Full Name: Bùi Văn Minh Triều

Student's ID: ITCSIU21241

1. Describe the behavior of RandomAgent from Step 7

- In the tinyMaze environment, Packman do not get to food straightforwardly. Because the code use the Random method, Packman move randomly and take much time to get food. Packman sometimes stop in random positions because of the Directions.Stop.

```
PS C:\Users\Admin\OneDrive\Máy tính\Programing\AI\search\search> python pacman.py --layout tinyMaze --pacman RandomAgent
Pacman emerges victorious! Score: -86
Average Score: -86.0
Scores:        -86.0
Win Rate:      1/1 (1.00)
Record:        Win
PS C:\Users\Admin\OneDrive\Máy tính\Programing\AI\search\search> python pacman.py --layout tinyMaze --pacman RandomAgent
Pacman emerges victorious! Score: 414
Average Score: 414.0
Scores:        414.0
Win Rate:      1/1 (1.00)
Record:        Win
PS C:\Users\Admin\OneDrive\Máy tính\Programing\AI\search\search> python pacman.py --layout tinyMaze --pacman RandomAgent
Pacman emerges victorious! Score: 79
Average Score: 79.0
Scores:        79.0
Win Rate:      1/1 (1.00)
Record:        Win
```

Figure 1: Results of RandomAgent in tinyMaze

➔ In three times playing Packman, the ratio of getting food with faster time is 2/3 cases.

- In the medium environment, Packman also move randomly. However, in this stage, Packman spend longer time compared with tinyMaze to get food.

```
PS C:\Users\Admin\OneDrive\Máy tính\Programing\AI\search\search> python pacman.py --layout mediumMaze --pacman RandomAgent
Pacman emerges victorious! Score: -5899
Average Score: -5899.0
Scores:        -5899.0
Win Rate:      1/1 (1.00)
Record:        Win
PS C:\Users\Admin\OneDrive\Máy tính\Programing\AI\search\search>
```

➔ The average score is -5899.0, it means packman move more than the score of food

2. A screen shot of your myLayout environment from Step 8

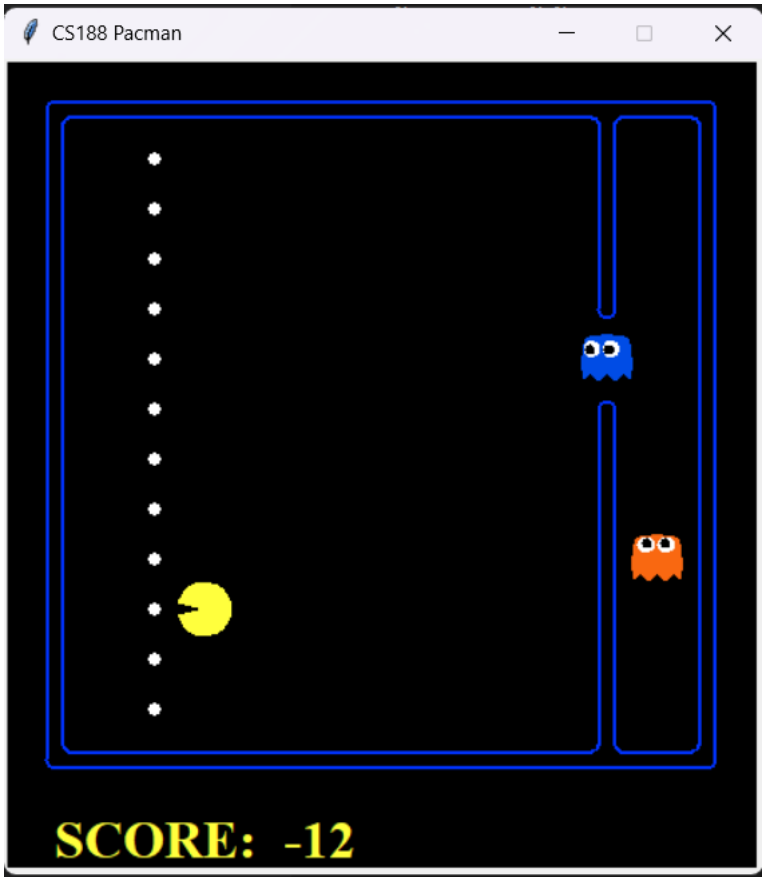

Figure 2: myLayout screenshot

3. Describe the behavior of RandomAgent from Step 9

- In the openSearch, the rate of eating food of Packman is faster than the previous version. However, Packman still take a few time to get all food in this layout.

```
PS C:\Users\Admin\OneDrive\Máy tính\Programing\AI\search\search> python pacman.py --layout openSearch --pacman BetterRandomAgent
Pacman emerges victorious! Score: -616
Average Score: -616.0
Scores:        -616.0
Win Rate:      1/1 (1.00)
Record:        Win
```

Figure 3: openSearch layout of BetterRandomAgent

- In the myLayout, the rate of eating food of Packman is also faster. However, there are an appearance of 2 ghost, Packman loses game.

```
PS C:\Users\Admin\OneDrive\Máy tính\Programing\AI\search\search> python pacman.py --layout myLayout --pacman BetterRandomAgent
Pacman died! Score: -630
Average Score: -630.0
Scores:        -630.0
Win Rate:      0/1 (0.00)
Record:        Loss
```

Figure 4: myLayout layout of BetterRandomAgent

4. Describe the behavior of ReflexAgent from Step 10

- In the myLayout, Packman tends to eat more food, but in the area that does not have food, Packman moves randomly. That is the reason why Packman meets 2 ghost, then lose the game.

```
PS C:\Users\Admin\OneDrive\Máy tính\Programing\AI\search\search> python pacman.py --layout myLayout --pacman ReflexAgent
Pacman died! Score: -521
Average Score: -521.0
Scores:        -521.0
Win Rate:      0/1 (0.00)
Record:        Loss
```

Figure 5: myLayout layout of ReflexAgent

- In the openSearch, Packman eat food faster and a lot with many combo food. However, in the area that Packman already eats which is empty, Packman moves like getting stuck in there. After that, when Packman eat any food, it is able to escape. Packman still takes much time to complete game.

```
PS C:\Users\Admin\OneDrive\Máy tính\Programing\AI\search\search> python pacman.py --layout openSearch --pacman ReflexAgent
Pacman emerges victorious! Score: -644
Average Score: -644.0
Scores:        -644.0
Win Rate:      1/1 (1.00)
Record:        Win
```

5. For each of the percepts listed in Step 10, show what command/code enables you to access it.

- Pac-man's postion: gameState.getPacmanPosition()
- The position of all the ghosts: gameState.getGhostPositions()
- The location of the walls: gameState.getWall()
- The position of the capsules: gameState.getCapsules()
- The position of each food pellet: gameState.getFood()
- The total number of food pellets still available: gameState.getNumFood()
- Whether is has won or lost the game: gameState.isLost() and gameState.isWin()
- Packman's current score in the game: gameState.getScore()