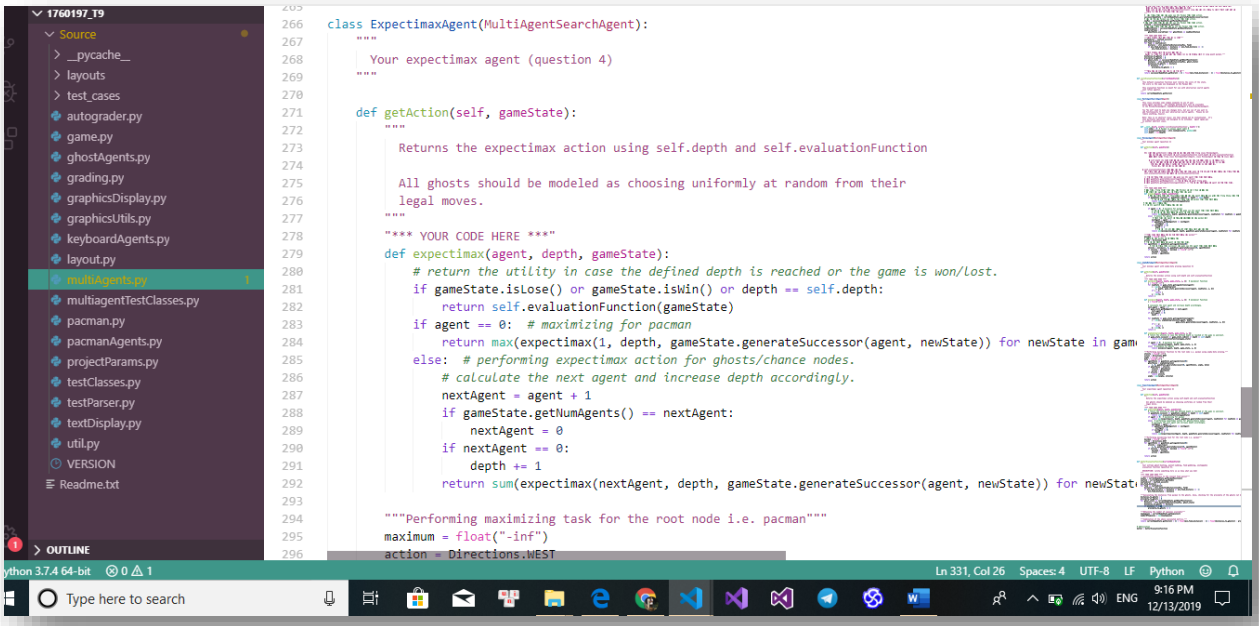
TÌM KIẾM ĐỐI KHÁNG

Thuật toán Expectimax

1> Demo code

Cả hai thuật toán Minimax và Alpha-Beta đều dự đoán hành động của đối thủ bằng cách giả định là đối thủ chơi tối ưu. Trên thực tế, không phải lúc nào đối thủ cũng chơi tối ưu, vậy nếu đối thủ chơi không tối ưu, ta sẽ dự đoán hành động của đối thủ bằng cách nào? Trong phần này mình sẽ tiến hành cài đặt thuật toán Expectimax để giải quyết vấn đề này.
Mình sẽ viết code vào phần **getAction()**,trong class **ExpectimaxAgent**

--- Hình ảnh code thuật toán:



```
283                 if agent == 0:  # maximizing for pacman
284                     return max(expectimax(1, depth, gameState.generateSuccessor(agent, newState)) for newState in gam
285                 else:  # performing expectimax action for ghosts/chance nodes.
286                     # calculate the next agent and increase depth accordingly.
287                     nextAgent = agent + 1
288                     if gameState.getNumAgents() == nextAgent:
289                         nextAgent = 0
290                     if nextAgent == 0:
291                         depth += 1
292                     return sum(expectimax(nextAgent, depth, gameState.generateSuccessor(agent, newState)) for newStat
293 
294             """Performing maximizing task for the root node i.e. pacman"""
295             maximum = float("-inf")
296             action = Directions.WEST
297             for agentState in gameState.getLegalActions(0):
298                 utility = expectimax(
299                     1, 0, gameState.generateSuccessor(0, agentState))
300                 if utility > maximum or maximum == float("-inf"):
301                     maximum = utility
302                     action = agentState
303 
304             return action
305 
306 
```

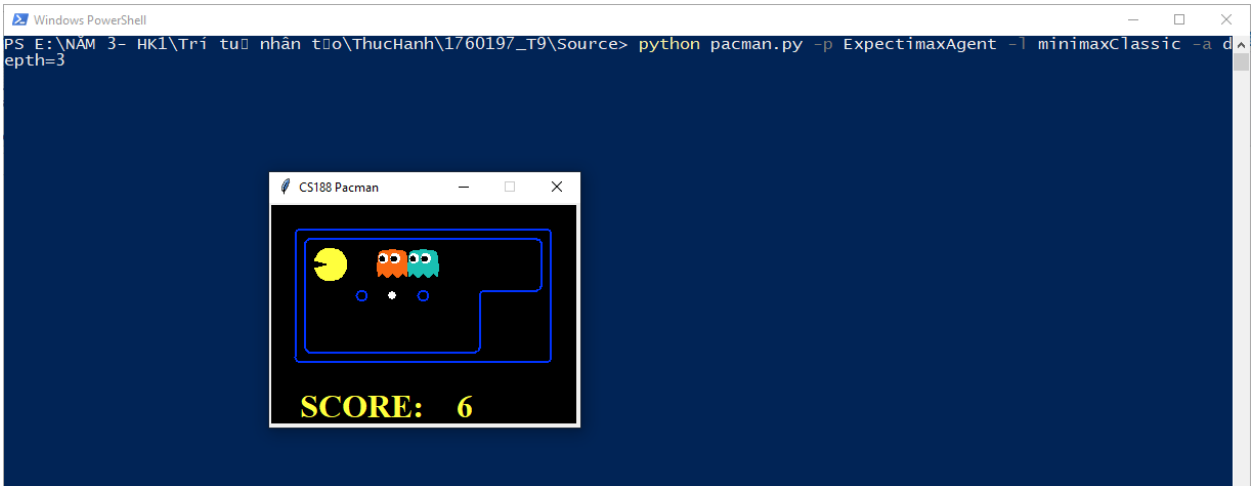
```
def betterEvaluationFunction(currentGameState):
    """
      Your extreme ghost-hunting, pellet-nabbing, food-gobbling, unstoppable
      evaluation function (question 5).

      DESCRIPTION: <write something here so we know what you did>
    """
    "*** YOUR CODE HERE ***"
    """Calculating distance to the closest food pellet"""
    newPos = currentGameState.getPacmanPosition()
    newFood = currentGameState.getFood()
    newFoodList = newFood.asList()
    min_food_distance = -1
    for food in newFoodList:
        distance = util.manhattanDistance(newPos, food)
        if min_food_distance >= distance or min_food_distance == -1:
            min_food_distance = distance

    """Calculating the distances from pacman to the ghosts. Also, checking for the proximity of the ghosts (at di
    distances_to_ghosts = 1
    proximity_to_ghosts = 0
    for ghost_state in currentGameState.getGhostPositions():
        distance = util.manhattanDistance(newPos, ghost_state)
        distances_to_ghosts += distance
        if distance <= 1:
            proximity_to_ghosts += 1

    """Obtaining the number of capsules available"""
    newCapsule = currentGameState.getCapsules()
    numberOfCapsules = len(newCapsule)

    """Combination of the above calculated metrics."""
```

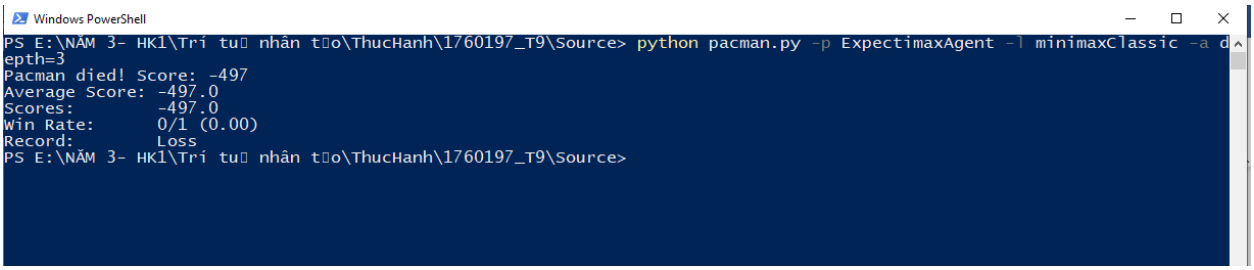
```
337
338     """Combination of the above calculated metrics."""
339     return currentGameState.getScore() + (1 / float(min_food_distance)) - (1 / float(distances_to_ghosts)) - prox
340
341
342 # Abbreviation
343 better = betterEvaluationFunction
344
```

2> Thực thi chương trình

-Sau khi code sẽ chạy các lệnh để demo lên chương trình:

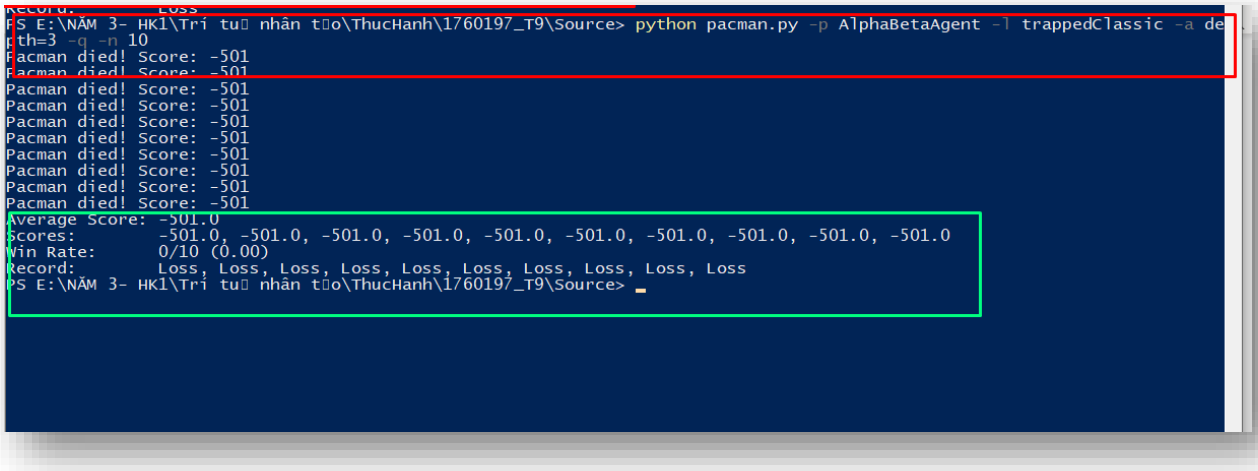
- python pacman.py -p ExpectimaxAgent -l minimaxClassic -a depth=3 và ảnh minh hoạ:





-Bạn có thể quan sát sự khác nhau giữa AlphaBetaAgent với ExpectimaxAgent bằng cách chạy hai lệnh sau:

- python pacman.py -p AlphaBetaAgent -l trappedClassic -a depth=3 -q -n 10



- python pacman.py -p ExpectimaxAgent -l trappedClassic -a depth=3 -q -n 10