CSE537 – Artificial Intelligence
Project 2 – Report

Rajesh Prabhakar **SBU ID: 112872762**
Udit Gupta **SBU ID: 112715403**

Question 1 – Stats

Python command	python autograder.py -q q1 --no-graphics
Average Score	1209.2
Running time	5.0 seconds

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

Pacman emerges victorious! Score: 1208
Pacman emerges victorious! Score: 1132
Pacman emerges victorious! Score: 1208
Pacman emerges victorious! Score: 1152
Pacman emerges victorious! Score: 1222
Pacman emerges victorious! Score: 1125
Pacman emerges victorious! Score: 1096
Pacman emerges victorious! Score: 1215
Pacman emerges victorious! Score: 1022
Pacman emerges victorious! Score: 1209
Average Score: 1158.9
Scores:        1208.0, 1132.0, 1208.0, 1152.0, 1222.0, 1125.0, 1096.0, 1215.0, 1022.0, 1209.0
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)
***     1158.9 average score (2 of 2 points)
```

For the evaluation function we used reciprocal of sum of min and max distance from the food list. To avoid pacman encountering ghost we sent negative values for all the actions of pacman which would land it in ghost position.

Question 2 – Stats

Python command	python autograder.py -q q2 –no-graphics
Average Score	84.0
Running time	2.0 seconds

```
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
```

Here we recursively found the min and max values for ghosts and pacman respectively and took the path actions which results in pacman having max value.

Question 3 – Stats

Python command	python autograder.py -q q3 --no-graphics
Average Score	84.0
Running time	1 second

```
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
```

Here also we recursively found the min and max values for ghosts and pacman respectively and took the path actions which results in pacman having max value. But we will prune the branches for which beta value is less than the alpha values.

Question 4 – Stats

Python command	python autograder.py -q q4 --no-graphics
Average Score	84.0
Running time	1 second

```
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
```

Python command	python pacman.py -p AlphaBetaAgent -l trappedClassic -a depth=3 -q -n 10
Average Score	-501
Record	Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss

As expected, the pacman died every time using the AlphaBeta approach.

Python command	python pacman.py -p ExpectimaxAgent -l trappedClassic -a depth=3 -q -n 10
Average Score	325.2
Record	Win, Win, Win, Win, Loss, Win, Win, Loss, Win, Win

With the Expectimax approach pacman won 8 out 10 games with an average score of 325.2.

Analysis:

The Expectimax agent performs better than the MinimaxAgent and AlphaBetaAgent in minimaxClassic and trappedClassic mazes and pacman emerges victorious only with Expectimax agent.

The running time for MinMax agent is better than the Reflex agent. Expectimax and AlphaBetaAgent (1 second) does better than MinMax (2 seconds) in time.

The Reflex agent always wins as it plays with deterministic outcome values whereas the MinMax, AlphaBeta and Expectimax plays with stochastic outcome values hence have a failure chance to.