HEURISTICS ANALYSIS

Adversarial Game Playing Agent for Isolation

Submitted By: Mayank Prakash

Udacity Artificial Intelligence Nanodegree

Introduction:

Isolation is a deterministic two player board game in which a player or agent alternately moves single piece from one cell to another on the board. An occupied cell stays blocked for the remainder of the game. The first player who doesn't have a valid position to move on the board loses. In this specific project, the movements of each of the pieces are restricted to an L-shape.

In order to implement an agent that plays the isolation game, following algorithms are implemented:

1. Minimax Search
2. Alpha-Beta Pruning
3. Iterative Deepening with Alpha-Beta

In order to score, evaluation function is implemented as heuristics in order to evaluate the performance of the agent. The heuristics defined are:

1. custom_score():
2. custom_score_2():
3. custom_score_3():

Evaluation:

tournament.py is run in order to evaluate the performance of the heuristic defined as 'Student'. custom_score() contains the best performing heuristic function. The opponent against which the 'Student' agent plays are as follows:

1. Random
2. MM_Open
3. MM_Center
4. MM_Improved
5. AB_Open
6. AB_Center
7. AB_Improved

Opponents:

1. Random: agent that randomly chooses from list of available options.
2. MM_Open: Minimax player with open_move_score heuristics with search depth 3.
3. MM_Center: Minimax player with center_move_score heuristics with search depth 3.
4. MM_Improved: Minimax player with improved_score heuristics with search depth 3.
5. AB_Open: AlphaBeta player using Iterative Deepening Alpha-Beta Search with open_move_score heuristics
6. AB_Center: AlphaBeta player using Iterative Deepening Alpha-Beta Search with center_move_score heuristics
7. AB_Improved: AlphaBeta player using Iterative Deepening Alpha-Beta Search with improved_score heuristics

Heuristics Functions:

1. Student 1: weighted heuristics in which for every wrong guess of agent or moves that leads to reduced chances of winning, the agent is penalized. It is evaluated as:

$$score = agents\_move^2 - 1.5 * opponents\_move^2$$

2. Student 2: this is more of a defensive approach where the agent tries to maximize its own legal number of moves. The evaluation is done as follows:

$$score = 1.5 * agents\_move - opponents\_move$$

3. Student 3: this is more of an aggressive approach where the agent tries chasing its opponent to try and minimize the opponent's legal moves. The evaluation is done as follows:

$$score = agents\_move - 1.5 * opponents\_move$$

Performance:

1. Student 1: custom_score_3()

```
*************************
  Playing Matches
*************************

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                      Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random      9  |  1     8  |  2     9  |  1     9  |  1
   2       MM_Open     6  |  4     8  |  2     8  |  2     7  |  3
   3      MM_Center    8  |  2    10  |  0     8  |  2     7  |  3
   4     MM_Improved   5  |  5     7  |  3     6  |  4     6  |  4
   5       AB_Open     5  |  5     8  |  2     3  |  7     3  |  7
   6      AB_Center    5  |  5     4  |  6     6  |  4     5  |  5
   7     AB_Improved   4  |  6     6  |  4     8  |  2     5  |  5
----------------------------------------------------------------------
          Win Rate:      60.0%        72.9%        68.6%        60.0%
```

2. Student 2: custom_score_2()

```
*************************
  Playing Matches
*************************

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                      Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random      9  |  1    10  |  0     9  |  1    10  |  0
   2       MM_Open     9  |  1     5  |  5     7  |  3     7  |  3
   3      MM_Center    9  |  1     8  |  2    10  |  0     8  |  2
   4     MM_Improved   7  |  3     4  |  6     6  |  4     6  |  4
   5       AB_Open     4  |  6     5  |  5     5  |  5     6  |  4
   6      AB_Center    7  |  3     7  |  3     5  |  5     5  |  5
   7     AB_Improved   3  |  7     7  |  3     5  |  5     4  |  6
----------------------------------------------------------------------
          Win Rate:      68.6%        65.7%        67.1%        65.7%
```

3. Student 3: custom_score()

```
                    *************************
                        Playing Matches
                    *************************

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                      Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random      10  |  0     10  |  0     10  |  0     9  |  1
   2       MM_Open      9  |  1      9  |  1      6  |  4     9  |  1
   3      MM_Center     9  |  1      8  |  2      8  |  2     8  |  2
   4     MM_Improved    7  |  3      6  |  4      7  |  3     7  |  3
   5       AB_Open      5  |  5      5  |  5      7  |  3     6  |  4
   6      AB_Center     6  |  4      7  |  3      6  |  4     7  |  3
   7     AB_Improved    4  |  6      5  |  5      6  |  4     6  |  4
----------------------------------------------------------------------
          Win Rate:      71.4%        71.4%        71.4%        74.3%
```

Result:

The result of the heuristics obtained for one instance is shown. The highest win percentage is obtained in case of aggressive player, i.e. Student 3 which is used as the heuristics evaluation and is selected as custom_score for the game playing agent.