ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

****

BÁO CÁO

ASSIGNMENT 3 – EVALUATION FUNCTION FOR MINIMAX/ ALPHABETA/ EXPECTIMAX

Môn học: Trí tuệ nhân tạo

Mã lớp học: CS106.M21.KHTN

Giảng viên giảng dạy: Lương Ngọc Hoàng

Sinh viên thực hiện: Nguyễn Tư Thành Nhân

Mã số sinh viên: 20520079

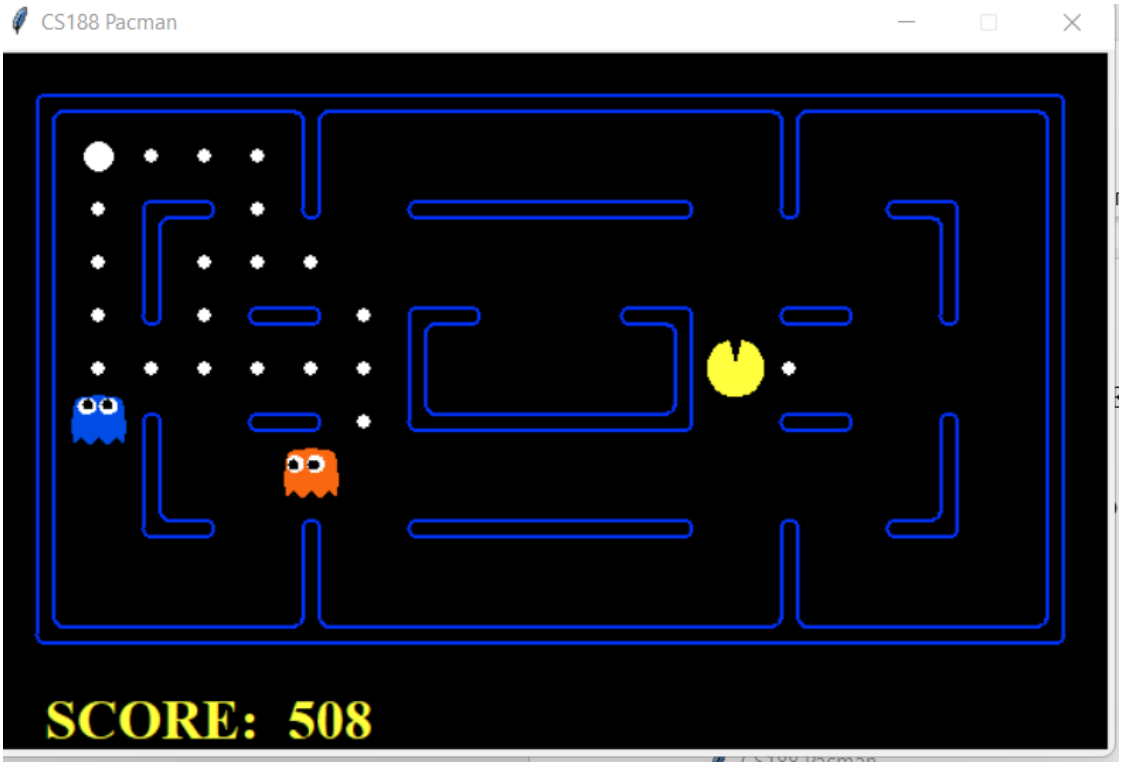
THÀNH PHỐ HỒ CHÍ MINH – 2022

MỤC LỤC

PHẦN 1
MÔ TẢ EVALUATION FUNCTION

Xem xét chạy thử với hàm đánh giá `betterEvaluationFunction` ta có nhận thấy một số nhược điểm như sau:

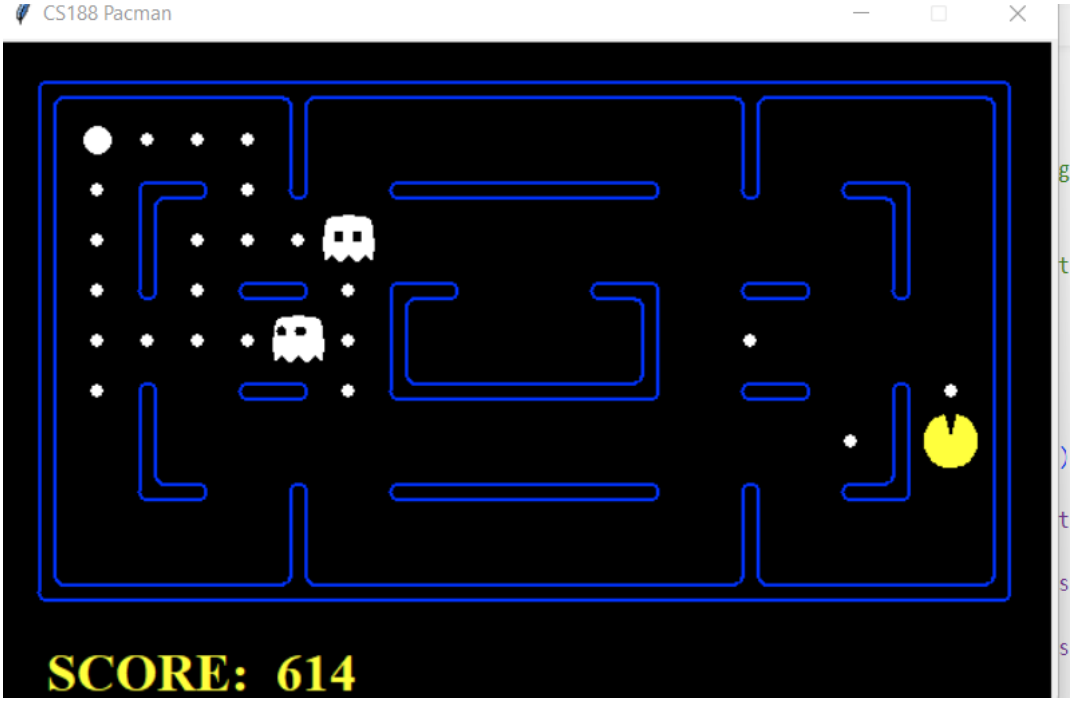
- **Nhược điểm thứ nhất,** khi đến trạng thái chỉ còn 1 chấm thức ăn tại khu vực lân cận của nó (khu vực này sẽ tùy thuộc vào `depth` mình setting) (xem minh họa tại *hình 1*) thì nó sẽ duy trì hành động **"`Stop`"** cho đến khi những con ma đến gần. Tức là phải do đặc trưng `ghost_distance` đủ lớn tác động vào thì con Pacman mới lựa chọn di chuyển. Điều này có thể giải thích là do đặc trưng `closestFood` là chủ yếu bởi vì nếu như Pacman lựa chọn ăn chấm thức ăn ở khu vực lân cận thì đặc trưng `closestFood` sẽ tăng lên đột ngột (trong khi đó trọng số của nó là -2) hoặc nếu nó di chuyển qua vị trí khác thì `closestFood` cũng sẽ tăng lên.



(Hình 1. Mô tả trạng thái chỉ còn 1 chấm thức ăn ở khu vực lân cận)

- **Nhược điểm thứ hai,** trong trạng thái vừa ăn xong chấm năng lượng (minh họa ở hình 2), thì Pacman sẽ không tận dụng được cơ hội ăn những con ma để lấy 500 điểm. Thay vì

đó, nó còn có thể rơi vào trạng thái đứng yên nếu những con ma này sẽ di chuyển đi xa ra so với Pacman và khu vực lân cận chỉ còn 1 chấm thức ăn (cái này đã giải thích vì sao tại ý trên)



(Hình 2. Mô tả trạng thái Pacman vừa mới ăn xong chấm thức ăn)

- **Nhược điểm thứ ba,** nó chưa thực sự quan trọng đến trạng thái thắng/ thua
- **Nhược điểm thứ tư,** là do độ đo khoảng cách chưa đủ tốt để ước lượng số bước đi tối thiểu giữa hai điểm.

Em sẽ giữ những đặc trưng `closestFood`, `ghost_distance` và `len(Foodlist)` và trọng số tương ứng ở hàm đánh giá `betterEvaluationFunction` (bản chất là để khắc **phục nhược điểm thứ hai** nhưng chưa đủ hiệu quả). Để khắc phục ba nhược điểm trên, trong hàm đánh giá `myEvaluationFunction` em đã đề xuất thêm ba đặc trưng là

- Sử dụng `close2ndFood` là khoảng cách đến chấm thức ăn gần nhất thứ 2 để khắc phục **nhược điểm thứ nhất**. Tuy nhiên, trong một số trường hợp đặc biệt nếu như trong vùng lân cận bao gồm hai chấm thức ăn có khoảng cách tỉ lệ nghịch với trọng số cho hai đặc trưng `closestFood`, `close2ndFood` thì nó sẽ giữ hành động đứng yên (xem ví dụ mô tả ở *hình 3*). Vì thế, chúng ta sẽ giữ trọng số `-2` cho `closestFood` và trọng số cho

`close2ndFood` sẽ là một số lẻ và nhỏ hơn `-2` (nhưng đủ có ý nghĩa đến kết quả của hàm đánh giá). Ở đây em sẽ để trọng số `-0.897` cho `closestFood`. Tất nhiên thêm điểm gần nhất thứ hai chưa đủ để giải quyết hết vấn đề này nhưng nó cũng đã có sự thay đổi đáng kể.

(Hình 3. Ví dụ mô tả trạng thái có hai chấm thức ăn ở khu vực lân cận và trọng số cho `closestFood = -2`*, trọng số cho* `closestFood = -1`*)*

- Đối với **nhược điểm thứ hai**, chúng ta sẽ sử dụng đặc trưng `can_eat_ghost` để khắc phục. Đặc trưng sẽ ước lượng một đại lượng nào đó để trong thời gian con ma đang sợ hãi thì mình có khả năng ăn nó nhất. Đồng thời, lợi dụng đặc điểm scare_timer của mỗi con ma sẽ được đưa từ một số tự nhiên về 0 thì sẽ làm tăng đột ngột giá trị `can_eat_ghost` này. Vì thế, em sẽ chọn trọng số là dương và đủ lớn (do giá trị mỗi con ma tương ứng lên đến 500). Ở đây em chọn là `10`
- **Nhược điểm thứ ba** thì em sẽ sử dụng thêm đặc trưng `score` với trọng số `= 1` vì nó giữ thái độ trung lập.

- Đối với **nhược điểm thứ tư**, thì chúng ta có thể thay thế độ đo khoảng cách Mahattan bằng số bước đi của thuật toán BFS hoặc UCS. Tuy nhiên, việc này sẽ tốn chi phí tính toán rất lớn. Nên chúng ta có thể chấp nhận tạm bỏ qua nhược điểm này.

PHẦN 2
CHẠY THỬ NGHIỆM

Lần lượt chạy thử nghiệm các thuật toán `Minimax`, `AlphaBeta`, `Expectimax` với các hàm đánh giá `scoreEvaluationFunction`, `betterEvaluationFunction` và `myEvaluationFunction` trên các map được cung cấp (không chạy trên 2 layout openClassic và trichkyClassic). Ta được kết quả điểm số là bảng thống kê như sau (trong đó flag `-f` và depth = 3 được sử dụng)

Layout	EvationFunction	Minimax	AlphaBeta	Expectimax
capsuleClassic	scoreEvaluationFunction	-477	-477	-478
	betterEvaluationFunction	-477	-477	-176
	myEvaluationFunction	-477	-477	-512
contestClassic	scoreEvaluationFunction	1299	1299	-70
	betterEvaluationFunction	30	30	103
	myEvaluationFunction	1461	1461	395
mediumClassic	scoreEvaluationFunction	-829	-829	-775
	betterEvaluationFunction	1118	1118	1107
	myEvaluationFunction	2060	2060	1902
minimaxClassic	scoreEvaluationFunction	-492	-492	511
	betterEvaluationFunction	-494	-494	508
	myEvaluationFunction	-492	-492	510
originalClassic	scoreEvaluationFunction	-107	-107	-147
	betterEvaluationFunction	320	320	2098
	myEvaluationFunction	2896	2896	288
powerClassic	scoreEvaluationFunction	614	614	2566
	betterEvaluationFunction	2342	2342	942
	myEvaluationFunction	1930	1930	1926
smallClassic	scoreEvaluationFunction	1530	1530	1530
	betterEvaluationFunction	972	972	985
	myEvaluationFunction	1360	1360	961

testClassic	scoreEvaluationFunction	528	528	528
	betterEvaluationFunction	561	561	561
	myEvaluationFunction	564	564	564
trappedClassic	scoreEvaluationFunction	-501	-501	532
	betterEvaluationFunction	532	532	532
	myEvaluationFunction	532	532	532

Những ô xanh lá là những ô cho kết quả thắng, những ô đỏ là những ô cho kết quả thua cuộc.

So sánh hiệu năng (efficiency) và hiệu quả (effectiveness) của 3 thuật toán Minimax, AlphaBeta, Expectimax:

- Về tính hiệu quả, do bản chất AlphaBeta là Minimax nhưng sử dụng thêm phương pháp cắt tỉa Alpha-Beta nên AlphaBeta và Minimax cho kết quả giống nhau. Tùy trường hợp cụ thể, thì có thể đánh giá hai thuật toán này có tốt hơn Expectimax hay không. Tuy nhiên, nếu như, có nhiều cơ hội để chạy random nhiều lần thì kết quả trung bình của Expectimax sẽ cho kết quả tốt hơn.
- Về tính hiệu năng, thời gian xử lí đối với mỗi trạng thái của thuật toán Minimax và Expectimax là tương đương nhau, do đó, nó tùy thuộc vào số lượng trạng thái mà mỗi thuật toán phải xử lí. Thời gian xử lí mỗi trạng thái của AlphaBeta sẽ nhanh hơn so với của Minimax và Expectimax, và số trạng thái của AlphaBeta và Minimax như nhau nên hiệu năng tổng thể của AlphaBeta sẽ hơn so với Minimax.

PHỤ LỤC

Source code được cung cấp kèm theo

Video record cho ván chơi đạt kết quả tốt nhất