

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

---



**BT4 - Minimax/AlphaBeta/Expectimax**  
**Trí tuệ nhân tạo**

---

**Sinh viên: Đỗ Phương Duy - 23520362**

Ngày ... tháng ... năm ...



## Mục lục

<b>1</b>	<b>Link Video Demo:</b>	<b>3</b>
<b>2</b>	<b>Thiết kế hàm betterEvaluationFunction:</b>	<b>3</b>
2.1	Code: . . . . .	3
2.2	Ý tưởng và chi tiết từng phần: . . . . .	4
2.2.1	Tổng quan ý tưởng: . . . . .	4
2.2.2	Chi tiết từng phần: . . . . .	4
2.2.3	Ưu điểm: . . . . .	5
<b>3</b>	<b>Kết quả thực nghiệm:</b>	<b>6</b>
<b>4</b>	<b>So sánh hiệu quả 2 hàm đánh giá:</b>	<b>8</b>
<b>5</b>	<b>So sánh hiệu quả Minimax AlphaBeta và Expectimax</b>	<b>8</b>
5.1	MiniMax: . . . . .	8
5.2	AlphaBeta: . . . . .	8
5.3	Expectimax: . . . . .	9



## 1 Link Video Demo:

[https://drive.google.com/file/d/162FPRzORE4\\_CN2xxDsXMymCeX5oWBHgH/view?usp=sharing](https://drive.google.com/file/d/162FPRzORE4_CN2xxDsXMymCeX5oWBHgH/view?usp=sharing)

## 2 Thiết kế hàm betterEvaluationFunction:

### 2.1 Code:

```
1 def betterEvaluationFunction(currentGameState):
2     """
3     Your extreme ghost-hunting, pellet-nabbing, food-gobbling, unstoppable
4     evaluation function (question 5).
5
6     DESCRIPTION: <write something here so we know what you did>
7     """
8     "*** YOUR CODE HERE ***"
9     #util.raiseNotDefined()
10    # Thông tin từ trạng thái hiện tại
11    pacmanPos = currentGameState.getPacmanPosition()
12    foodList = currentGameState.getFood().asList()
13    ghostStates = currentGameState.getGhostStates()
14    scaredTimes = [ghost.scaredTimer for ghost in ghostStates]
15    capsules = currentGameState.getCapsules()
16
17    # Trường hợp kết thúc game
18    if currentGameState.isWin():
19        return float('inf')
20    if currentGameState.isLose():
21        return float('-inf')
22
23    # ----- Phần đánh giá điểm -----
24
25    score = currentGameState.getScore()
26
27    # --- Thức ăn ---
28    if foodList:
29        minFoodDist = min(manhattanDistance(pacmanPos, food) for food in
30                           ↪ foodList)
31        foodScore = -1.5 * minFoodDist - 4 * len(foodList)
32    else:
33        foodScore = 0
34
35    # --- Viên năng lượng ---
36    if capsules:
37        minCapDist = min(manhattanDistance(pacmanPos, cap) for cap in capsules)
38        capsuleScore = -2 * minCapDist - 20 * len(capsules)
39    else:
```



```
39         capsuleScore = 0
40
41     # --- Ma ---
42     ghostScore = 0
43     for ghost, scaredTime in zip(ghostStates, scaredTimes):
44         dist = manhattanDistance(pacmanPos, ghost.getPosition())
45         if scaredTime > 0:
46             # Nếu ma đang sợ, khuyến khích lại gần để ăn
47             ghostScore += 10 / (dist + 1)
48         else:
49             # Nếu ma bình thường, tránh xa
50             if dist < 2:
51                 ghostScore -= 200 # quá gần => nguy hiểm
52             else:
53                 ghostScore -= 1.5 / dist
54
55     # --- Trường hợp đứng yên ---
56     def isStop():
57         if pacmanPos == currentGameState.getPacmanPosition():
58             return float(-9999)
59
60     # --- Tổng điểm ---
61     finalScore = score + foodScore + capsuleScore + ghostScore + isStop()
62     return finalScore
```

---

## 2.2 Ý tưởng và chi tiết từng phần:

### 2.2.1 Tổng quan ý tưởng:

Mục tiêu của hàm là ước lượng "giá trị" của một trạng thái trò chơi để Pacman có thể chọn hành động tốt nhất trong tương lai. Hàm này tích hợp:

- Khoảng cách đến thức ăn
- Khoảng cách đến viên năng lượng (capsules)
- Tình trạng của ma (ghost) – sợ hay không
- Hình phạt khi Pacman quá gần ma
- Phần thưởng cho việc ăn ma sợ
- Tránh đứng yên

### 2.2.2 Chi tiết từng phần:

#### Khoảng cách đến thức ăn

- Nếu Pacman thắng, trạng thái này là tối ưu.
- Nếu thua, trả về âm vô cực.



### Khoảng cách đến viên năng lượng (capsules)

- minFoodDist: Ưu tiên những trạng thái mà Pacman gần viên thức ăn nhất.
- len(foodList): Hình phạt cho mỗi thức ăn còn lại.
- Kết hợp khuyến khích vừa ăn nhanh vừa đi đúng hướng có thức ăn gần.

### Tình trạng của ma (ghost) – sợ hay không

- Tương tự như thức ăn, nhưng trừ điểm nặng hơn.
- Vì capsules rất quan trọng để chuyển ghost sang trạng thái sợ nên ưu tiên mạnh.

### Phần thưởng ăn ma sợ và hình phạt khi Pacman quá gần ma

- Nếu ghost đang sợ, điểm được cộng khi đến gần → khuyến khích ăn ghost.
- Nếu ghost không sợ:
  - Trừ điểm nặng nếu quá gần (tránh chết).
  - Trừ nhẹ nếu ở xa → khuyến khích giữ khoảng cách.

### Tránh đứng yên

- Nếu Pacman không di chuyển thì phạt rất nặng vì việc không di chuyển nó kéo dài game khiến game rất lâu mới kết thúc cũng như bị trừ mất rất nhiều điểm trong thời gian không di chuyển

### 2.2.3 Ưu điểm:

Đa yếu tố, phản ánh được nhiều chiến lược: Ăn thức ăn, ưu tiên capsule, né ma thường, săn ma sợ.

Khuyến khích hành vi chủ động, né tránh nguy hiểm.



### 3 Kết quả thực nghiệm:

#### Medium Classic

Random seed	ScoreEvaluation			BetterEvaluation		
	Score	Result	Runtime	Score	Result	Runtime
<b>MiniMax</b>						
23520362	-1116	Win	113.00	1997	Win	26.00
23520363	-1372	Win	119.00	2005	Win	26.00
23520364	209	Win	19.00	686	Win	35.00
23520365	-474	Win	77.00	2091	Win	20.00
23520366	1611	Win	41.00	1907	Win	18.00
<b>AlphaBeta</b>						
23520362	-1116	Win	130.00	1997	Win	20.00
23520363	-1372	Win	131.00	2005	Win	21.00
23520364	209	Win	19.00	686	Win	28.00
23520365	-474	Win	94.00	2091	Win	15.00
23520366	1611	Win	46.00	1907	Win	14.00
<b>Expectimax</b>						
23520362	-1118	Win	122.00	1997	Win	26.00
23520363	-4784	Win	444.00	2005	Win	29.00
23520364	481	Win	44.00	1683	Win	36.00
23520365	-824	Win	126.00	1986	Win	31.00
23520366	1612	Win	52.00	797	Win	20.00

#### Small Classic

Random seed	ScoreEvaluation			BetterEvaluation		
	Score	Result	Runtime	Score	Result	Runtime
<b>MiniMax</b>						
23520362	-115	Win	24.00	1736	Win	14.00
23520363	221	Win	14.00	1681	Win	21.00
23520364	-257	Win	39.00	1667	Win	20.00
23520365	-64	Win	32.00	1284	Win	19.00
23520366	-495	Win	16.00	1316	Win	16.00
<b>AlphaBeta</b>						
23520362	-115	Win	29.00	1736	Win	11.00
23520363	221	Win	12.00	1681	Win	18.00
23520364	-257	Win	41.00	1667	Win	16.00
23520365	-64	Win	31.00	1284	Win	14.00
23520366	-495	Win	14.00	1316	Win	13.00
<b>Expectimax</b>						
23520362	1366	Win	10.00	1543	Win	15.00
23520363	218	Win	15.00	1038	Win	25.00
23520364	-284	Win	50.00	1432	Win	26.00
23520365	-119	Win	44.00	1282	Win	17.00
23520366	-1116	Win	93.00	1304	Win	16.00



## Capsule Classic

Random seed	ScoreEvaluation			BetterEvaluation		
	Score	Result	Runtime	Score	Result	Runtime
<b>MiniMax</b>						
23520362	-477	Win	2.00	-433	Win	3.00
23520363	-481	Win	8.00	-136	Win	9.00
23520364	-429	Win	3.00	-437	Win	4.00
23520365	-438	Win	3.00	-438	Win	2.00
23520366	-463	Win	6.00	-482	Win	8.00
<b>AlphaBeta</b>						
23520362	-477	Win	3.00	-433	Win	2.50
23520363	-481	Win	7.00	-136	Win	8.00
23520364	-429	Win	2.00	-437	Win	3.00
23520365	-438	Win	3.00	-438	Win	1.50
23520366	-463	Win	5.00	-482	Win	7.00
<b>Expectimax</b>						
23520362	-478	Win	2.00	-434	Win	4.00
23520363	-481	Win	12.00	-136	Win	9.00
23520364	-429	Win	3.00	-438	Win	3.00
23520365	-463	Win	5.00	-283	Win	28.00
23520366	-463	Win	8.00	-482	Win	9.00

## Contest Classic

Random seed	ScoreEvaluation			BetterEvaluation		
	Score	Result	Runtime	Score	Result	Runtime
<b>MiniMax</b>						
23520362	-217	Win	9.00	2171	Win	14.00
23520363	-289	Win	25.00	1933	Win	21.00
23520364	-240	Win	55.00	1397	Win	20.00
23520365	888	Win	62.00	2588	Win	19.00
23520366	-194	Win	18.00	931	Win	16.00
<b>AlphaBeta</b>						
23520362	-217	Win	7.00	2171	Win	12.00
23520363	-289	Win	22.00	1933	Win	19.00
23520364	-240	Win	41.00	1397	Win	19.00
23520365	888	Win	52.00	2588	Win	19.00
23520366	-194	Win	6.00	931	Win	16.00
<b>Expectimax</b>						
23520362	545	Win	38.00	1543	Win	15.00
23520363	531	Win	100.00	1038	Win	25.00
23520364	-33	Win	17.00	1432	Win	26.00
23520365	1195	Win	29.00	1282	Win	17.00
23520366	-194	Win	14.00	1304	Win	16.00



## Open Classic

Random seed	ScoreEvaluation			BetterEvaluation		
	Score	Result	Runtime	Score	Result	Runtime
<b>MiniMax</b>						
23520362	-	Unknown	3000.00	-	Unknown	3000.00
23520363	-	Unknown	3000.00	-	Unknown	3000.00
23520364	-	Unknown	3000.00	-	Unknown	3000.00
23520365	-	Unknown	3000.00	-	Unknown	3000.00
23520366	-	Unknown	3000.00	-	Unknown	3000.00
<b>AlphaBeta</b>						
23520362	-	Unknown	3000.00	-	Unknown	3000.00
23520363	-	Unknown	3000.00	-	Unknown	3000.00
23520364	-	Unknown	3000.00	-	Unknown	3000.00
23520365	-	Unknown	3000.00	-	Unknown	3000.00
23520366	-	Unknown	3000.00	-	Unknown	3000.00
<b>Expectimax</b>						
23520362	-	Unknown	3000.00	1349	Win	58 .00
23520363	-	Unknown	3000.00	-	Unknown	3000.00
23520364	-	Unknown	3000.00	-	Unknown	3000.00
23520365	-	Unknown	3000.00	-	Unknown	3000.00
23520366	-	Unknown	3000.00	-	Unknown	3000.00

## 4 So sánh hiệu quả 2 hàm đánh giá:

Nhìn chung hàm betterEvaluation cải tiến hơn, hiệu năng tốt hơn hàm evaluation có sẵn khi tỷ lệ thắng cao hơn đáng kể cũng như thời gian chạy tốt hơn.

Đặc biệt ở Seed 23520362, hàm betterEvaluation thắng được map Open Classic với thời gian nhanh điều mà những agent và hàm đánh giá còn lại không làm được trong 3000s.

## 5 So sánh hiệu quả Minimax AlphaBeta và Expectimax

### 5.1 MiniMax:

- **Score** của MiniMax có thể khá thấp, cho thấy rằng thuật toán này không luôn tìm ra các kết quả tối ưu so với các thuật toán khác.
- **Runtime** của MiniMax có thể dài hơn do không có cắt tỉa (so với AlphaBeta), đặc biệt khi không gian trạng thái lớn.

### 5.2 AlphaBeta:

- **Score** của AlphaBeta rất gần với MiniMax, nhưng **Runtime** thấp hơn đáng kể, điều này chứng tỏ AlphaBeta hiệu quả hơn trong việc tối ưu hóa tìm kiếm bằng cách cắt tỉa nhánh không cần thiết.
- Với AlphaBeta, bạn có thể thấy thời gian tính toán trung bình thấp hơn





### 5.3 Expectimax:

- **Score** của Expectimax không phải lúc nào cũng là tối ưu nhất so với MiniMax hoặc AlphaBeta (nhưng vẫn có kết quả tốt trong những trường hợp ngẫu nhiên).
- **Runtime** của Expectimax có thể dài hơn, đặc biệt là trong các trường hợp có yếu tố ngẫu nhiên, vì thuật toán phải tính toán kỳ vọng thay vì chỉ tối đa hóa hay tối thiểu hóa.

#### Kết luận:

- AlphaBeta là lựa chọn tối ưu nhất về mặt **hiệu suất** khi cần cắt tỉa và giảm **Runtime** trong các trò chơi có đối thủ đối kháng.
- MiniMax có thể tốn thời gian hơn nhưng vẫn thích hợp trong các trường hợp không có yếu tố ngẫu nhiên.
- Expectimax hoạt động tốt trong môi trường có yếu tố ngẫu nhiên nhưng sẽ có **Runtime** lâu hơn so với hai thuật toán còn lại.