PACMAN Project

1 Introduction

PAC-MAN is one of the popular games. The goal of this game is to accumulate points by eating all the Pac-Dots in the maze, completing that 'stage' of the game and starting the next stage and maze of Pac-dots. There are several ghosts roaming the maze, trying to kill Pac-Man. If any of the ghosts hit Pac-Man, he loses a life; when all lives have been lost, the game is over. In this project, I have to implement adversarial search using alpha-beta pruning, minmax and expectimax search and evaluation function in the Multi-Agent PACMan.

2 Multi-Agent Pacman

2.1 Alpha-Beta Pruning

In this task Alpha-Beta Pruning was implemented here. In this case the assumption is that there is only one ghost. In this way, it become two player game with pacman as the max node and the ghost as the min node. In this case the depth refers the number of moves.

Here following is the report for each case:

```
Question q1
===========

Pacman emerges victorious! Score: 1257
Pacman emerges victorious! Score: 1255
Pacman emerges victorious! Score: 1251
Pacman emerges victorious! Score: 1262
Pacman emerges victorious! Score: 1261
Pacman emerges victorious! Score: 1262
Pacman emerges victorious! Score: 1256
Pacman emerges victorious! Score: 1262
Pacman emerges victorious! Score: 1265
Pacman emerges victorious! Score: 1265
Average Score: 1259.6
Scores:        1257.0, 1255.0, 1251.0, 1262.0, 1261.0, 1262.0, 1256.0, 1262.0, 1265.0, 1265.0
```

```
Question q2
===========

*** PASS: test_cases\q2\0-lecture-6-tree.test
*** PASS: test_cases\q2\0-small-tree.test
*** PASS: test_cases\q2\1-1-minmax.test
*** PASS: test_cases\q2\1-2-minmax.test
*** PASS: test_cases\q2\1-3-minmax.test
*** PASS: test_cases\q2\1-4-minmax.test
*** PASS: test_cases\q2\1-5-minmax.test
*** PASS: test_cases\q2\1-6-minmax.test
*** PASS: test_cases\q2\1-7-minmax.test
*** PASS: test_cases\q2\1-8-minmax.test
*** PASS: test_cases\q2\2-1a-vary-depth.test
*** PASS: test_cases\q2\2-1b-vary-depth.test
*** PASS: test_cases\q2\2-2a-vary-depth.test
*** PASS: test_cases\q2\2-2b-vary-depth.test
*** PASS: test_cases\q2\2-3a-vary-depth.test
*** PASS: test_cases\q2\2-3b-vary-depth.test
*** PASS: test_cases\q2\2-4a-vary-depth.test
*** PASS: test_cases\q2\2-4b-vary-depth.test
*** PASS: test_cases\q2\2-one-ghost-3level.test
*** PASS: test_cases\q2\3-one-ghost-4level.test
*** PASS: test_cases\q2\4-two-ghosts-3level.test
*** PASS: test_cases\q2\5-two-ghosts-4level.test
*** PASS: test_cases\q2\6-tied-root.test
*** PASS: test_cases\q2\7-1a-check-depth-one-ghost.test
*** PASS: test_cases\q2\7-1b-check-depth-one-ghost.test
*** PASS: test_cases\q2\7-1c-check-depth-one-ghost.test
*** PASS: test_cases\q2\7-2a-check-depth-two-ghosts.test
*** PASS: test_cases\q2\7-2b-check-depth-two-ghosts.test
*** PASS: test_cases\q2\7-2c-check-depth-two-ghosts.test
*** Running MinimaxAgent on smallClassic 1 time(s).
Pacman died! Score: 84
```

```
Question q3
===========

*** PASS: test_cases\q3\0-lecture-6-tree.test
*** PASS: test_cases\q3\0-small-tree.test
*** PASS: test_cases\q3\1-1-minmax.test
*** PASS: test_cases\q3\1-2-minmax.test
*** PASS: test_cases\q3\1-3-minmax.test
*** PASS: test_cases\q3\1-4-minmax.test
*** PASS: test_cases\q3\1-5-minmax.test
*** PASS: test_cases\q3\1-6-minmax.test
*** PASS: test_cases\q3\1-7-minmax.test
*** PASS: test_cases\q3\1-8-minmax.test
*** PASS: test_cases\q3\2-1a-vary-depth.test
*** PASS: test_cases\q3\2-1b-vary-depth.test
*** PASS: test_cases\q3\2-2a-vary-depth.test
*** PASS: test_cases\q3\2-2b-vary-depth.test
*** PASS: test_cases\q3\2-3a-vary-depth.test
*** PASS: test_cases\q3\2-3b-vary-depth.test
*** PASS: test_cases\q3\2-4a-vary-depth.test
*** PASS: test_cases\q3\2-4b-vary-depth.test
*** PASS: test_cases\q3\2-one-ghost-3level.test
*** PASS: test_cases\q3\3-one-ghost-4level.test
*** PASS: test_cases\q3\4-two-ghosts-3level.test
*** PASS: test_cases\q3\5-two-ghosts-4level.test
*** PASS: test_cases\q3\6-tied-root.test
*** PASS: test_cases\q3\7-1a-check-depth-one-ghost.test
*** PASS: test_cases\q3\7-1b-check-depth-one-ghost.test
*** PASS: test_cases\q3\7-1c-check-depth-one-ghost.test
*** PASS: test_cases\q3\7-2a-check-depth-two-ghosts.test
*** PASS: test_cases\q3\7-2b-check-depth-two-ghosts.test
*** PASS: test_cases\q3\7-2c-check-depth-two-ghosts.test
*** Running AlphaBetaAgent on smallClassic 1 time(s).
Pacman died! Score: 84
Average Score: 84.0
```

```
Question q4
===========

*** PASS: test_cases\q4\0-expectimax1.test
*** PASS: test_cases\q4\1-expectimax2.test
*** PASS: test_cases\q4\2-one-ghost-3level.test
*** PASS: test_cases\q4\3-one-ghost-4level.test
*** PASS: test_cases\q4\4-two-ghosts-3level.test
*** PASS: test_cases\q4\5-two-ghosts-4level.test
*** PASS: test_cases\q4\6-1a-check-depth-one-ghost.test
*** PASS: test_cases\q4\6-1b-check-depth-one-ghost.test
*** PASS: test_cases\q4\6-1c-check-depth-one-ghost.test
*** PASS: test_cases\q4\6-2a-check-depth-two-ghosts.test
*** PASS: test_cases\q4\6-2b-check-depth-two-ghosts.test
*** PASS: test_cases\q4\6-2c-check-depth-two-ghosts.test
*** Running ExpectimaxAgent on smallClassic 1 time(s).
Pacman died! Score: 84
Average Score: 84.0
```

```
### Question q5: 0/6 ###


Finished at 17:28:09

Provisional grades
==================
Question q1: 4/4
Question q2: 5/5
Question q3: 5/5
Question q4: 5/5
Question q5: 0/6
------------------
Total: 19/25
```

2.2 Evaluation Function

In this task the evaluation function used here evaluates the states as whole. The score is calculated using the following steps:

1. Calculate iteratively minimum distance of food from Pacman position and keep adding

inverse of it to state score and replacing Pacman position with nearest food till exhausting food list.
2. Do the above same for capsules as well.
3. If minimum ghost distance is 1 or less then return a large negative value as state score. Otherwise subtract its inverse from score.
4. Add eight times score of game state to state score.
5. Subtract six times total food plus total capsules remaining.

In this case after using betterEvaluationFunction we can improve the pacman score for layer 5 and the pacman score got increased after using better Evaluation function.

```
Question q5
===========

Pacman emerges victorious! Score: 1069
Pacman emerges victorious! Score: 1077
Pacman emerges victorious! Score: 1147
Pacman emerges victorious! Score: 1166
Pacman emerges victorious! Score: 1172
Pacman emerges victorious! Score: 1120
Pacman emerges victorious! Score: 952
Pacman emerges victorious! Score: 914
Pacman emerges victorious! Score: 1171
Pacman emerges victorious! Score: 938
Average Score: 1072.6
Scores:        1069.0, 1077.0, 1147.0, 1166.0, 1172.0, 1120.0, 952.0, 914.0, 1171.0, 938.0
```

```
Finished at 17:36:09

Provisional grades
==================
Question q1: 4/4
Question q2: 5/5
Question q3: 5/5
Question q4: 5/5
Question q5: 6/6
------------------
Total: 25/25
```

3. References

http://ai.berkeley.edu/multiagent.html