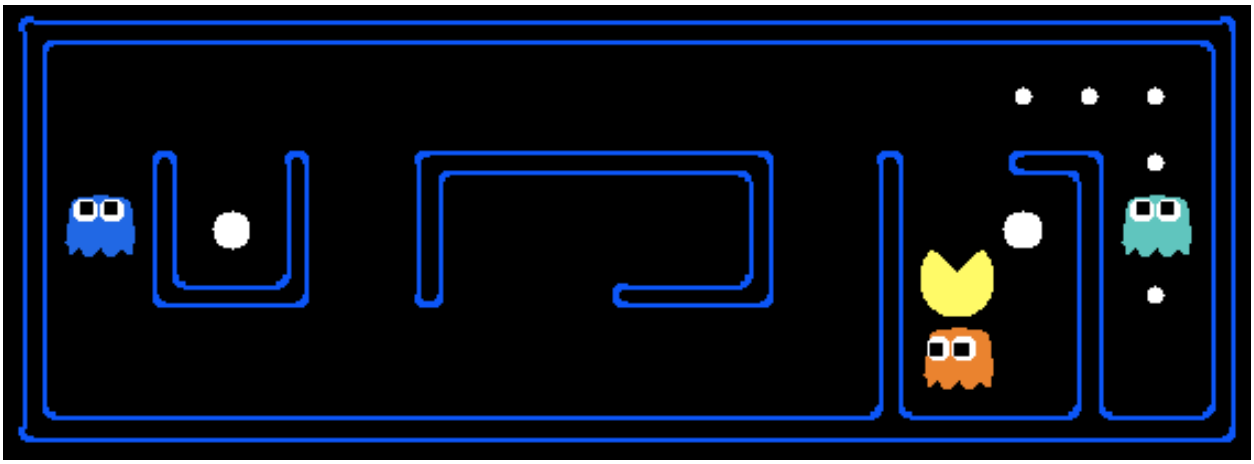
Project 01
SEARCH

1. Description

You are given a file which describe Pac-man World. Propose or apply learned algorithms to help Pac-Man to find foods without dying by monsters.



Pacman or monsters only moves in 4 direction: left, right, bottom, up and cannot move over or through the wall. The game has four levels:

- Level 1: Pac-man know the food's position in map and monsters do not appear in map. There is only one food in the map.
- Level 2: monsters stand in the place ever (never move around). If Pac-man pass through the monster or vice versa, game is over. There is still one food in the map and Pac-man know its position.
- Level 3: Pac-man cannot see the foods if they are outside Pacman's nearest **three-step**. It means that Pac-man just only scan all the adjacent him (8 tiles x 3). There are many foods in the map. Monsters just move one step in any valid direction (if any) around the initial location at the start of the game. Each step Pacman go, each step Monsters move.
- Level 4 (difficult): map is opened. Monsters will seek and kill Pac-man. Pac-man want to get food as much as possible. Pacman will die if at least one monster passes him. It is ok for monsters go through each other. Each step Pacman go, each step Monsters move. The food is so many.

Game points is calculated as following rules:

- Each moving step, your point will be decreased by 1.
- Each food you take, 20 points will be given for you.

You may need to run your algorithm on many different graphs to make a comprehensive comparison of these algorithms' performance regarding the following aspects:

- Time to finished
- The length of the discovered paths

Specially, you should generate some difficult maps such as Pac-man is stay among two monster or wall is around in all side.

2. Specifications

- **Input:** the given graph is represented by its adjacency matrix, which is stored in the input file, for example, **map1.txt**. The input file format is described as follows:
  - The first line contains two integers N x M, which is the size of map.
  - N next lines represent the N × M map matrix. Each line contains M integers. The number at [i, j] (row i, column j) determines whether wall, food or monster is set. If there is wall at this position, we will have value 1. If there is food, we will have value 2. If there is monster, we will have 3. Otherwise (empty path), we will have 0.
  - The last line stores a pair of integers describing the indices of the **Pacman** position (indices start from 0).
- **Output**:
  - If you don't use graphic, you can display a result map in text file with pathfinding for Pacman, path length, game point such as **result1.txt**. You can display each step of moving or display all steps in one map. However, in case that monster can move, you must separate steps clearly.
  - I recommend you should use some graphic library for display results.

3. 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 in console screen or use any other graphical library.	10%
6	Generate at least 5 maps with difference in number and structure of walls, monsters, and food.	10%
7	Report your algorithm, experiment with some reflection or comments.	30%
Total		100%

4. Notice

- This is a **GROUP** assignment. Each group has a maximum of 4 members.
- Duration: about 3 or 4 weeks.
- Your group can use any programming language to do.
- Beside above requirements, report must also give the following information:
  - Your detail 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)
- Any plagiarism, any tricks, or any lie will have 0 point for course grade.