Maze Solver that compares stack and queue searching (DFS and BFS)

1. Project Background

This project may be applicable to simulations or pathfinding if expanded. As is, the concept involves determining the optimal path through a maze, comparing two data structure frameworks and their efficiency. Shortest path calculations are important for delivery route planning or can be abstracted and applied to process optimization (translating maze sections into process duration).

2. Motivation

I chose this project for a simpler premise than my first exam's concept, but still maintaining an interesting topic for me. Process optimization, and optimization more generally, are something that interests me. This is especially true for side projects and hobbies. Certain robotics competitions are centered around mazes which also strongly draw my interest.

3. Problem Statement

Some types of robotics competitions focus on getting a robot to solve a maze then drive through it as fast as possible. This project aims to design and implement a simplified version of this, that finds the optimal path through a random maze and compares results using two different data structure frameworks.

4. Objectives

Use an algorithm to find a path through the maze.

Repeat to compare two data structure frameworks (stack vs. queue).

5. Key Data Structures

Finding a path through the maze can be done with a queue and array, where the queue finds a path, and the array holds the maze. This could be compared against a different system, using a stack for the pathing and the same array to hold the maze. Both the queue and stack are built on linked-lists.

6. Scope and Limitations

The scope of the project includes a random maze generator based on user-input size for the maze, Depth-first search for the path using a stack, Breadth-first search using a queue, and a timing comparison between the two. The original scope included an optimality test for the paths.

7. Expected Outcomes

The outcome of this project is to compare the efficiency of two different maze solve frameworks, being a stack vs queue and array. This will be evaluated by tracking the time spent solving the randomly generated maze(s).