

Họ và tên: Nguyễn Văn Thông

MSSV: 1760197

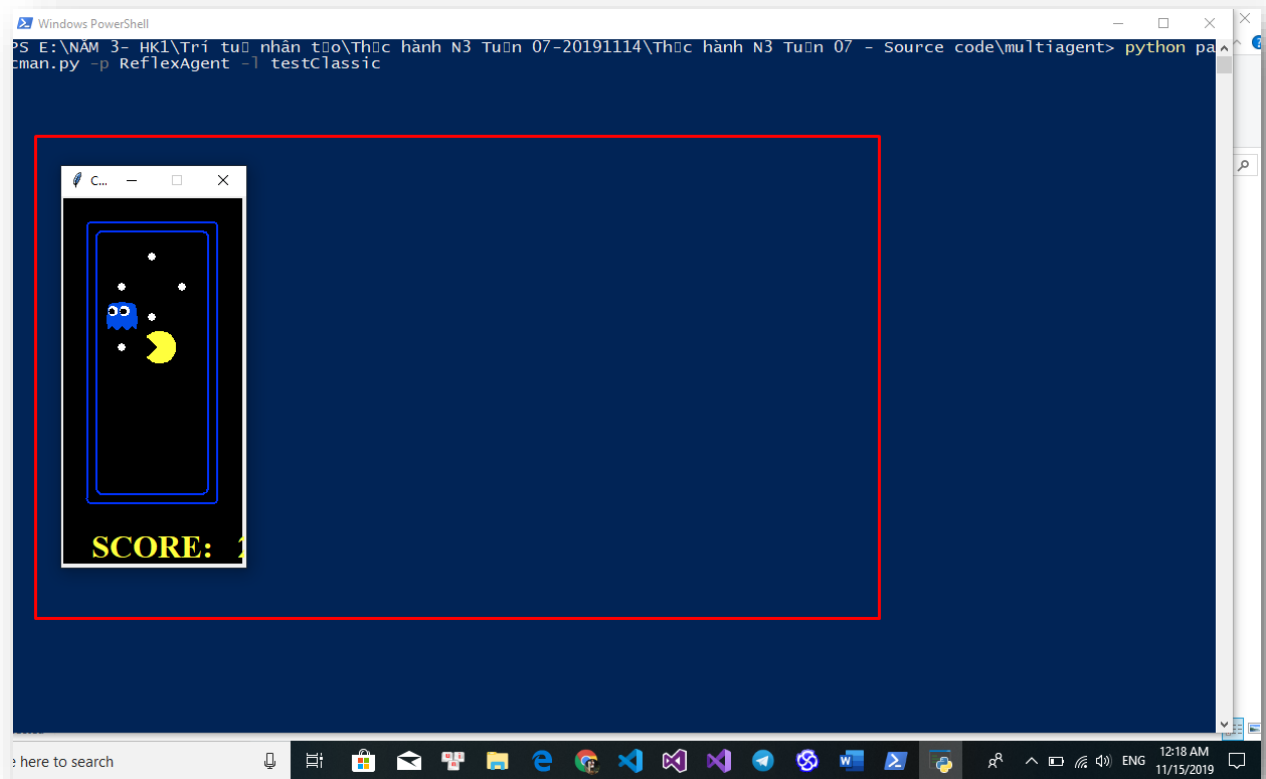
Lớp: 17ck2

Báo cáo tìm kiếm đối kháng

1. Reflex Agent

Xin chào thầy và các bạn, hôm nay mình có làm về thuật toán tìm kiếm đối kháng về phần Reflex Agent .

Đầu tiên, mình sẽ test game khi chưa cài hàm **evaluationFunction()** qua hình ảnh minh hoạ sau:



Qua hình ảnh này cho thấy, Pacman khá là kém trong việc di chuyển và có thể dễ dàng bị ma tiêu diệt.

----Bây giờ, mình sẽ cài thêm hàm **evaluationFunction()** xem thử ra sao nhe!

```
def evaluationFunction(self, currentGameState, action):
    """
    Hàm evaluationFunction() nhận đầu vào là trạng thái hiện tại của game và một hành động mà Pacman
    có thể thực hiện từ trạng thái này. Kết quả trả về sẽ là điểm số ước lượng của hành động này,
    điểm số càng cao thì hành động càng được ưu tiên.
    Đoạn code có sẵn trong hàm evaluationFunction() cung cấp sẵn cho chúng ta cách trích xuất một số
    thông tin hữu ích từ trạng thái của game
    """

    # Lấy trạng thái mới của game sau khi Pacman thực hiện action.
    successorGameState = currentGameState.generatePacmanSuccessor(action)
    # Lấy vị trí của Pacman sau khi Pacman thực hiện action.
    newPos = successorGameState.getPacmanPosition()
    # Cập nhật phần thức ăn còn lại sau khi Pacman thực hiện action.
    newFood = successorGameState.getFood()
    # Cập nhật trạng thái mới của ma sau khi Pacman thực hiện action.
    newGhostStates = successorGameState.getGhostStates()
    newScaredTimes = [
        ghostState.scaredTimer for ghostState in newGhostStates]
```

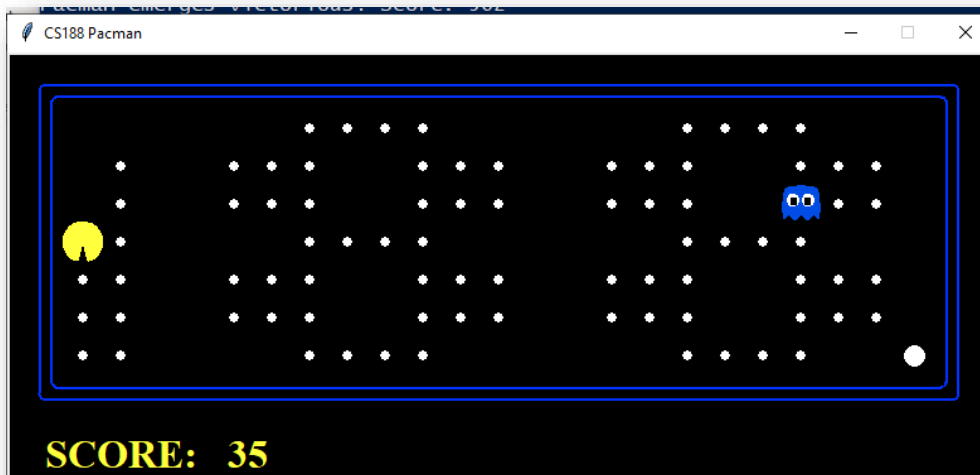
Code demo ở dưới đây:

```

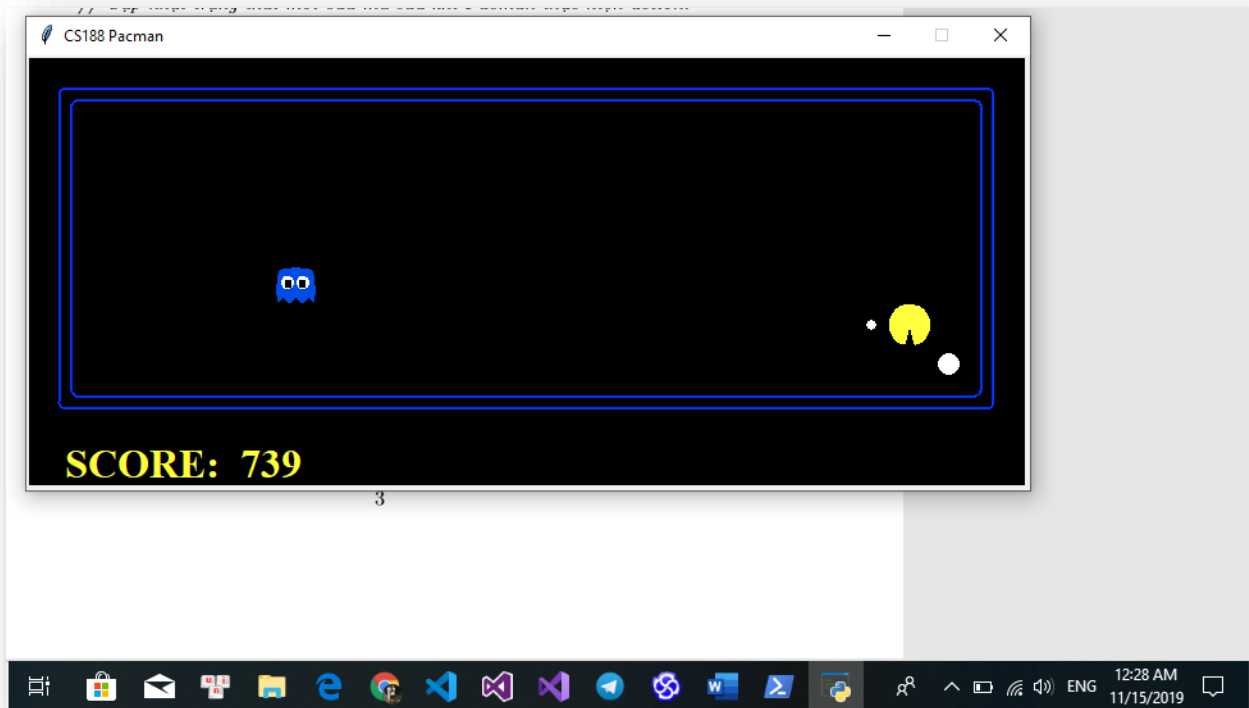
78     """ YOUR CODE HERE """
79     """tính khoảng cách đến thức ăn xa nhất"""
80     newFoodList = newFood.asList()
81     min_food_distance = -1
82     for food in newFoodList:
83         distance = util.manhattanDistance(newPos, food)
84         if min_food_distance >= distance or min_food_distance == -1:
85             min_food_distance = distance
86
87     """Tính khoảng cách từ pacman đến hồn ma.
88     Ngoài ra, kiểm tra sự gần gũi của những con ma (ở khoảng cách 1) xung quanh pacman."""
89     distances_to_ghosts = 1
90     proximity_to_ghosts = 0
91     for ghost_state in successorGameState.getGhostPositions():
92         distance = util.manhattanDistance(newPos, ghost_state)
93         distances_to_ghosts += distance
94         if distance <= 1:
95             proximity_to_ghosts += 1
96
97     """Tổng hợp số liệu vừa tìm dc sẽ trả về"""
98     return successorGameState.getScore() + (1 / float(min_food_distance)) - (1 / float(distances_to_ghosts))
99
100

```

Sau khi cài đặt xong, bạn có thể chạy lệnh: **python autograder.py -q q1**

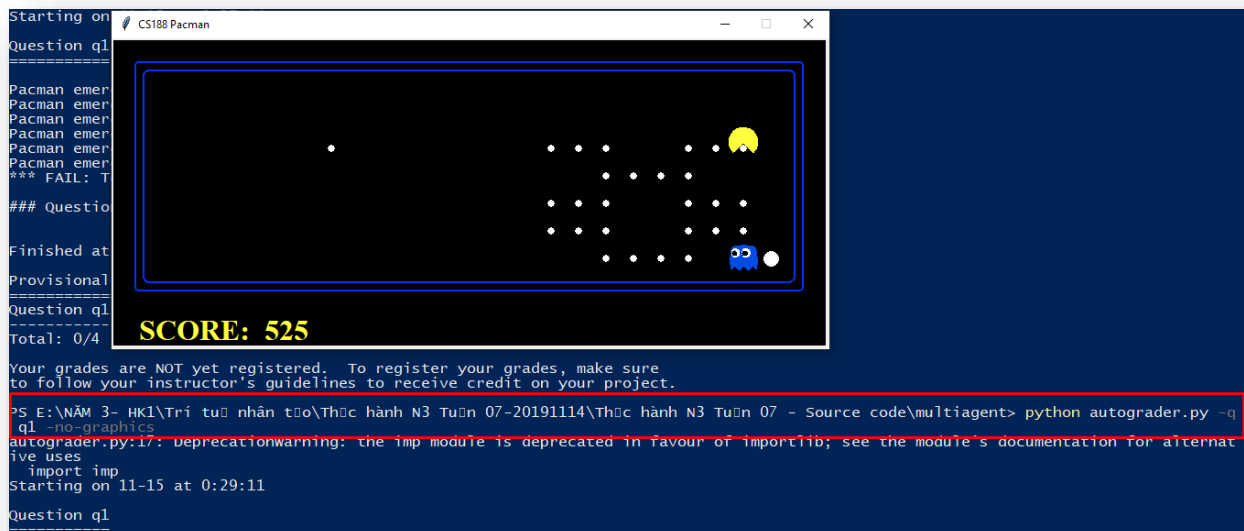


Tuần 07 - Source code of importlib; see the



Bạn có thể sử dụng câu lệnh **python autograder.py -q q1 --no-graphics** để thực hiện đánh giá điểm một cách nhanh chóng hơn. Câu lệnh này sẽ không hiển thị lên màn hình quá trình chơi game của Pacman.

Kết quả qua hình ảnh:



2. Minimax

Trong phần này, mình sẽ viết code vào hàm **getAction()**, trong **class MinimaxAgent**. Thuật toán minimax thiết kế nên đảm bảo chạy được với số lượng ma là bất kì. Lượt chơi của Pacman và ma luân phiên nhau như sau: Pacman đi, rồi lần lượt từng ma đi, rồi Pacman đi, ...

---Code minh hoạ qua hình ảnh có kèm giải thích:

```
def getAction(self, gameState):  
    """  
    Khi viết hàm getAction() (cũng như là các hàm phụ trợ) trong class MinimaxAgent,  
    bạn có thể gọi self.depth và self.evaluationFunction (self.depth và self.evaluationFunction  
    được định nghĩa trong class MultiAgentSearchAgent, class MinimaxAgent kế thừa từ class này):  
  
    • self.depth cho biết biết độ sâu giới hạn của cây (số bước nhìn xa về tương lai).  
    Lưu ý: một đơn vị độ sâu ở đây gồm một lần đi của Pacman và tất cả các ma sau đó;  
    như vậy, độ sâu bằng 2 nghĩa là: Pacman đi, tất cả các ma lần lượt đi,  
    Pacman đi, tất cả các ma lần lượt đi  
  
    • self.evaluationFunction nhận đầu vào là  
    một trạng thái mà không phải là trạng thái kết thúc game và trả về giá trị ước lượng của trạng thái đó.  
    self.evaluationFunction mặc định là scoreEvaluationFunction  
  
    // trả về trạng thái successor của game sau khi agent thực hiện hành động.  
    • Lệnh gameState.generateSuccessor(agentIndex, action):  
    • Lệnh gameState.getNumAgents(): // trả về tổng số agent trong game.  
    • Lệnh gameState.getLegalActions(agentIndex): // Trả về các hành động mà agent có thể thực hiện.  
    """
```

```

    """*** YOUR CODE HERE ***"""
    # Đầu tiên, trong thuật toán này, hàm Minimax sẽ có 3 tham số đầu vào
    # Đó chính là: agent, độ sâu, và trạng thái của game
    def minimax(agent, depth, gameState):
        # Nếu mà trạng thái này thua/thắng hoặc độ sâu của agent bằng độ sâu giới hạn trong khung chơi thì
        if gameState.isLose() or gameState.isWin() or depth == self.depth:
            # Trả về giá trị ước lượng của trạng thái mà pacman thực hiện hành động
            return self.evaluationFunction(gameState)
        # tại đây có 2 trường hợp:
        # 1 là khi agent=0 hoặc trường hợp còn lại

        if agent == 0: # maximize for pacman
            # trả về trạng thái successor của game sau khi agent thực hiện hành động
            # và trả về minimax: minimax agent có thể thực hiện
            return max(minimax(1, depth, gameState.generateSuccessor(agent, newState)) for newState in gameState.getLegalActions(agent))
        else: # minize for ghosts
            # Tính toán các Agent và tăng độ sâu(bước đi của pacman đi)
            nextAgent = agent + 1
            if gameState.getNumAgents() == nextAgent:
                nextAgent = 0
            if nextAgent == 0:
                depth += 1
            # trả về chi phí ước lượng ít nhất bằng cách gọi lại hàm
            return min(minimax(nextAgent, depth, gameState.generateSuccessor(agent, newState)) for newState in gameState.getLegalActions(agent))

```

```

        else: # minize for ghosts
            # Tính toán các Agent và tăng độ sâu(bước đi của pacman đi)
            nextAgent = agent + 1
            if gameState.getNumAgents() == nextAgent:
                nextAgent = 0
            if nextAgent == 0:
                depth += 1
            # trả về chi phí ước lượng ít nhất bằng cách gọi lại hàm
            return min(minimax(nextAgent, depth, gameState.generateSuccessor(agent, newState)) for newState in gameState.getLegalActions(agent))

    """Thực hiện hành động tối đa hoá hành động của pacman"""
    maximum = float("-inf")
    # hướng đi của pacman sẽ là hướng tây
    action = Directions.WEST
    # Trả về các hành động mà agent có thể thực hiện.
    for agentState in gameState.getLegalActions(0):
        # trả về trạng thái successor của game sau khi agent thực hiện hành động
        utility = minimax(1, 0, gameState.generateSuccessor(0, agentState))
        if utility > maximum or maximum == float("-inf"):
            maximum = utility
            action = agentState

    return action

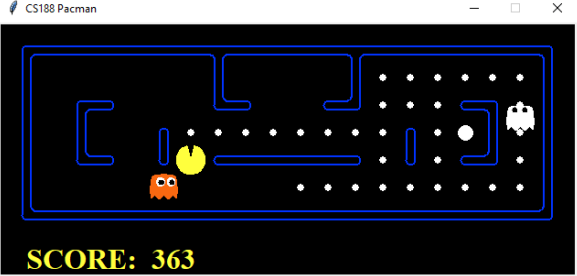
```

Để debug, kiểm tra code và đánh giá chương trình của mình, mình sẽ chạy chương trình bằng lệnh **python autograder.py -q q2**. Và có thể chạy lệnh: **python autograder.py -q q2 --no-graphics** để chạy chương trình mà không cần hiển thị graphics.

---Hình ảnh minh hoạ khi chạy:

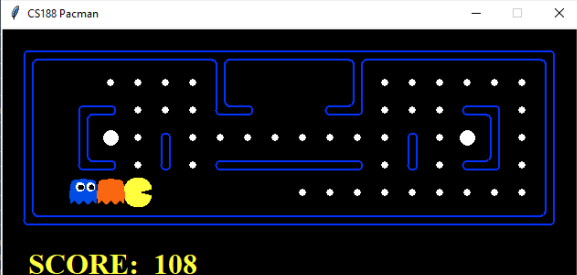
```
PS E:\NAM 3- HK1\Trí tuệ nhân tạo\Thực hành N3 Tuần 07-20191114\Thực hành N3 Tuần 07 - Source code\multiagent> python autograder.py -q q2
autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
import imp
Starting on 11-15 at 0:38:07

Question q2
=====
*** PASS: test_cases\q2\0-eval-function-lo
*** PASS: test_cases\q2\0-eval-function-lo
*** PASS: test_cases\q2\0-eval-function-wi
*** PASS: test_cases\q2\0-eval-function-wi
*** PASS: test_cases\q2\0-lecture-6-tree.t
*** PASS: test_cases\q2\0-small-tree.test
*** PASS: test_cases\q2\1-1-minimax.test
*** PASS: test_cases\q2\1-2-minimax.test
*** PASS: test_cases\q2\1-3-minimax.test
*** PASS: test_cases\q2\1-4-minimax.test
*** PASS: test_cases\q2\1-5-minimax.test
*** PASS: test_cases\q2\1-6-minimax.test
*** PASS: test_cases\q2\1-7-minimax.test
*** PASS: test_cases\q2\1-8-minimax.test
*** PASS: test_cases\q2\2-1a-vary-depth.test
*** PASS: test_cases\q2\2-1b-vary-depth.test
*** PASS: test_cases\q2\2-2a-vary-depth.test
*** PASS: test_cases\q2\2-2b-vary-depth.test
*** PASS: test_cases\q2\2-3a-vary-depth.test
*** PASS: test_cases\q2\2-3b-vary-depth.test
*** PASS: test_cases\q2\2-4a-vary-depth.test
*** PASS: test_cases\q2\2-4b-vary-depth.test
*** PASS: test_cases\q2\2-one-ghost-3level.test
*** PASS: test_cases\q2\3-one-ghost-4level.test
*** PASS: test_cases\q2\4-two-ghosts-3level.test
*** PASS: test_cases\q2\5-two-ghosts-4level.test
*** PASS: test_cases\q2\6-tied-root.test
*** PASS: test_cases\q2\7-1a-check-depth-one-ghost.test
*** PASS: test_cases\q2\7-1b-check-depth-one-ghost.test
```



```
PS E:\NAM 3- HK1\Trí tuệ nhân tạo\Thực hành N3 Tuần 07-20191114\Thực hành N3 Tuần 07 - Source code\multiagent> python autograder.py -q q2 --no-graphics
autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
import imp
Starting on 11-15 at 0:39:37

Question q2
=====
*** PASS: test_cases\q2\0-eval-func
*** PASS: test_cases\q2\0-eval-func
*** PASS: test_cases\q2\0-eval-func
*** PASS: test_cases\q2\0-lecture-6
*** PASS: test_cases\q2\0-small-tre
*** PASS: test_cases\q2\1-1-minimax.
*** PASS: test_cases\q2\1-2-minimax.
*** PASS: test_cases\q2\1-3-minimax.
*** PASS: test_cases\q2\1-4-minimax.
*** PASS: test_cases\q2\1-5-minimax.
*** PASS: test_cases\q2\1-6-minimax.
*** PASS: test_cases\q2\1-7-minimax.
*** PASS: test_cases\q2\1-8-minimax.
*** PASS: test_cases\q2\2-1a-vary-d
*** PASS: test_cases\q2\2-1b-vary-d
*** PASS: test_cases\q2\2-2a-vary-d
*** PASS: test_cases\q2\2-2b-vary-dep
*** PASS: test_cases\q2\2-3a-vary-dep
*** PASS: test_cases\q2\2-3b-vary-dep
*** PASS: test_cases\q2\2-4a-vary-dep
*** PASS: test_cases\q2\2-4b-vary-dep
*** PASS: test_cases\q2\2-one-ghost-3
*** PASS: test_cases\q2\3-one-ghost-4
*** PASS: test_cases\q2\4-two-ghosts-3
*** PASS: test_cases\q2\5-two-ghosts-4
*** PASS: test_cases\q2\6-tied-root.test
*** PASS: test_cases\q2\7-1a-check-dep
*** PASS: test_cases\q2\7-1b-check-dep
```



```

*** PASS: test_cases\q2\7-1c-check-depth-one-ghost.test
*** PASS: test_cases\q2\7-2a-check-depth-two-ghosts.test
*** PASS: test_cases\q2\7-2b-check-depth-two-ghosts.test
*** PASS: test_cases\q2\7-2c-check-depth-two-ghosts.test
*** Running MinimaxAgent on smallClassic 1 time(s).
Pacman died! Score: 84
Average Score: 84.0
Scores:      84.0
Win Rate:    0/1 (0.00)
Record:      Loss
*** Finished running MinimaxAgent on smallClassic after 49 seconds.
*** Won 0 out of 1 games. Average score: 84.000000 ***
*** PASS: test_cases\q2\8-pacman-game.test

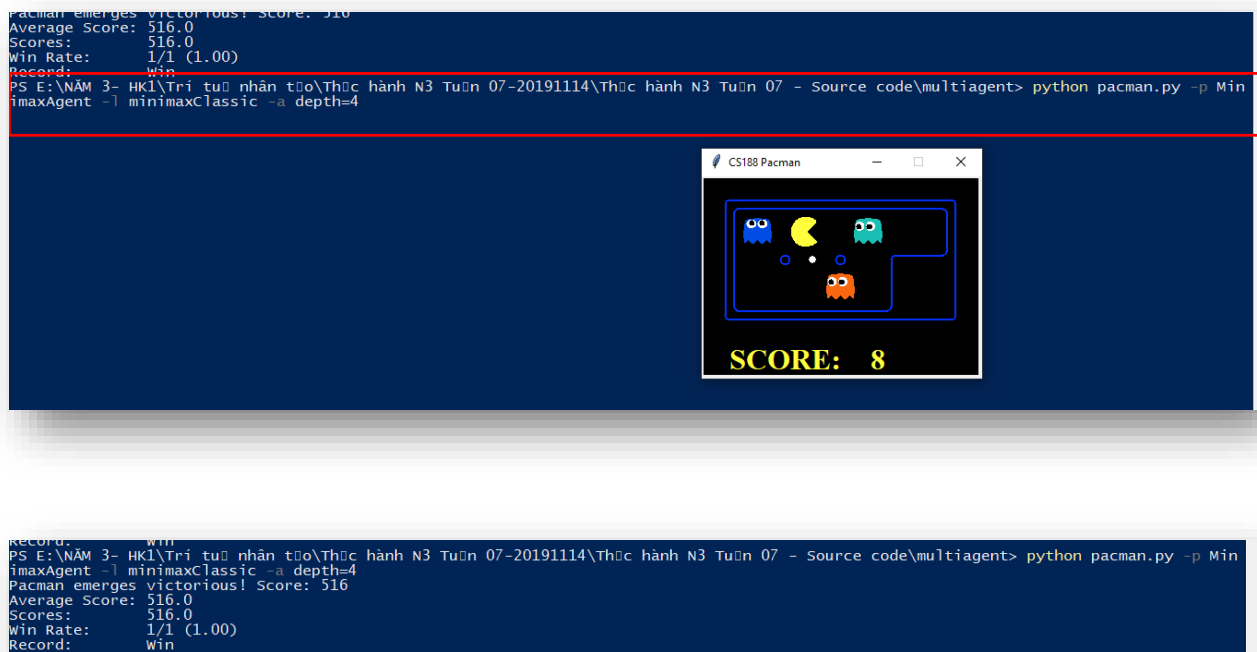
### Question q2: 5/5 ###

Finished at 0:40:26
Provisional grades
=====
Question q2: 5/5
-----
Total: 5/5

```

Có thể test thử bằng lệnh sau: **python pacman.py -p MinimaxAgent -l minimaxClassic -a depth=4.**

---- Hình ảnh minh họa:



The image shows a terminal window with the following output:

```

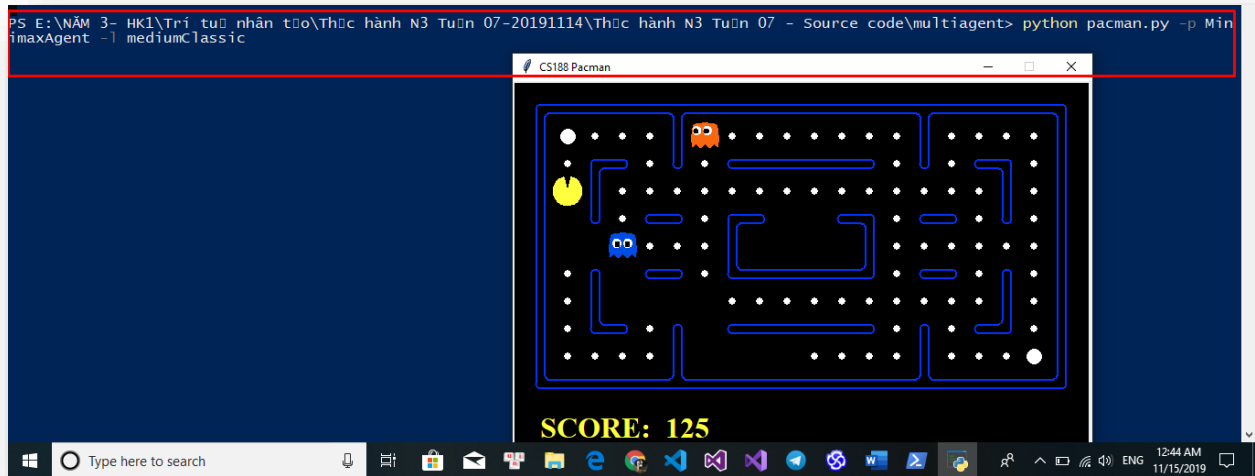
Pacman emerges victorious! Score: 516
Average Score: 516.0
Scores:      516.0
Win Rate:    1/1 (1.00)
Record:      Win
PS E:\NAM 3- HK1\Trí tuệ nhân tạo\Thực hành N3 Tuần 07-20191114\Thực hành N3 Tuần 07 - Source code\multiagent> python pacman.py -p MinimaxAgent -l minimaxClassic -a depth=4

```

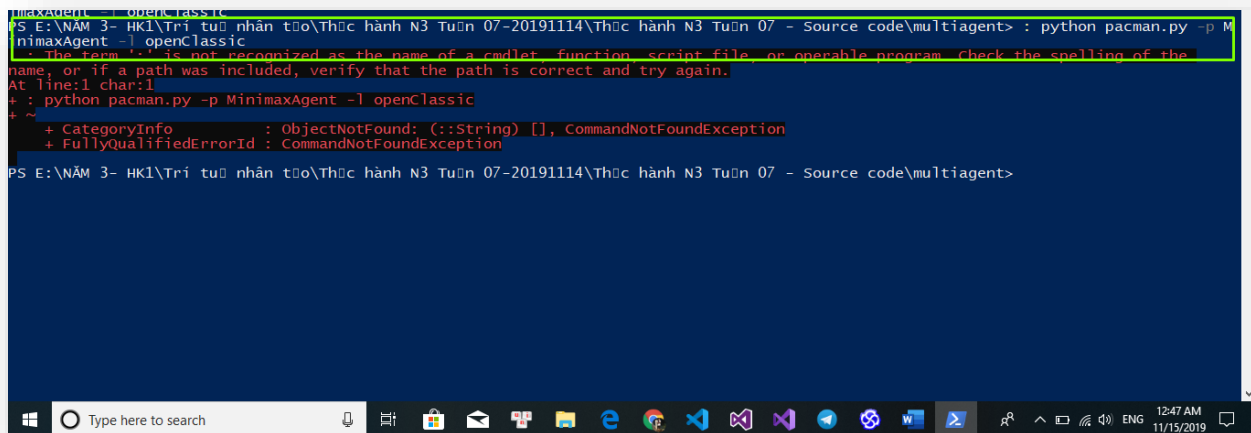
Below the terminal output, there is a small window titled "CS188 Pacman" showing a Pacman game. The game is in progress, with Pacman (yellow) at the bottom and three ghosts (blue, green, and orange) at the top. The score is displayed as "SCORE: 8".

- Khi bạn chạy trên các layout lớn hơn như openClassic và mediumClassic, bởi lệnh: **python pacman.py -p MinimaxAgent -l openClassic** hoặc **python pacman.py -p MinimaxAgent -l mediumClassic**

-----Hình ảnh minh hoạ: `python pacman.py -p MinimaxAgent -l mediumClassic`

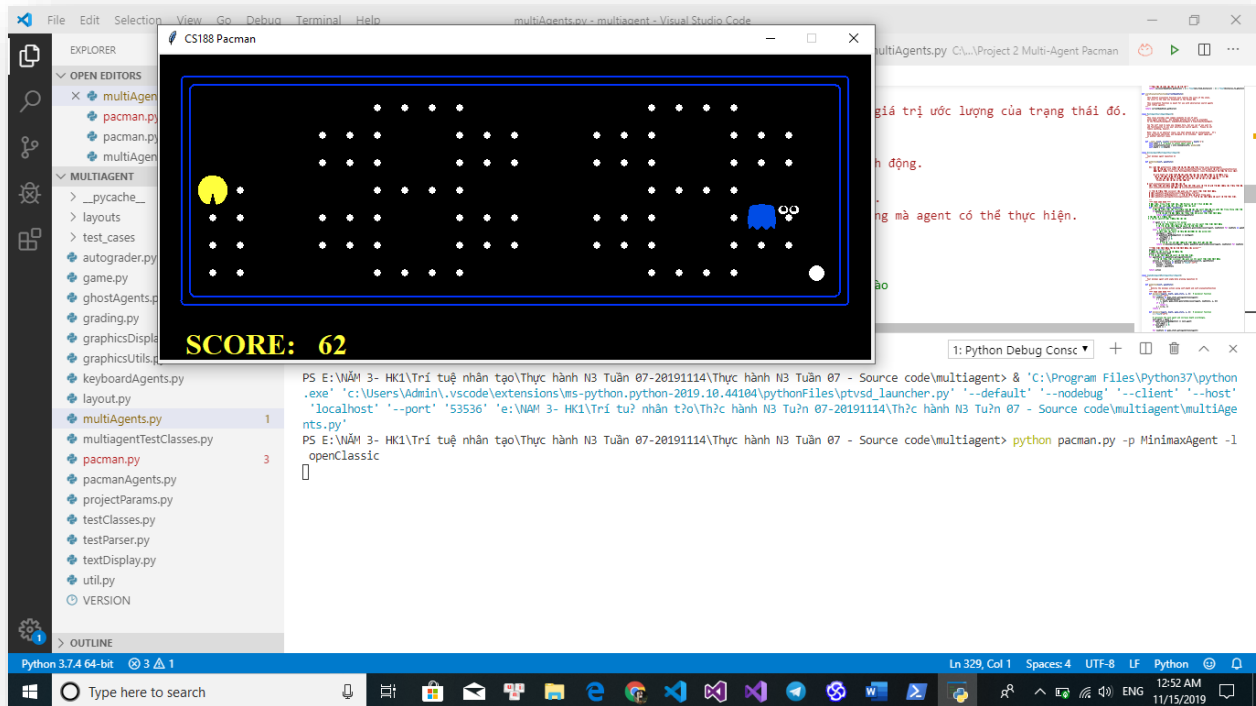


---- Hình ảnh minh hoạ: `python pacman.py -p MinimaxAgent -l openClassic`



Trong lượt chạy này, mình đã bị bug rồi mấy bạn và thầy ơi, mình sẽ cố gắng fixed bug này trong thời gian sớm nhất có thể !

-À, mình đã fix được rồi nè!



CẢM ƠN CÁC BẠN VÀ THẦY ĐÃ XEM BÀI CỦA EM NHE!