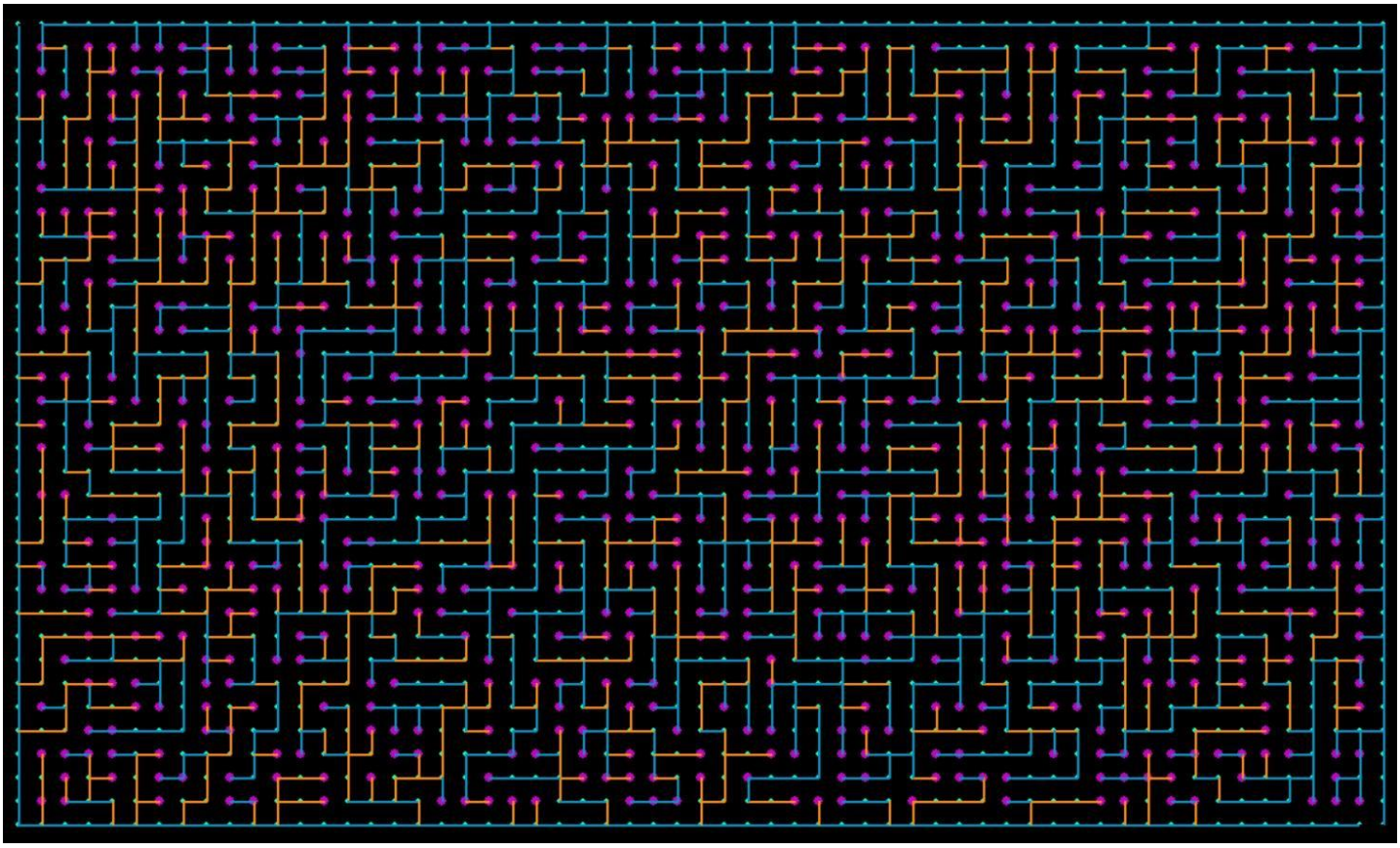
Computer Science Project

2020-21

MAZE GENERATOR



CONTENTS

PROJECT SYNOPSIS

We have created an algorithm that draws complex mazes. These mazes are randomized and there are millions of different possible mazes with our algorithm. There is no bias towards any type of solution in the mazes.

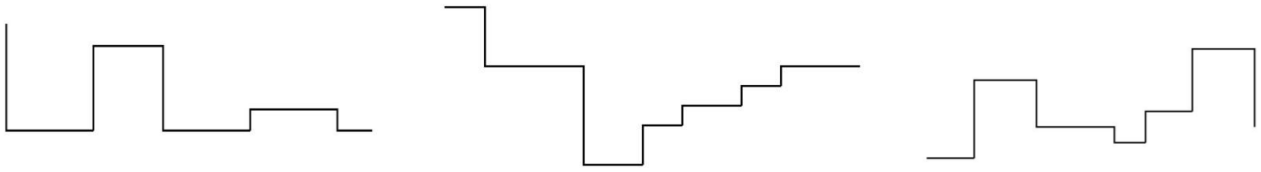
We have developed a game where you can test your maze solving skills against the clock. You can choose from 3 different difficulty levels. But be warned, a wrong move can result in a loss. However, you can reset your solution and try again until you complete the maze.

You can save and retry a maze you have already solved, or try a new maze instead. Can you solve a level 3 maze in under 30 seconds? Each user has a separate account and a unique high score saved.

ALGORITHM

The algorithm is completely original. It is based on a method used to draw mazes on paper. It is faster and generates more random mazes than the already existing maze generation algorithms. We have taken all the factors into consideration to ensure that all the mazes are solvable. These mazes will always have only one solution. However, larger mazes take much longer to generate.

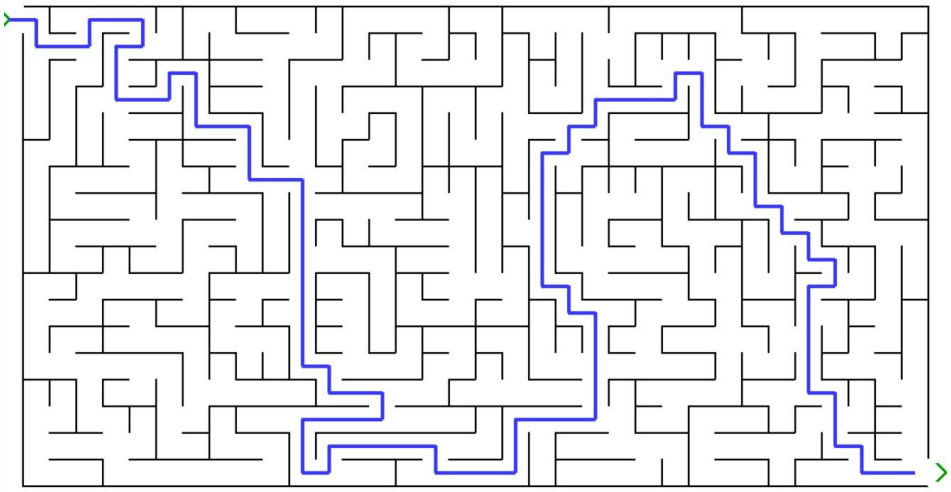
1. A 2D list of coordinates is created to keep track of the locations within the maze. These are the points at which a line can begin. All coordinates with lines on them are marked 'closed', and the remaining are marked 'open'.
2. The program begins by drawing a box with two openings- the starting point and the ending point. (the points through which these lines pass through are now marked 'closed')
3. A random coordinate is chosen from the list of 'open' coordinates.
4. The program randomly generates a list of '*maze like snake*' moves for the walls of the maze.
5. ***How snake moves work-***
    5.1. Snake moves were created to avoid structures like spirals, excessive stairs and long corridors. These lines also help us avoid creating completely isolated areas
    5.2. The main direction of the snake is chosen. For example, let us choose up as the main direction. Now, the line cannot move down
    5.3. If the line goes left, it can continue to go left or go up. The number of lefts and rights are regulated to avoid stairs or very diagonal lines.
    5.4. The line takes a randomly generated number of steps in the chosen direction to avoid long corridors.
    5.5. While stairs and long corridors can be created by the merging of multiple snakes, they are now less probable and contribute more to the difficulty of the maze.
    5.6. The resultant line, i.e. the snake line is drawn until it joins another line.
6. More snake lines are drawn until all the coordinates have been filled.
7. This results in a maze. Since each line is connected to only one other line, there is always a solution. Since no line is left not connected, there is only one solution.



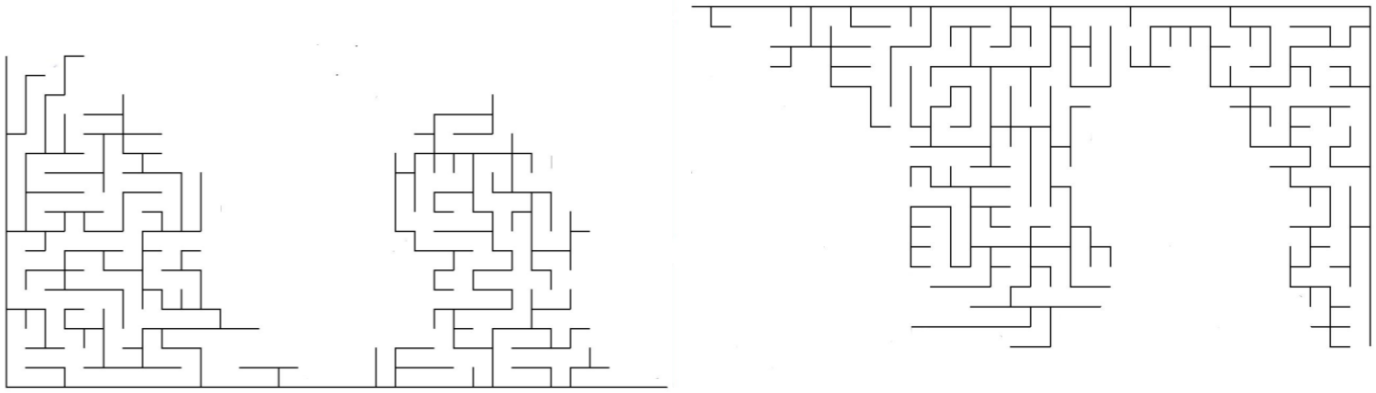
The above are a few examples of snake lines generated for the maze

The solution of a maze

A perfect maze always has only one solution. There will always be only one path between any two points on the maze. This is how the algorithm always generates a perfect maze.



This maze can be split into two halves as shown below.



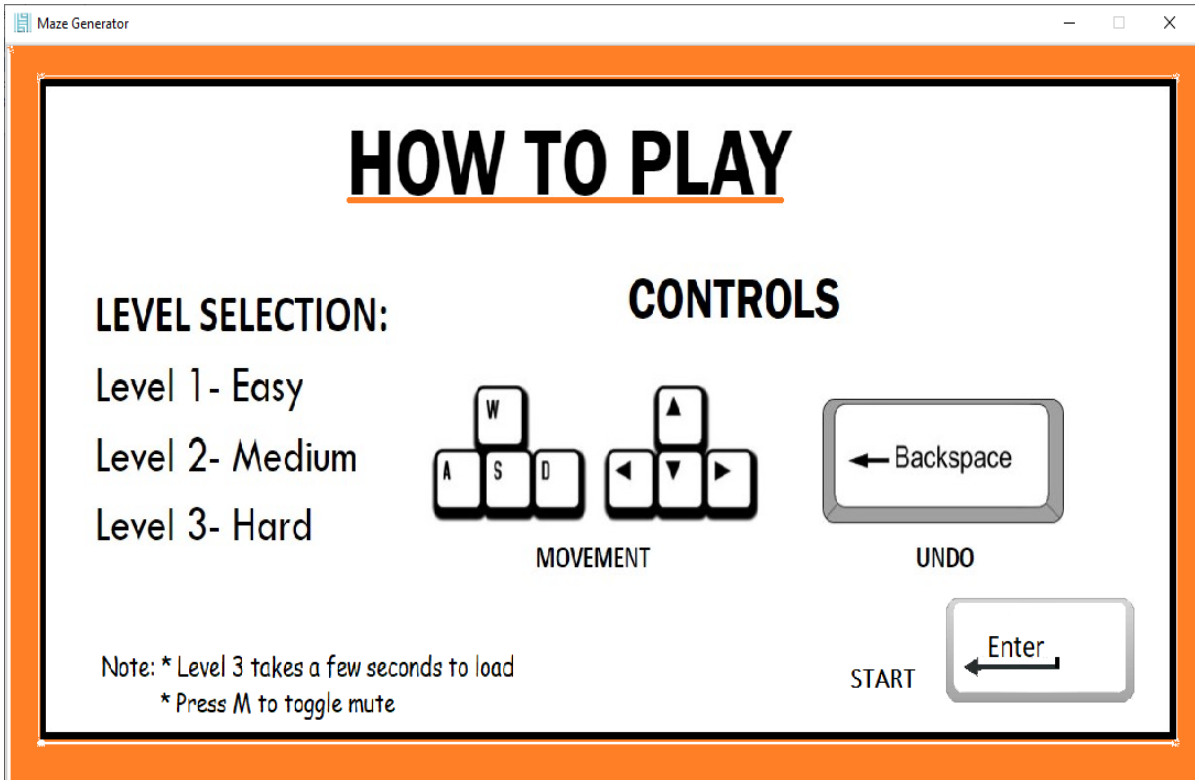
Every line in these halves is interconnected. When joined, the halves form a maze. The only solution to the maze is right between the two halves.

SYSTEM REQUIREMENTS

- PYTHON - System should be able to run python programs
- PYGAME - An additional Pygame module should be installed
- OPERATING SYSTEM – Windows 8/ 9/ 10

MODULES & FUNCTIONS

- Pygame is used to both in maze generation and user interface
- Time is used to measure how long an event takes, like a stopwatch
- Random is used to randomize the maze each time



INSTRUCTION MANUAL

1. The aim of the game is to complete the maze in the shortest time possible.

2. When you run the game, a small screen will pop up, with the words, ‘Maze Generator’. To begin, press enter on the keyboard, or click on the green button which says ‘Play’.

3. You will see a screen with buttons. Select a level by clicking on one of the buttons, or press the level number on the keyboard. You will see a screen with 10 colours you can draw with. Click on any of the squares to select that colour.

4. You will see a maze in a few seconds. As soon it appears, the timer will start. The 2 green arrows represent the start point and end point. You can start moving using the keyboard.

5. Remember, you are not allowed to go back the same way you came. If you want to undo the last move, press backspace. If you accidentally go back, or hit a line, you lose.

6. You will see 2 small buttons at the top of the screen. If you want to restart the same maze, click on ‘Try again’ and start moving from the start point. Otherwise, if you’re done playing, click on ‘Done’.

7. If you click on ‘Done’, or if you successfully complete the maze, you will see a new screen, with 2 green buttons.

8. If you want to try a new maze of the same level, click on ‘New maze of same level’. If you want to change your level, click on ‘Select level and colour’.

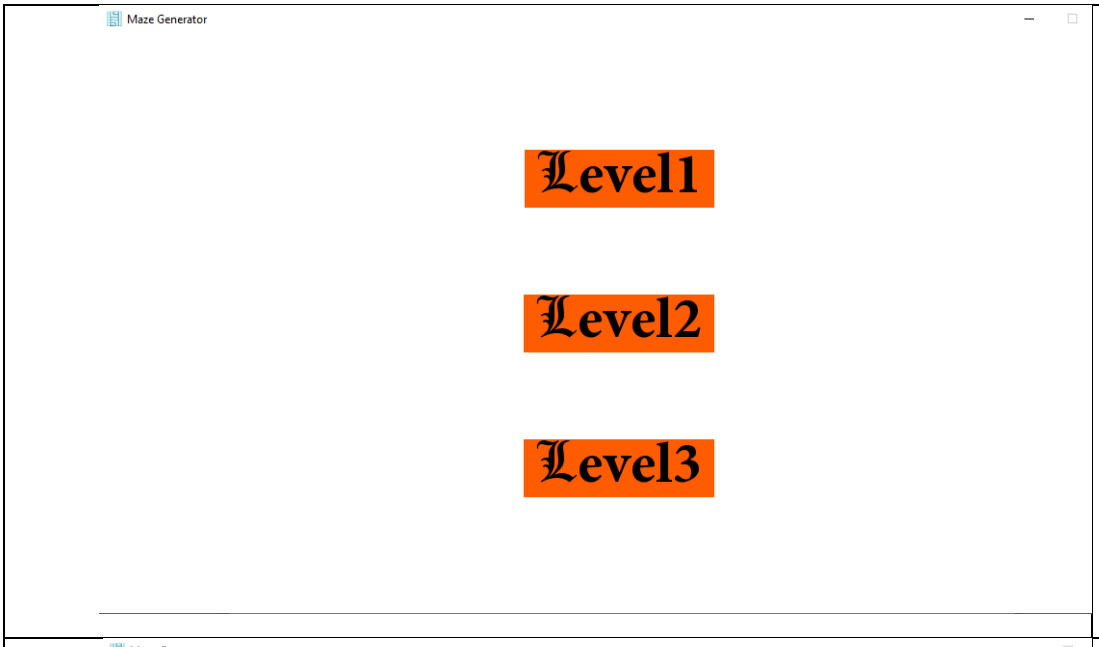
9. Replay as many times as you want, and enjoy! But if you want to exit anytime, just click on the red cross on the top right. You may also press escape on your keyboard.

10. **Game creator’s special tip**: Instead of mentally following a path to see if you and win, at any turn, select a random direction. If that direction turns out to be wrong, you can always press backspace and try again.

PROJECT OUTPUT

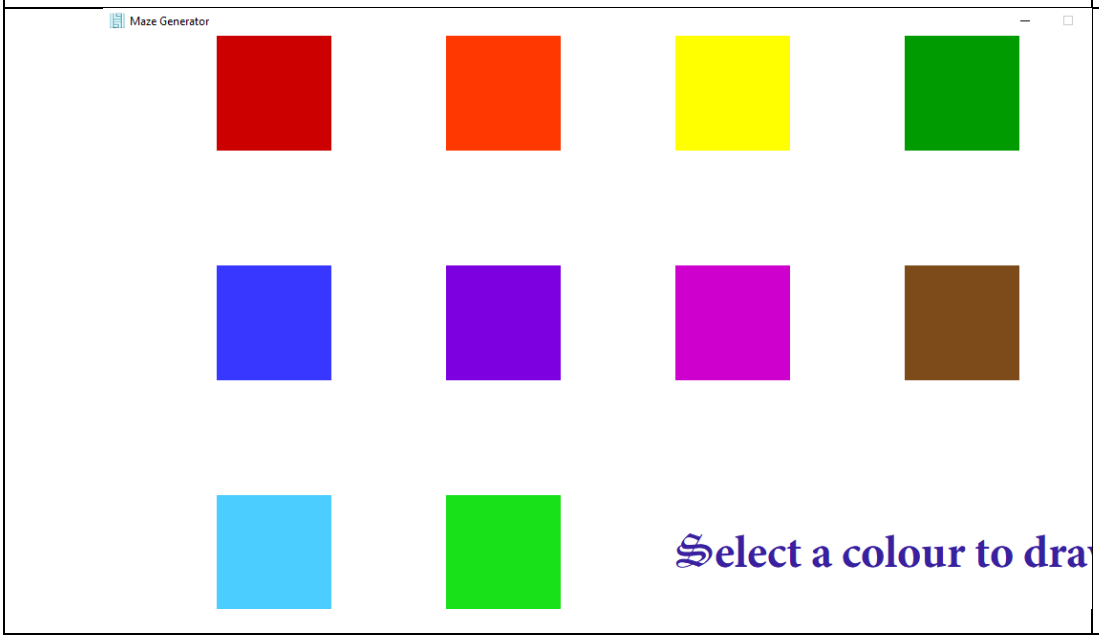
Screen 1

Game Icon/ Logo



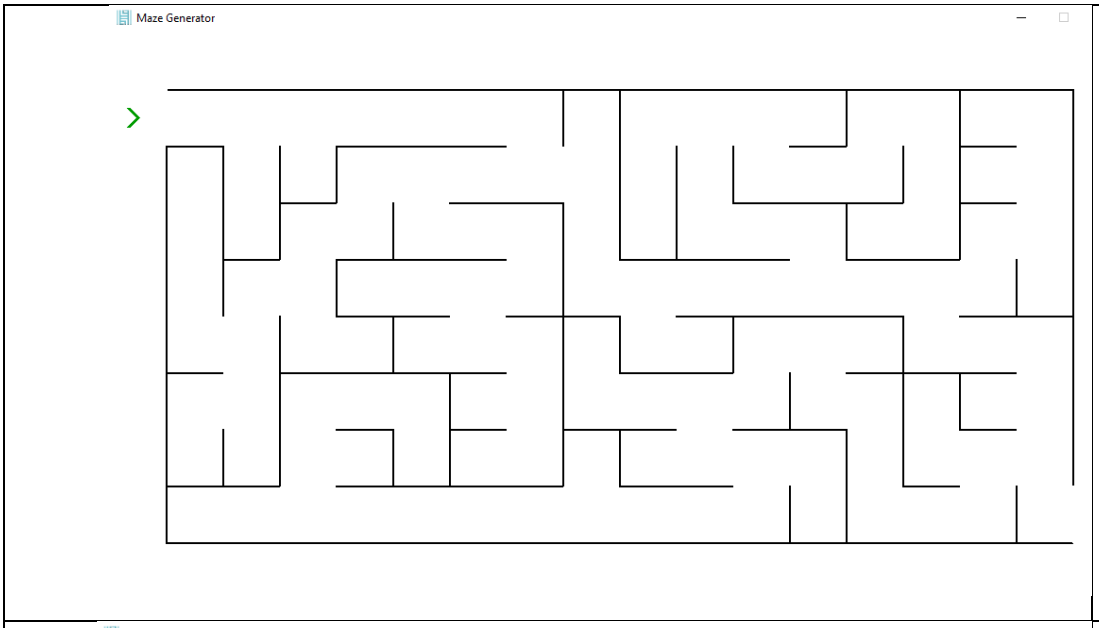
Screen 3

Level selection

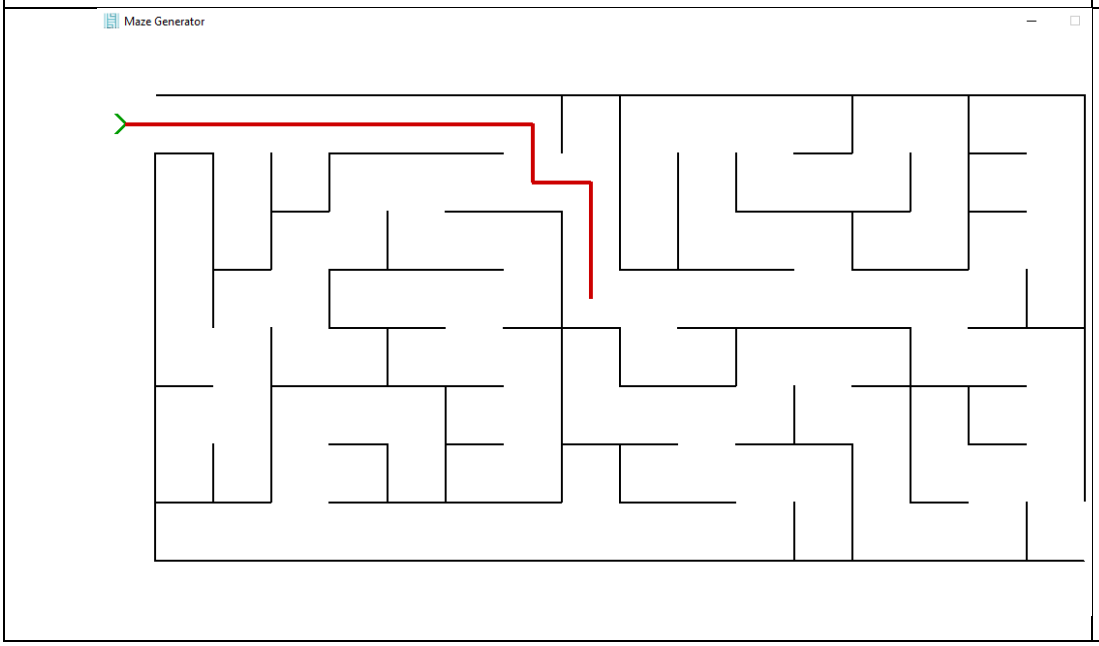


Screen 4

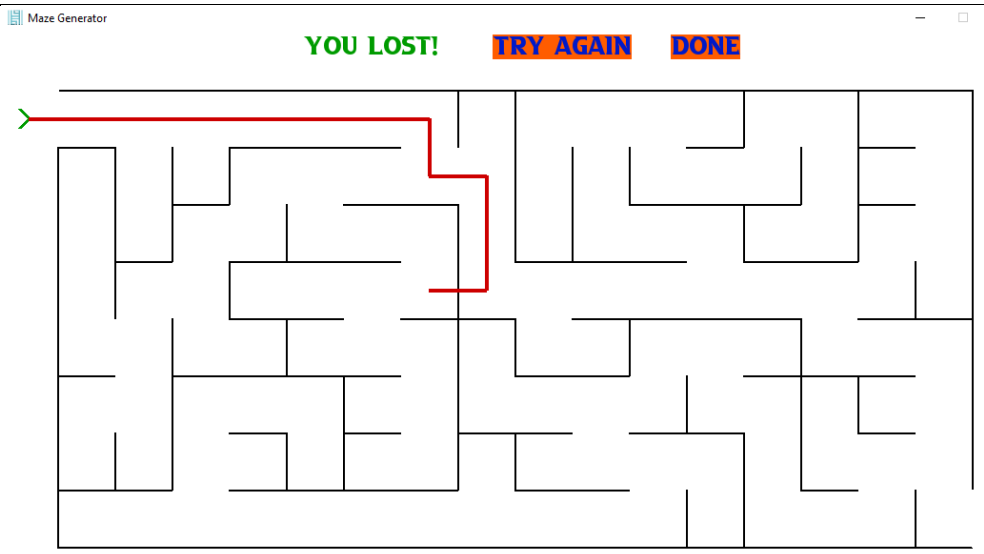
Color selectio



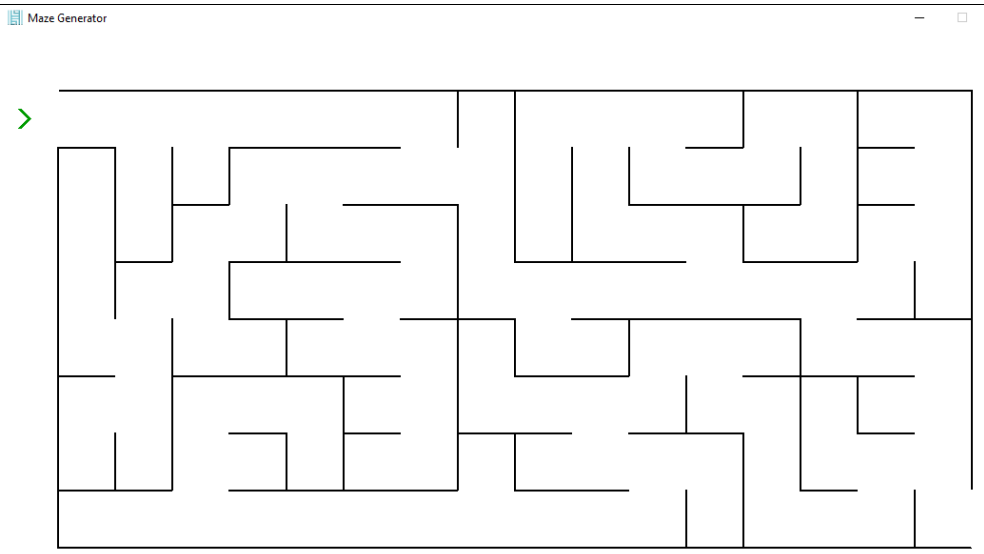
Level 1 Maze example



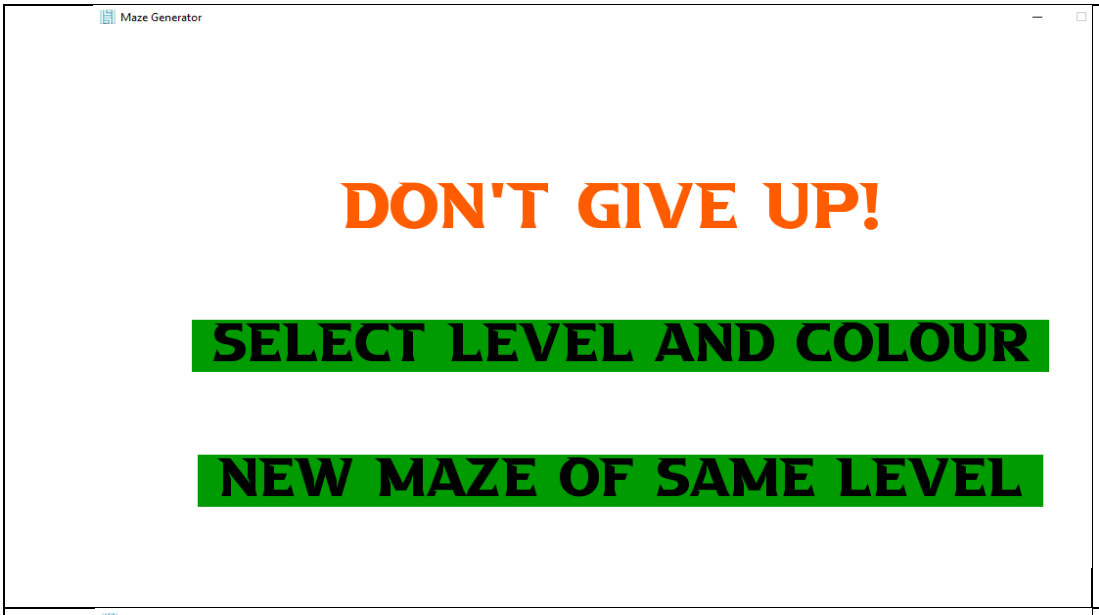
Attempting a level 1 maze



Losing after hitting a lin

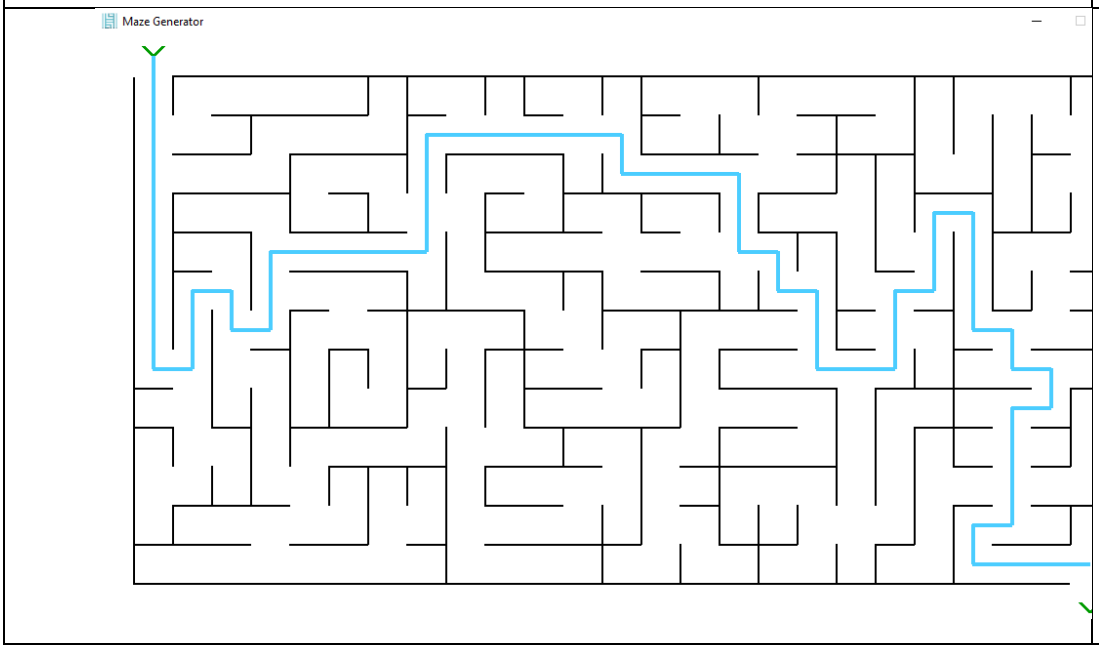


Screen displaye after choosing ‘Try Again’



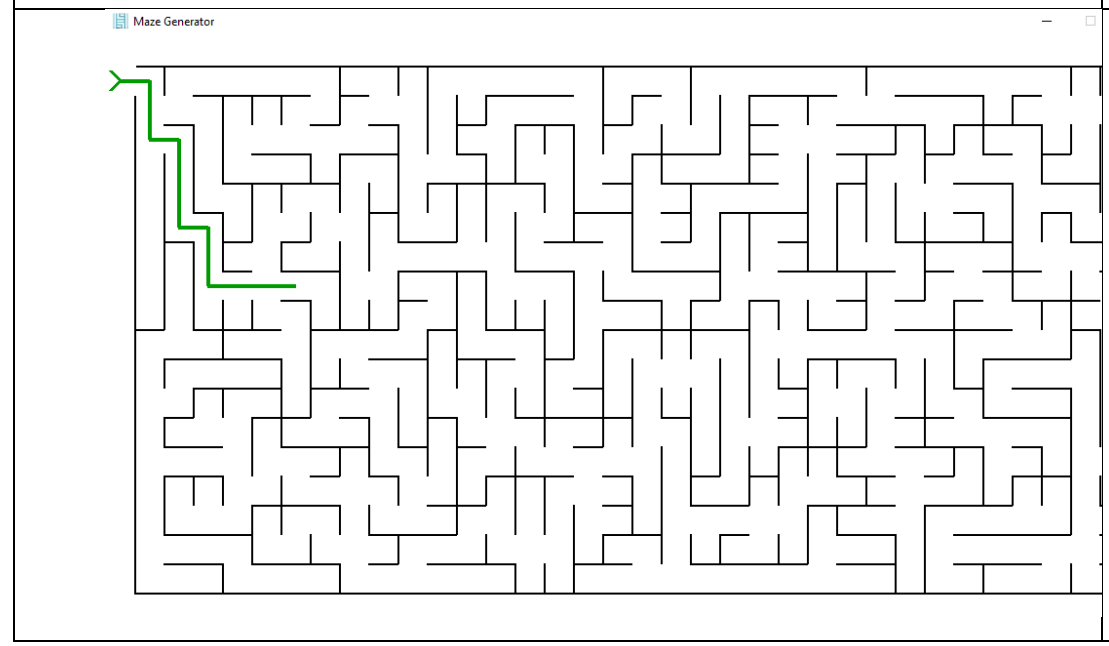
After choosing

'Done'

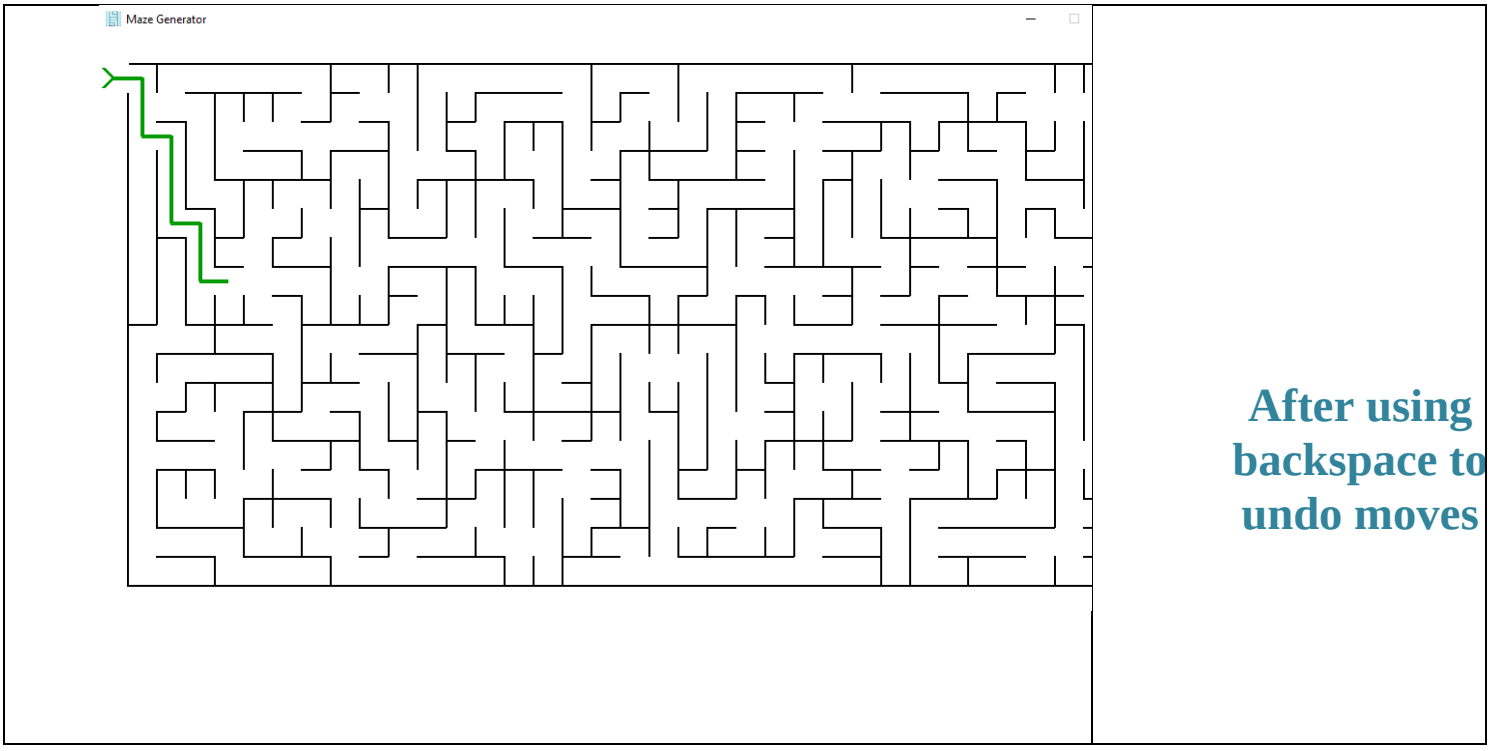


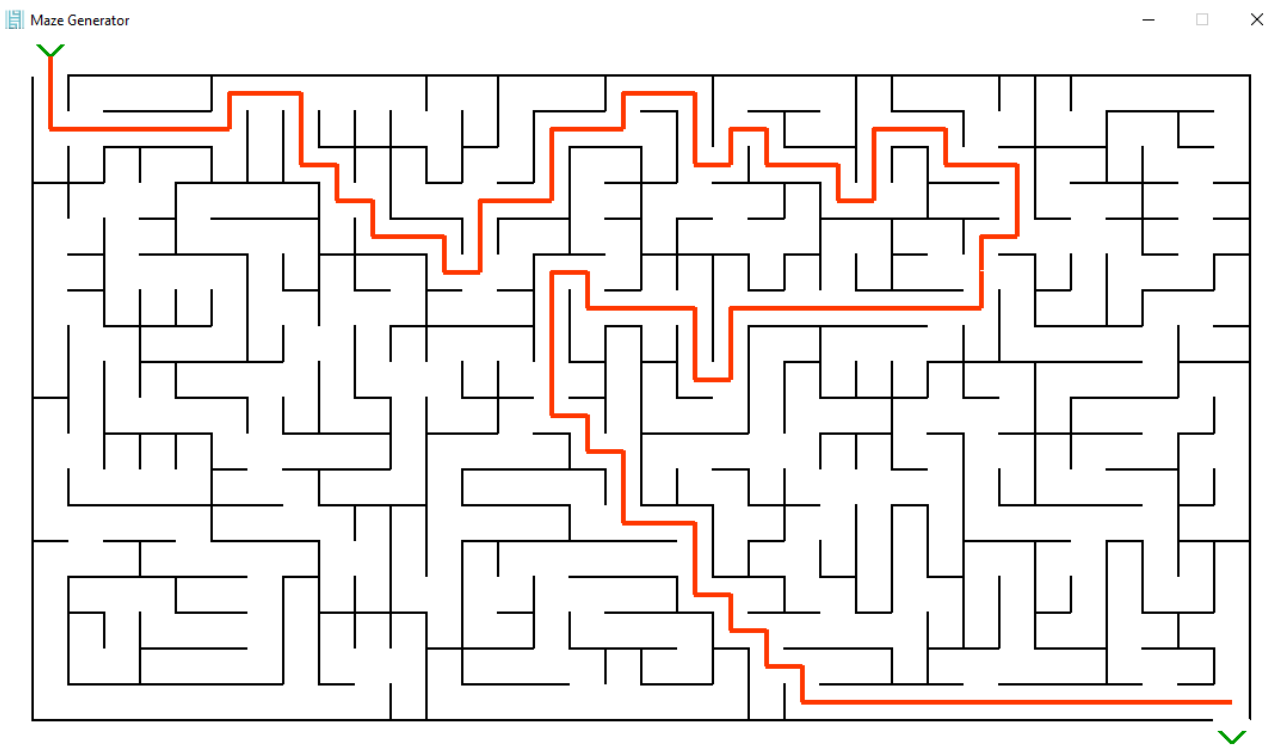
Moving in a Level 2 Maze after selecting blue colour

Winning Scree



Moving in a
Level 3 Maze
after choosing
Green colour





One Last Example!

LIMITATIONS

Right now, there's a lot of scope for improvement in our project. For example, complicated mazes take a lot of time to load. There are many additional features which may be added in later versions of the game. Here are a few:

- Collecting coins/ points on the way
- Other shapes of Mazes like triangle, circle
- Multiplayer game
- A character that moves around, while drawing
- Converting this game into an executable file
- Other puzzles/ mini-games
- 3D mazes

BIBLIOGRAPHY

These are the websites that helped us complete this project:

- https://www.youtube.com
- https://www.geeksforgeeks.org
- https://www.w3schools.com
- https://stackoverflow.com
- https://www.pygame.org/docs