Heuristic Analysis

Heuristic 1:

Heuristic 1 just maximizes the number of moves we have against our opponent at leaves of the search tree. This works under the assumption that having more moves than you opponent is winning.

$$score = number\ of\ our\ moves - number\ of\ our\ opponents\ moves$$

Heuristic 2:

Heuristic 2 attempts to prevent the agent from getting stuck in a corner in the late game. The corner avoiding behavior is achieved by first counting the number of moves that are in the corners for each player, and then scaling the difference between our opponent's corner moves and our corner moves by minimum of the fraction of total space to blank spaces. The scaling factor, *game_stat*e, allows the corner strategy to not dominate until late game where it becomes more meaningful, as fewer spaces are available. *game_state* is also capped at 6 to prevent the corner moves from over dominating.

$$game\ state = min(\frac{spaces\ left}{total\ spaces}, 6)$$

$$score = \#\ our\ moves - \#\ opponents\ moves + game\ state * (\#\ opponents\ corners - \#\ our\ corners)$$

Heuristic 3:

Heuristic 3 attempts the same behaviour as Heuristic 2, but uses the distance to the center of the end moves instead of the corner moves. Its behaviour maximizes the number of moves we have over our opponent while the board is less than 75% filled, and when the board is 75% filled or the number of moves we have equals our opponents the heuristic will reward moves that are closer to the center of the board in *block distance* by maximizing the opponents block distance over our own.

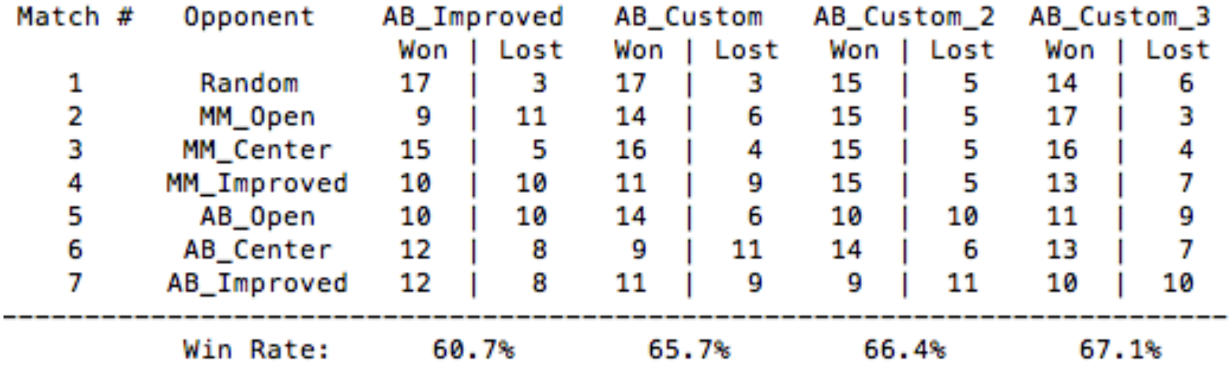
$$\begin{aligned} block\ distance = &|player\ position\ y\ coordinate - center\ y\ coordinate| \\ &+ |player\ position\ x\ coordinate - center\ x\ coordinate| \end{aligned}$$

$$score\ if < 75\%\ used = \#\ of\ our\ moves - \#\ of\ opponents\ moves$$

$$score\ if \geq 75\%\ used = block\ distance(opponent) - block\ distance(us)$$

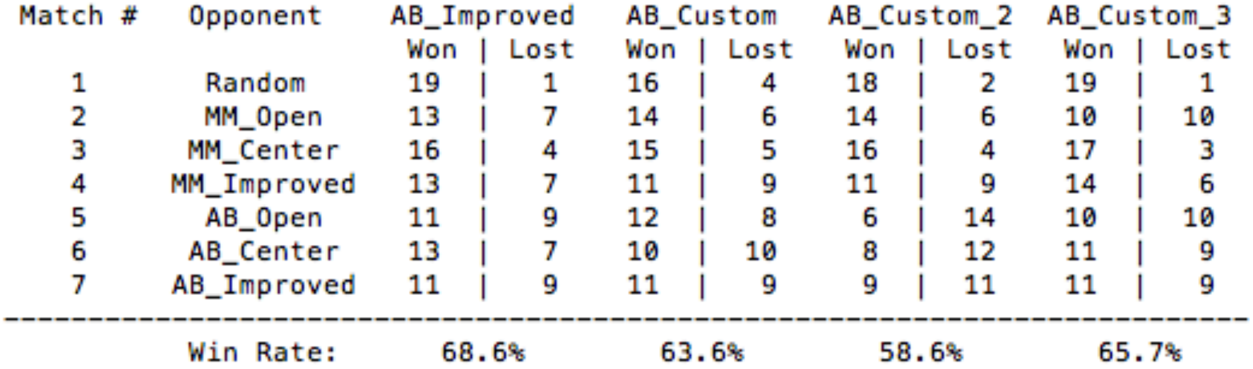
Decisions:

I would choose between Heuristic 2 and Heuristic 3. Heuristic 3 seems to perform best (see Figures 1 and 2 for example runs from tournament.py), but it is just Heuristic 1 with an extra strategy tacked on. There might be additional strategies that could be added to improve it, such as the implementation of opening strategies.



```
 Match #     Opponent     AB_Improved    AB_Custom     AB_Custom_2   AB_Custom_3
                          Won | Lost    Won | Lost    Won | Lost    Won | Lost
    1         Random      17  |   3     17  |   3     15  |   5     14  |   6
    2         MM_Open      9  |  11     14  |   6     15  |   5     17  |   3
    3        MM_Center    15  |   5     16  |   4     15  |   5     16  |   4
    4       MM_Improved   10  |  10     11  |   9     15  |   5     13  |   7
    5         AB_Open     10  |  10     14  |   6     10  |  10     11  |   9
    6        AB_Center    12  |   8      9  |  11     14  |   6     13  |   7
    7       AB_Improved   12  |   8     11  |   9      9  |  11     10  |  10
--------------------------------------------------------------------------
           Win Rate:      60.7%         65.7%         66.4%         67.1%
```

Figure 1: Example Run 1



```
 Match #     Opponent     AB_Improved    AB_Custom     AB_Custom_2   AB_Custom_3
                          Won | Lost    Won | Lost    Won | Lost    Won | Lost
    1         Random      19  |   1     16  |   4     18  |   2     19  |   1
    2         MM_Open     13  |   7     14  |   6     14  |   6     10  |  10
    3        MM_Center    16  |   4     15  |   5     16  |   4     17  |   3
    4       MM_Improved   13  |   7     11  |   9     11  |   9     14  |   6
    5         AB_Open     11  |   9     12  |   8      6  |  14     10  |  10
    6        AB_Center    13  |   7     10  |  10      8  |  12     11  |   9
    7       AB_Improved   11  |   9     11  |   9      9  |  11     11  |   9
--------------------------------------------------------------------------
           Win Rate:      68.6%         63.6%         58.6%         65.7%
```

Figure 2: Example Run 2