

# BÁO CÁO BÀI TẬP 3

## Evaluation functions for Minimax/AlphaBeta/Expectimax

Tên: Hoàng Đình Hữu

MSSV: 20521384

Lớp: CS106.N21.KHCL

Giảng viên: Lương Ngọc Hoàng



### **I. Ý tưởng thiết kế evaluation function:**

➤ Evaluation function sẽ sử dụng các thông số như sau:

Đặc trưng	Định nghĩa
gscore	Điểm thưởng của ma
fdist	Khoảng cách từ pacman đến các chấm thức ăn
nfood	Số lượng chấm thức ăn
bonus	Điểm thưởng
duration	Chu kì khi ăn được điểm chấm lớn
rs	Giá trị của trạng thái

Ngoài ra còn có các thông số khác như nghost, gdist, nearestGdist, nearestFdist, ncapsule và cdist (được định nghĩa đầy đủ trong comment line của source code)

➤ Ý tưởng về cách tính giá trị của trạng thái:

- Đầu tiên, kiểm tra trạng thái của game nếu thắng trả về dương vô cùng nếu thua trả về âm vô cùng (vô cùng cho xấp xỉ 999999).
- Thứ hai, tính các thông số cơ bản như khoảng cách(manhattan) từ pacman đến tất cả các chấm thức ăn, con ma và điểm chấm lớn,

trong quá trình tính thì tìm ra được khoảng cách gần nhất tới chấm thức ăn, ma và điểm chấm lớn.

- Tính điểm bonus:
  - o Dựa trên khoảng cách tới con ma gần nhất nếu khoảng cách trong khoảng  $[0;3]$  thì  $\text{bonus} = -1200/\text{nearestGdist}$ . Bởi vì khi càng gần ma thì sẽ tự động tìm hướng đi để tránh đụng phải ma. Ngược lại nếu nằm ngoài 3 ô thì sẽ tăng điểm bonus vì ma đang ở xa nên có thể tự do chọn hướng đi.
  - o Dựa trên số lượng chấm thức ăn và khoảng cách tới các chấm thức ăn, đặt điều kiện nếu số lượng thức ăn còn ít hơn 4 và khoảng cách tới các chấm thức ăn lớn hơn 0 thì điểm bonus sẽ tăng ( $\text{bonus} += 5000/\text{fdist}$ ). Vì hướng đi của pacman phụ thuộc khá nhiều vào tọa độ của ma nên chủ yếu sẽ tránh né ma cộng thêm nếu khoảng cách tới các chấm thức ăn còn lại quá xa thì sẽ gây ra tình trạng chạy vô hạn không thể kết thúc trò chơi.
- Tính điểm của ma, nếu có ma thì tính bằng khoảng cách tới các con ma chia cho bình phương số lượng ma. Điểm của ma càng cao thì có nghĩa là khoảng cách trung bình của các con ma tới pacman càng lớn ( $\text{gscore} = \text{gdist}/(\text{nghost})^{**2}$ )
- Tính chu kì nếu pacman ăn được điểm chấm lớn, được tính dựa trên 2 lần tổng thời gian hoảng sợ của các con ma, nếu không con ma nào hoảng sợ thì tính theo khoảng cách tới các điểm chấm lớn ( $\text{duration} = 10 * \text{cdist}$ )
- Cuối cùng, tính giá trị trạng thái dựa trên thứ tự ưu tiên là tránh ma( $\text{gscore}$ ) -> ăn thức ăn -> ăn điểm chấm lớn -> ăn ma.

```
rs = gscore - 7*fdist/nfood + 10*bonus + currentGameState.getScore() - duration - 5000*nfood
```

Note: các trọng số trong bài là do em tự thiết kế dựa trên mức độ quan trọng của từng đặc trưng.

## II. Kết quả chạy thực nghiệm các thuật toán đã cài đặt với evaluation function:

- Chọn 5 map: Capsule classic, Contest classic, Medium classic, Minimax classic, Test classic.
- 2 hàm đánh giá: scoreEvaluationFunction (SEF) và betterEvaluationFunction (BEF).
- Map Capsule classic:

Thuật toán	Evaluation function	Lần 1	Lần 2	Lần 3	Lần 4	Lần 5	Win-rate
Minimax	SEF	-463	-438	-430	-478	-477	0%
	BEF	1236	972	-625	-434	877	60%
AlphaBeta	SEF	80	-279	-367	-156	-130	0%
	BEF	-446	-481	793	815	-255	40%
Expectimax	SEF	-427	56	146	558	157	0%
	BEF	-179	-450	53	-452	1197	25%

- Map Contest classic:

Thuật toán	Evaluation function	Lần 1	Lần 2	Lần 3	Lần 4	Lần 5	Win-rate
Minimax	SEF	-76	350	377	-164	-202	0%
	BEF	2210	1731	355	1607	2056	60%
AlphaBeta	SEF	-89	-360	-250	-256	-252	0%
	BEF	2398	1556	2400	1113	1143	100%
Expectimax	SEF	216	1094	563	-202	235	0%
	BEF	1628	1848	1811	1621	2278	100%

- Map Medium classic:

Thuật toán	Evaluation function	Lần 1	Lần 2	Lần 3	Lần 4	Lần 5	Win-rate
Minimax	SEF	-883	1284	-3308	152	-589	40%
	BEF	1735	1390	1530	1806	1603	100%
AlphaBeta	SEF	-28	69	-29	129	-340	0%
	BEF	1783	1510	1906	1890	1793	100%
Expectimax	SEF	-361	1860	-1100	-956	-136	20%
	BEF	1690	868	1782	1488	1387	80%

- Map Minimax classic:

Thuật toán	Evaluation function	Lần 1	Lần 2	Lần 3	Lần 4	Lần 5	Win-rate
Minimax	SEF	-492	514	511	-496	511	60%
	BEF	-492	-492	-492	-492	-492	0%
AlphaBeta	SEF	511	-492	-495	512	-495	40%
	BEF	-492	510	509	512	510	80%
Expectimax	SEF	511	-501	511	-505	507	60%
	BEF	-505	-494	515	515	508	60%

- Map Test classic:

Thuật toán	Evaluation function	Lần 1	Lần 2	Lần 3	Lần 4	Lần 5	Win-rate
Minimax	SEF	516	550	544	538	528	100%
	BEF	562	492	538	564	484	100%
AlphaBeta	SEF	490	516	516	536	520	100%
	BEF	536	564	564	554	366	100%
Expectimax	SEF	528	532	502	522	560	100%
	BEF	536	484	468	522	522	100%

- Bảng giá trị trung bình và độ lệch chuẩn:

Map	Evaluation Function	Minimax	AlphaBeta	Expectimax
Capsule Classic	SEF	$-457.2 \pm 22.17$	$-170.4 \pm 169.55$	$98 \pm 351.7$
	BEF	$405.2 \pm 865.97$	$85.2 \pm 661.84$	$33.8 \pm 683.51$
Contest Classic	SEF	$57 \pm 283.66$	$-241.4 \pm 97.07$	$381.2 \pm 482.16$
	BEF	$1591.8 \pm 732.62$	$1722 \pm 642.33$	$1837.2 \pm 267.21$
Medium Classic	SEF	$-688.8 \pm 1695.93$	$-39.8 \pm 180.77$	$-138.6 \pm 1187.04$
	BEF	$1612.8 \pm 164.96$	$1776.4 \pm 158.89$	$1443 \pm 357.67$
Minimax Classic	SEF	$109.6 \pm 551.01$	$-91.8 \pm 550.74$	$104.6 \pm 554.66$
	BEF	$-492 \pm 0$	$309.8 \pm 448.22$	$107.8 \pm 554.4$
Test Classic	SEF	$535.2 \pm 13.46$	$515.6 \pm 16.51$	$528.8 \pm 20.9$
	BEF	$528 \pm 38.03$	$516.8 \pm 85.07$	$506.4 \pm 28.89$

- Nhận xét:

- Thuật toán cho ra kết quả tốt nhất là Expectimax, tỉ lệ thắng khá cao (trừ 2 map đầu tiên với SEF).
- Hàm BEF hoạt động tốt hơn SEF ở hầu hết các màn chơi (win-rate của BEF khá cao so với SEF), đối với map 5 tỉ lệ thắng của map này 100% nên ta có thể dễ dàng so sánh điểm số cao nhất mà 2 hàm giá trị này có thể đạt được.
  - Minimax: BEF > SEF (562>550)
  - AlphaBeta: BEF > SEF (564>536)
  - Expectimax: BEF < SEF (536<560)

Tuy nhiên, mọi màn chơi đều cho điểm khác nhau và ngẫu nhiên nên chưa chắc chắn được.

Link màn chơi chạy khá tốt:

<https://drive.google.com/file/d/1MH3JvwLKvc5s41ulZJcxm2JrTUU4lwJ9/view?usp=sharing>