

HW2 Project Report

1. Question 1 Reflex Agent:

- **Explanation:** A reflex agent chooses an action at each choice point by examining its alternatives using its evaluate function. I have used the current Pacman game state to retrieve the food list in Pacman world, the ghost positions, ghost states and current states. If Pacman's position and ghost position are same then we stop proceeding further and return the default minimum possible value. For the available food in the grid, I have used manhattanDistance function to get the locations and find the minimum distance from Pacman's current position. And finally, we return this distance to proceed so that Pacman can eat the foods.
- **Agent Result:**

Python Command	Result
python pacman.py -p ReflexAgent -l testClassic --frameTime 0	Pacman emerges victorious! Score: 562 Average Score: 562.0 Scores: 562.0 Win Rate: 1/1 (1.00) Record: Win
python pacman.py --frameTime 0 -p ReflexAgent -k 1 --frameTime 0	Pacman emerges victorious! Score: 1452 Average Score: 1452.0 Scores: 1452.0 Win Rate: 1/1 (1.00) Record: Win
python pacman.py --frameTime 0 -p ReflexAgent -k 2 --frameTime 0	Pacman died! Score: 276 Average Score: 276.0 Scores: 276.0 Win Rate: 0/1 (0.00) Record: Loss

- **Autograder Output:**

python autograder.py -q q1 --no-graphics

Question q1

=====

Pacman emerges victorious! Score: 1246
Pacman emerges victorious! Score: 1237
Pacman emerges victorious! Score: 1235
Pacman emerges victorious! Score: 1245
Pacman emerges victorious! Score: 1233
Pacman emerges victorious! Score: 1231
Pacman emerges victorious! Score: 1235
Pacman emerges victorious! Score: 1253
Pacman emerges victorious! Score: 1231
Pacman emerges victorious! Score: 1236

Average Score: 1238.2
 Scores: 1246.0, 1237.0, 1235.0, 1245.0, 1233.0, 1231.0, 1235.0, 1253.0, 1231.0, 1236.0
 Win Rate: 10/10 (1.00)
 Record: Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
 *** PASS: test_cases/q1/grade-agent.test (4 of 4 points)
 *** 1238.2 average score (2 of 2 points)
 *** Grading scheme:
 *** < 500: 0 points
 *** >= 500: 1 points
 *** >= 1000: 2 points
 *** 10 games not timed out (0 of 0 points)
 *** Grading scheme:
 *** < 10: fail
 *** >= 10: 0 points
 *** 10 wins (2 of 2 points)
 *** Grading scheme:
 *** < 1: fail
 *** >= 1: 0 points
 *** >= 5: 1 points
 *** >= 10: 2 points

 ### Question q1: 4/4 ###

Finished at 2:27:02

Provisional grades
 =====
 Question q1: 4/4

 Total: 4/4

2. Question 2 Minimax:

- Explanation:** The concept of this search algorithm is present in its name. In a 2 player game we will with max node and at each depth we will find the values of successor nodes depending on the type of node(min or max). By using provided depth and evaluation utility, I have developed this algorithm which will start from max node and search for nodes in depth wise. Starting with the max node, we will generate the next successor with valid moves and pass it to the next min node. The min node will recursively call itself and max node depending on the ghost position and their availability. The final result will be bestMoves and state score to reach the leaf node. This algorithm will finally return the best move from the starting/root node which is max node for a player.
- Agent Result:**

Python Command	Result
python pacman.py -p MinimaxAgent -l minimaxClassic -a depth=4 --frameTime 0	Pacman died! Score: -499 Average Score: -499.0 Scores: -499.0 Win Rate: 0/1 (0.00)

	Record: Loss
python pacman.py -p MinimaxAgent -l trappedClassic -a depth=3 --frameTime 0	Pacman died! Score: -479 Average Score: -479.0 Scores: -479.0 Win Rate: 0/1 (0.00) Record: Loss

- **Autograder Output:**

python autograder.py -q q2 --no-graphics

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
Average Score: 84.0
Scores:        84.0
Win Rate:     0/1 (0.00)
Record:       Loss
*** Finished running MinimaxAgent on smallClassic after 1 seconds.
*** Won 0 out of 1 games. Average score: 84.000000 ***

```

*** PASS: test_cases/q2/8-pacman-game.test

Question q2: 5/5

Finished at 2:50:25

Provisional grades

=====

Question q2: 5/5

Total: 5/5

3. Question 3 Alpha-Beta Pruning:

- **Explanation:** In this algorithm, we will start with max start node with and all the nodes including max node will have 2 values included – alpha(-infinity) and beta(+infinity). Similar to minimax this will also return minimax action using depth and evaluationFunction utility. Starting node(max) will find all the valid moves it can take and find the next node accordingly, which is a min node. This min node will recursively call itself and max node depending on the ghost position and their availability. The main advantage of Alpha-Beta pruning search is that at each node will compare the values of alpha and beta and if at any point $\alpha \geq \beta$ then we will stop further proceeding. This will result not to visit all the successor node of a parent. Min node will return the minimum score and max node will return the maximum score. Finally, we will retrieve the best move a player can take using this search algorithm.

- **Agent Result:**

Python Command	Result
python pacman.py -p AlphaBetaAgent -a depth=3 -l smallClassic --frameTime 0	Pacman died! Score: 147 Average Score: 147.0 Scores: 147.0 Win Rate: 0/1 (0.00) Record: Loss

- **Autograder Output:**

python autograder.py -q q3 --no-graphics

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
Scores:      84.0
Win Rate:    0/1 (0.00)
Record:      Loss
*** Finished running AlphaBetaAgent on smallClassic after 1 seconds.
*** Won 0 out of 1 games. Average score: 84.000000 ***
*** PASS: test_cases/q3/8-pacman-game.test

```

Question q3: 5/5

Finished at 23:33:43

Provisional grades

=====

Question q3: 5/5

Total: 5/5

4. Question 4 Expectimax:

- Explanation:** This search algorithm searches for best move to be taken by max node based on the value of the chance node which is average of the values of its successor nodes. This will return minimax action using depth and evaluationFunction utility. Starting with root or max node if this is not terminal node then find all the valid moves max can take and generate the successor nodes. Successor min node will recursively call itself and max node based on the ghost position and availability and eventually find the average minimum value and return the same to the root node. Finally, root max node will be able to retrieve the best move based on the chance node's values and player will be able to take proper move to proceed.

- **Agent Result:**

Python Command	Result
python pacman.py -p ExpectimaxAgent -l minimaxClassic -a depth=3 --frameTime 0	Pacman died! Score: -496 Average Score: -496.0 Scores: -496.0 Win Rate: 0/1 (0.00) Record: Loss
python pacman.py -p ExpectimaxAgent -l trappedClassic -a depth=3 -q -n 10 --frameTime 0	Pacman died! Score: -502 Pacman died! Score: -502 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 emerges victorious! Score: 532 Pacman emerges victorious! Score: 532 Average Score: 118.4 Scores: -502.0, -502.0, -502.0, 532.0, 532.0, 532.0, -502.0, 532.0, 532.0, 532.0 Win Rate: 6/10 (0.60) Record: Loss, Loss, Loss, Win, Win, Win, Loss, Win, Win, Win

- **Autograder Output:**

python autograder.py -q q4 --no-graphics

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
Scores:    84.0
Win Rate:  0/1 (0.00)
Record:    Loss
*** Finished running ExpectimaxAgent on smallClassic after 1 seconds.
*** Won 0 out of 1 games. Average score: 84.000000 ***
*** PASS: test_cases/q4/7-pacman-game.test
```

Question q4: 5/5

Finished at 23:36:02

```
Provisional grades
=====
Question q4: 5/5
-----
Total: 5/5
```

Autograder Report:

python autograder.py

Question q1
=====

```
Pacman emerges victorious! Score: 1246
Pacman emerges victorious! Score: 1237
Pacman emerges victorious! Score: 1235
Pacman emerges victorious! Score: 1245
Pacman emerges victorious! Score: 1233
Pacman emerges victorious! Score: 1231
Pacman emerges victorious! Score: 1235
Pacman emerges victorious! Score: 1253
Pacman emerges victorious! Score: 1231
Pacman emerges victorious! Score: 1236
Average Score: 1238.2
Scores:    1246.0, 1237.0, 1235.0, 1245.0, 1233.0, 1231.0, 1235.0, 1253.0, 1231.0, 1236.0
Win Rate:  10/10 (1.00)
Record:    Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
*** PASS: test_cases/q1/grade-agent.test (4 of 4 points)
***    1238.2 average score (2 of 2 points)
***    Grading scheme:
***    < 500: 0 points
***    >= 500: 1 points
***    >= 1000: 2 points
```

```
*** 10 games not timed out (0 of 0 points)
***   Grading scheme:
***   < 10: fail
***   >= 10: 0 points
*** 10 wins (2 of 2 points)
***   Grading scheme:
***   < 1: fail
***   >= 1: 0 points
***   >= 5: 1 points
***   >= 10: 2 points
```

Question q1: 4/4

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
Average Score: 84.0
Scores:      84.0
Win Rate:    0/1 (0.00)
Record:      Loss
*** Finished running MinimaxAgent on smallClassic after 1 seconds.
```


*** Won 0 out of 1 games. Average score: 84.000000 ***
*** PASS: test_cases/q2/8-pacman-game.test

Question q2: 5/5

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
Scores: 84.0
Win Rate: 0/1 (0.00)
Record: Loss
*** Finished running AlphaBetaAgent on smallClassic after 1 seconds.
*** Won 0 out of 1 games. Average score: 84.000000 ***
*** PASS: test_cases/q3/8-pacman-game.test

Question q3: 5/5

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
Scores:      84.0
Win Rate:    0/1 (0.00)
Record:      Loss
*** Finished running ExpectimaxAgent on smallClassic after 1 seconds.
*** Won 0 out of 1 games. Average score: 84.000000 ***
*** PASS: test_cases/q4/7-pacman-game.test
```

Question q4: 5/5

Question q5
=====

```
*** Method not implemented: betterEvaluationFunction at line 415 of multiAgents.py
*** FAIL: Terminated with a string exception.
```

Question q5: 0/6

Finished at 23:36:39

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