1. Enter the score from the auto grader.

Q1: 4/4  
Q2: 5/5  
Q3: 5/5  
Q4: 5/5  
Q5: 5/6  
Total: 24/25

2. Describe briefly how you solved the problems. Make sure that you describe in detail your evaluation function! You can include your code.

Q1: For the reflex agent, I considered pacman’s distance from ghosts, his distance to pellets, and his action. I wanted pacman to always be at least 1 step away from ghosts unless they were scared (and therefore edible). I used the manhattan distance to calculate the distance to every ghost and picked the minimum distance. If this distance was less than 2, I returned a large negative number. I also punished pacman for stopping, so if his action was STOP I also returned a large negative number. Lastly I also calculated the distance to the closest pellet or scared ghost and returned the negative of this distance. Therefore, the score becomes larger (and better) the closer to pellets/scared ghosts that pacman is.

Q2: I implemented my minimax based on the pseudocode shown in class. The recursion was done using self.depth and counting down after all agents were iterated through. We want to maximize the score for pacman while minimizing ghost scores.

Q3: My alphabeta function was very similar to minimax. Except that this time I counted up from 0 to self.depth. I start alpha at negative infinity and beta at positive infinity and update the values as alphabeta recurses. The alpha and beta are used as tie breakers.

Q4: In the expectimax function, we no longer assume that the ghosts will always make the smartest choice. Instead

Q5: In my better evaluation function, if pacman won I return positive infinity and if pacman lost, I return negative infinity. Next, I calculate the distance to the closest ghost. The further from the ghost pacman is the better, so I add the distance to the final score returned. Next, I calculate the distance to the closest pellet/scared ghost. The further away from food that pacman is the worse, so I subtract this distance from the final score. If the distance to the closest pellet/scared ghost is zero, I return positive infinity because this means that pacman can eat and increase his score. Lastly I also take into account the amount of food left in the maze. The more food left the worse, so I subtract this number from the final score. I also multiply these values by a weight of 2, because through testing this resulted in better pacman performance. My code is shown below:

newPos = currentGameState.getPacmanPosition()

newGhostStates = currentGameState.getGhostStates()

pellets = currentGameState.getFood().asList()

if currentGameState.isWin():

return float("inf")

if currentGameState.isLose():

return float("-inf")

# we always want to be at least 1 away from any ghosts unless we can eat them

distanceToClosestGhost = max([1] +[util.manhattanDistance(ghost.getPosition(), newPos) for ghost in newGhostStates if ghost.scaredTimer == 0])

# we want pacman to eat pellets and ghosts if he can

distanceToClosestPellet = min([util.manhattanDistance(pellet, newPos) for pellet in pellets +

[ghost.getPosition() for ghost in newGhostStates if ghost.scaredTimer != 0]]) \

if len(pellets) != 1 else util.manhattanDistance(pellets[0], newPos)

if distanceToClosestPellet == 0:

return float("inf")

foodLeft = currentGameState.getNumFood()

pelletWeight, ghostWeight, foodWeight = 2, 2, 2

return (scoreEvaluationFunction(currentGameState)) + (-distanceToClosestPellet \* pelletWeight) + (distanceToClosestGhost \* ghostWeight) + (-foodLeft \* foodWeight)

3. Paste the output from auto grader.

Starting on 10-15 at 20:18:09

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

===========

Pacman emerges victorious! Score: 1098

Pacman emerges victorious! Score: 908

Pacman emerges victorious! Score: 789

Pacman emerges victorious! Score: 850

Pacman emerges victorious! Score: 1078

Pacman emerges victorious! Score: 843

Pacman emerges victorious! Score: 967

Pacman emerges victorious! Score: 933

Pacman emerges victorious! Score: 923

Pacman emerges victorious! Score: 1140

Average Score: 952.9

Scores: 1098.0, 908.0, 789.0, 850.0, 1078.0, 843.0, 967.0, 933.0, 923.0, 1140.0

Win Rate: 10/10 (1.00)

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

\*\*\* FAIL: test\_cases\q5\grade-agent.test (5 of 6 points)

\*\*\* 952.9 average score (1 of 2 points)

\*\*\* Grading scheme:

\*\*\* < 500: 0 points

\*\*\* >= 500: 1 points

\*\*\* >= 1000: 2 points

\*\*\* 10 games not timed out (1 of 1 points)

\*\*\* Grading scheme:

\*\*\* < 0: fail

\*\*\* >= 0: 0 points

\*\*\* >= 10: 1 points

\*\*\* 10 wins (3 of 3 points)

\*\*\* Grading scheme:

\*\*\* < 1: fail

\*\*\* >= 1: 1 points

\*\*\* >= 5: 2 points

\*\*\* >= 10: 3 points

### Question q5: 5/6 ###

Finished at 20:18:33

Provisional grades

==================

Question q1: 4/4

Question q2: 5/5

Question q3: 5/5

Question q4: 5/5

Question q5: 5/6

------------------

Total: 24/25

Your grades are NOT yet registered. To register your grades, make sure

to follow your instructor's guidelines to receive credit on your project.

4. Paste the git log.

$ git log

commit d039f161a7ed49c54ab929c658f803db0be04843

Author: Rebeca Amaya <rebeca\_amaya@bellsouth.net>

Date: Sun Oct 7 17:08:55 2018 -0400

almost working betterevaluationfunction

commit 84df3665b4c410e821d5ec36298161e71178daf1

Author: Rebeca Amaya <rebeca\_amaya@bellsouth.net>

Date: Sun Oct 7 13:47:05 2018 -0400

added solutions for alphabeta and expectimax

commit fb5de7ca0c44660d16332636269ebb7aa0872ea5

Author: Rebeca Amaya <rebeca\_amaya@bellsouth.net>

Date: Thu Oct 4 21:14:29 2018 -0400

almost working solution for alphabeta

commit 846adad4cfa7ccf934faf729f5a4bc19d73b420f

Author: Rebeca Amaya <rebeca\_amaya@bellsouth.net>

Date: Tue Oct 2 20:10:01 2018 -0400

added solution for minimax

commit 8880f1eee915f1478ebaff4016f57e305a7eec92

Author: Rebeca Amaya <rebeca\_amaya@bellsouth.net>

Date: Sun Sep 30 18:10:40 2018 -0400

added solution for reflex agent