers / Computer Scientists /

## Background

The maze will consist of multiple python modules that will interact with each other in a modularity system where each function does not need to be used during run time. It will have features such as: Maze-Solving, Exportation of results, Importing mazes, SVG / PNG conversion, Maze-generating.

As the complexity, of the program has not been fully identified by me, I believe that features needed may arise during the development of the program.

## Objectives

Create a maze generator

Create a maze solver, that runs independent of a maze generator (not built-in) (modular and separate)

Create a GUI application that runs both MG and MS whenever it needs to depend on user parameters such as a given maze file / generated maze.

Optional User preferences such as colour of application and mazes, the exportation of time taken to solve a maze given as data in order to be interacted with mathlib in order to generate a report.

Implement different algorithms that can be encapsulated within each maze generator / solver

## SMART objectives

## Modelling (diagrams)

N/A for now…

## Prototyping and critical path

Given the time limit set by my Teacher, I believe that the program will be finished by March. The majority of this time taken for the development of software, however in order to allow the report to also be finished by that time. I will use an agile-based approach that will ensure, using order of modularity, that my program will progress through a decomposed approach where I will improve and develop the software “bit-by-bit”.

.

# DESIGN

## structure / hierarchy chart

## System flowchart

## DFD



## Class diagram

## Other diagrams

## Algorithms (pseudocode/flowcharts)

## Data structures

## File structure and organisation

## Database design

## HCI

## Hardware selection/design

# TECHNICAL SOLUTION