

## Heuristic Analysis

*****									
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	10	0
2	MM_Open	7	3	9	1	9	1	8	2
3	MM_Center	10	0	10	0	6	4	8	2
4	MM_Improved	7	3	8	2	9	1	8	2
5	AB_Open	8	2	5	5	5	5	8	2
6	AB_Center	6	4	7	3	6	4	6	4
7	AB_Improved	5	5	5	5	4	6	5	5
-----									
Win Rate:		75.7%		77.1%		70.0%		75.7%	

### Custom score 1

The heuristic simply computes the difference in number of legal moves left between the own player and the opponent player moves multiplied by a factor of two. If player and its opponent have the same number of moves, then the returned value is zero. If the returned value is positive (negative), then player is doing better (worse) than its opponent. If the returned value is "inf" ("-inf"), then player has won (lost) the game. Its benefits are that it's easily interpretable and fast to compute. On the downside, it doesn't exploits the specific mechanism of the game: for instance it doesn't consider the positional advantage .

### Custom score 2

The second heuristic is a variation of the first one. Instead of computing the difference, we compute the ratio between the moves available for the own player and the opponent, the division error is handled by adding a 1 to the denominator. This Heuristic tend to choose portion of the table where our agent has more moves available. Also this heuristic has downsides, in the sense that it doesn't exploits the specific mechanism of the game, for instance it doesn't consider the positional advantage.

### Custom score 3

This heuristic computes the differences in number of moves available for both players, but if players have the same number of moves we also look for a positional advantage by computing the manhattan distance from the center of the board. We are assuming that moving closer to the center is a better choice than moving near the edge of the board. The choice of the manhattan distance is due to the faster runtime execution compared to the euclidean distance.

## Results

The heuristics that works best is the `custom_score_1`. This heuristics simply computes the difference in number of moves between the two players. This evaluation function causes the computer player to seek moves with the most options while trying to get in the way of the opponent's moves. In this case we put some weight in opponent part of the formula to try to encourage more aggressive game play. In particular, multiplying `opponent_moves` by two will cause our computer player to chase after the opponent. Moreover this evaluation function it is simple and easy to compute, instead a possible downside is that it doesn't exploit any specific mechanism of the game.

Regarding the third heuristic, it seems that considering also that try to stay close to the center of the board it doesn't improve the performance very much.