Heuristic Evaluation

Custom Score 1
Takes the number of possible moves for opponent and takes double this value away from the number of possible moves for the current player. The idea behind this is to encourage moves thats minimises your opponents moves. This can been seen as moves that block your opponents moves.

Custom Score 2
This heuristic uses a similar approach to the above heuristic but the player plays more aggressively as the game goes on. It rewards minimising opponents moves as the game go one. The value used to determine aggression is one of three values that is determined by the percentage of moves left on the board.

Custom Score 3
This heuristics still uses the difference between the number of moves the opponent has and the number of moves the current player has. It however maximised the number of moves the current player has.

Result
Tests from the tournament python script can be seen below

	AB_Improved	AB_Custom	AB_custom2	AB_Custom3
	Win Rate	Win Rate	Win Rate	Win Rate
Random	70%	80%	90%	80%
MM_Open	90%	90%	80%	50%
MM_Center	80%	60%	70%	80%
MM_Improved	50%	60%	70%	70%
AB_Open	70%	40%	80%	60%
AB_Center	40%	50%	50%	80%
AB_Improved	40%	40%	40%	50%
Total Win Rate	62.9%	60.0%	68.6%	67.1%

From the results it seems that all customer heuristics performed optimally against a random opponent. In general It can be said that the alphabeta players out performed the minimax algorithm set. Except in the case where AB_Improved came up against the MM_Improved where the algorithms came up tied, which is no surprise as alphabeta is simply a faster minimax algorithm.

When my custom heuristics came up against other alphabeta algorithms the results varied. Custom heuristic 2 and custom heuristic 3 seemed to come out on top against AB_OPEN and AB_improved. The interesting results came from when my custom heuristics came up against AB_Improved. Only AB_Custom3 had an equal number or wins. Where as the other custom

heuristic did not win the majority. This is disappointing as a good way of evaluating heuristics is to put it up against your current best heuristic (AB_Improved).

From the data known AB_Custom2 is the evaluation function that should be used. Because :

- it had the overall highest win rate.
- It is trivial to implement
- It performed consistently under the given time constraints of 150 milli seconds
- it addresses of the issues faced in custom score 1, the agent shouldn't always look to block opponents from the very start. As in the first couple of moves this is extremely hard. By increasing aggression towards the end of the game, you are awarding blocking moves a higher score when they are more like to happen, towards the end of the game.