

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH



Assignment 3

Evaluation functions for Minimax/AlphaBeta/Expectimax

Môn: Trí tuệ nhân tạo
Lớp: CS106.N21.KHCL
Giảng viên hướng dẫn: Lương Ngọc Hoàng
Sinh viên: Bùi Mạnh Hùng - 21522110

Ngày 27 tháng 5 năm 2023

Mục lục

1	Ý tưởng hàm evaluation function	3
2	Kết quả thực nghiệm	4
3	Kết luận	5
3.1	Độ hiệu quả của hai hàm đánh giá	5
3.2	Độ hiệu quả của 3 thuật toán	5
3.3	Trường hợp tốt nhất	5

1 Ý tưởng hàm evaluation function

Giá trị trả về của hàm lượng giá:

```
score_weight * score + \
food_weight * minFoodDistance + \
ghost_weight * min_ghost_distance + \
remaining_food_weight * remainingFood + \
proximity_weight * proximity_to_ghosts + \
capsule_weight * totalCapsule
```

Hàm lượng giá dựa vào các yếu tố:

- **score**: Số điểm đạt được
- **minFoodDistance**: Khoảng cách từ Pacman đến chấm thức ăn nhỏ nhất
- **min_ghost_distance**: Tổng khoảng cách từ Pacman đến các Ghosts
- **remainingFood**: Số chấm thức ăn còn lại
- **proximity_to_ghosts**: Số lượng Ghosts Pacman chạm vào
- **totalCapsule**: Tổng số viên Capsule còn lại

Tương ứng với các trọng số khởi tạo:

- `score_weight = 10`
- `food_weight = -10`
- `ghost_weight = 5`
- `remaining_food_weight = -20`
- `capsule_weight = -200`
- `proximity_weight = -300`

1.1 Ý nghĩa từng yếu tố được lựa chọn

1.1.1 Score

Mục đích: Khuyến khích Pacman ăn các chấm thức ăn giá trị điểm sẽ tăng 20 lần. Đặc biệt, khi ăn các Ghost_Scare (Ghost đang ở trạng thái sợ hãi) thì điểm số sẽ tăng nhiều hơn nữa.

1.1.2 minFoodDistance

```
minFoodDistance = 0
for food in newFood.asList():
    dis_Pacman_Food = manhattanDistance(newPos, food)
    if dis_Pacman_Food < minFoodDistance or minFoodDistance == 0:
        minFoodDistance = dis_Pacman_Food
```

Mục đích: Khuyến khích Pacman tiếp cận các chấm thức ăn. Khi càng xa chấm thức ăn giá trị giảm càng lớn (-20 lần)

1.1.3 min_ghost_distance & proximity_to_ghosts & isGreater4

```
proximity_to_ghosts = 0
min_ghost_distance = 0
isGreater4 = 1
for ghost in newGhostPositions:
    distance = manhattanDistance(newPos, ghost)
    if distance <= 4:
        isGreater4 = 0
    if min_ghost_distance == 0 or min_ghost_distance > distance:
        min_ghost_distance = distance
    if distance == 0:
        proximity_to_ghosts += 1
```

min_ghost_distance: Ghost có khoảng cách gần nhất càng gần Pacman thì Pacman càng cần phải tránh xa nó

proximity_to_ghosts: Nếu trạng thái đó chạm vào Ghost thì Pacman tránh action đó.

isGreater4: Nếu như Pacman đều cách Ghost > 4 đơn vị (manhattan) thì giá trị trả về càng lớn khi Pacman ăn các chấm thức ăn. Ngược lại thì Pacman cần phải tránh xa Ghosts hơn.

```
if isGreater4:
    remaining_food_weight = -40
else:
    ghost_weight = 10
```

1.1.4 totalCapsule

Mục đích: Khuyến khích Pacman ăn Capsule để Ghost biến thành trạng thái scare. Nhưng nếu như số lượng ghosts scare > 0 thì không nên ăn tiếp Capsule để có thể ăn nhiều ghost scare nhất có thể.

```
if len(ghostScare) > 0:
    capsule_weight = 150
```

2 Kết quả thực nghiệm

Tiến hành thực nghiệm với các tham số:

python pacman.py -l<layout> -p<algo_func> -depth=3,evalFn=<eval_func> -frameTime 0

Map	Minimax			AlphaBeta			Expectimax			Evaluation function
	Average score	Average runtime	Win rate	Average score	Average runtime	Win rate	Average score	Average runtime	Win rate	
contestClassic	1926.0	51.05	3/5	1905.0	42.494	2/5	3066.6	43.576	5/5	betterEvaluation
	-81.2	65.134	0/5	-176.2	17.976	0/5	135.8	82.438	0/5	scoreEvaluation
mediumClassic	1595.0	23.088	5/5	1567.6	16.89	4/5	1679	24.058	5/5	betterEvaluation
	407.2	96.28	2/5	491.6	26.352	1/5	-192.2	181.71	2/5	scoreEvaluation
originalClassic	1890.2	638.608	3/5	1968.4	272.074	3/5	1651.8	576.4	2/5	betterEvaluation
	NULL	NULL	NULL	NULL	NULL	NULL	395.4	343.65	0/5	scoreEvaluation
openClassic	1266.8	60.978	5/5	1209.0	34.602	5/5	1338	54.754	5/5	betterEvaluation
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	scoreEvaluation
powerClassic	1984.6	272.52	4/5	2978.4	102.336	5/5	2132.2	300.402	4/5	betterEvaluation
	NULL	NULL	NULL	1241.4	500.704	1/5	NULL	NULL	NULL	scoreEvaluation

Bảng 1: Bảng thống kê kết quả thực nghiệm

Chú thích:

NULL: game chạy > 5p chưa hoàn thành

3 Kết luận

3.1 Độ hiệu quả của hai hàm đánh giá

Hàm **betterEvaluation** đạt độ hiệu quả vượt trội hơn hẳn. Tỷ lệ thắng cao hơn.

3.2 Độ hiệu quả của 3 thuật toán

Minimax & AlphaBeta có tỉ lệ thắng, điểm số khá tương đương nhau. Tuy nhiên thời gian thực thi AlphaBeta nhanh hơn.

Expectimax đạt được điểm số, thời gian thực thi vượt trội hơn hẳn 2 thuật toán kia.

3.3 Trường hợp tốt nhất

Sử dụng map contestClassic + ExpectimaxAgent + depth = 3
=> đạt được điểm số: **3600**