

# Report Thực Hành Về Nhà Tuần 07

## Cài Đặt Hoàn Chính Phần 2.1, 2.2 Cho Bài Toán Tìm Kiếm Đối Kháng

Họ tên: Nguyễn Sanh Tuấn.

MSSV: 1760227.

Lớp: 17CK2 – ca 3 thực hành chiều t7.

### 1. Reflex Agent

#### \*Mục tiêu:

Thiết kế một chức năng đánh giá tốt hơn ở đây.

- Chức năng đánh giá mất trong sự kế thừa hiện tại và đề xuất GameStates (pacman.py) và trả về một con số, nơi số càng cao càng tốt.
  - Đoạn code dưới đây trích xuất một số thông tin hữu ích từ nhà nước, giống như còn lại thực phẩm (newFood) và vị trí Pacman sau khi di chuyển (newPos) newScaredTimes nắm giữ số lần di chuyển mà mỗi con ma sẽ vẫn sợ hãi vì Pacman đã ăn một viên quyền lực.
  - In ra các biến để xem những gì bạn đang nhận được, sau đó kết hợp chúng để tạo một hàm đánh giá bậc thầy.
- ⇒ 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.

#### \*Code:

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. Cụ thể như:

```
// Lấy vị trí của Pacman sau khi Pacman thực hiện action.  
newPos = successorGameState.getPacmanPosition()  
// Lấy trạng thái mới của game sau khi Pacman thực hiện action.  
successorGameState = currentGameState.generatePacmanSuccessor(action)  
// 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()
```

⇒ Tiến hành cài đặt hoàn chỉnh hàm evaluationFunction().

```
**** YOUR CODE HERE ****  
"""tính khoảng cách đến thức ăn xa nhất"""  
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  
  
"""Tính khoảng cách từ pacman đến hồn ma.
```

```

    Ngoài ra, kiểm tra sự gần gũi của những con ma (ở khoảng cách 1) xung
    quanh pacman."""
    distances_to_ghosts = 1
    proximity_to_ghosts = 0
    for ghost_state in successorGameState.getGhostPositions():
        distance = util.manhattanDistance(newPos, ghost_state)
        distances_to_ghosts += distance
        if distance <= 1:
            proximity_to_ghosts += 1

    """Tổng hợp số liệu vừa tìm được sẽ trả về"""
    return successorGameState.getScore() + (1 / float(min_food_distance))
- (1 / float(distances_to_ghosts)) - proximity_to_ghosts

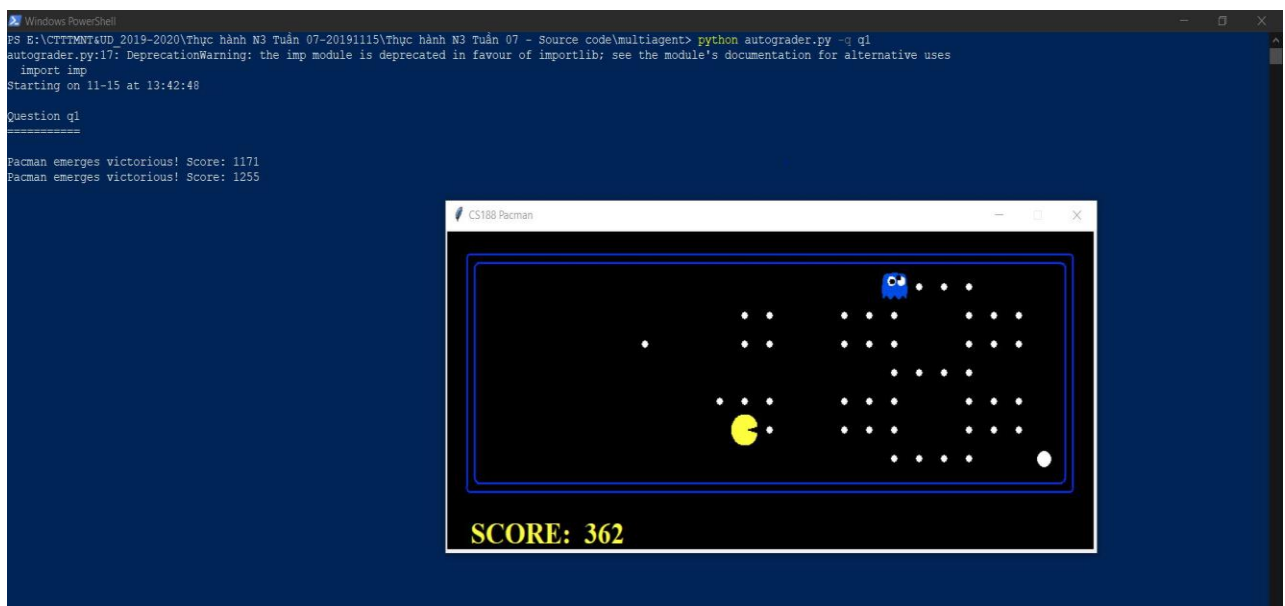
```

### ***\*Chạy lệnh kiểm tra thuật toán đã cài đặt.***

`python autograder.py -q q1`

Lệnh này sẽ thực hiện chạy Pacman trên openClassic layout 10 lần khác nhau. Nếu bạn bị time out trong quá trình chạy, bạn sẽ không thể nào thắng được ma, khi đó chương trình của bạn sẽ nhận được 0 điểm. Bạn sẽ nhận được 1 điểm nếu Pacman thắng ma ít nhất 5 lần, 2 điểm nếu Pacman thắng ma tất cả 10 lần. Bạn sẽ nhận thêm được 1 điểm, nếu điểm trung bình sau các lần chơi game của bạn lớn hơn 500 điểm. Bạn sẽ nhận được 2 điểm, nếu điểm trung bình sau các lần chơi game lớn hơn 1000 điểm.

⇒ **Kết quả khi chạy lệnh kiểm tra cài đặt:**



```

Windows PowerShell
PS E:\CTTTMNT\UD 2019-2020\Thực hành N3 Tuần 07-20191115\Thực hành N3 Tuần 07 - Source code\multiagent> python autograder.py -q q1
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 13:42:48

Question q1
=====

Pacman emerges victorious! Score: 1171
Pacman emerges victorious! Score: 1255
Pacman emerges victorious! Score: 1243
Pacman emerges victorious! Score: 1256
Pacman emerges victorious! Score: 1248
Pacman emerges victorious! Score: 1247
Pacman emerges victorious! Score: 1251
Pacman emerges victorious! Score: 1255
Pacman emerges victorious! Score: 1252
Pacman emerges victorious! Score: 1228
Average Score: 1240.6
Scores:      1171.0, 1255.0, 1243.0, 1256.0, 1248.0, 1247.0, 1251.0, 1255.0, 1252.0, 1228.0
Win Rate:    10/10 (1.00)
Record:      Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
*** PASS: test_cases/q1/grade-agent.test (4 of 4 points)
*** 1240.6 average score (2 of 2 points)
***   Grading scheme:
***     < 500: 0 points
***     >= 500: 1 points
***     >= 1000: 2 points
*** 10 games not timed out (0 of 0 points)
***   Grading scheme:
***     < 10: fail
***     >= 10: 0 points
*** 10 wins (2 of 2 points)
***   Grading scheme:
***     < 1: fail
***     >= 1: 0 points
***     >= 5: 1 points
***     >= 10: 2 points

### Question q1: 4/4 ###

Finished at 13:44:52

Provisional grades
=====
Question q1: 4/4
-----
Total: 4/4

```

`python autograder.py -q q1 --no-graphics`

Câu lệnh này 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 chạy câu lệnh kiểm tra thuật toán:**

```

Windows PowerShell
PS E:\CTTTMNT\UD 2019-2020\Thực hành N3 Tuần 07-20191115\Thực hành N3 Tuần 07 - Source code\multiagent> python autograder.py -q q1 --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 13:50:16

Question q1
=====

Pacman emerges victorious! Score: 1171
Pacman emerges victorious! Score: 1255
Pacman emerges victorious! Score: 1243
Pacman emerges victorious! Score: 1256
Pacman emerges victorious! Score: 1248
Pacman emerges victorious! Score: 1247
Pacman emerges victorious! Score: 1251
Pacman emerges victorious! Score: 1255
Pacman emerges victorious! Score: 1252
Pacman emerges victorious! Score: 1228
Average Score: 1240.6
Scores:      1171.0, 1255.0, 1243.0, 1256.0, 1248.0, 1247.0, 1251.0, 1255.0, 1252.0, 1228.0
Win Rate:    10/10 (1.00)
Record:      Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
*** PASS: test_cases/q1/grade-agent.test (4 of 4 points)
*** 1240.6 average score (2 of 2 points)
***   Grading scheme:
***     < 500: 0 points
***     >= 500: 1 points
***     >= 1000: 2 points
*** 10 games not timed out (0 of 0 points)
***   Grading scheme:
***     < 10: fail
***     >= 10: 0 points
*** 10 wins (2 of 2 points)
***   Grading scheme:
***     < 1: fail
***     >= 1: 0 points
***     >= 5: 1 points
***     >= 10: 2 points

### Question q1: 4/4 ###

Finished at 13:50:19

Provisional grades
=====
Question q1: 4/4
-----
Total: 4/4

```

## 2. Minimax

## **\*Mục Tiêu:**

Trả về hành động minimax từ hiện tại tại GameState sử dụng self.depth và self.evaluationFunction.

Dưới đây là một số cuộc gọi phương pháp đó có thể có ích khi thực hiện minimax.

- gameState.getLegalActions (agentIndex):  
⇒ Trả về các hành động mà agent có thể thực hiện. agentIndex = 0 có nghĩa là Pacman, bóng ma là > = 1
- gameState.generateSuccessor (agentIndex, hành động):  
⇒ Trả về trạng thái game sau khi agent thực hiện hành động
- gameState.getNumAgents ():  
⇒ Trả về tổng số agent trong game
- gameState.isWin ():  
⇒ Lợi nhuận hay không tình trạng trò chơi là một trạng thái chiến thắng
- gameState.isLose ():  
⇒ Lợi nhuận hay không tình trạng trò chơi là một trạng thái giảm điểm

### **Tìm hiểu ý nghĩa của:**

- **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: nhận đầu vào là một trạng thái và trả về giá trị ước lượng là điểm số của game tại trạng thái đó.

## **\*Cài đặt thuật toán:**

```
""" 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 sa  
u 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 các hành động mà 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(0, 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, action))
        if utility > maximum or maximum == float("-inf"):
            maximum = utility
            action = agentState
    return action

util.raiseNotDefined()

```

### **\*Kiểm tra kết quả cài đặt thuật toán:**

- chạy chương trình bằng lệnh: `python autograder.py -q q2`

Windows PowerShell

```

PS E:\CTTTM74UD_2019-2020\Thực hành N3 Tuần 07-20191115\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 14:39:43

Question q2
=====

*** PASS: test_cases\q2\0-eval-function-lose-states-1.test
*** PASS: test_cases\q2\0-eval-function-lose-states-2.test
*** PASS: test_cases\q2\0-eval-function-win-states-1.test
*** PASS: test_cases\q2\0-eval-function-win-states-2.test
*** PASS: test_cases\q2\0-lecture-6-tree.test
*** 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\3-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
*** 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).

```

CS188 Pacman

SCORE: 515

```

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 19 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 14:40:03

Provisional grades
=====
Question q2: 5/5
-----
Total: 5/5

Your grades are NOT yet registered. To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.

```

- Chạy lệnh: `python autograder.py -q q2 --no-graphics` để chạy chương trình mà không cần hiển thị graphics.

```

Windows PowerShell
PS E:\CTTTMT\UD 2019-2020\Thực hành N3 Tuần 07-2019\115\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 14:44:45

Question q2
=====
*** PASS: test_cases\q2\0-eval-function-lose-states-1.test
*** PASS: test_cases\q2\0-eval-function-lose-states-2.test
*** PASS: test_cases\q2\0-eval-function-win-states-1.test
*** PASS: test_cases\q2\0-eval-function-win-states-2.test
*** PASS: test_cases\q2\0-lecture-6-tree.test
*** 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\3-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
*** 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 1 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 14:44:47

Provisional grades
=====
Question q2: 5/5
-----
Total: 5/5

Your grades are NOT yet registered. To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.

```

-----THE END-----