Daniel Matar, Neel Vishalbhai Lodhiya, Aman Bhardwaj, and Samuel Borough Kamau

SIT215 Computational Intelligence

Assignment 1: Search for Agent Navigation

Contents

[Introduction 2](#_Toc130831455)

[Problem Description 3](#_Toc130831456)

[Results 4](#_Toc130831457)

[Analysis 4](#_Toc130831458)

[Conclusions 5](#_Toc130831459)

[Lessons Learned 5](#_Toc130831460)

[Acknowledgement of External Assistance (If Applicable) 6](#_Toc130831461)

[References 7](#_Toc130831462)

# Introduction

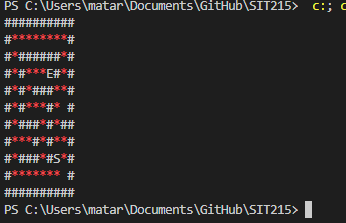
This is the first assignment task of SIT215 Computational Intelligence. This report details our findings in completing this task, what problems we faced, our results, and lessons learned.

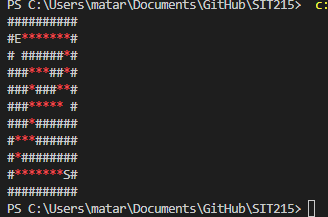
# Problem Description

In this project, we will implement the Depth-First Search (DFS) algorithm to help a nonplayable character (NPC) navigate a maze. The maze is represented as a 2D grid, with walls represented as blocks and open paths represented as empty spaces. The NPC is represented by an ASCII character in the game, and can move in four directions (up, down, left, and right). Our task is to implement the DFS algorithm to find a path for the NPC to reach the endpoint in the maze. The NPC must navigate around walls to reach the endpoint. We will also implement a visualization of the NPC’s movement in the maze.

# Results

Results of our solution show that DFS finds the path (Though it does not remove vertices that are not part of the solution afterwards). But it does not always find the shortest path to the endpoint.





Though as mentioned our solution does not remove the vertices not apart of the path, which makes things hard to see which is the correct path. To remedy this, we simply checked to see if the next vertex with DFS returns true or false and set the vertex back to its original character if it was false.

# Analysis

# Conclusions & Lessons Learned

DFS is quite simple and if we wish to improve on our solution we would be better off using BFS which is much better at finding the best/shortest path for these types of problems. We were able to find a path to the endpoint from different starting points, visualised this by showing the path in a different colour compared to the walls.

# Acknowledgement of External Assistance (If Applicable)

We have used other unit material to get some information on DFS and BFS from SIT320 Advanced Algorithms (Full document included in GitHub), that was written by one of the students in this group (Daniel Matar) which is based off that unit’s material.

# References

Blades, A. (2020, March 08). *Solving Mazes with Depth-First Search*. Retrieved from Medium: https://medium.com/swlh/solving-mazes-with-depth-first-search-e315771317ae

Zaidi, N. (n.d.). *Deakin Sync Module 3a Graphs Part 1 - DFS.* Retrieved July 29, 2021, from https://d2l.deakin.edu.au/d2l/le/content/1031061/viewContent/5742250/View