

# Assignment 03 - Evaluation functions for Minimax/AlphaBeta/Expectimax Report

Sinh viên thực hiện: **Huỳnh Thiện Tùng (19522492)**

Lớp: **CS106.M11.KHCL**

**Báo cáo ngắn gọn mô tả ý tưởng/chiến thuật mình tự thiết kế evaluation function. Những đặc trưng mình sử dụng để ước lượng giá trị trạng thái là gì? Trọng số của các đặc trưng đó như thế nào? Tại sao?**

Ý tưởng: Nhận thấy chỉ sử dụng score để ước lượng cho mỗi trạng thái trung gian của Pacman thì kết quả chưa thật sự tốt lắm. Chiến thuật chạy độc lập từng đặc trưng như: score, khoảng cách từ pacman đến thức ăn, khoảng cách từ Pacman đến các con ma,... Ghi nhận lại từng đặc trưng có đóng góp như thế nào, cuối cùng tổng hợp tuyến tính các đặc trưng đó lại với nhau cho hàm ước lượng. Giai đoạn cuối cùng là điều chỉnh các hệ số từng đặc trưng sao cho mang lại kết quả tốt nhất.

Các đặc trưng sử dụng:

- Score (là giá trị ban đầu cho hàm ước lượng, sẽ trừ dồn vào các đặc trưng còn lại)
- Khoảng cách Manhattan từ Pacman tới thức ăn (trọng số 1.0)
- Khoảng cách Manhattan từ Pacman tới con ma hung dữ (trọng số 2.0)
- Khoảng cách Manhattan từ Pacman tới con ma hiền lành, ngây thơ (trọng số 4.0)
- Số chấm thức ăn còn lại (trọng số 10.0)
- Số chấm capsule còn lại (trọng số 20.0)

Giải thích:

```
""" Sử dụng đặc trưng điểm của currentState cho hàm đánh giá """

""" Tính khoảng cách từ Pacman tới tất cả các vị trí thức ăn. Ưu tiên cho Pacman ăn chấm thức ăn gần nhất """
food_distances = [manhattanDistance(newPos, food_position) for food_position in newFood]
if len(food_distances) != 0:
    closest_food = min(food_distances)
    score -= 1.0 * closest_food

""" Lấy thông tin, vị trí các con ma trong trạng thái game hiện tại """
for ghost in newGhostStates:
    if ghost.scaredTimer != 0:
        enemy_ghosts.append(ghost)
    else:
        scared_ghosts.append(ghost)

for enemy_ghost in enemy_ghosts:
    enemy_ghost_pos.append(enemy_ghost.getPosition())

for scared_ghost in scared_ghosts:
    scared_ghost_pos.append(scared_ghost.getPosition())
```

```
""" Với con ma chiến (hung dữ) thì chúng ta không nên để Pacman lại gần, mục tiêu cho Pacman đi xa con ma hung dữ đó
Xác định khoảng cách gần nhất từ Pacman tới con ma hung dữ đó, sau đó trừ một lượng nghịch đảo khoảng cách.
Ta chọn hệ số trừng phạt là 2.0. Vì chúng ta cần minimize giá trị ước lượng nên cần tăng khoảng cách từ Pacman
tới con ma hung dữ đó
"""
if len(enemy_ghost_pos) != 0:
    distance_from_enemy_ghost = [manhattanDistance(newPos, enemy_ghost_position) for enemy_ghost_position in enemy_ghost_pos]
    closest_enemy_ghost = min(distance_from_enemy_ghost)
    score -= 2.0 * (1 / closest_enemy_ghost)

""" Với con ma hiền lành, ngay thơ thì chúng ta nên để Pacman lại gần để ăn nó, sẽ đóng góp một lượng điểm lớn,.
Ta chọn hệ số trừng phạt là 10.0 (ý nghĩa là không khuyến khích Pacman đi xa con ma, thay vào đó đi gần lại sẽ có cơ hội ăn điểm)
Trái với trường hợp trên, vì đi lại gần con ma hiền lành đó sẽ có cơ hội ăn điểm, nên phải giảm khoảng cách tới con ma hiền lành đó
"""
if len(scared_ghost_pos) != 0:
    distance_from_scared_ghost = [manhattanDistance(newPos, scared_ghost_position) for scared_ghost_position in scared_ghost_pos]
    closest_scared_ghost = min(distance_from_scared_ghost)
    score -= 4.0 * closest_scared_ghost

""" Cho hệ số vào trước đặc trưng capsule còn lại và thức ăn còn lại lớn, là 20 và 5. Ý nghĩa mong muốn Pacman phải cố minimize
giá trị ước lượng, nên khuyến khích nó nên ăn nhiều capsule và chấm thức ăn nhất có thể"""
score -= 20.0 * capsule_left
score -= 10.0 * food_left
return score

# Abbreviation
better = betterEvaluationFunction
```

**Chạy thử nghiệm các thuật toán đã cài đặt với evaluation function mình đã thiết kế (có thể so sánh với evaluation function có sẵn scoreEvaluationFunction trong đó chỉ sử dụng điểm số của trạng thái để ước lượng giá trị trạng thái). Hiệu năng/Hiệu quả của Minimax và AlphaBeta và Expectimax so với nhau thế nào? Nên thực nghiệm trên nhiều map khác nhau (trong thư mục layouts).**

Video quay lại ván chơi đạt điểm cao nhất trong quá trình thực nghiệm

<https://drive.google.com/file/d/1tRl8Qq0qTOUagaDbELhYESOCWMI3bya0/view?usp=sharing>

Table of Scores							
		betterEvaluationFunction			scoreEvaluationFunction		
Map	Iteration	Minimax	AlphaBeta	Expectimax	Minimax	AlphaBeta	Expectimax
capsuleClassic	1	-353	371	-48	-429	-421	-441
	2	-443	221	0	-477	-505	-439
	3	-97	1383	-477	-452	-452	-338
contestClassic	1	736	-239	764	726	-130	99
	2	1580	572	121	620	-332	224
	3	437	-160	1557	-256	432	432
mediumClassic	1	1681	1711	2074	-952	90	1167
	2	1711	1314	1683	-1104	-54	684
	3	1433	1635	1819	-1503	779	-1362
minimaxClassisc	1	-492	-496	-497	-495	-495	-502
	2	-492	508	513	513	-495	-494
	3	-492	-496	-497	-495	-492	510
Average of each column		434.0833333	527	584.3333333	-358.6666667	-172.9166667	-38.3333333
Average of each group		515.1388889			-189.9722222		

Nhận xét: Đối với hàm ước lượng **betterEvaluationFunction**, Expectimax cho hiệu năng cao nhất, số điểm trung bình trong 4 map cao nhất. Đứng thứ 2 là AlphaBeta, cuối cùng là Minimax. Kết quả này cũng tương tự trong hàm ước lượng `scoreEvaluationFunction`.

Tuy nhiên, vẫn còn một số map vẫn chưa thực sự giải được, điều này có thể là do hàm ước lượng chưa thật sự tốt. Hầu hết các trường hợp xấu nhất khi sử dụng hàm ước lượng **scoreEvaluationFunction** rơi vào trường hợp “loop”, nghĩa là Pacman không di chuyển mà cứ xoay trái rồi xoay phải, phải đợi Ghost lại sát mới di chuyển nhưng bước di chuyển vẫn không mang lại ý nghĩa.

Tổng thể, nếu chỉ sử dụng score để làm đặc trưng cho hàm ước lượng thì sẽ không mang lại kết quả tốt, điểm số rất nhỏ, số trường hợp không giải được nhiều,... Nhưng khi cải tiến hàm ước lượng bằng việc thêm một số đặc trưng khác đã đóng góp một phần hiệu suất đáng kể. Vấn đề này cần được nghiên cứu và cải tiến thêm.