**Lab 1: Designing Pac-Man Agents**

Name: Nguyễn Đình Khánh Ngân

ID: ITCSIU22236

1. **Describe the behavior of RandomAgent from Step 7**

Class RandomAgent

A screen shot of a computer program

AI-generated content may be incorrect.

Terminal run

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

The output behavior shows that the RandomAgent behaves completely unpredictably, simply picks one of the legal moves at random at each step.

* Random Choice of Actions: At every position, the agent randomly selects from the set of legal actions. There is no strategy involved in the agent behavior.
* Oscillatory Movement: The log shows the agent often returns to previous positions. This kind of back-and-forth movement is common with random decisions.
* Frequent Stops: The agent sometimes chooses the “Stop” action, meaning it doesn’t move even when other options are available. This contributes to periods where it appears to be “idle” or stuck in place.
* No Directed Progress: Without any evaluation of where the food is, the agent might eventually move in the right direction by chance. However, the overall behavior is simply a random walk with no guarantee that it will ever reach the food or avoid unnecessary loops.

1. **A screen shot of your myLayout environment from Step 8**

myLayout.lay

A screenshot of a computer screen

AI-generated content may be incorrect.

Terminal run

A screen shot of a video game

AI-generated content may be incorrect.

1. **Describe the behavior of RandomAgent from Step 9**

Class BetterRandomAgent

A computer screen shot of a program code

AI-generated content may be incorrect.

Terminal run

openSearch layout

A screen shot of a game

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

myLayout

A screen shot of a video game

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

The behavior shows that the BetterRandomAgent consistently avoids the 'Stop' action when other moves are available. Instead, it chooses one of the four directional moves at random.

* Non-Stop Movement: At every step, the agent selects a move from the available directions, never opting to "Stop" if there's an alternative. This ensures that the agent is always in motion.
* Random Choice: The agent’s actions are chosen randomly from the set of valid moves.
* Oscillatory Behavior: The log indicates that after moving to a new position, the agent sometimes returns to its previous location. This suggests that the agent does not have a directional preference or long-term strategy, it simply picks from the available options.
* Continuous Movement: The agent alternates between directions showing that it is actively exploring the maze without pausing. Although this random wandering may eventually lead to discovering food, there is no deliberate progression toward any specific goal.

Overall, the BetterRandomAgent’s behavior is characterized by constant, random movement that avoids staying still, resulting in a path that can sometimes loop back on itself without clear progress.

1. **Describe the behavior of ReflexAgent from Step 10**

Class ReflexAgent

A screen shot of a computer program

AI-generated content may be incorrect.

Terminal run

openSearch layout

A screen shot of a game

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

myLayout layout

A screen shot of a video game

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

The behavior shows that the ReflexAgent continuously scans its immediate surroundings for food and then acts accordingly:

* Immediate Food Detection: When one of the available moves would place Pac-Man directly on a cell with a food pellet, the agent immediately selects that action. For example, messages like "Food detected. Taking the next action: North" indicate that the cell to the north contained food, so the agent moved in that direction.
* Fallback to Random Movement: In situations where no immediate adjacent cell has food, the agent chooses a random move (ignoring the 'Stop' action). This random choice helps the agent explore when no food is immediately accessible.
* Overall Behavior: The agent mostly follows a reactive strategy, it makes decisions based solely on the current, local environment. This causes Pac-Man to often change direction depending on where food is available, leading to a mix of deliberate moves and random moves (when cannot detected the food). Ultimately, this reflex-based strategy allows Pac-Man to successfully clear the level, as indicated by the final win message and score.

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

* Pac-Man’s position:

A screen shot of a computer program

AI-generated content may be incorrect.

* Positions of All the Ghosts:

A screen shot of a computer program

AI-generated content may be incorrect.

* The Locations of the Walls:

A screenshot of a computer screen

AI-generated content may be incorrect.

* The Positions of the Capsules: A black screen with white text

  AI-generated content may be incorrect.
* The Positions of Each Food Pellet: A screenshot of a computer program

  AI-generated content may be incorrect.
* The Total Number of Food Pellets Still Available:

A black background with white text

AI-generated content may be incorrect.A black background with white text

AI-generated content may be incorrect.

* Whether It Has Won or Lost the Game:

A screenshot of a computer program

AI-generated content may be incorrect.

* His Current Score in the Game:

state.getScore()

A black background with text

AI-generated content may be incorrect.