VIETNAM NATIONAL UNIVERSITY

UNIVERSITY OF SCIENCE

TOPIC

Project 01

MEMBERS

Lê Thanh Bình	18127041
Lư Ngọc Liên	18127046
Nguyễn Thị Anh Đào	18127272
Vũ Công Thành	18127218

Course: Artificial Intelligence

Ho Chi Minh City– 2020

Contents

Assignment Plan

Tasks	Lê Thanh Bình	Lư Ngọc Liên	Nguyễn Thị Anh Đào	Vũ Công Thành
Create text files	No	No	5 maps for each levels (level 3 and level 4) 5%	5 maps for each levels (level 1 and level 2) 5%
Read files	No	Function to read maze from file 1%	No	No
Graphic	Draw map (monster, food and pacman) from file. 5%	Pacman move follow path 4%	No	No
Level 1 and level 2	No	No	A* function to find the path for pacman 15%	BFS function to find the path for pacman 15%
Level 3	Monsters move randomly 5%	No	Find path for pacman 10%	No
Level 4	Monster change path when pacman move. 1%	Using A* to find path for Monsters 4%		
Report	Write report 15% for each member		No	No

Self rating:

Project: 80% overall.

For each member:

+ Lê Thanh Bình: 26%

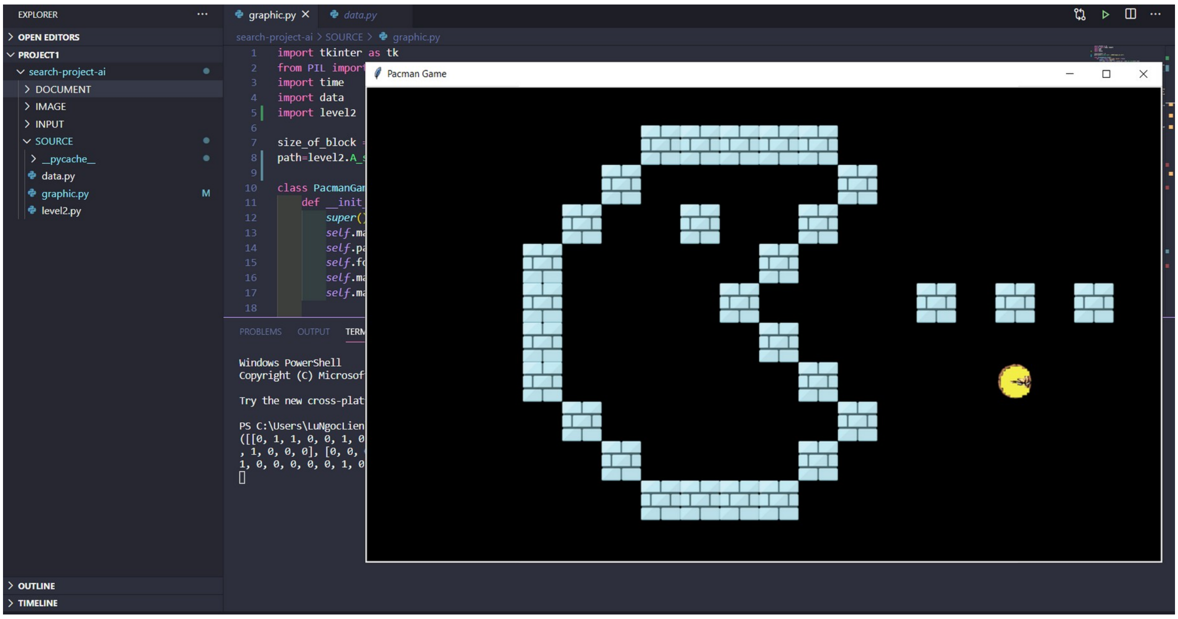
+ Lư Ngọc Liên: 24%

+ Nguyễn Thị Anh Đào: 30%

+ Vũ Công Thành: 20%

Environment to compile and run the program

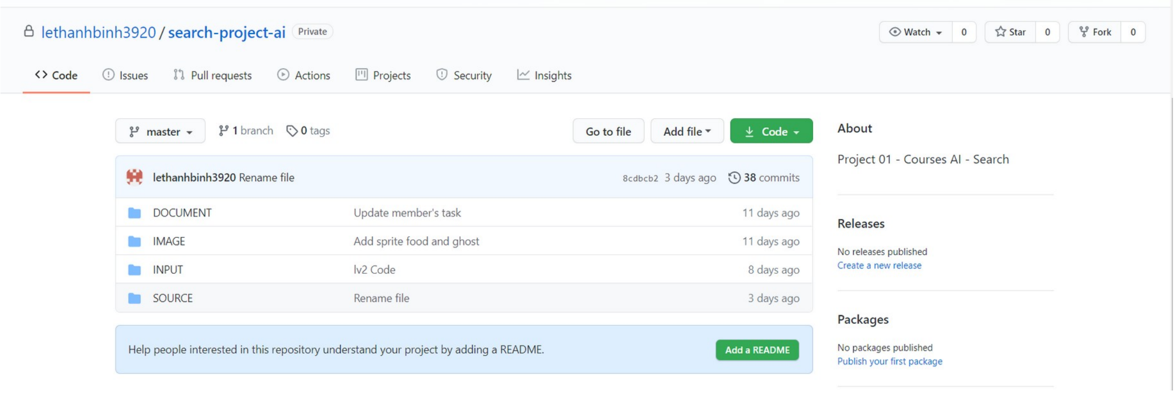
Environment to compile and run the program: Visual Code



```
import tkinter as tk
from PIL import
import time
import data
import level2

size_of_block
path=level2.A_

class PacmanGar
    def __init_
        super()
        self.ma
        self.pa
        self.fo
        self.ma
        self.ma
```

```
Windows PowerShell
Copyright (C) Microsof

Try the new cross-plat

PS C:\Users\LuNgocLien
([[0, 1, 1, 0, 0, 1, 0
, 1, 0, 0, 0], [0, 0,
1, 0, 0, 0, 0, 0, 1, 0
```

Version control: GitHub



lethanhbinh3920 Rename file 8cdbcb2 3 days ago 38 commits

DOCUMENT	Update member's task	11 days ago
IMAGE	Add sprite food and ghost	11 days ago
INPUT	lv2 Code	8 days ago
SOURCE	Rename file	3 days ago

Help people interested in this repository understand your project by adding a README.

About

Project 01 - Courses AI - Search

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Estimating the degree of completion level for each requirement

1. Level 1 and level 2

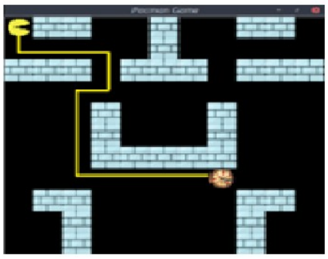
The idea of level 1 and level 2 is the same because at level 2 the monster does not move, so we consider the monster as a wall. The mission is pacman finds the food.

We use two searching algorithms:

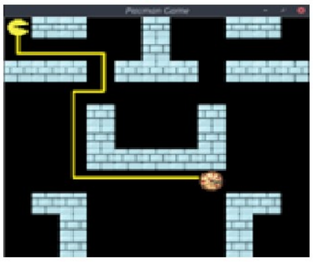
- A* search
- BFS.

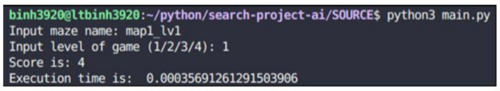
- MAP 1
  - LEVEL 1:

A*

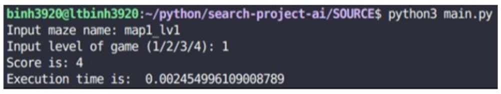


BFS





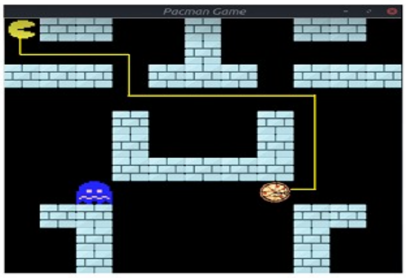
```
binh3920@ltbinh3920:~/python/search-project-ai/SOURCE$ python3 main.py
Input maze name: map1_lv1
Input level of game (1/2/3/4): 1
Score is: 4
Execution time is:  0.00035691261291503906
```



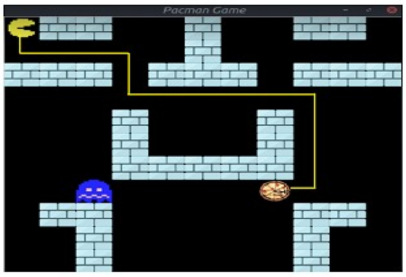
```
binh3920@ltbinh3920:~/python/search-project-ai/SOURCE$ python3 main.py
Input maze name: map1_lv1
Input level of game (1/2/3/4): 1
Score is: 4
Execution time is:  0.002454996109008789
```

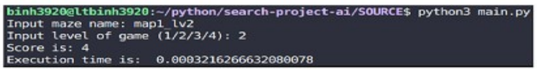
  - LEVEL 2:

A*

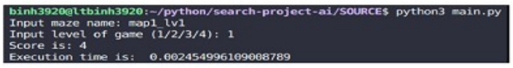


BFS





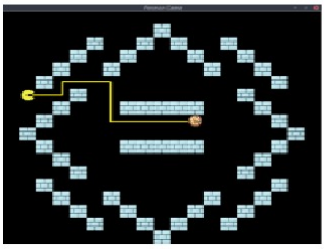
```
binh3920@ltbinh3920:~/python/search-project-ai/SOURCE$ python3 main.py
Input maze name: map1_lv2
Input level of game (1/2/3/4): 2
Score is: 4
Execution time is:  0.0003216266632080078
```

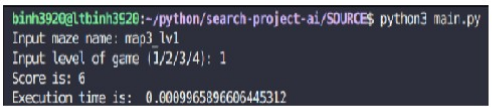


```
binh3920@ltbinh3920:~/python/search-project-ai/SOURCE$ python3 main.py
Input maze name: map1_lv1
Input level of game (1/2/3/4): 1
Score is: 4
Execution time is:  0.002454996109008789
```

- MAP 2
  - LEVEL 1:

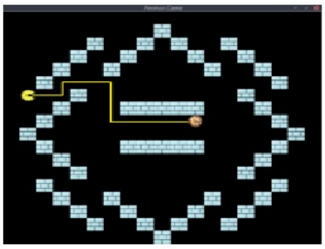
A*

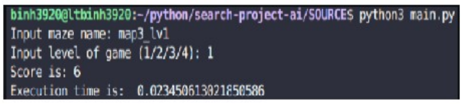




```
binh3920@ltbinh3920:~/python/search-project-ai/SOURCE$ python3 main.py
Input maze name: map3_lvl
Input level of game (1/2/3/4): 1
Score is: 6
Execution time is:  0.0009965896606445312
```

BFS

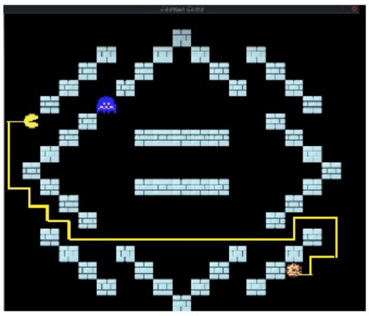


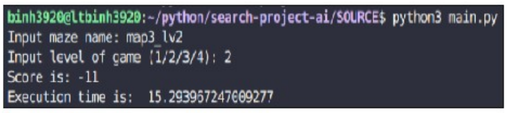


```
binh3920@ltbinh3920:~/python/search-project-ai/SOURCE$ python3 main.py
Input maze name: map3_lvl
Input level of game (1/2/3/4): 1
Score is: 6
Execution time is:  0.023450613021850586
```

  - LEVEL 2:

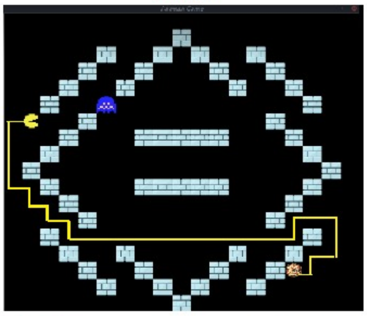
A*

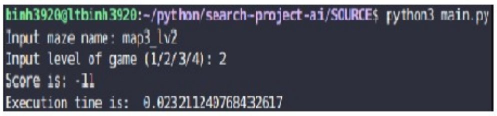




```
binh3920@ltbinh3920:~/python/search-project-ai/SOURCE$ python3 main.py
Input maze name: map3_lv2
Input level of game (1/2/3/4): 2
Score is: -11
Execution time is:  15.293967247009277
```

BFS





```
binh3920@ltbinh3920:~/python/search-project-ai/SOURCE$ python3 main.py
Input maze name: map3_lv2
Input level of game (1/2/3/4): 2
Score is: -11
Execution time is:  0.023211240768432617
```

Summary:

In runtime implement, A* is better than BFS. But in some special situations (level 2 map 2) A* takes long time to get the path than BFS.

2. Level 3

The idea:

- Pacman: In area 7x7, if the pacman see the food, move directly to it and eat. Based on heuristic (move follow the step which has the smallest heuristic between pacman position and monsters position. However, in some situations, the pacman will stop to wait for monster's move to get a better heuristic). If not, pacman moves randomly to the next step. This action repeat until there is no more food in maze or pacman touch the monster.
- Monster: when pacman moves, the monster moves randomly around the its position (up, down, left, right) until pacman eat all food in maze or catch pacman.

3. Level 4

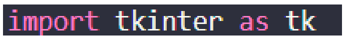
The idea:

- Pacman is the same as level 3.
- Monster: monster move directly to the pacman position by using heuristic between pacman position and its position until the pacman eat all food in maze or catch pacman. Each monster will wait for a different time before it scan for pacman again.

4. Graphical demonstration

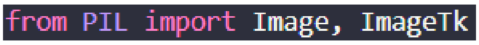
We use graphical library included:

+ Canvas of Tkinter



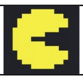
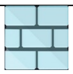
```
import tkinter as tk
```

+ Image of PIL

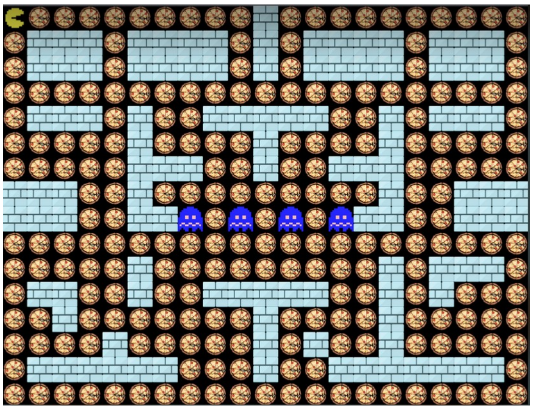


```
from PIL import Image, ImageTk
```

Images:

Food	Monster	Pacman	Block
			

Some example maps:



5. Report

We use Microsoft Word to write down what, how we have done the project by using text, image to illustrate.

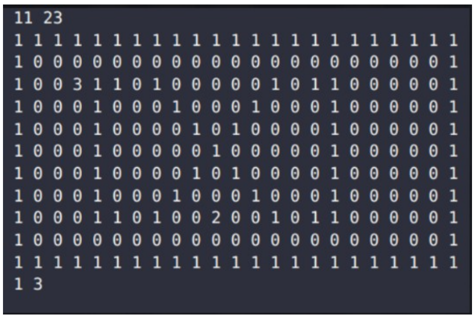
6. Data

We have 5 different maps for level 1 and level 2. These are quite the same where we put the food, block and pacman. The different is in level 2, the map has one monster.

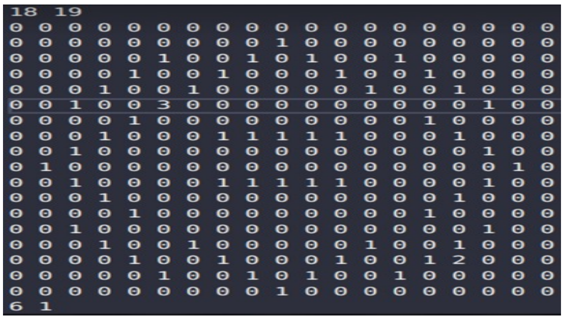
In level 3 and level 4, we have 5 completely different maps.

The first line is the size of maze, the last line is the start position of pacman, number 1 is block, number 2 is food and number 3 is monster

Some example maps:



```
11 23
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 3 1 1 0 1 0 0 0 0 0 1 0 1 1 0 0 0 0 0 1
1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1
1 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 1
1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1
1 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 1
1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1
1 0 0 0 1 1 0 1 0 0 2 0 0 1 0 1 1 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 3
```



```
18  19
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
0  0  0  0  0  1  0  0  1  0  1  0  0  1  0  0  0  0  0
0  0  0  0  1  0  0  1  0  0  0  1  0  0  1  0  0  0  0
0  0  0  1  0  0  1  0  0  0  0  0  1  0  0  1  0  0  0
0  0  1  0  0  3  0  0  0  0  0  0  0  0  0  0  1  0  0
0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  0
0  0  0  1  0  0  0  1  1  1  1  1  0  0  0  1  0  0  0
0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
0  0  1  0  0  0  0  1  1  1  1  1  0  0  0  0  1  0  0
0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  0
0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
0  0  0  1  0  0  1  0  0  0  0  0  1  0  0  1  0  0  0
0  0  0  0  1  0  0  1  0  0  0  1  0  0  1  2  0  0  0
0  0  0  0  0  1  0  0  1  0  1  0  0  1  0  0  0  0  0
0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
6  1
```

Instruction

To play the game, you should follow steps below.

1. Run the main.py file.
2. Input maze you want to play by entering the syntax *“map[1]_lv[2]”*
   - [1] is the maze number(from 1 to 5)
   - [2] is the level number (from 1 to 4)
3. Input the level you want to play by entering the number from 1 to 4
4. Choose the movement speed of pacman. The higher number, the slower pacman moves. For example: *250* means 250 milliseconds.
5. Enjoy the game.

During your time game, the score will caculate follow some rules:

1. Each step pacman moves, your score decrease 1 point.
2. When you get food, your socre increase 20 point.
3. You can see the score on the right side on the game screen.
4. Console will print real time execution. This is time for algorithms return path for pacman.