

CSE 537 Project 2 : Multi Agent PacMan

Rohan Bharadwaj (109758985)

Shashank Jain (109956091)

Ashish Goel (109753528)

Question 1

Reflex Agent :

python pacman.py -p ReflexAgent -l openClassic -n 10 -q

Implementation Details :

We've considered the following parameters for evaluation function:-

- 1) Minimum distance of closest active ghost :- Farther the distance of minimum distance active ghost, more would be the evaluation score.
- 2) Minimum distance of closest scared ghost :- Lesser the distance of nearest scary ghost, pacman would tend to move towards it because the score would be reduced by less amount as compared to other scary ghost.
- 3) Closest distance of food :- Lesser the distance of closest food, then the evaluation score would be reduced by the least amount.
- 4) If remaining number of foods in new state are less than previous state, then we increment the score.
- 5) To encourage pacman for moving towards the capsules, we give pacman a score boost for the same.

The Reflex Agent on running 10 times on open classic emerges as victorious all the times. Sometimes we observed that it is 9/10. But never below 9.

Average Score: 1247.4

Scores: 1247.0, 1235.0, 1249.0, 1254.0, 1241.0, 1256.0, 1245.0, 1253.0, 1239.0, 1255.0

Win Rate: 10/10 (1.00)

Record: Win, Win, Win, Win, Win, Win, Win, Win, Win, Win

Question 2

Minimax Agent :

python pacman.py -p MinimaxAgent -l minimaxClassic -a depth=4 -n 20

Average Score: 61.4

Scores: -492.0, -495.0, -492.0, -495.0, 513.0, 513.0, 513.0, -492.0, 516.0, 516.0, 516.0, 516.0, 516.0, -492.0, -495.0, -492.0, 514.0, 516.0, -492.0, 516.0

Win Rate: 11/20 (0.55)

Record: Loss, Loss, Loss, Loss, Win, Win, Win, Loss, Win, Win, Win, Win, Win, Loss,
Loss, Loss, Win, Win, Loss, Win

**python pacman.py -p MinimaxAgent -l minimaxClassic -a depth=4 --numGames 1000
--frameTime 0 --fixRandomSeed --textGraphics**

Average Score: 118.14

Win Rate: 606/1000 (0.61)

The PacMan is very slow in eating dots, as no pruning is being performed here.

For 1000 games the time taken is **1150 secs**.

Question 3

Alpha Beta Pruning Agent :

python pacman.py -p AlphaBetaAgent -a depth=3 -l smallClassic -n 20

Average Score: 325.7

Scores: -76.0, 1035.0, -200.0, -379.0, -126.0, 1124.0, 1473.0, -221.0, -396.0, -222.0,
1010.0, 1121.0, 1693.0, -379.0, 248.0, 850.0, -397.0, 1120.0, -392.0, -372.0

Win Rate: 8/20 (0.40)

Record: Loss, Win, Loss, Loss, Loss, Win, Win, Loss, Loss, Loss, Win, Win, Win, Loss,
Loss, Win, Loss, Win, Loss, Loss

**python pacman.py -p AlphaBetaAgent -l minimaxClassic -a depth=4 --numGames 1000
--frameTime 0 --fixRandomSeed --textGraphics**

Average Score: 118.14

Win Rate: 606/1000 (0.61)

The PacMan Alpha Beta agent is faster in eating dots as compared to minimax, because some nodes have been pruned.

For 1000 games the time take is **190 secs**.