Udacity Artificial Intelligence Nanodegree

Heuristic Analysis

For an adversarial game playing agents for Isolation

1. Maximising the number of moves available for the player

This heuristic is the simplest implementation which aims to maximise the number of available moves for the player for every given game state. The algorithm is as follows:

```
moves available for the player - moves available for the opponent
```

The results show that using this heuristic function, the algorithm has won 6 out of 10 games against AB_Improved. And has had an overall 81.4% winning rate against various algorithmic opponents.

2. Maximising the number of moves available for the player - weighted

This heuristic is a slight adjustment to the first heuristic. It attempts to define a more aggressive play method by increasing the weight of the opponent move when calculating the number of available moves for the player. The algorithm is as follows:

```
moves available for the player - moves available for the opponent*1.5
```

The results show that using this heuristic function, the algorithm has won 6 out of 10 games against AB_Improved. And compared to the previous heuristic, has had an poorer overall winning rate of 75.7% against various algorithmic opponents.

3. The ratio of player’s moves to the opponent's

The last heuristic takes a different approach by calculating the value of each game state by calculating the ratio of moves for every player and opponent.

```
moves available for the player / moves available for the opponent
```

The results show that using this heuristic function, the algorithm has only won 2 out of 10 games against AB_Improved. And compared to the previous heuristic, has had an poorer overall winning rate of 51.4% against various algorithmic opponents.

Results

Having run the tests to evaluate each of the above heuristics, the basic utility function that aims to maximise the player's available moves (Heuristic 1) performs much better than the more sophisticated heuristics (Heuristic 1 and 2).

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	7	3
2	MM_Open	7	3	10	0	7	3	3	7
3	MM_Center	10	0	10	0	9	1	9	1
4	MM_Improved	8	2	8	2	6	4	4	6
5	AB_Open	6	4	4	6	6	4	5	5
6	AB_Center	9	1	9	1	9	1	6	4
7	AB_Improved	6	4	6	4	6	4	2	8
	Win Rate:	80.0%		81.4%		75.7%		51.4%	

Based on the results here, using Heuristic 1 is recommended. In addition to the higher win rate (6 percent more than Heuristic 2 and 30 percent more than Heuristic 3), the implementation is a very simple and easy to reason about.