Satish Boggarapu

Udacity AIND – Isolation Heuristics Analysis

Custom Score 1:

My best custom score gets the current player's and opponent's legal moves and squares. Then the function returns the difference between the two values.

Custom Score 2:

This custom score function returns the difference between the number of legal moves of both players

Custom Score3:

This custom score function returns the number of legal moves of the player.

Result:

```
                    **************************
                         Playing Matches
                    **************************

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                      Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random      6  |   4    10  |   0    8  |   2    10  |   0
   2       MM_Open     9  |   1     7  |   3    5  |   5     6  |   4
   3      MM_Center    6  |   4     9  |   1   10  |   0     8  |   2
   4     MM_Improved   5  |   5     7  |   3    8  |   2     8  |   2
   5       AB_Open     5  |   5     4  |   6    4  |   6     4  |   6
   6      AB_Center    7  |   3     8  |   2    4  |   6     5  |   5
   7     AB_Improved   3  |   7     3  |   7    8  |   2     2  |   8
--------------------------------------------------------------------------
          Win Rate:      58.6%        68.6%        67.1%        61.4%
```

After testing the custom score functions it was clear that the custom_score() method consistently had higher win rates then the other functions. There were test runs where the custom_score() function had a win rate of above 70%. I believe squaring the number of legal moves for each player emphasized nodes with a higher number of moves as possible winning nodes. For this reason, I choose to use this my best custom score function.