

DSA Project Documentation

Terminal-Based Maze Solver

Objective:

To develop a terminal-based application that generates a random maze, finds a path from the start to the end, and visualizes the maze and path in the terminal.

Prerequisites:

1. Basic understanding of $2D$ arrays and loops.
2. Familiarity with terminal-based input/output.

Specifications:

Input:

1. Size of the maze ($n * n$).
2. Users choose to either print the path, generate another puzzle, or exit the game.

Output:

1. A visual representation of the generated maze in the terminal.

2. A visual representation of the path from start to end, if it exists.

Rules:

1. The maze is represented as a $2D$ array where each cell can be either a wall or an open space.
2. The top-left corner is the start (S) and the bottom-right corner is the end (E).
3. The application should provide options to print the path, generate another puzzle, or exit the game.
4. The number of random walls should be restricted to be less than or equal to 25 % of the total cells to increase the likelihood of a solvable maze.

Guidelines:

1. Generate a random maze of size $n * n$ with walls and open spaces.
2. Implement a pathfinding algorithm to find a path from the start to the end.
3. Visualize the maze and the path in the terminal.
4. Provide options to the user for further actions.

Code Structure:

- **Maze Generation Function:** Generates a random maze and returns it as a $2D$ array.
- **Maze Printing Function:** Prints the maze in the terminal.
- **Pathfinding Function:** Finds a path from the start to the end using a suitable algorithm like BFS or DFS.
- **Path Printing Function:** Prints the path in a readable format.
- **Path Marking Function:** Marks the path on the maze for visualization.

Sample Output:

The maze should be visualized in the terminal with distinct characters or colors for:

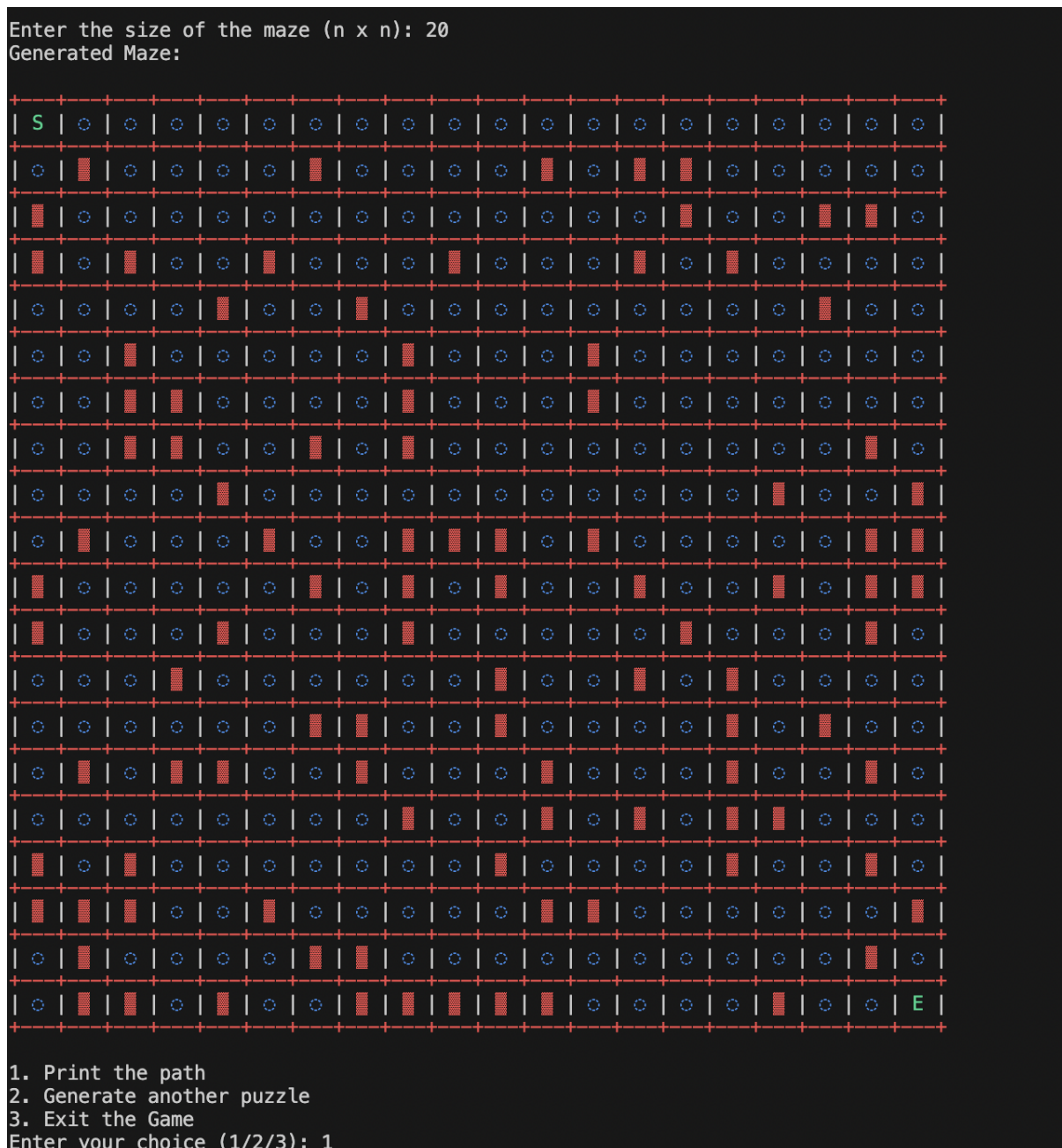
- Start (S) and End (E)
- Walls : ■ (red)
- Open Space ○ (Blue)

- Path (Green)

Future Perspective:

1. Implement additional maze generation algorithms.
2. Implement additional pathfinding algorithms for comparison.

Screenshots:



[illegible]