

Heuristic Analysis

During this project, I developed a heuristic that would take into account a score that I called the centrality. This number would consist of using the distance formula to give the player a score based on how close the next move was to the center. I thought that sense the next move would be close to the center that would give the player a positional advantage.

To test this heuristic, I ran `python run_match.py -f -r 10` for three rounds against both Minimax agents and Greedy agents so that I could get an accurate result.

	Round 1	Round 2	Round 3	Average	
					Central Depth
Minimax	65	65	65	65	4
Greedy	87.5	82.5	85	85	
					Normal
Minimax	62.5	65	70	65.8333333	Depth 4
Greedy	87.5	90	82.5	86.6666667	
					Central Depth
Minimax	55	50	47.5	50.8333333	3
Greedy	90	87.5	80	85.8333333	
					Normal
Minimax	55	47.5	50	50.8333333	Depth 3
Greedy	82.5	85	82.5	83.3333333	

Based on my results, my custom heuristic was not very advantageous compared to the normal `#my_moves - #opp_moves`. When the depth was set to 3, the centrality heuristic performed slightly better than the normal heuristic. When the depth was set to 4, the normal heuristic performed better. When the depth was increase to 4, the time that each match took was increased but the percentage of wins was greatly increased. The increase in depth did not change the agent's ability to play in the allotted amount of time. If my computer had faster processing speeds I would be able to test the program with even more increased depths.