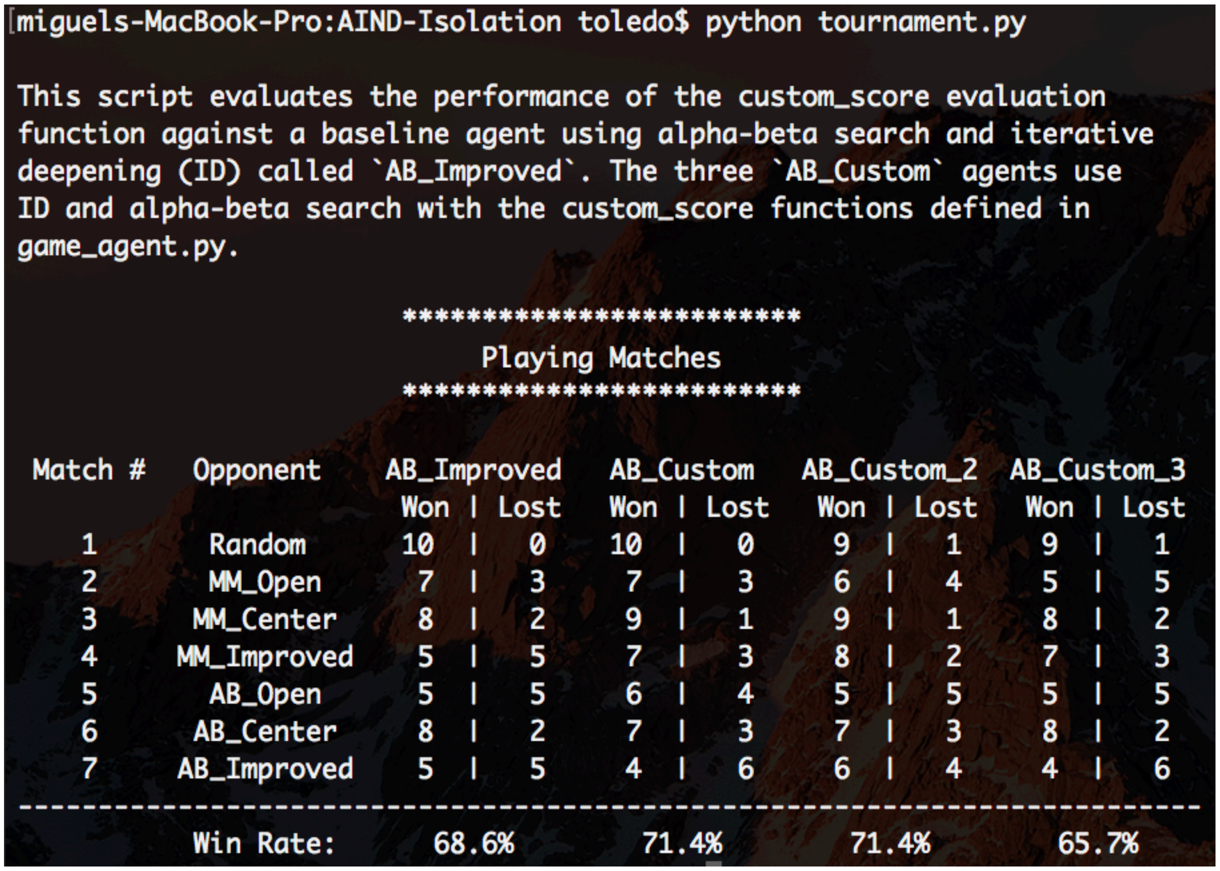
Heuristic Analysis

First of all, I was not expecting that coming up a heuristic would be as difficult as it proved to be. My approach to solving was first looking at sample_players.py and examining how those heurisitics were implemented. After that I had a somewhat good understanding about how to approach the heuristic. Below is a table of my results.



```
miguels-MacBook-Pro:AIND-Isolation toledo$ python tournament.py

This script evaluates the performance of the custom_score evaluation
function against a baseline agent using alpha-beta search and iterative
deepening (ID) called `AB_Improved`. The three `AB_Custom` agents use
ID and alpha-beta search with the custom_score functions defined in
game_agent.py.

                        *************************
                             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     9  |   1     9  |   1
    2       MM_Open      7  |   3     7  |   3     6  |   4     5  |   5
    3      MM_Center     8  |   2     9  |   1     9  |   1     8  |   2
    4     MM_Improved    5  |   5     7  |   3     8  |   2     7  |   3
    5       AB_Open      5  |   5     6  |   4     5  |   5     5  |   5
    6      AB_Center     8  |   2     7  |   3     7  |   3     8  |   2
    7     AB_Improved    5  |   5     4  |   6     6  |   4     4  |   6
--------------------------------------------------------------------------
           Win Rate:      68.6%        71.4%        71.4%        65.7%
```

Analysis 1:

For my first heuristic, my function would calculate the difference between the current player's and opponent's position from the center of the grid. From playing the game as you are closer to the center you have more possibilities to be around legal moves. As you approach the edges you remove possibilities from the number of legal moves you have. Calculating the distance was merely applying the distance formula for each player's current location coordinates. After playing around with this heuristic I also added the difference of available moves of the player and opponent to the heuristic which resulted in better performance. My thinking behind this, is if you have more available moves than your opponent then you should have a better chance of winning. This resulted as the best heuristic overall and I believe that was the case because of the simplicity yet effectiveness of the heuristic. As a user, you can tell that the board has more possible values if you're closer to the center. And as a bonus if you have more possible moves then you have a greater chance of not losing.

Analysis 2:

My idea behind this heuristic was quite simple. If the current player has moves that the opposing opponent has then this would work in their favor by having the possibility to make that move and thus eliminating a possible move from the opponent. Calculating this would consist of running a for loop and every time an opponent move appears in the current player's possible moves add one to a counter. At the end of the function just return the difference of possible moves between the player and opponent plus the number of moves they share. This resulted as the second best heuristic overall even though in the table of results it appears as a tie with the first heuristic.

Analysis 3:

My last heuristic did not consist of having a clear reason as to why I chose this, instead this heuristic came from trial and error. Playing around with the possible values and changing numbers gave a mix range of results and ultimately due to time constraint I settled on the one I implemented. This heuristic adds the both the number of possible values from the current player and opponent and then subtracts this from the difference of the number of spaces available and the number of moves that have occurred in the game. This function on average resulted in a result of a little over 60% win rate.