# ARTIFICIAL INTELLIGENCE

**LAB 3’S REPORT**

**Evaluation functions for**

**Minimax/AlphaBeta/Expectimax**

**1. Mô tả ý tưởng/chiến thuật mình tự thiết kế evaluation function:**

Hàm lượng giá bao gồm các feature: game\_score (điểm số của trò chơi), closet\_food (khoảng cách chấm thức ăn gần nhất), avg\_capsule (khoảng cách viên capsule nhất, hunter\_ratio (tổng tỉ số bị ma đuổi), stop\_penalty.

Ý tưởng:

Game\_score: đó chính là sẽ gồm một phần điểm của trò chơi.

1/avg\_capsule:

* Bên cạnh đó chúng ta muốn khuyến khích pacman sẽ ăn các chấm năng lượng, chính vì vậy ta sẽ bổ sung một feature là avg\_capsule để biểu thị cho khoảng cách trung bình các viên năng lượng. bên cạnh đó bổ sung thêm một feature nữa là capsule\_count, mục đích nhằm khuyến khích các pacman ăn các viên capsule.
* Khi số lượng các viên năng lượng hết, thì tỉ số avg\_capsule sẽ bé nhất. Tuy nhiên trong thời gian các viên năng lượng còn tác dụng, ta sẽ muốn pacman không để dành các viên capsule chính vì vậy, ta sẽ không quan tâm tới chỉ số này mà sẽ cho avg\_capsule lên bé nhất.

Capsule\_count:

* Là số lượng capsule còn lại, khi pacman còn càng nhiều capsule thì sẽ trừ điểm càng nhiều. 1/closet\_food:
* Bằng khoảng cách bé nhất tới chấm thức ăn gần nhất, nếu số chấm thức ăn = 0 thì closet\_food sẽ lớn nhất.
* Khoảng cách tới chấm thức ăn càng ngắn thì điểm sẽ càng nhiều vì chúng ta muốn pacman ăn hết các chấm thức ăn.

Hunter\_ratio:

* Nếu bất kì ma nào có khoảng cách <1 khi scare\_time = 0 thì chúng ta muốn pacman sẽ chạy đi.

1. **So sánh các hàm evaluation function, Hiểu quả của Minimax và AlphaBeta và Expectimax**

So với scoreEvaluationFunction, thì pacman sẽ bị đứng lại rất nhiều lần. Chỉ khi nào bị ma đuổi hoặc khoảng cách thức ăn gần nhất = depth thì mới di chuyển, chính vì vậy sẽ bị tốn rất nhiều điểm.

EvaluationFunction tự thiết mặc dù đã khắc phục được lỗi bị dừng, bên cạnh đó là việc chỉ di chuyển liên tục hai hướng lặp lại liên tục (lên xuống, trái phải), tuy nhiên khi ăn các chấm năng lượng thì pacman vẫn không tìm được các con ma, vì lúc này ghostAgent không tìm thấy, bên cạnh đó việc ăn các chấm thức ăn gần nhất vẫn không phải là tốt nhất vì chưa đôi lúc sẽ tốt nhiều thời gian (1 quãng đường đi tới 2 lần để ăn 2 chấm thức ăn gần nhau). Và heuristic để tránh khỏi ma vẫn còn naïve, chính vì vậy đối với các corner case, những trường hợp bị chặn hai đầu vẫn không thể thoát được.

Đối kết quả, AlphaBeta, tương tự với Minimax nhưng thời gian chạy nhanh hơn rất đáng kể, tuy nhiên đối với trappedClassic, thì AlphaBeta với dept >=3 sẽ chắc chắn tự tử còn đối với expectmax sẽ chạy tới viên thức ăn, chính vì vậy cho ra kết quả cao hơn.

1. **Source code**

Bổ sung thêm phần search, để tìm vị trí đến các điểm chính xác.

Trong file search.py, bổ sung thuật toán tìm kiếm A\*, dựa vào hai vị trí trên mê cung, tìm ra chính xác khoảng cách ngắn nhất giữa 2 điểm.

Bên cạnh đó ở file multiAgent.py, các agent được bổ sung thêm một biến là is\_stop và previous\_state, để check xem pacman có đứng yên hay không