User’s Manual

# Maze Solver

# Table of contents

# General Information

## 1.0 General Information

This section just explains the system in general terms and the purpose that it is intended for.

## 1.1 System Overview

The main idea of this project revolves around various path finding algorithms, such as BFS, DFS, Greedy, A*, and Dijkstra's algorithm. The purpose is to visualize how these algorithms work internally to provide a better understanding for anyone who has difficulty grasping the various algorithms.

The software implements maze generation. The program will allow for a user to generate a grid of a range of sizes, draw their own grid in real-time, determine animation speed and generate random mazes for use of ease.

## 1.2 Format of the User Manual

The user manual for this program will be made up of 5 sections: General Information, System Summary, Getting Started, Using the Program, and Reporting.

The system summary provides the general overview and layout of the system. This section will include information on both hardware and software requirements.

Getting Started simply explains where to download the maze solver and how to install it onto your computer.

The Using the Program section will underline a detailed the description of the programs functions.

The reporting section will describe a detailed explanation of what the user will be seeing as the run the program.

# System summary

## 2.0 System Summary

The system summary provides the general overview and layout of the system. This section will include information on both hardware and software requirements.

## 2.1 System Configuration

Maze Solver will work on computers with 32/64-bit Windows 7, 8, or 10 with Intel Core™ i3, i5, or i7. The computer must have Java™ Runtime Environment 8 or higher. and it will require at least 4 GB of DDR3/4 RAM. The display of the computer must be set at 1920 x 1080 to fully display all buttons in the program. Maze Solver is not required to be connected to the internet.

## 2.2 User Access Levels

Anyone that downloads the program will be able to use it if their computer meets the specifications required to run the program. To gain access to the download you must be a member of our gitlab team “2018-CA326-dajayi-mazesolver”.

# Getting Started

## 3.0 Getting Started

Getting Started simply explains where to download the maze solver and how to install it onto your computer.

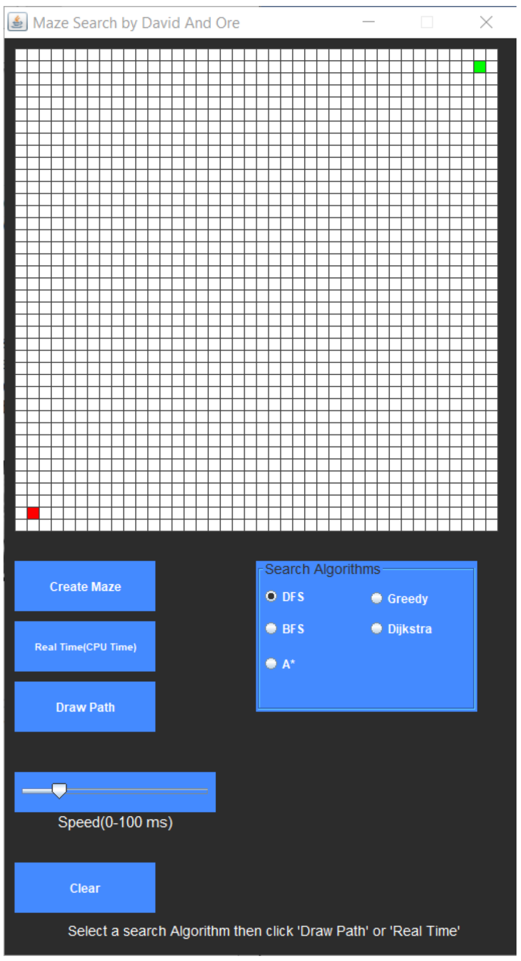
## 3.1 Download and Installation

To download the program, simply just click on the link provided. Here are the steps to installing the program:

1. Download either maze_70x70.jar or maze_70x70_clearn.jar from the provided download link.
2. Double click on the file to open when it is installed.

## 3.2 Program layout

If you have downloaded and installed the program correctly, the layout in image below (Figure 1) is what you should see.



Select a search Algorithm then click 'Draw Path' or 'Real Time'

Figure 1. Program Layout

If all the buttons in the program are not displayed, go to the display setting and set your screen resolution to 1920 x 1080.

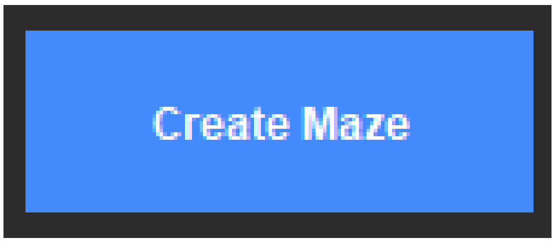
# Using the Program

## 4.0 Using the Program

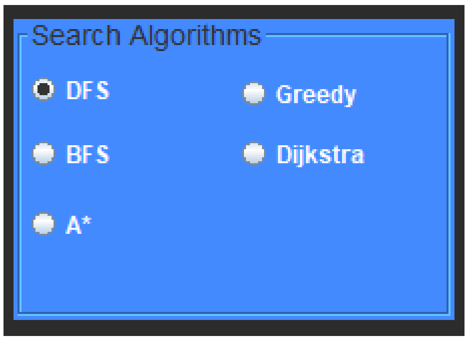
The Using the Program section will underline a detailed the description of the programs functions.

## 4.1 Buttons

- Create Maze – This button allows the user to create a random maze. This is the first button the user should click.



- Search Algorithms – This section of the program layout consists of all the different algorithms the user can select from to search the maze. There are 5 algorithms to select from.

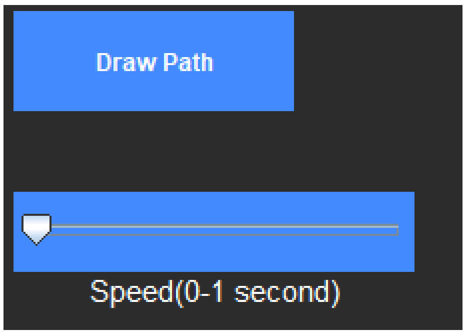

Search Algorithms

[✓ DFS] Greedy

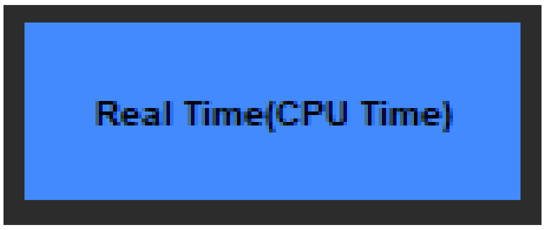
BFS Dijkstra

A*

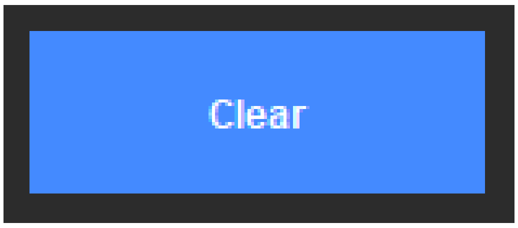
- Draw Path – This button allows the user to choose the speed (in ms) at which the robot searches the maze by. The speed can be adjusted by the user as shown below.



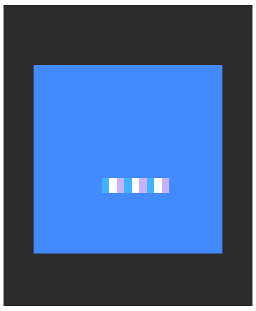
- Real Time (CPU Time) – The user can click this button to search the maze in real time (instant speed). After this button has been clicked, in order to search again the user must click the Create Maze button or the Clear button.



- Clear – The user must clear the maze before they can begin to search it again. Clicking it twice will clear the generated Maze as well as the calculated displayed path.



- Back Button – After the robot has finished searching the maze, some information will be displayed below on the algorithm they chose (described in the Reporting section of the User Manual). The user can click this button to go back to the main menu.



# Reporting

## 5.0 Reporting

The reporting section will describe a detailed explanation of what the user will be seeing as the run the program.

## 5.1 Detailed explanation of the output

After the robot has finished searching the maze, the image below (Figure 2) will be outputted to the user.

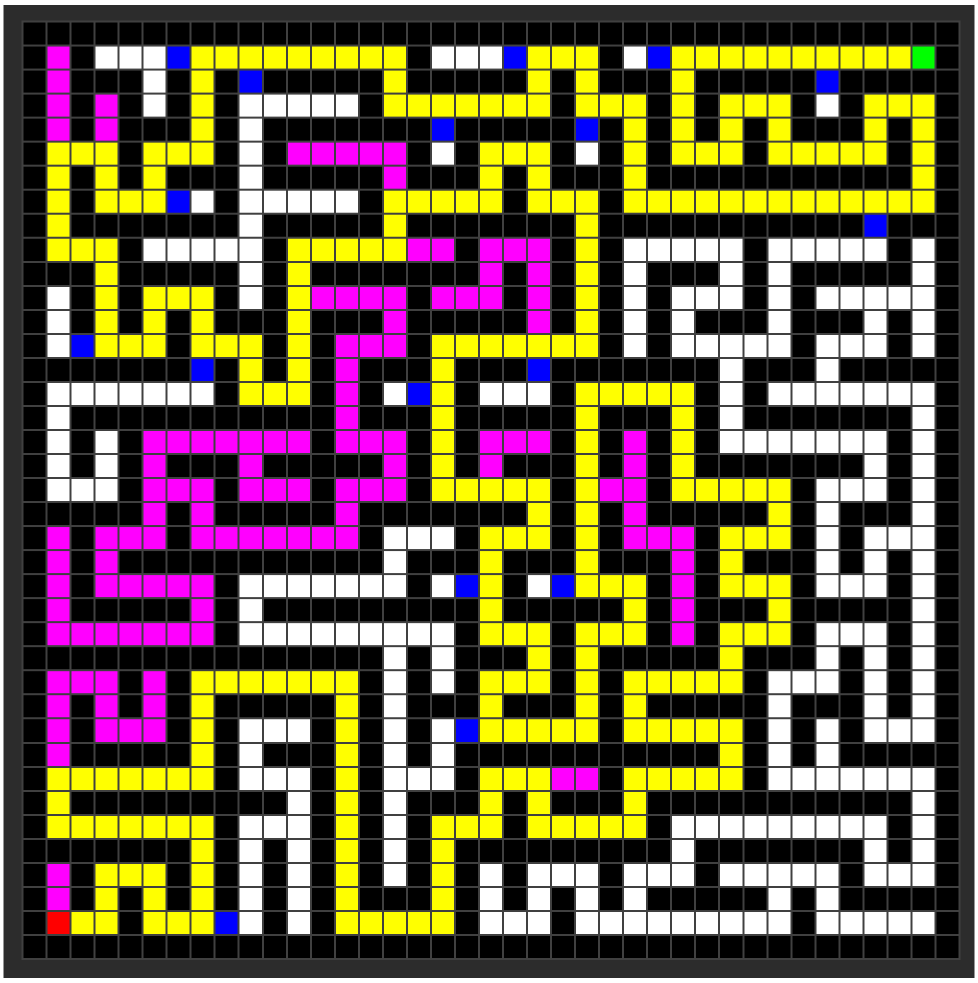
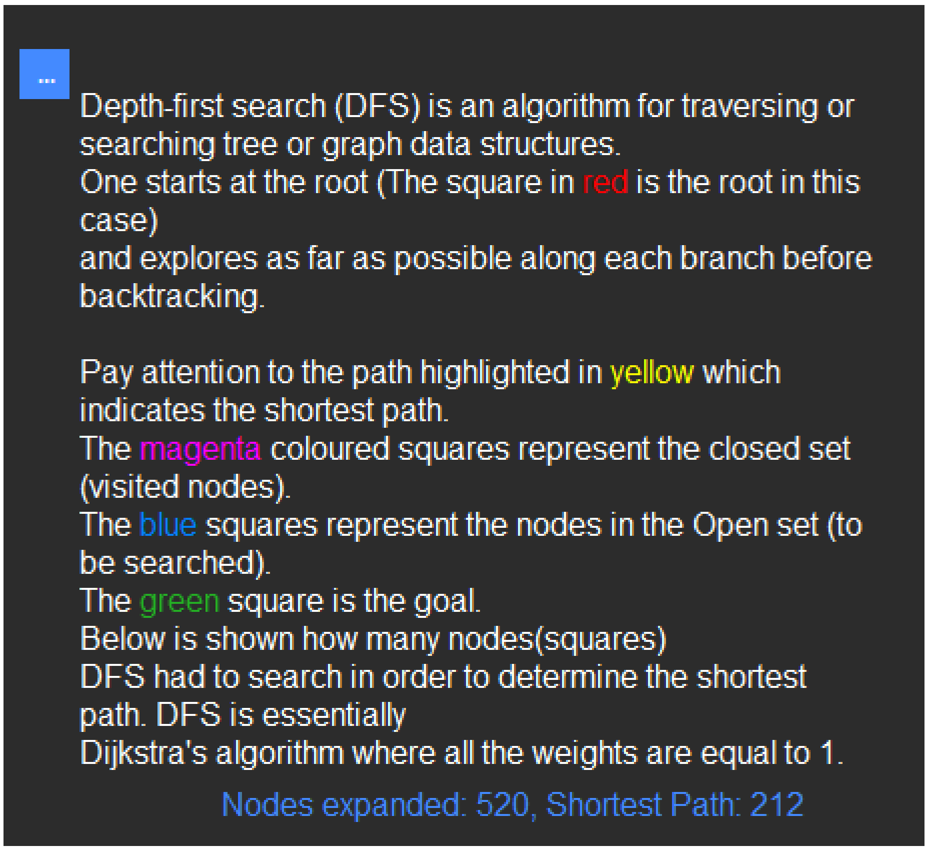


Figure 2. Output

- The node that it is red is the root and is also the robots initial position in the maze.
- The green node is the goal.
- The path highlighted in yellow is the shortest path to the goal.
- The magenta coloured squares represent the closed set (visited nodes).
- The blue squares represent the nodes in the Open set(to be searched).

The user will also see a short description of the algorithm they have chosen, as well as the total number of nodes expanded and the number of squares taken by the shortest path.



Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures.
One starts at the root (The square in red is the root in this case)
and explores as far as possible along each branch before backtracking.

Pay attention to the path highlighted in yellow which indicates the shortest path.
The magenta coloured squares represent the closed set (visited nodes).
The blue squares represent the nodes in the Open set (to be searched).
The green square is the goal.
Below is shown how many nodes(squares)
DFS had to search in order to determine the shortest path. DFS is essentially
Dijkstra's algorithm where all the weights are equal to 1.

Nodes expanded: 520, Shortest Path: 212