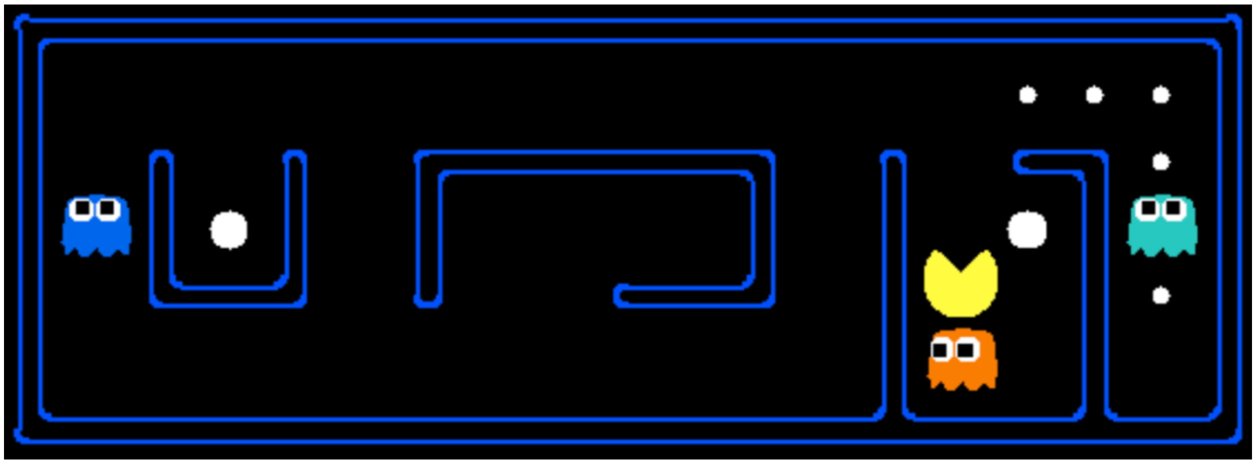
CSC14003 – Artificial Intelligence

PROJECT 01: SEARCH

Project description



You are given a file that describes Pac-man World. Suggest or implement learned algorithms to assist Pac-Man in finding food without getting killed by monsters.

In the game Pac-Man, both Pac-Man and the monsters are constrained to moving in four directions: left, right, up, and down. They are not able to move through walls. The game is divided into four distinct levels, and each level has its own set of rules.

- Level 1: Pac-Man is aware of the food's position on the map, and there are no monsters present. There is only one food item on the map.
- Level 2: Monsters are stationary and do not move around. If Pac-Man and a monster collide with each other, the game ends. There is still one food item on the map, and Pac-Man knows its position.
- Level 3: Pac-Man's visibility is limited to its nearest three steps. Foods outside this range are not visible to Pac-Man. Pac-Man can only scan the adjacent tiles within the 8 tiles x 3 range. There are multiple food items spread throughout the map. Monsters can move one step in any valid direction around their initial location at the start of the game. Both Pac-Man and monsters move one step per turn.

- Level 4 (difficult) involves an enclosed map where monsters relentlessly pursue Pac-Man. Pac-Man must gather as much food as possible while avoiding being overtaken by any monster. The monsters have the ability to pass through each other. Both Pac-Man and the monsters move one step per turn, and the map contains a multitude of food items.

The calculation of game points follows these rules:

- Each movement deducts 1 point from your score.
- Collecting each food item awards you 20 points.

To comprehensively compare the performance of different algorithms, it is recommended to run them on various graphs and evaluate them based on the following aspects:

- Time is taken to complete the task.
- Length of the discovered paths.

It is particularly important to generate challenging maps, such as placing Pac-Man between two monsters or creating a scenario where walls surround Pac-Man on all sides. This will test the algorithms' abilities to handle difficult situations.

Specifications

Input: The given graph is represented by its adjacency matrix, which is stored in the input file (e.g., map1.txt). The format of the input file is as follows:

- The first line contains two integers N x M, indicating the size of the map.
- The next N lines represent the N x M map matrix. Each line contains M integers. The value at position [i, j] (row i, column j) determines the presence of a wall, food, or monster. A value of 1 represents a wall, 2 represents food, 3 represents a monster, and 0 represents an empty path.
- The last line contains a pair of integers indicating the indices of Pacman's position (indices start from 0).

Output:

- If a graphical display is not used, the result can be stored in a text file, such as result1.txt. The file may include the pathfinding for Pacman, the path length, and the game points. Each step of movement can be displayed individually or all steps can be displayed on a single map. However, when monsters are able to move, steps must be clearly separated.

- It is recommended to utilize a graphic library for displaying the results.

Requirements

No.	**Specifications**	**Scores**
1	Finish level 1 successfully.	15%
2	Finish level 2 successfully.	15%
3	Finish level 3 successfully.	10%
4	Finish level 4 successfully.	10%
5	Graphical demonstration of each step of the running process. You can demo on the console screen or use any other graphic library.	10%
6	Generate at least 5 maps with differences in the number and structure of walls, monsters, and food.	10%
7	Report the algorithm, and experiment with some reflections or comments.	30%
Total		100%

Notice

This assignment will be completed in **groups**, with a maximum of 4 members per group. To prepare, you will need to create a folder that contains various subfolders, including source, input, output, and document. The report must give the following information:

- Your detailed information (Student Id, Full Name)
- Assignment Plan
- Environment to compile and run your program.
- Estimating the degree of completion level for each requirement.
- References (if any)

Your team can use any programming language to be, but Python is encouraged

Any plagiarism, tricks, or any lie will have 0 points for the course grade.

Contact this email if you have any questions about project (quochuyy2000@gmail.com).