**Maze Solving Application**

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

##### Contents

###### ANALYSIS

Defining Project Research

###### DESIGN

Prototyping (DPS Maze Generation)

###### TESTING

###### MAINTENANCE

###### USER GUIDE

###### EVALUATION

##### **ANALYSIS**

Maze solving and generating is an idea that I have had for a long time now. I believe it encapsulates the very core of computer science, where logical thinking, mathematical constraints and (word where you can see something being done in real time) is used to present the usability and practicality of an application. Furthermore, the maze solver itself is inclusive of high-level practical programming, such as: OOP, stacks

##### **Design**

As per requested by my teacher, a prototype design of our program was required for the teacher to assess how we can produce a program that satisfies the application we decide to develop for the NEA. This maze was developed in order for me to understand the foundations required for a maze generation, and how I can tinker with certain parameters in order to fit the maze for my needs.



The designing and developing of the program can be outlined as such:

* Creating and developing several automated algorithms within python that are able to both generate and solve the mazes.
* Create a GUI that is front-end page for the user, it should contain features such as, saving Mazes that have been generated; use different maze generators and solve, record generating and solving times; (Additionally, we can use mathlib to create a mapping tool that can import and export data onto a graph)
* Once all features are implemented, a more ease design can be collaborated with peer review that is much easier to operate the software.

