

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA KHOA HỌC MÁY TÍNH  
BÀI TẬP TRÍ TUỆ NHÂN TẠO

## Assignment 3 - Evaluation functions for Minimax/AlphaBeta/Expectimax

Giáo viên hướng dẫn: Lương Ngọc Hoàng

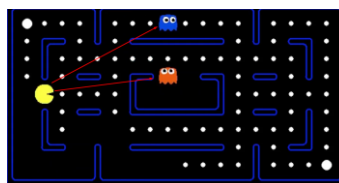
Sinh viên thực hiện : Trần Văn Lực MSSV: 20521587

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

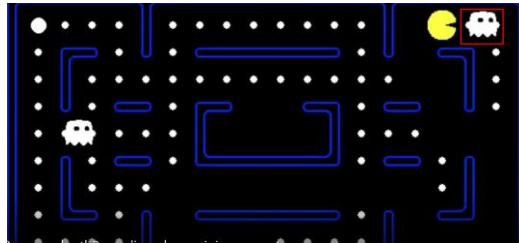
- ✚ Ý tưởng thiết kế hàm lượng giá (evaluation function) để ước lượng giá trị của các trạng thái trung gian:
- ❖ Điểm số của trò chơi là một đặc trưng để ước lượng giá trị của trạng thái hiện tại vì trạng thái mà có điểm số cao điểm trạng thái kết thúc sẽ cao hơn nhằm tối ưu số điểm khi kết thúc và trọng số của đặc trưng này sẽ cao nhất để tối ưu điểm số và chọn hướng đi ăn ghost khi ở trạng thái scared.



- ❖ Vị trí cũng như khoảng cách giữa pacman và ghost cũng được dùng để ước lượng giá trị của trạng thái bởi nếu ở các trạng thái gần ghost thì có thể dẫn đến các trường hợp xấu hơn như ngõ cụt và dẫn đến Pacman died do đó khoảng cách giữa ghost và pacman là một đặc trưng để ước lượng khi các ghost không ở trạng thái sợ hãi, với mỗi ghost :
  - khi ghost ở trạng thái bình thường tức `ghost.scaredTimer == 0` và `ghost_distance != 0` → `ghost_score += - 2 / ghost_distance` (càng gần ghost giá trị trạng thái càng thấp)
  - và khi `ghost_distance == 0` (Pacman died) → `ghost_score += -1000`



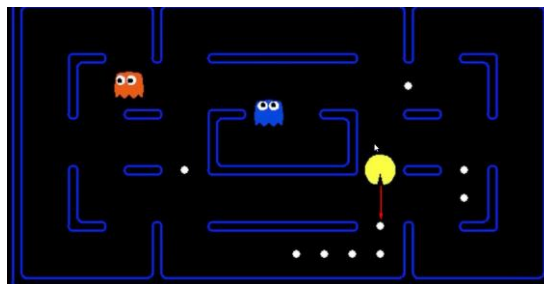
- ❖ vì pacman có thể ăn ghost sau khi ăn capsule và khi các ghost ở trạng thái sợ hãi ta có thể cho pacman ưu tiên ăn ghost để tối ưu số điểm và khi đó khoảng cách của pacman với ghost [ `ghost_distance` ] càng nhỏ thì giá trị đặc trưng này càng lớn để action của pacman sẽ ưu tiên lại gần các ghost ở trạng thái sợ hãi hơn để ăn chúng nhằm tối ưu điểm số, với mỗi ghost ở trạng thái sợ hãi  $\Leftrightarrow \text{ghost.scaredTimer} \neq 0$ 
  - nếu `ghost_distance`  $\neq 0 \rightarrow \text{ghost\_score} += 20 / \text{ghost\_distance}$  (khoảng cách càng gần giá trị của trạng thái càng cao)
  - nếu `ghost_distance`  $= 0 \rightarrow \text{ghost\_score} += 500$



- ❖ Số lượng thức ăn còn lại [ `len(foodList)` ] cũng được tính vào số điểm của trạng thái hiện tại ở trạng thái là lượng thức ăn còn lại trên map ít hơn thì điểm của trạng thái sẽ cao hơn nên trọng số của đặc trưng này sẽ là số âm (pacman ăn càng nhiều giá trị trạng thái đó càng cao)



- ❖ Mức độ gần của trạng thái với chấm thức ăn [ `closestFood` ] cũng là một đặc trưng để ước lượng giá trị của trạng thái vì nếu trạng thái càng gần với chấm thức ăn gần nhất sẽ có giá trị trạng thái cao hơn để pacman đi tới chấm thức ăn gần nhất, đặc trưng này sẽ được tính bằng khoảng cách của chấm thức ăn gần nhất tới vị trí pacman



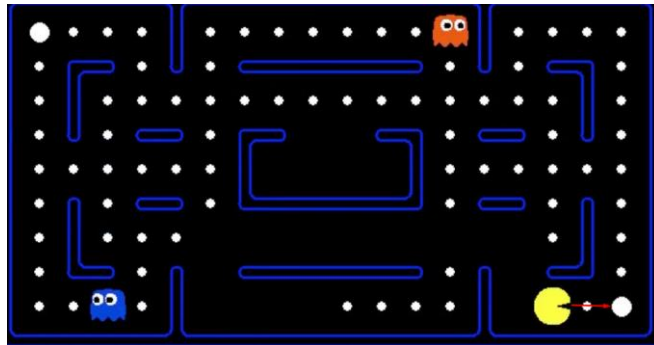
- ❖ Độ gần của các viên capsule [ `closest_capsule` ] so với pacman cũng ảnh hưởng đến điểm của trạng thái và khi ăn được thì điểm trạng thái sẽ càng cao hơn vì khi ăn

được capsule vị trí của các ghost sẽ không còn ảnh hưởng nhiều đến pacman giúp pacman tối ưu đường đi để ăn hết các chấm thức ăn nhanh hơn và còn có thể ăn cả ghost để gia tăng điểm số của trạng thái hiện tại,

- `closestCapsule` = khoảng cách tới Capsule gần nhất so với pacman

khi `closestCapsule != 0` → `closest_capsule_score = 3 / closestCapsule`

khi `closestCapsule == 0` → `closest_capsule_score = 100`



## 2. Những đặc trưng mình sử dụng để ước lượng giá trị trạng thái là gì? Trọng số của các đặc trưng đó như thế nào? Tại sao?

- Điểm số trò chơi ở trạng thái hiện tại [ `game_score` ] trọng số là **20** vì để tối ưu điểm số nên với trạng thái mà có điểm số trò chơi hiện tại lớn hơn thì điểm số của trạng thái cũng sẽ lớn hơn
- Độ gần với chấm thức ăn [ `closestFood` ] trọng số là **-2** vì càng gần thì điểm số trừ sẽ càng ít hơn tức là gần chấm thức ăn thì giá trị trạng thái sẽ cao hơn và ngược lại.
- Khoảng cách với ghost [ `ghost_score` ] trọng số là **1** vì giá trị của `ghost_distance` sẽ khác nhau tại các trạng thái khác nhau
- Lượng chấm thức ăn còn lại (`foodList`) trọng số là **-10** vì chấm thức ăn càng ít giá trị trạng thái càng cao và vì mục tiêu để thắng là ăn hết chấm thức ăn nên trọng số của đặc trưng này là bằng một nửa so với việc tối ưu điểm số với đặc trưng `game_score`
- Điểm số của độ gần so với capsule `closest_capsule_score` trọng số là **1** vì đây là một đặc trưng để tăng điểm số nên giá trị sẽ lớn hơn đặc trưng `closestFood` một chút

## 3. Chạy thử nghiệm các thuật toán đã cài đặt với evaluation function mình 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).

Layout	MinimaxAgent	AlphaBetaAgent	ExpectimaxAgent
mediumClassic	Average Score: 1976.0 Scores: 1926.0, 1732.0, 2144.0, 2140.0, 1938.0 Win Rate: 5/5 (1.00) Record: Win, Win, Win, Win, Win	Average Score: 1910.6 Scores: 2183.0, 1722.0, 1934.0, 1919.0, 1875.0 Win Rate: 5/5 (1.00) Record: Win, Win, Win, Win, Win	Average Score: 1666.2 Scores: 2124.0, 119.0, 2109.0, 1923.0, 2856.0 Win Rate: 4/5 (0.80) Record: Win, Loss, Win, Win, Win
capsuleClassic	Average Score: 164.0 Scores: 1252.0, -458.0, -383.0, 198.0, 211.0 Win Rate: 1/5 (0.20) Record: Win, Loss, Loss, Loss, Loss	Average Score: 214.6 Scores: -430.0, 1252.0, -442.0, -383.0, 1076.0 Win Rate: 2/5 (0.40) Record: Loss, Win, Loss, Loss, Win	Average Score: 296.2 Scores: 2832.0, -434.0, 3.0, -63.0, -57.0 Win Rate: 1/5 (0.20) Record: Win, Loss, Loss, Loss, Loss
contestClassic	Average Score: 2088.2 Scores: 2086.0, 1805.0, 2474.0, 1793.0, 1883.0 Win Rate: 5/5 (1.00) Record: Win, Win, Win, Win, Win	Average Score: 1783.4 Scores: 1685.0, 1483.0, 2279.0, 1886.0, 1664.0 Win Rate: 5/5 (1.00) Record: Win, Win, Win, Win, Win	Average Score: 2118.4 Scores: 2910.0, 1130.0, 2284.0, 2601.0, 1667.0 Win Rate: 4/5 (0.80) Record: Win, Loss, Win, Win, Win
minimaxClassic	Average Score: -294.4 Scores: -497.0, -495.0, 512.0, -496.0, -496.0 Win Rate: 1/5 (0.20) Record: Loss, Loss, Win, Loss, Loss	Average Score: -293.0 Scores: -493.0, -497.0, 513.0, -495.0, -493.0 Win Rate: 1/5 (0.20) Record: Loss, Loss, Win, Loss, Loss	Average Score: 102.4 Scores: -502.0, 512.0, 515.0, -510.0, 497.0 Win Rate: 3/5 (0.60) Record: Loss, Win, Win, Loss, Win
openClassic	Average Score: 1316.0 Scores: 1431.0, 1239.0, 1250.0, 1419.0, 1241.0 Win Rate: 5/5 (1.00) Record: Win, Win, Win, Win, Win	Average Score: 1284.0 Scores: 1251.0, 1258.0, 1424.0, 1246.0, 1241.0 Win Rate: 5/5 (1.00) Record: Win, Win, Win, Win, Win	Average Score: 1244.8 Scores: 1252.0, 1247.0, 1233.0, 1242.0, 1250.0 Win Rate: 5/5 (1.00) Record: Win, Win, Win, Win, Win
smallClassic	Average Score: 1164.0 Scores: 970.0, 967.0, 1175.0, 1371.0, 1337.0 Win Rate: 5/5 (1.00) Record: Win, Win, Win, Win, Win	Average Score: 1185.2 Scores: 1367.0, 1343.0, 957.0, 697.0, 1562.0 Win Rate: 4/5 (0.80) Record: Win, Win, Win, Loss, Win	Average Score: 1242.8 Scores: 977.0, 1362.0, 1537.0, 1361.0, 977.0 Win Rate: 5/5 (1.00) Record: Win, Win, Win, Win, Win
trappedClassic	Average Score: -501.0 Scores: -501.0, -501.0, -501.0, -501.0, -501.0 Win Rate: 0/5 (0.00) Record: Loss, Loss, Loss, Loss, Loss	Average Score: -501.0 Scores: -501.0, -501.0, -501.0, -501.0, -501.0 Win Rate: 0/5 (0.00) Record: Loss, Loss, Loss, Loss, Loss	Average Score: -501.0 Scores: -501.0, -501.0, -501.0, -501.0, -501.0 Win Rate: 0/5 (0.00) Record: Loss, Loss, Loss, Loss, Loss

## B1.KẾT QUẢ THỬ NGHIỆM VỚI EVALUATIONFUNCTION MÌNH ĐÃ THIẾT KẾ VỚI -N 5 VÀ DEPTH = 3

Layout	MinimaxAgent	AlphaBetaAgent	ExpectimaxAgent
mediumClassic	Pacman crashed Average Score: 812.4 Scores: 336.0, 336.0, 1320.0, 1231.0, 839.0 Win Rate: 1/5 (0.20) Record: Loss, Loss, Win, Loss, Loss	Average Score: 506.8 Scores: 295.0, 358.0, 979.0, 665.0, 237.0 Win Rate: 0/5 (0.00) Record: Loss, Loss, Loss, Loss, Loss	Pacman crashed Average Score: 524.8 Scores: 384.0, 789.0, 743.0, 517.0, 271.0 Win Rate: 0/5 (0.00) Record: Loss, Loss, Loss, Loss, Loss
capsuleClassic	Average Score: -378.6 Scores: -453.0, -55.0, -492.0, -410.0, -483.0 Win Rate: 0/5 (0.00) Record: Loss, Loss, Loss, Loss, Loss	Average Score: -237.4 Scores: -511.0, -457.0, -461.0, 684.0, -442.0 Win Rate: 1/5 (0.20) Record: Loss, Loss, Loss, Win, Loss	Average Score: -344.2 Scores: -434.0, -262.0, -423.0, -161.0, -441.0 Win Rate: 0/5 (0.00) Record: Loss, Loss, Loss, Loss, Loss
contestClassic	Average Score: 846.8 Scores: 475.0, 1069.0, 1183.0, 1935.0, -420.0 Win Rate: 1/5 (0.20) Record: Loss, Loss, Loss, Win, Loss	Average Score: 1199.2 Scores: 920.0, 1482.0, 1534.0, 1485.0, 575.0 Win Rate: 0/5 (0.00) Record: Loss, Loss, Loss, Loss, Loss	Pacman crashed Average Score: 1226.2 Scores: 996.0, 1008.0, 991.0, 1262.0, 1874.0 Win Rate: 0/5 (0.00) Record: Loss, Loss, Loss, Loss, Loss
minimaxClassic	Average Score: 109.6 Scores: 514.0, 511.0, -495.0, 514.0, -496.0 Win Rate: 3/5 (0.60) Record: Win, Win, Loss, Win, Loss	Average Score: -91.0 Scores: 512.0, 512.0, -495.0, -495.0, -493.0 Win Rate: 2/5 (0.40) Record: Win, Win, Loss, Loss, Loss	Average Score: 308.4 Scores: -498.0, 503.0, 512.0, 513.0, 512.0 Win Rate: 4/5 (0.80) Record: Loss, Win, Win, Win, Win
openClassic	Pacman crashed Average Score: 58.4 Scores: 30.0, 41.0, 69.0, 79.0, 73.0 Win Rate: 0/5 (0.00) Record: Loss, Loss, Loss, Loss, Loss	Average Score: -17.2 Scores: -184.0, -107.0, -110.0, -84.0, 399.0 Win Rate: 0/5 (0.00) Record: Loss, Loss, Loss, Loss, Loss	Pacman crashed Average Score: 42.6 Scores: 56.0, 44.0, 24.0, 41.0, 48.0 Win Rate: 0/5 (0.00) Record: Loss, Loss, Loss, Loss, Loss
smallClassic	Average Score: 230.0 Scores: -171.0, 305.0, -100.0, 273.0, 843.0 Win Rate: 1/5 (0.20) Record: Loss, Loss, Loss, Loss, Win	Average Score: 797.0 Scores: 58.0, 427.0, 1556.0, 796.0, 1148.0 Win Rate: 3/5 (0.60) Record: Loss, Loss, Win, Win, Win	Pacman crashed Average Score: 709.6 Scores: 587.0, 1475.0, 72.0, 1149.0, 265.0 Win Rate: 2/5 (0.40) Record: Loss, Win, Loss, Win, Loss
trappedClassic	Average Score: -501.0 Scores: -501.0, -501.0, -501.0, -501.0, -501.0 Win Rate: 0/5 (0.00) Record: Loss, Loss, Loss, Loss, Loss	Average Score: -501.0 Scores: -501.0, -501.0, -501.0, -501.0, -501.0 Win Rate: 0/5 (0.00) Record: Loss, Loss, Loss, Loss, Loss	Average Score: -295.2 Scores: -502.0, -502.0, 532.0, -502.0, -502.0 Win Rate: 1/5 (0.20) Record: Loss, Loss, Win, Loss, Loss

## B2.KẾT QUẢ THỬ NGHIỆM VỚI SCOREEVALUATIONFUNCTION VỚI -N 5 VÀ DEPTH = 3

- Theo bảng B1 và B2 ta có :

### B3: HIỆU NĂNG/HIỆU QUẢ CỦA MINIMAX VÀ ALPHABETA VÀ EXPECTIMAX

		MinimaxAgent	AlphaBetaAgent	ExpectimaxAgent
EvaluationFunction đã thiết kế	Average score	<b>833</b>	<b>797</b>	<b>881</b>
	Win rate	<b>22/35 ~ 0.63</b>	<b>22/35 ~ 0.63</b>	<b>22/35 ~ 0.63</b>
scoreEvaluationFunction	Average score	<b>240</b>	<b>242</b>	<b>310</b>
	Win rate	<b>6/35 ~ 0.17</b>	<b>6/35 ~ 0.17</b>	<b>7/35 ~ 0.2</b>

- Dựa vào bảng B2 có thể thấy Hiệu năng/Hiệu quả của ExpectimaxAgent cao hơn so với MinimaxAgent và AlphaBetaAgent
- Và với evaluation function có sẵn scoreEvaluationFunction trong đó chỉ sử dụng điểm số của trạng thái để ước lượng giá trị trạng thái thì cả Average score và winrate thấp hơn nhiều so với score EVALUATION FUNCTION tự thiết kế vì chỉ sử dụng điểm số trạng thái để ước lượng nên khi ở các trạng thái khác nhau mà điểm số bằng nhau thì pacman dễ bị crashed và đi sai hướng dẫn đến hiệu quả thấp.

🌟 Link record ván chơi kết quả tốt nhất:

<https://drive.google.com/drive/folders/1Ny-qNsojGpZbYNnoKgEhIV7VAPCt7U2n?usp=sharing>