

**Tên:** Châu Tấn

**MSSV:** 20520926

**Lớp:** CS106.M21 (Môn: Trí tuệ nhân tạo)

**GVHD:** TS. 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 em tự thiết kế evaluation function. Những đặc trưng em 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 em 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).

#### Nội dung:

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

Mục tiêu của việc thiết kế Evaluation Function là làm sao cho Pacman ăn hết các chấm thức ăn để kết thúc trò chơi và đạt được điểm cao nhất có thể.

Hàm lượng giá do bản thân thiết kế

$$\text{score} = 2 * \text{currentGameState.getScore()} - (\text{len}(\text{foodList})) * 2 - 3 * \text{foodDistanceEuclid} - 3 * (\text{closestCapsule}) - 2 * \text{effectGhostDistance}$$

Các tham số trong hàm lượng giá

`currentGameState.getScore()`: Hàm chọn các state để tối ưu điểm số ở thời điểm của trạng thái game hiện tại.

`len(foodList)`: Đặc trưng lưu trữ số lượng của các chấm thức ăn còn lại.

`foodDistanceEuclid`: Đặc trưng chỉ hướng cho pacman về chấm thức ăn gần nhất dựa theo khoảng cách Euclid.

closestCapsule: Đặc trưng chỉ hướng cho pacman đến chấm Capsule gần nhất.

(Capsule là chấm to hơn chấm thức ăn)

effectGhostDistance: Sau khi tiêu thụ chấm Capsule thì ma sẽ bị nhận một hiệu ứng đặc biệt trong 40s. Pacman sẽ ăn được ma trong thời gian đó và có thêm 200 điểm. Đặc trưng này sẽ khuyến khích pacman ăn ma thay vì chạy trốn ma ở thời điểm sau khi ăn Capsule.

### Chiến thuật thiết kế hàm Evaluation function:

Mỗi hành động của Pacman thực hiện đều có sự ảnh hưởng đến điểm số của hàm lượng giá và điểm số tổng thể. Do đó ta cần thiết kế trong hàm lượng giá có các tham số của các sự kiện mà các hành động Pacman sẽ có thể tạo ra.

Có rất nhiều sự kiện mà Pacman có thể gặp phải trong quá trình Pacman di chuyển. Tuy nhiên, những sự kiện đó có thể gây xung đột lẫn nhau dẫn đến làm giảm kết quả của Pacman. Do đó phải chọn những sự kiện và thông số hợp lý để đưa vào hàm lượng giá. Việc chọn các trọng số là kết quả của việc chạy thực nghiệm nhiều lần với nhiều trọng số khác nhau. Dẫu biết trọng số vẫn chưa đem lại kết quả tốt nhưng tham số hiện tại là tham số tốt nhất trong những lần thử nghiệm.

Do bản thân em có phần hơi ưu tiên đến chiến thắng của ván chơi nên em ưu tiên cho đặc trưng len(foodList) có hàm bậc 2. Lý do là em thử nghiệm với nhiều trọng số cho len(foodList) và nhìn thấy kết quả có vẻ không được tốt ở các map. Ứng với các map thì số chấm thức ăn sẽ khác nhau nếu đặt một tham số cố định thì kết quả sẽ không được khách quan, do đó em chọn trọng số của len(foodList) là chính nó ( $\text{len}(\text{foodList}) * \text{len}(\text{foodList})$ )

### Nhận xét về hàm lượng giá này

Hàm lượng giá này chưa thể đem lại kết quả tốt hoàn toàn ở mọi layout. Tại vì mỗi layout thì lời giải, số chấm thức ăn, số lượng ma, diện tích bản đồ,... sẽ khác nhau. Một hàm lượng giá có thể chạy tốt ở layout này nhưng cũng có thể chạy khá tệ ở layout khác. Nhưng đây là hàm lượng giá cho ra kết quả mà em hài lòng nhất trong tất cả các thử nghiệm trước đó.

- 2. Chạy thử nghiệm các thuật toán đã cài đặt với evaluation function em 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).**

Để cho việc thống kê kết quả của các thuật toán thuận lợi hơn thì em có viết script AutomaticCMD.py để ghi vào file script.sh. Kết quả sau khi chạy file script.sh sẽ ra một file csv thống kê kết quả của các lần chạy.

### Chạy thử nghiệm các thuật toán đã cài đặt với evaluation function mặc định

Sau khi chạy thử nghiệm cả 3 thuật toán với hàm lượng giá mặc định trong source code. Sau đó em tổng kê lại được bảng sau.

<b>Map</b>	<b>Avg Score</b>	<b>Scores</b>	<b>Win</b>	<b>Times</b>	<b>Algorithhm</b>	<b>Evaluation</b>
capsuleClassic	-281	[-281.0]	0	1	MinimaxAgent	scoresEvaluation Function
	-408	[-408.0]	0	1	AlphaBetaAgent	
	-250	[-250.0]	0	1	ExpectimaxAgent	
contestClassic	154	[154.0]	0	1	MinimaxAgent	
	-299	[-299.0]	0	1	AlphaBetaAgent	
	799	[799.0]	0	1	ExpectimaxAgent	
mediumClassic	-794	[-794.0]	0	1	MinimaxAgent	
	507	[507.0]	1	1	AlphaBetaAgent	
	510	[510.0]	1	1	ExpectimaxAgent	
minimaxClassic	-497	[-497.0]	0	1	MinimaxAgent	
	-409	[-409.0]	0	1	AlphaBetaAgent	
	-380	[-380.0]	0	1	ExpectimaxAgent	
openClassic	< -16000	< -16000	0	1	MinimaxAgent	
	< -16000	< -16000	0	1	AlphaBetaAgent	
	< -16000	< -16000	0	1	ExpectimaxAgent	
originalClassic	476	[476.0]	0	1	MinimaxAgent	
	859	[859.0]	0	1	AlphaBetaAgent	
	172	[172.0]	0	1	ExpectimaxAgent	
powerClassic	-58	[-58.0]	0	1	MinimaxAgent	
	1308	[1308.0]	0	1	AlphaBetaAgent	
	761	[761.0]	0	1	ExpectimaxAgent	
smallClassic	-370	[-370.0]	0	1	MinimaxAgent	
	-380	[-380.0]	0	1	AlphaBetaAgent	
	-372	[-372.0]	0	1	ExpectimaxAgent	
testClassic	544	[544.0]	1	1	MinimaxAgent	
	530	[530.0]	1	1	AlphaBetaAgent	
	468	[468.0]	1	1	ExpectimaxAgent	
trappedClassic	-502	[-502.0]	0	1	MinimaxAgent	
	-502	[-502.0]	0	1	AlphaBetaAgent	
	532	[532.0]	1	1	ExpectimaxAgent	
trickyClassic	-69	[-69.0]	0	1	MinimaxAgent	
	1488	[1488.0]	0	1	AlphaBetaAgent	
	399	[399.0]	0	1	ExpectimaxAgent	

Chạy thực nghiệm thuật toán đã cài đặt với Evaluation Function do em thiết kế  
Sau khi chạy thử nghiệm cả 3 thuật toán với hàm lượng giá mặc định trong source code. Sau đó em tổng kê lại được bảng sau.

<b>Map</b>	<b>Avg Score</b>	<b>Scores</b>	<b>Win</b>	<b>Times</b>	<b>Algorithhm</b>	<b>Evaluation</b>
capsuleClassic	-231	[-231.0]	0	1	MinimaxAgent	
	-748	[-748.0]	0	1	AlphaBetaAgent	
	-581	[-581.0]	0	1	ExpectimaxAgent	
contestClassic	-433	[-433.0]	0	1	MinimaxAgent	
	-309	[-309.0]	0	1	AlphaBetaAgent	
	-396	[-396.0]	0	1	ExpectimaxAgent	
mediumClassic	-440	[-440.0]	0	1	MinimaxAgent	
	-512	[-512.0]	0	1	AlphaBetaAgent	

	-415	[-415.0]	0	1	ExpectimaxAgent	<b>betterEvaluation Function</b>
<i>minimaxClassic</i>	516	[516.0]	1	1	MinimaxAgent	
	-493	[-493.0]	0	1	AlphaBetaAgent	
	516	[516.0]	1	1	ExpectimaxAgent	
<i>openClassic</i>	1238	[1238.0]	1	1	MinimaxAgent	
	1434	[1434.0]	1	1	AlphaBetaAgent	
	-97	[-97.0]	0	1	ExpectimaxAgent	
<i>originalClassic</i>	-548	[-548.0]	0	1	MinimaxAgent	
	-370	[-370.0]	0	1	AlphaBetaAgent	
	-117	[-117.0]	0	1	ExpectimaxAgent	
<i>powerClassic</i>	-25	[-25.0]	0	1	MinimaxAgent	
	1878	[1878.0]	0	1	AlphaBetaAgent	
	-446	[-446.0]	0	1	ExpectimaxAgent	
<i>smallClassic</i>	-289	[-289.0]	0	1	MinimaxAgent	
	-353	[-353.0]	0	1	AlphaBetaAgent	
	-419	[-419.0]	0	1	ExpectimaxAgent	
<i>testClassic</i>	564	[564.0]	1	1	MinimaxAgent	
	562	[562.0]	1	1	AlphaBetaAgent	
	564	[564.0]	1	1	ExpectimaxAgent	
<i>trappedClassic</i>	-502	[-502.0]	0	1	MinimaxAgent	
	532	[532.0]	1	1	AlphaBetaAgent	
	532	[532.0]	1	1	ExpectimaxAgent	
<i>trickyClassic</i>	-336	[-336.0]	0	1	MinimaxAgent	
	-283	[-283.0]	0	1	AlphaBetaAgent	
	-348	[-348.0]	0	1	ExpectimaxAgent	

### So sánh kết quả của betterEvaluationFunction với scoreEvaluationFunction có sẵn

Ở map *openClassic* khi chạy với scoreEvaluationFuction thì cả 3 thuật toán đều không thể cho ra kết quả hoặc rất lâu để cho ra kết quả (Nó lâu đến mức để máy cho nó chạy đến khi điểm nó bé hơn -16000 điểm vẫn chưa chạy ra kết quả). Trong khi đó map này khi chạy với betterEvaluationFuction thì lại cho ra kết quả rất nhanh.

Xét về mặt bằng chung thì số ván thắng khi chạy scoreEvaluationFuction thấp hơn so với betterEvaluationFuction. Thời gian để chạy scoreEvaluationFuction lại lâu hơn betterEvaluationFuction khá nhiều.

Nguyên nhân có thể là do hàm scoreEvaluationFuction chỉ có mỗi thuộc tính `currentGameState.getScore()` sẽ dẫn đến tình trạng Pacman không thể di chuyển nếu độ sâu thấp và các không gian trạng thái xung quanh không giúp gia tăng điểm số.

Xét mặt bằng chung thì điểm số của betterEvaluationFuction tốt hơn điểm số của scoreEvaluationFuction. Tuy nhiên không phải layout nào betterEvaluationFuction cũng chạy tốt hơn scoreEvaluationFuction.

### So sánh kết quả theo 3 thuật toán Minimax, Alpha-Beta Pruning và Expectimax.

Trong hầu hết mọi trường hợp thì Minimax và Alpha-Beta Pruning có phần sẽ chạy tốt hơn Expectimax. Nguyên nhân là do ở Expectimax thì việc phân chia xác suất sẽ không giúp pacman nhận biết được nguy hiểm và kết thúc game do ghost.

Tuy nhiên có một số trường hợp mà Expectimax có thể chiến thắng trò chơi hoặc kết thúc trò chơi với điểm số cao hơn 2 thuật toán còn lại. Điển hình là ở trappedClassic của scoreEvaluationFuncion.

Trong hai thuật toán Minimax và Alpha-Beta Pruning thì thuật toán Alpha-Beta Pruning chạy hiệu quả hơn Minimax một chút.

#### **NOTE:**

**Đường [link](#) dẫn tới video mà pacman chạy tốt nhất theo hàm của em.**

\_\_\_\_\_ KẾT THÚC \_\_\_\_\_