VIETNAM NATIONAL UNIVERSITY

HO CHI MINH UNIVERSITY OF SCIENCE

FACULTY INFORMATION TECHNOLOGY

PROJECT 01

SEARCHING

TEAM MEMBERS

Phạm Ngọc Thùy Trang	18127022
Võ Trần Quang Tuấn	18127248
Nguyễn Gia Hưng	18127044

Course: Introduction to Artificial Intelligence

Ho Chi Minh City – 2020

VIETNAM NATIONAL UNIVERSITY

HO CHI MINH UNIVERSITY OF SCIENCE

FACULTY INFORMATION TECHNOLOGY

PROJECT PACMAN GAME

| TOPIC |

| LECTURERS |

Mr. Le Ngoc Thanh

Ms. Ho Thi Thanh Tuyen

Ms. Nguyen Ngoc Thao

TABLE OF CONTENTS

REPORT'S TABLE OF CONTENTS

ACKNOWLEDGEMENTS

We would like to express our deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude we give to our major Introduction to AI's teacher, Ms. Nguyen Ngoc Thao, whose contribution in stimulating suggestions and encouragement, helped us to coordinate my project especially in writing this report.

Furthermore we would also like to acknowledge with much appreciation the crucial role of the staff of Mr. Le Ngoc Thanh and Ms. Ho Thi Thanh Tuyen, who gave the permission to use all required equipment and the necessary material to complete the project "Searching".

We have to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comment and advices.

Regards,

Team Representative,

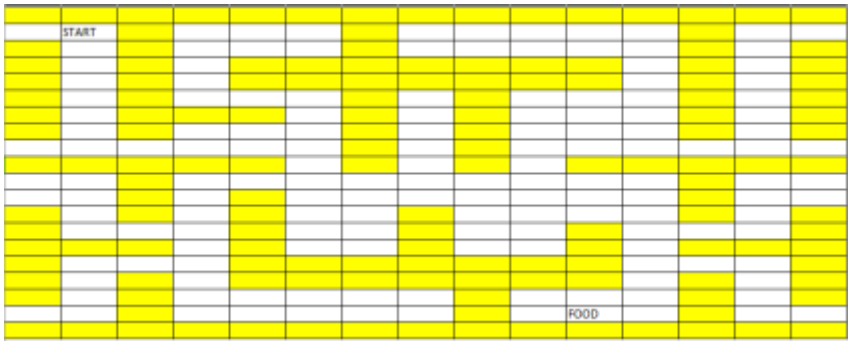
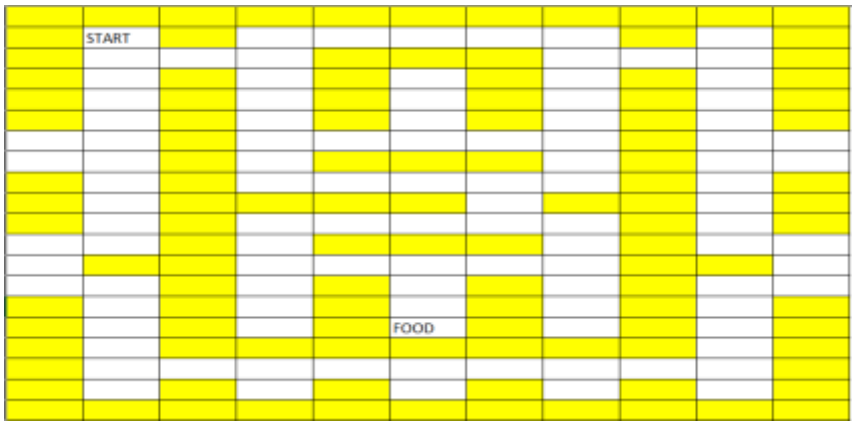
Trang

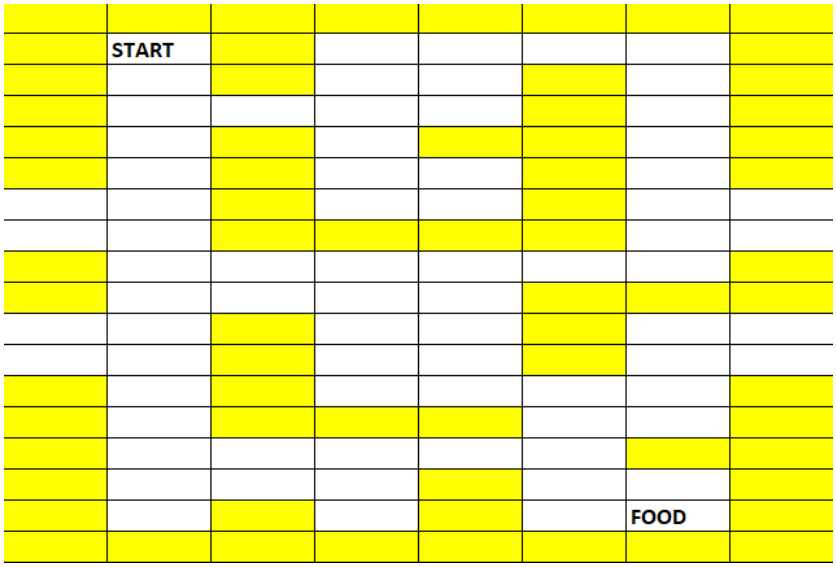
Pham Ngoc Thuy Trang

ASSIGNMENT PLAN

- PREPARATION

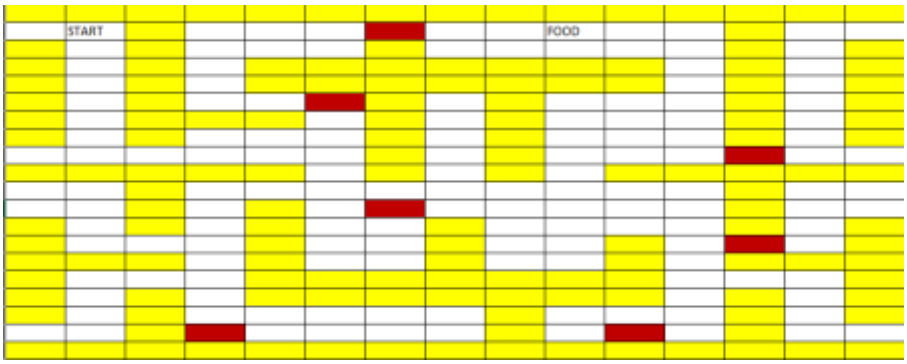
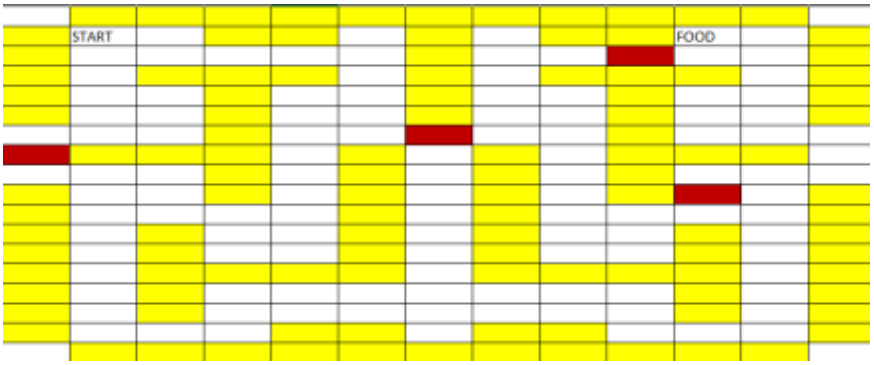
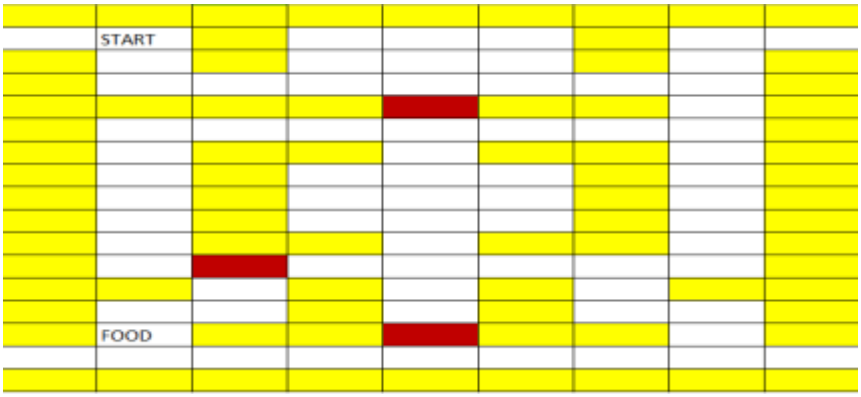
Pseudo Maze for Level 1

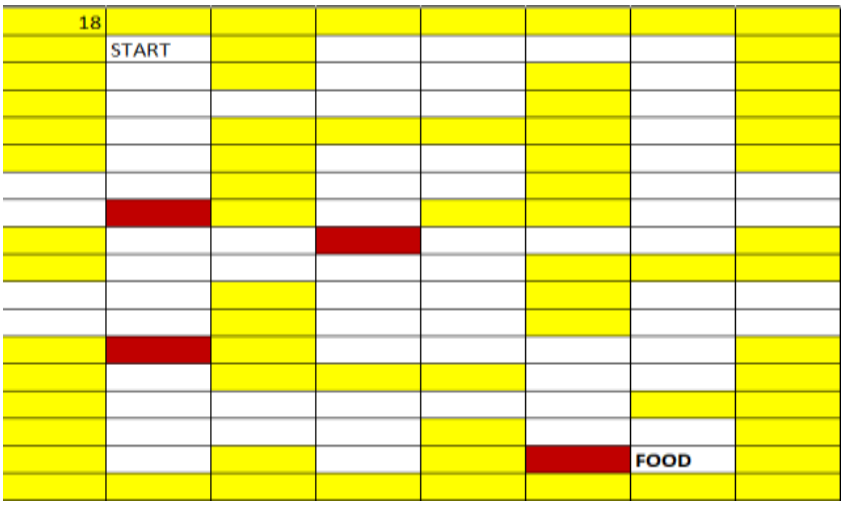
Maze	Size	Description (Yellow Cells are walls)
1	20x15	
2	18x13	
3	20x11	

4	17x9	
5	18x8	

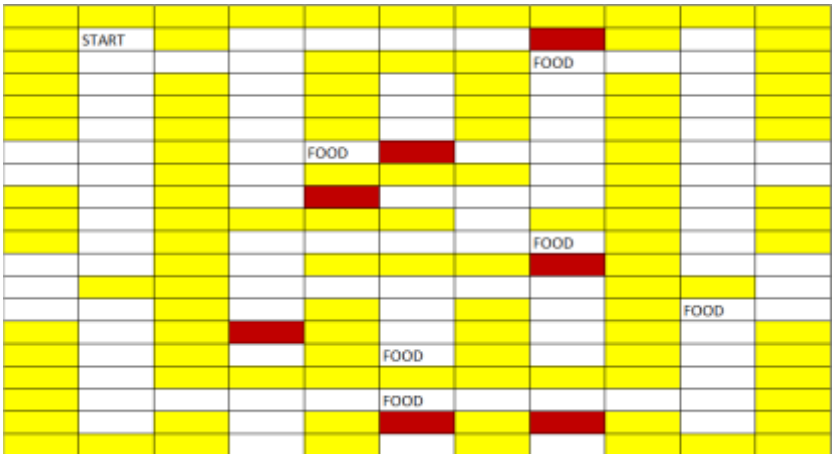
Pseudo Maze for Level 2

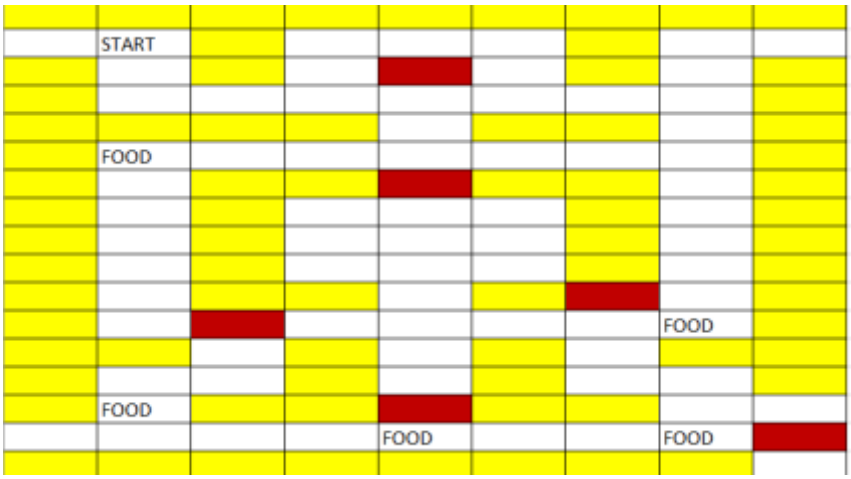
Maze	Size	Description (Yellow Cells are walls, Red Cells are monsters)

1	20x15	
2	18x13	
3	20x11	
4	17x9	

5	18x8	

Pseudo Maze for Level 3

Maze	Size	Description (Yellow Cells are walls, Red Cells are monsters)
1	20x11	

2	17x9	

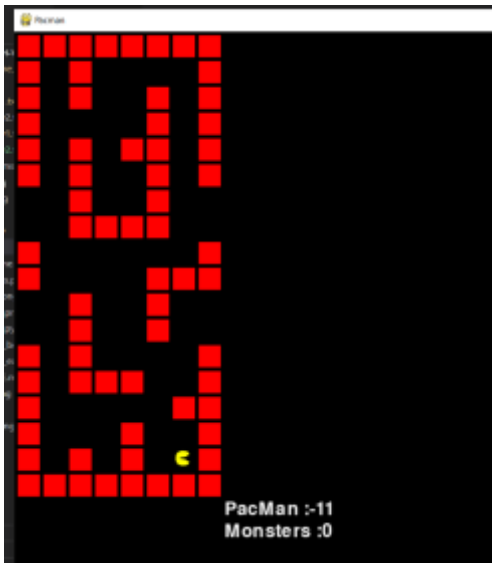
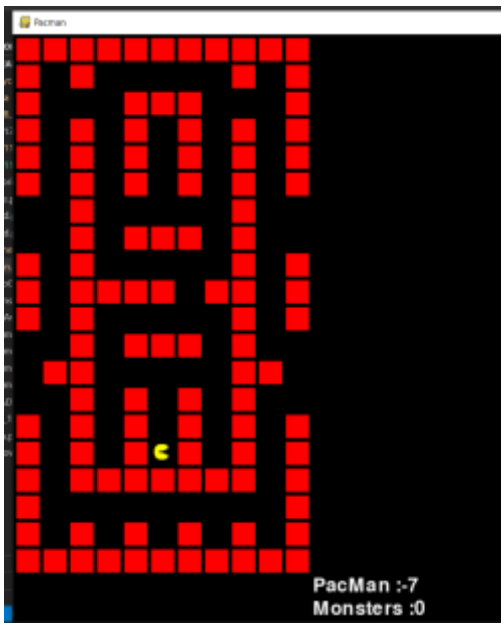
Basic Workflow

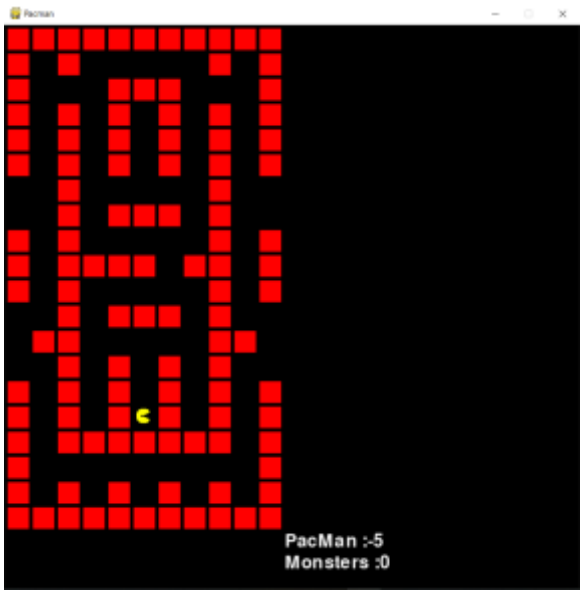
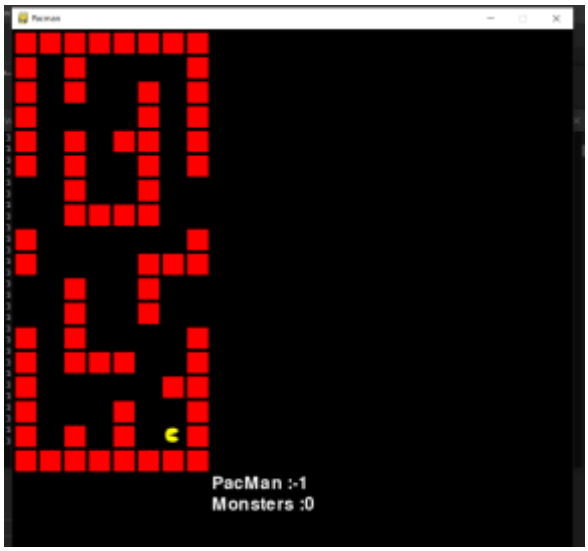
- PLANNING FOR EACH LEVEL

LEVEL 1

We will implement and compare between **2 search informed search algorithms** includes: GBFS (Greedy best first search), A* graph search to find which search is the best for Pacman at level 1 to find food with the minimum finished time and has the shortest path. ***Because in this level, there is only one food in the map and Pacman know where the food's position is***, we only need to implement two above algorithms to compare them. (We don't need to implement uninformed searches because the pacman know the position of food)

Algorithm	Pacman's score	Illustration

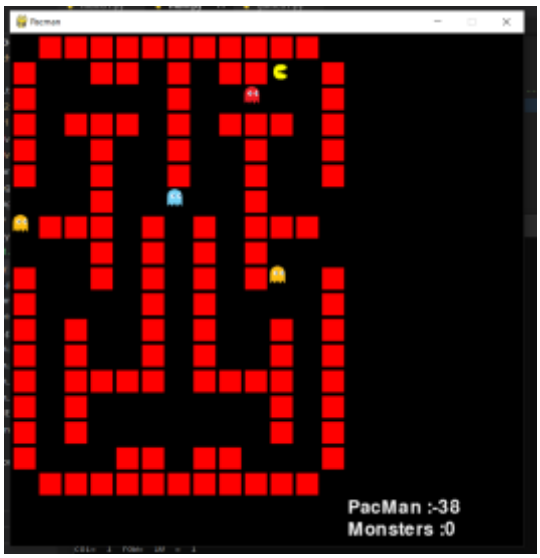
GBFS (Greedy Best First Search)	**-11** (map 18x8) **-7** (map 20x11)	 

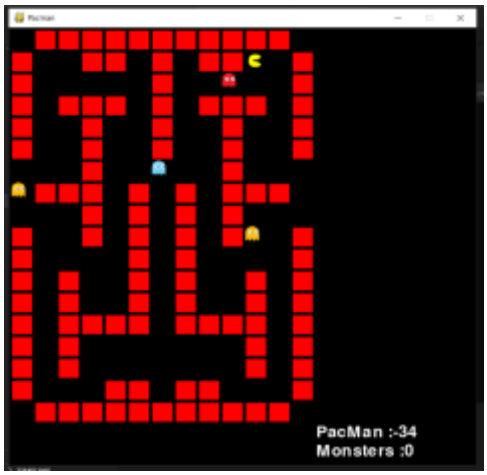
A*	**-5** (map 20x11) **-1** (map 18x8)	 
Overall	We can see with the same searching maps, ***A* search algorithm runs better than GBFS search***. Both of them are algorithms that can use both the knowledge acquired so far while exploring the search space, **denoted by g(n)**, and a heuristic function, **denoted by h(n)**, which estimates the distance to the goal node, for each **node n** in the search space (often represented as a graph)	

	But in these cases, ***GBFS is not complete and not optimal***, that is, the path found may not be the optimal one. An all nodes on the *border* (or fringe or frontier) are kept in memory, and nodes that have already been expanded do not need to be stored in memory and can therefore be discarded. Because GBFS doesn't use the past knowledge, so its connotation greedy On the other hands, because A* uses an admissible heuristic function which means A* is optimal, that is, it's always finds the optimal path between the starting node and the goal node. Compare with GBFS, ***A* is not only complete but also optimal.*** Though using more memory than Greedy BFS but if the searching space is not huge, A* will be often practical. With our maps, the searching spaces aren't huge so that A* is better than GBFS for pacman to find food. With map 20x11, pacman with A* search has score is **better** than pacman with GBFS search's score (-5 < - 7) With map 18x8, pacman with A* search has score is **better** than pacman with GBFS seach's score (-1 < -11)

LEVEL 2

We will implement and compare between **2 search informed search algorithms** includes: GBFS (Greedy Best First Search), A* graph search to find which search is the best for Pacman at level 2 to find food with the minimum finished time and has the shortest path. ***Because in this level, there is only one food in the map and we also have monsters in the place ever, however, monsters won't move around so we can see them like walls of the map and Pacman know where the food's position is***, we only need to implement 2 above algorithms and compare them. (We don't need to implement uninformed searches because we know the position of food)

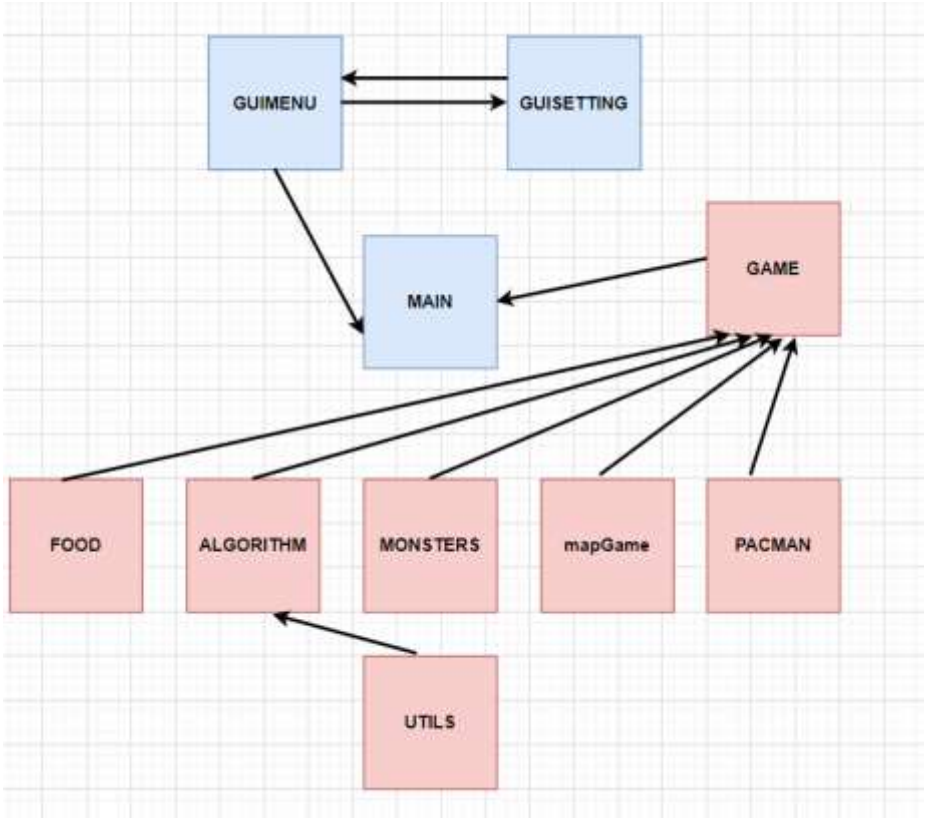
Algorithm	Pacman's score	Illustration
GBFS	**-38** (map 18x13) **-47**(map 20x11)	

		
A*	**-34** (map 18x13) **-45**(map 20x11)	 
Overall	We can see with the same searching maps, ***A* search algorithm runs better than GBFS search***. Both of	

	them are algorithms that can use both the knowledge acquired so far while exploring the search space, **denoted by g(n)**, and a heuristic function, **denoted by h(n)**, which estimates the distance to the goal node, for each **node n** in the search space (often represented as a graph) But in these cases, ***GBFS is not complete and not optimal***, that is, the path found may not be the optimal one. An all nodes on the *border* (or fringe or frontier) are kept in memory, and nodes that have already been expanded do not need to be stored in memory and can therefore be discarded. Because GBFS doesn't use the past knowledge, so its connotation greedy On the other hands, because A* uses an admissible heuristic function which means A* is optimal, that is, it's always finds the optimal path between the starting node and the goal node. Compare with GBFS, ***A* is not only complete but also optimal.*** Though using more memory than Greedy BFS but if the searching space is not huge, A* will be often practical. With our maps, the searching spaces aren't huge so that A* is better than GBFS for pacman to find food. With map 18x13, pacman with A* search has score is **better** than pacman with GBFS search's score (-34 < -38) With map 20x11, pacman with A* search has score is **better than** pacman with GBFS seach's score (-45 < -47)

- Planning UI and Workflow for game

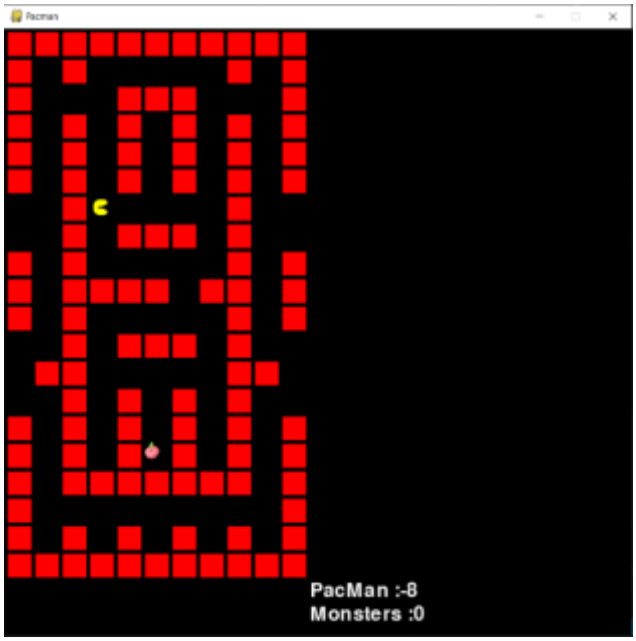
Workflow for game:



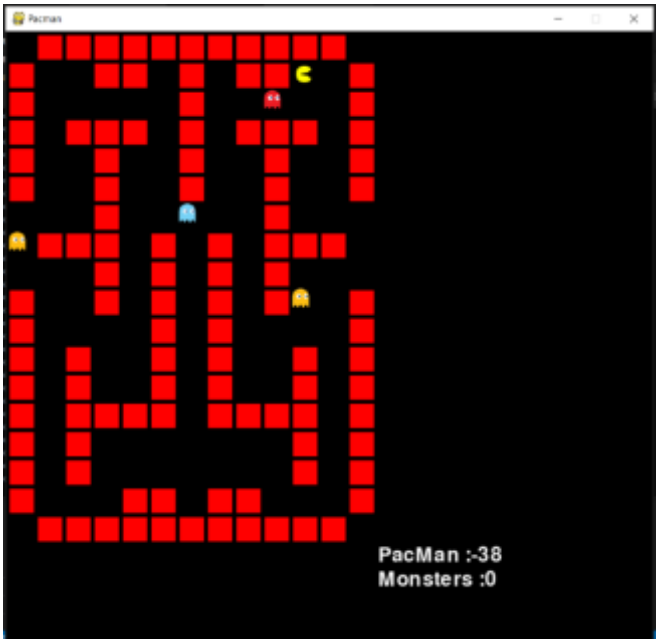
- Explain some important classes and function.
  - Node (mapGame.py): this class is the typical of a node in the graph, we use this to indicate a position in the map. At that position, we have monster or pacman, wall or empty path,…
  - Map (mapGame.py): this class indicates to the map of the game, it contains size of the map game, list of node of this map, list of food, monsters, empty path and node where pacman standing.
  - utils.py: contains some utilities data structure for implementing AI algorithms.

- Game(game.py): indicate the game of this project, it contains all important information of this game: pacman, map of game, monsters, food and some AI algorithms using for cheating game.
- Algorithms.py: implementing some basic AI algorithms such as BFS, GBFS, IDS, DFS, DFS limit search, A*
- guiMenu.py and guiSettings.py: using for show the graphical game with pygame package in python and pyQt5 in anaconda environment.
- Some function using for read file input and preprocessing that file.

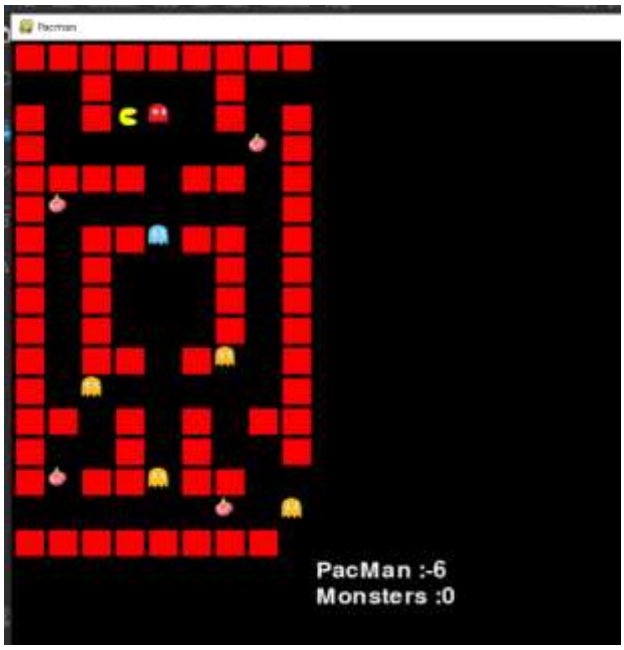
Map for LV1



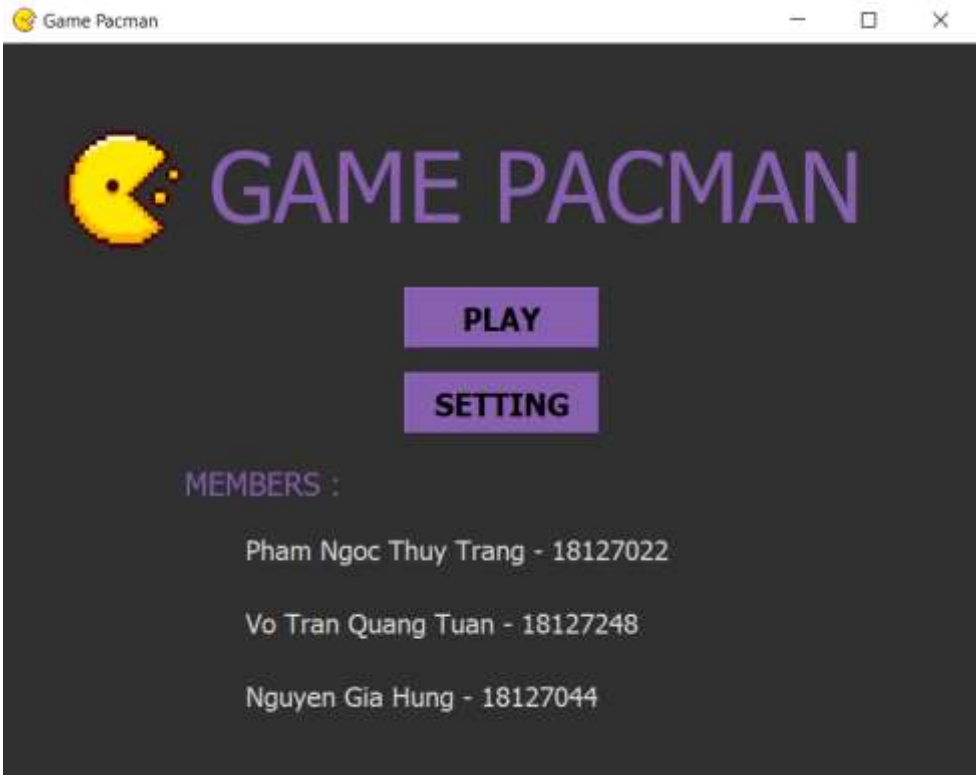
Map for LV2

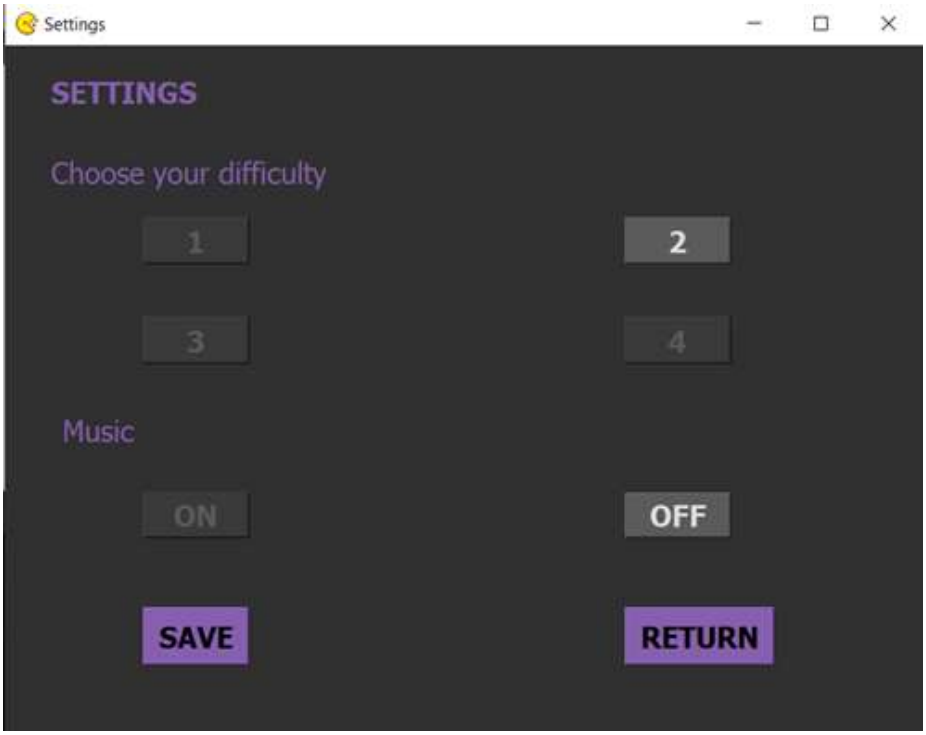


Map for lv3 (but monster cannot move)



- Graphics for menu game and for setting. Can choose one of these settings for music and level and then save to play game.


Game Pacman
GAME PACMAN
PLAY
SETTING
MEMBERS :
Pham Ngoc Thuy Trang - 18127022
Vo Tran Quang Tuan - 18127248
Nguyen Gia Hung - 18127044


Settings
SETTINGS
Choose your difficulty
1
2
3
4
Music
ON
OFF
SAVE
RETURN

ENVIRONMENT

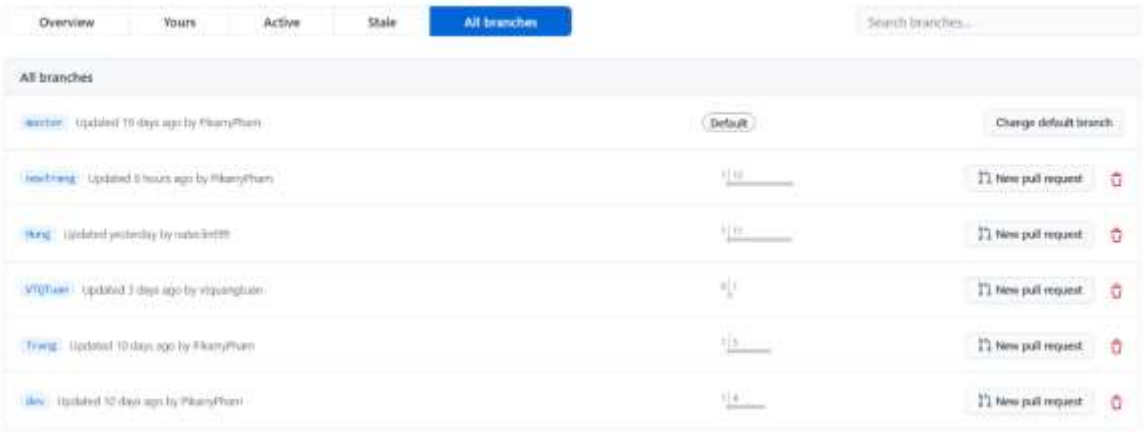
What we use for our project are:

- **IDEs**: Visual Code, Vim
- Programming language: Python
- Outsource Platform for Python: Anaconda
- **Libraries for Project**: PyQt5, pygame (for graphics), heapq (to use heap data structure in python), sys, re (Regular Expression - restrict the way how user input information), os
- **Version Control System:** Git (using GitHub to store projects and teamwork).

➢ The primary branch is branch “master” and parallel to this branch is another branch called “dev”.

➢ When the team’s source code in the “dev” branch reaches a stable point and is ready to be released, all of changes will be merged back into “master” branch.

➢ Members will also have their own branch which is named after their name. These branches are used for edit/delete/update their functions….



ESTIMATE THE DEGREE OF COMPLETION LEVEL

ESTIMATE THE DEGREE OF COMPLETION FOR EACH LEVEL

Level	What have done	What have not done yet	Issues
1	Implement algorithms and find path for pacman	No	No
	Design graphics for lv1 with food, pacman and map	No	No

2	Implement algorithms and find path for pacman	No	No
	Design graphics for lv1 with food, pacman, monsters and map	No	No
3	No	Design graphics for lv3 with food, pacman, monsters and map, apply the AI algorithm for pacman and moving for monster	
4	No	Design graphics for lv4 with food, pacman, monsters and map, apply the AI algorithm for pacman and monster	

OVERALL ESTIMATE

Level/Tasks	Is Complete/Is not Complete?	Rating
1	15	/15
2	15	/15
3	0	/10
4	0	/10
At least 5 maps with different in number and structure of walls, monsters, food	10	/10
UI for Game	10	/10
Report	30	/30
Overall	80	/100

REFERENCES

[1]: Artificial Intelligence – A Modern Approach 3rd Edition Russel and Peter Norvig

[2]: https://ai.stackexchange.com/questions/8902/what-are-the-differences-between-a-and-greedy-best-first-search

[3]: http://ashishgupta.me/articles/2019-05-02-AI-Search-Algorithms-Implementations/