

Tên: Trương Văn Khải

MSSV: 21520274

Lớp: CS106.N21 (Môn: Trí tuệ nhân tạo)

GVHD: Dr. Lương Ngọc Hoàng



BÁO CÁO BÀI TẬP 3:

EVALUATION FUNCTIONS FOR MINIMAX/ALPHABETA/EXPECTIMAX

Yêu cầu:

1. Báo cáo 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?
2. Chạy thử nghiệm các thuật toán đã cài đặt với evaluation function mình đã thiết kế (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? Thực nghiệm trên ít nhất 05 map khác nhau (trong thư mục layouts).
 1. Chọn một mốc chi phí tính toán phù hợp với máy tính của mỗi bạn (ví dụ tối đa 3 phút cho mỗi lần chạy).
3. Source Code.
4. Có thể submit 1 video record lại 1 ván chơi (link google drive) mà em thấy là cho kết quả tốt nhất.

1. Báo cáo 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 chiến thuật thiết kế Evaluation Functions:

- Nhận thấy ở hàm **scoreEvaluationFunction**, hàm chỉ trả về điểm số của trạng thái hiện tại, điều này sẽ cho ra kết quả không được tốt lắm.
- Lấy ý tưởng khác, đề ra 1 chiến thuật hiệu quả hơn là chạy độc lập từng đặc trưng: **Score**, **Khoảng cách từ PacMan đến chấm thức ăn**, **Khoảng cách từ PacMan đến con ma ...**
- Sau đó ghi nhận từng trọng số ảnh hưởng của từng đặc trưng đến hàm đánh giá => tổ hợp tuyến tính các đặc trưng đó lại với nhau tạo thành hàm ước lượng.
- Điều chỉnh trọng số sao cho phù hợp.

- Các đặc trưng được sử dụng:

- Khoảng cách Manhattan từ PacMan tới “Chấm Thức Ăn”: **closest_food** (trọng số 1.0).
- Khoảng cách Manhattan từ PacMan tới “Con Ma Bình Thường”: **closest_enemy_ghost** (trọng số 2.0).
- Khoảng cách Manhattan từ PacMan tới “Con Ma Hiên Lành”: **closest_scared_ghost** (trọng số 4.0).
- Số viên năng lượng còn lại: **capsule_left** (trọng số 20.0).
- Số viên thức ăn còn lại: **food_left** (trọng số 10.0).

- Giải thích tại sao phải chọn Trọng số các đặc trưng đó:

```
""" Khởi tạo các giá trị đặc trưng """
closest_food = float('inf')
closest_scared_ghost = float('inf')
closest_enemy_ghost = float('inf')

""" Đặc trưng thứ 1 là số chấm thức ăn còn lại và số viên năng lượng còn lại """
food_left = len(newFood.asList())
capsule_left = len(newCapsules)

""" Đặc trưng thứ 2 là những con ma bình thường và con ma sợ hiên """
enemy_ghosts = list()
scared_ghosts = list()
enemy_ghost_pos = list()
scared_ghost_pos = list()

""" 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

""" Hàm lấy 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 Bình Thường thì chúng ta không nên để Pacman lại gần -> đưa Pacman đi xa con ma hung dữ đó
Xác định khoảng cách gần nhất từ Pacman tới con ma đó, sau đó trừ một lượng nghịch đảo khoảng cách.
Ta chọn trọng số là 2.0. Vì chúng ta cần minimize giá trị ước lượng nên cần tăng khoảng cách từ Pacman
tới con ma 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 thì chúng ta nên để Pacman lại gần để ăn nó, sẽ được lượng điểm lớn,.
Ta chọn hệ số trừng phạt là 4.0 (ý nghĩa là không khuyến khích Pacman đi xa con mà, thay vào đó đi gần lại sẽ có cơ hội ăn điểm)
"""
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 cho viên năng lượng và thức ăn còn lại lớn, là 20 và 10. Ý nghĩa mong muốn Pacman phải cố minimize
giá trị ước lượng, nên khuyến khích nó nên ăn nhiều viên năng lượng và chấm thức ăn nhất có thể"""
score -= 20.0 * capsule_left
score -= 10.0 * food_left
return score

# Abbreviation
better = betterEvaluationFunction

```

- Giải thích:

- **Trọng số là 1.0** đối với khoảng cách tới chấm thức ăn: Ưu tiên cho Pacman ăn chấm thức ăn gần nhất
- **Trọng số là 2.0** đối với khoảng cách tới con ma bình thường: Với con ma Bình Thường thì chúng ta không nên để Pacman lại gần -> đưa Pacman đi xa con ma hung dữ đó. Xác định khoảng cách gần nhất từ Pacman tới con ma đó, sau đó trừ một lượng nghịch đảo khoảng cách. Ta chọn trọng số là 2.0. Vì chúng ta cần minimize giá trị ước lượng nên cần tăng khoảng cách từ Pacman tới con ma đó.
- **Trọng số là 4.0** đối với khoảng cách tới con ma hiền lành: Với con ma hiền lành thì chúng ta nên để Pacman lại gần để ăn nó, sẽ được lượng điểm lớn. Ta chọn hệ số trừng phạt là 4.0 (ý nghĩa là không khuyến khích Pacman đi xa con mà, thay vào đó đi gần lại sẽ có cơ hội ăn điểm).
- **Trọng số là 10.0 và 20.0** đối với chấm thức ăn và viên năng lượng: Cho hệ số vào trước đặc trưng cho viên năng lượng 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 viên năng lượng và chấm thức ăn nhất có thể.

2. Chạy thử nghiệm các thuật toán đã cài đặt với evaluation function mình đã thiết kế (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? Thử nghiệm trên ít nhất 05 map khác nhau (trong thư mục layouts).

- Bảng điểm trung bình trong 7 map: **capsuleClassic**, **contestClassic**, **mediumClassic**, **MinimaxClassic**, **OpenClassic**, **smallClassic**, **trappedClassic**:

MAP	ALGORITHM DEPTH = 3	Evaluation function					
		scoreEvaluationFunction			betterEvaluationFunction		
		Loss/ Win: Score					
		Lần 1	Lần 2	Lần 3	Lần 1	Lần 2	Lần 3
capsuleClassic	Minimax	-430	-442	-556	-549	-234	-443
	Alpha-Beta Pruning	-406	104	-431	192	692	-431
	Expectimax	23	-463	-206	1118	326	165
ContestClassic	Minimax	-428	-502	-437	1171	1302	1323
	Alpha-Beta Pruning	1679	39	218	-294	560	1725
	Expectimax	-126	-238	1023	1555	1469	627
MediumClassic	Minimax	1349	-345	1401	1508	1473	-22
	Alpha-Beta Pruning	-349	377	-380	1465	1473	1613
	Expectimax	-6987	1400	-348	2119	1778	1769
MinimaxClassic	Minimax	-492	-495	-492	510	-498	-497
	Alpha-Beta Pruning	512	-496	513	511	-495	-497
	Expectimax	513	512	504	508	510	511
OpenClassic	Minimax	489	735	937	1387	1332	1138
	Alpha-Beta Pruning	1032	-384	-493	1036	1119	582
	Expectimax	188	937	102	1090	1329	1155
smallClassic	Minimax	-1809	1339	287	1291	1338	1234
	Alpha-Beta Pruning	1328	1441	-89	1287	1737	-339
	Expectimax	789	1713	1042	86	1177	-38
trappedClassic	Minimax	-501	-501	-501	531	531	-502
	Alpha-Beta Pruning	-501	-501	-501	-502	-502	531
	Expectimax	-502	532	-502	-502	531	-502

Chú thích: Điểm số là màu đỏ nếu ván đấu đó là ván thắng, màu đen nếu ván đó là ván thua.

- **Bảng tổng kết kết quả chạy:**

scoreEvaluationFunction			betterEvaluationFunction		
Average of MiniMax	Average of Alpha-Beta Pruning	Average of Expectimax	Average of MiniMax	Average of Alpha-Beta Pruning	Average of Expectimax
-66.38095238	129.1428571	-4.476190476	634.4761905	545.8571429	799.0952381
19.42857143			659.8095238		

- **Nhận xét:**

- So với hàm ước lượng **scoreEvaluationFunction**, **betterEvaluationFunction** cho hiệu năng vượt trội hẳn, điểm trung bình cao hơn hẳn (~659.8 >> ~19.4). Trong đó **Expectimax**

cho hiệu năng tốt nhất trong tất cả các map, đứng thứ 2 là **Minimax**, cuối cùng là **AlphaBeta**.

- Tuy nhiên vẫn còn một số map vẫn chưa thực sự giải được, hầu hết các trường hợp xấu nhất rơi vào trường hợp loop, nghĩa là PacMan cứ di chuyển qua lại 1 chỗ sát theo con ma hoặc đứng yên 1 chỗ, phải đợi con ma tiến lại gần mới di chuyển nhưng bước di chuyển vẫn không mang lại nhiều ý nghĩa. Nhưng các trường hợp rơi vào trường hợp loop hầu như đều xảy ra khi sử dụng hàm ước lượng **scoreEvaluationFunction**.
- Tóm lại, nếu chỉ sử dụng score để làm đặc trưng cho hàm ước lượng thì kết quả cho ra sẽ rất xấu. Thay vào đó, ta thêm 1 số đặc trưng khác cho hàm ước lượng, có thể thấy kết quả cho ra tốt hơn, hiệu suất vượt trội.

3. Submit video mà bản thân cho là ấn tượng

- **Video1:** map **mediumClassic**, **ExpectimaxAgent**, sử dụng **betterEvaluationFunction**. (Điểm số ấn tượng)
- **Link:** <https://drive.google.com/drive/folders/1y3WCB-mIUS-MoLdI5oi3H9c96-kh1yVL?usp=sharing>