Name: Tạ Thị Thuỳ Dương ID: ITCSIU21053

Artificial Intelligence Lab 1

What to submit:

Your submission should include the following:

1. Lab report answers to the following questions:
   a. Describe the behavior of **RandomAgent** from Step 7
   b. A screen shot of your **myLayout** environment from Step 8
   c. Describe the behavior of **RandomAgent** from Step 9
   d. Describe the behavior of **ReflexAgent** from Step 10
   e. For each of the percepts listed in Step 10, show what command/code enables you to access it. For example:
      Pac-man's postion: **gameState.getPacmanPosition()**
2. Source code + README (how to compile and run your code)
3. Please create a folder called "yourname_studentID" that includes all the required files and generate a zip file called "yourname_studentID.zip".
4. Please submit your work (.zip) to Blackboard.

1. Lab report answers to the following questions:

a. Describe the behavior of RandomAgent from Step 7

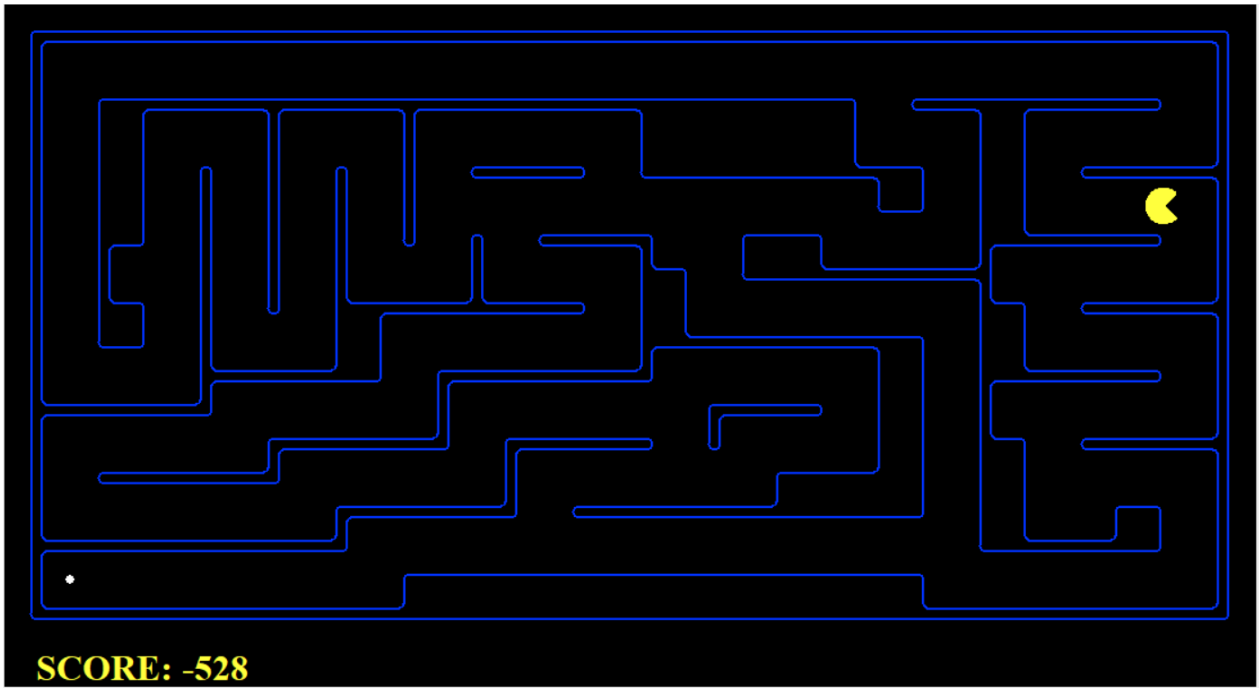

Figure 1: Step 7 result

Description:

Pacman only moves randomly around a small area on the right side of the screen. This could be because the random Agents results are evenly distributed, causing Pacman to not have consecutive similar results to go far. Another reason is that the random time interval is too short, causing Packman to not go far to the left side of the maze. It gets to the food randomly, it can win the game if lucky, which is clearer in the tinyMaze as figure3.

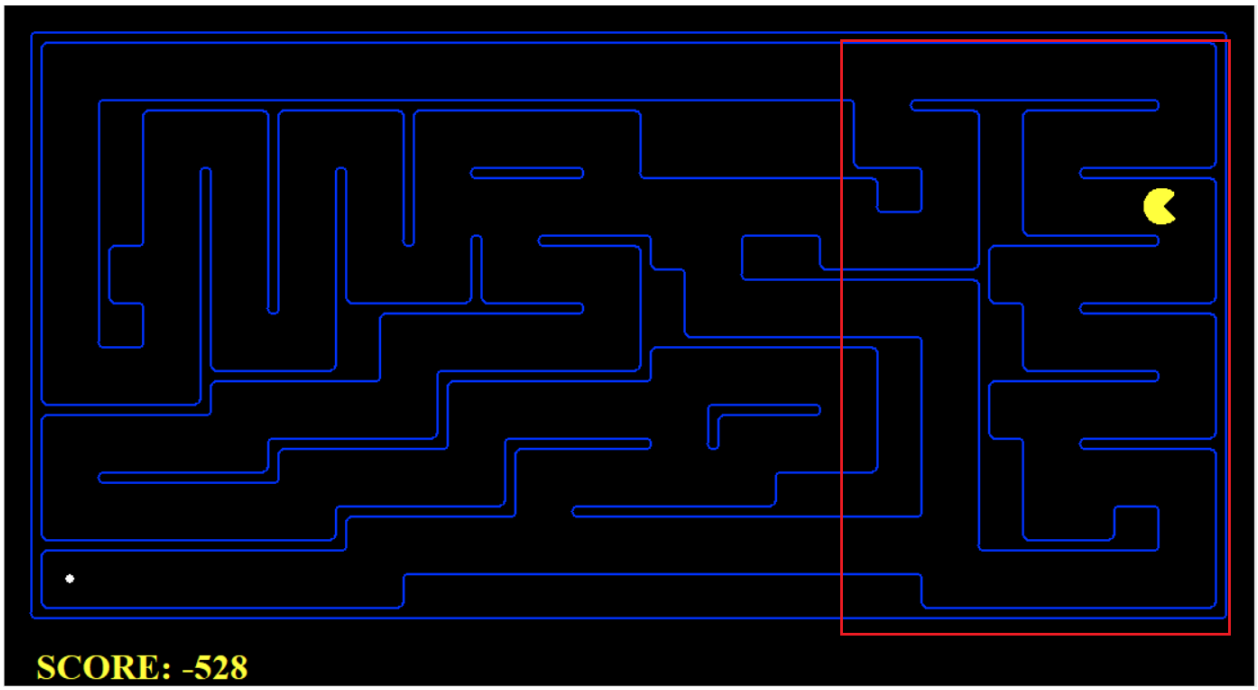

Figure 2: The area where pacman moves during the first few minutes of running the program

Figure 3: The agent run in the tinyMaze environment

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

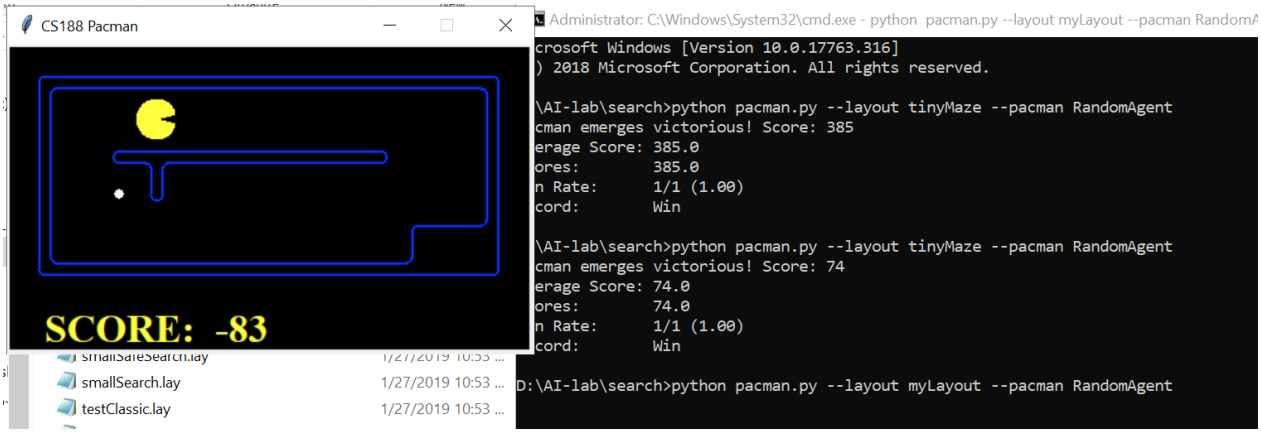

Figure 4: The agent run in the myLayout environment

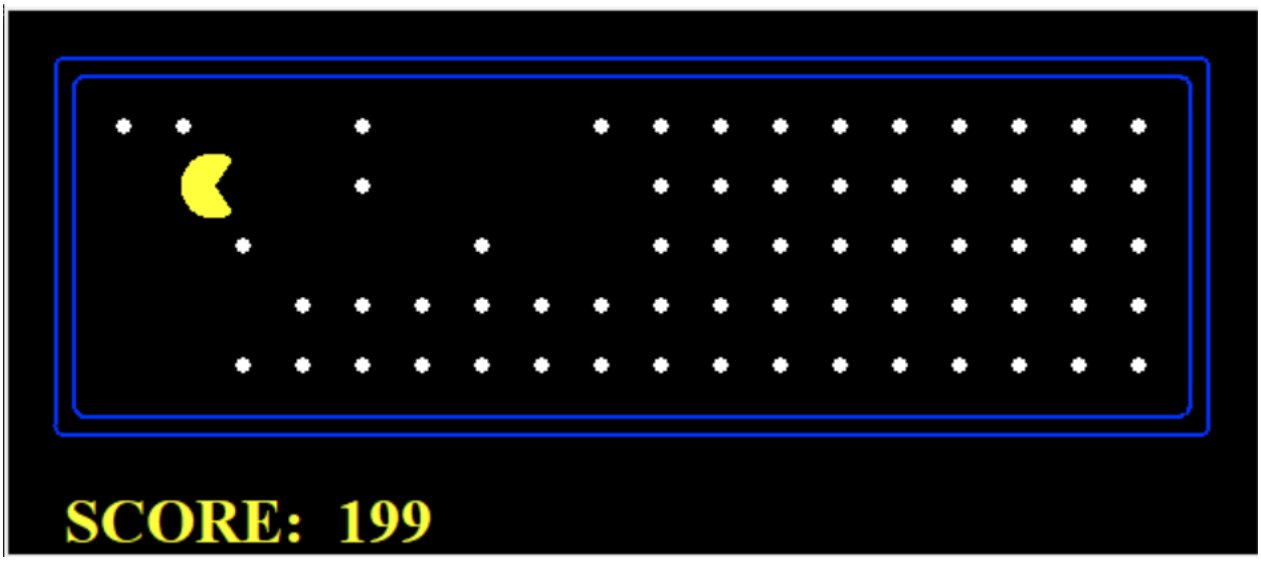

Figure 5: The agent run in the openSearch environment

Description:

Pacman moves and grabs food randomly. It moves through areas (where it has already visited and eaten food) multiple times. Sometimes it leaves one or a few dots of food and wanders around before coming back and grabbing them all.

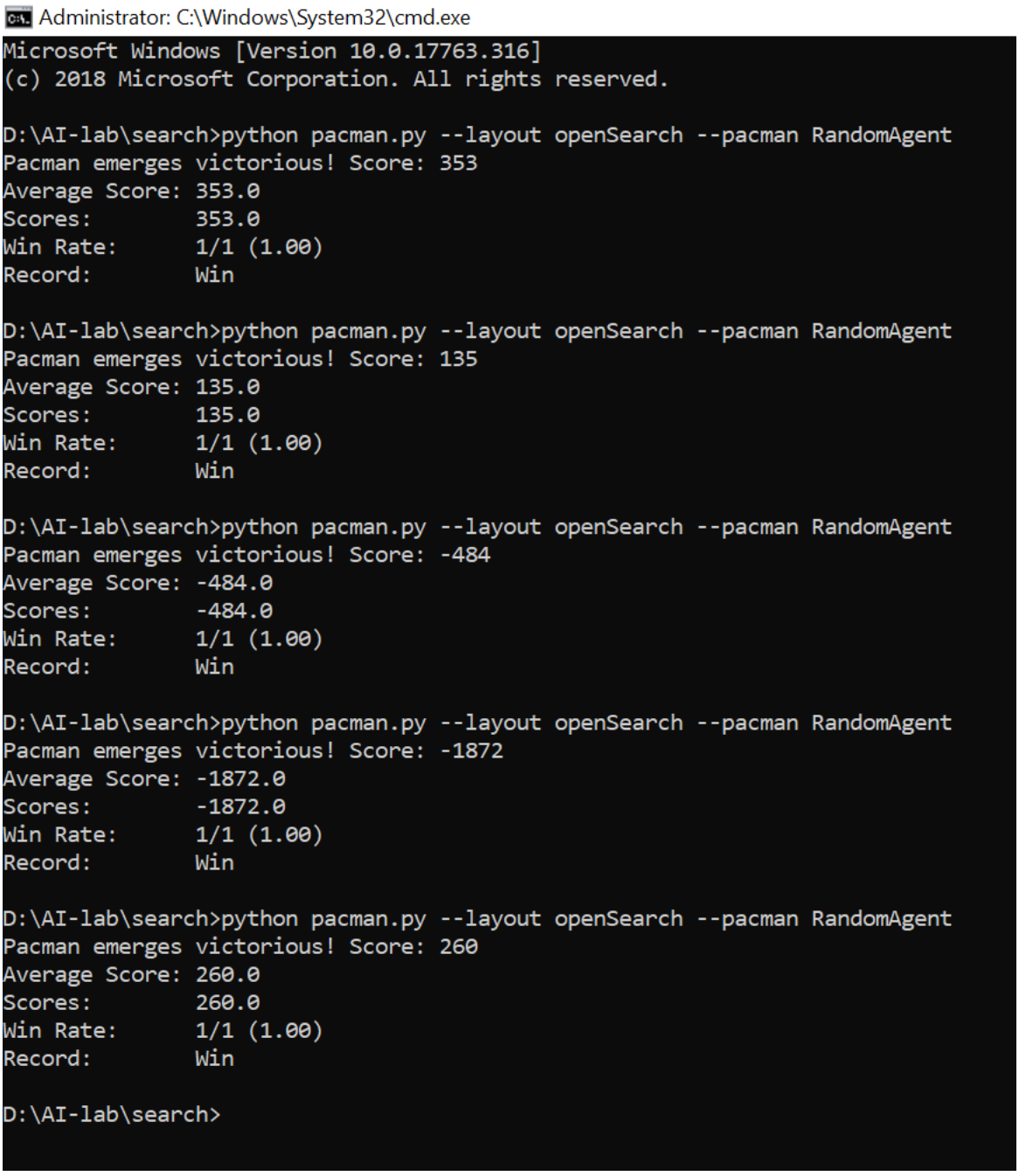

Figure 6: The results of running agent in the openSearch environment

Average score: (353.0 + 135.0 – 484.0 – 1872.0 + 260.0)/5 = -321.6

C. Describe the behavior of RandomAgent from Step 9

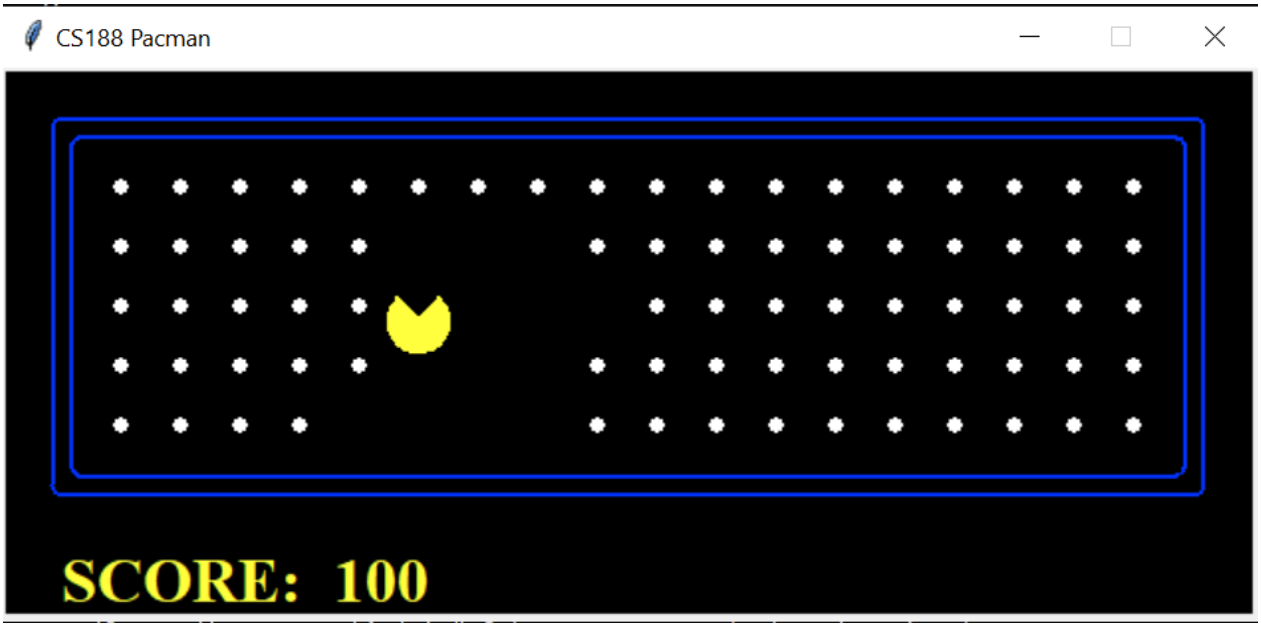

Figure 7: BetterRandomAgent helps pacman collect foods faster than the RandomAgent because of reducing stop

d. Describe the behavior of ReflexAgent from Step 10

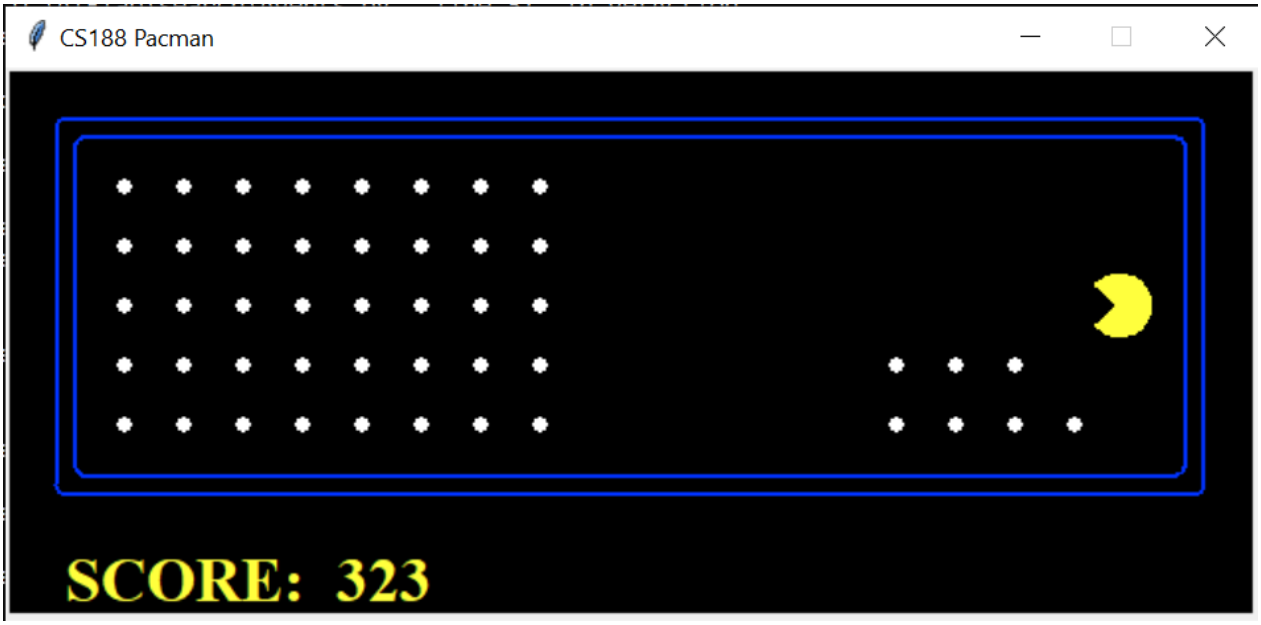

Figure 8: ReflexAgent *run in the openSearch environment*

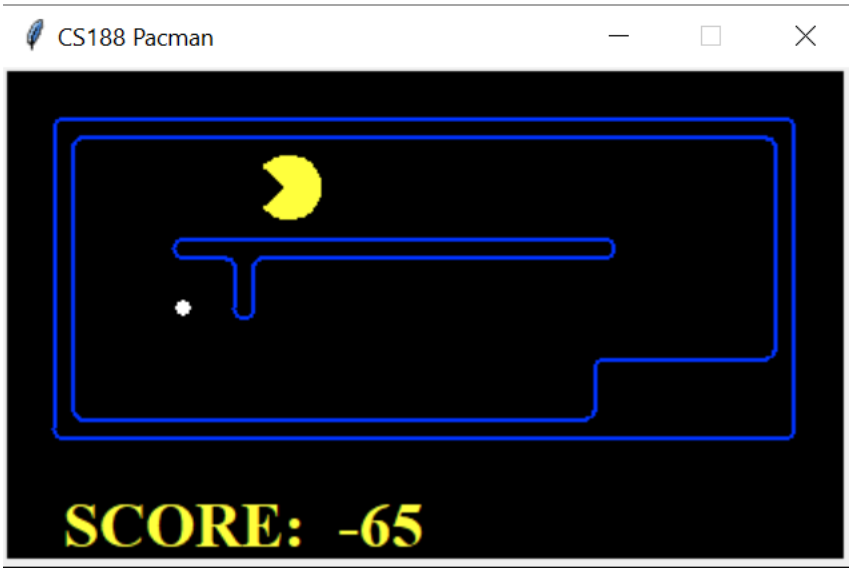

Figure 9: ReflexAgent *run in the myLayout environment*

e. For each of the percepts listed in Step 10, show what command/code enables you to access it. For example: Pac-man's postion: gameState.getPacmanPosition()

Pac-Man can perceive:

• His position: gameState.getPacmanPosition()

• The position of all the ghosts: : gameState.getGhostPositions()

• The locations of the walls: gameState.getWalls()

• The positions of the capsules: gameState.getCapsules()

• The positions of each food pellet: gameState.getFood()

• The total number of food pellets still available: gameState.getNumFood()

• Whether it has won or lost the game: gameState.isWin() and gameState.isLose() ,respectively

• His current score in the game: gameState.getScore()