Homework #2 Report

土木所交通組 R12521502 陳冠言

Part A： Show your autograder results and describe each algorithm.

Q1. Reflex Agent

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

Pacman emerges victorious! Score: 1171
Pacman emerges victorious! Score: 1255
Pacman emerges victorious! Score: 1243
Pacman emerges victorious! Score: 1256
Pacman emerges victorious! Score: 1248
Pacman emerges victorious! Score: 1247
Pacman emerges victorious! Score: 1251
Pacman emerges victorious! Score: 1255
Pacman emerges victorious! Score: 1252
Pacman emerges victorious! Score: 1228
Average Score: 1240.6
Scores:        1171.0, 1255.0, 1243.0, 1256.0, 1248.0, 1247.0, 1251.0, 1255.0, 1252.0, 1228.0
Win Rate:      10/10 (1.00)
Record:        Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
*** PASS: test_cases\q1\grade-agent.test (30.0 of 30.0 points)
***     1240.6 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: 30/30 ###


Finished at 18:15:09

Provisional grades
==================
Question q1: 30/30
------------------
Total: 30/30
```

對於每個 food 的位置，計算 Pacman 到該食物的曼哈頓距離，並找出最短的距離，同時對於每個 ghost 的位置，計算 Pacman 到該 ghost 的曼哈頓距離，另外檢查 pacman 周圍 ghost 的接近程度得到 Pacman 到所有 ghost 的總距離，最後計算綜合的結果以形成最終的評分，該分數用於指導 Pacman 在遊戲中做出決策。

Q2. Minimax

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

*** PASS: test_cases\q2\0-eval-function-lose-states-1.test
*** PASS: test_cases\q2\0-eval-function-lose-states-2.test
*** PASS: test_cases\q2\0-eval-function-win-states-1.test
*** PASS: test_cases\q2\0-eval-function-win-states-2.test
*** 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 0 seconds.
*** Won 0 out of 1 games. Average score: 84.000000 ***
*** PASS: test_cases\q2\8-pacman-game.test

### Question q2: 30/30 ###
```

首先對於 Pacman 可用行動使用 minimax 函數，並計算其值。然後，選擇具有最大值的行動作為最終的最佳行動，這代表著 Pacman 在搜索深度內的最佳行動。最後，返回所選擇的最佳行動。其中，建立 Minimax function，是為了使用遞歸方式來搜索找出最佳的行動。每次遞歸步驟中，會先檢查是否達到終止條件，如果是，則返回當前遊戲狀態的評估值。如果尚未達到終止條件，則根據 Pacman 或 Ghost 來選擇最大化或最小化數值。若選擇 Pacman，則根據 Pacman 的行動，使用 minimax function 並傳遞生成的後繼遊戲狀態，然後取最大值。若選擇 ghost，則根據 ghost 的行動使用 minimax 函數並取最小值。

Q3. Alpha-Beta Pruning

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

*** PASS: test_cases\q3\0-eval-function-lose-states-1.test
*** PASS: test_cases\q3\0-eval-function-lose-states-2.test
*** PASS: test_cases\q3\0-eval-function-win-states-1.test
*** PASS: test_cases\q3\0-eval-function-win-states-2.test
*** 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 0 seconds.
*** Won 0 out of 1 games. Average score: 84.000000 ***
*** PASS: test_cases\q3\8-pacman-game.test

### Question q3: 30/30 ###
```

將原本的 minimax function 拆成 max_value 和 min_value function，分別代表 Pacman 和 Ghost 的行動，max_value 嘗試最大化其值，min_value 嘗試最小化其值，同時設定 a 和 b 兩參數分別去代表 max_value 中 Pacman 目前獲得的最大值及 min_value 中 Ghost 目前獲得的最小值，且透過設定 a 和 b 值可讓演算法在搜尋過程中，動態地刪除一些不可能產生最佳解的節點，從而減少搜尋的數量。

Part B： Describe the idea of your design about evaluation function in Q1.

Answer：

主要目的為幫助 agent 在選擇行動時做出最佳的決策，使得 agent 能夠有效地移動並最大化其獲得的分數，因此需要評估的指標包括 Pacman 到最近 food 的距離和 Pacman 到 ghost 的距離，並結合遊戲狀態的分數，以獲得對當前狀況的總體評價。

Part C： Demonstrate the speed up after the implementation of pruning.

Answer：

主要是對 Minimax 演算法進行優化，利用已經得到的數值來減少對某些子節點的搜索，即利用 a 表示 max_value 中的最大值和 b 表示 min_value 節點中的最小值，也因此 agent 只搜索了部分遊戲狀態的空間，從而大大提高了搜尋效率。

Appendix

python autograder.py

```
Finished at 18:18:57

Provisional grades
==================
Question q1: 30/30
Question q2: 30/30
Question q3: 30/30
------------------
Total: 90/90
```