Künstliche Intelligenz, SoSe21

Project 2

Jonny Lam & Thore Brehmer

23. Juni 2023

---

1 Q1: Why does Pacman rush to the closest ghost in this case?

- The question was already somewhat explained in the project sheet. 'When Pacman believes that his death is unavoidable, he will try to end the game as soon as possible because of the constant penalty for living.'
- It should also be noted that we assume that the ghost always take the best path for themselves (the path in which we lose the fastest). Hence the closest ghost is the quickest way to lose, resulting in losing the fewest points.

2 Q2: Compare this solution with the previous minimax and comment on that.

- Both algorithms will return the same actions and thus run the same path. The only difference is that the Alpha-Beta Pruning will skip the calculation of unnecessary paths and hence it will run faster.

3 Q3: Investigate the results of these two scenarios and comment on them:

```
#python3 pacman.py -p AlphaBetaAgent -l trappedClassic -a depth=3 -q -n 10
#python3 pacman.py -p ExpectimaxAgent -l trappedClassic -a depth=3 -q -n 10
```

- For the 1. command pacman always runs right into the closest ghost and thus always loses the game.
- Output for 1. command:

```
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Average Score: -501.0
Scores:        -501.0, -501.0, -501.0, -501.0, -501.0, -501.0, -501.0, -501.0, -501.0,
    -501.0
Win Rate:      0/10 (0.00)
Record:        Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss
```

- For the 2. command pacman always tries to eat the foot on the left path.
- He either runs into the opposing ghost on this path, if the ghost runs towards, resulting in pacman and losing the game
- or he is able to eat all food and win the game, if the ghost runs away from pacman.
- Output for 2. command:

```
Pacman died! Score: -502
Pacman emerges victorious! Score: 532
Pacman emerges victorious! Score: 532
Pacman emerges victorious! Score: 532
Pacman died! Score: -502
Pacman emerges victorious! Score: 532
Pacman died! Score: -502
Pacman died! Score: -502
Pacman emerges victorious! Score: 532
Pacman died! Score: -502
Average Score: 15.0
Scores:        -502.0, 532.0, 532.0, 532.0, -502.0, 532.0, -502.0, -502.0, 532.0,
    -502.0
Win Rate:      5/10 (0.50)
Record:        Loss, Win, Win, Win, Loss, Win, Loss, Loss, Win, Loss
```

4 Q4: Make sure you understand why the behavior here differs from the minimax case and comment on it.

- **Minimax**
  - In minmax we assume that the ghost always take the best path for themselves (the path in which we lose the fastest). Hence the closest ghost is the quickest way to lose, resulting in losing the fewest points. That is why pacman moves right.
  - If pacman would move left on the otherhand, the opposing ghost on that path would just move towards pacman. Pacman would not be able to eat any food, but he would live longer, resulting in losing more points. => Worse score than moving right.
  - Score moving right = -501, while score moving left = -502
- **Expectimax**
  - In expectimax we assume that the ghost takes a random path. With that we calculate the best path (highest score) for pacman. We do that by taking the path with the highest average points.
  - The right path will always result in -501 points. While the left path can either result in -502 or 532 points.
  - So the average of the left path is 15 points. This is much better than the average of the right path, which is -502. Thats why pacman always goes left.