Heuristics Analysis

The heuristics aspect of this project was definitely the most challenging part of the AIND to date. I was very surprised with how difficult it was to come up with a "good" heuristic. What I came to realize was that it's particularly difficult to beat a fast heuristic. Take the *improved_score* heuristic for example. It's fast due to its simplicity, *own_moves - opp_moves*. I think the reason a fast heuristic is so difficult to beat is because speed plays a big factor in a game like this, where a time constraint exists and a simple heuristic allows the search algorithm to search deeper in the tree than a more complicated heuristic.

I started the heuristics portion of the assignment by going through test runs *(game.play())* with my *AlphaBetaPlayer* and the provided *GreedyPlayer*. An observation I made was that it was an invariably bad decision for a player to move to a corner, especially later in a game. It was even bad at times to be caught along the perimeter if a partition was forming because it severally limited the number of legal moves available to the user. It seemed wise to stay away from the perimeter and the corners for as long as possible.

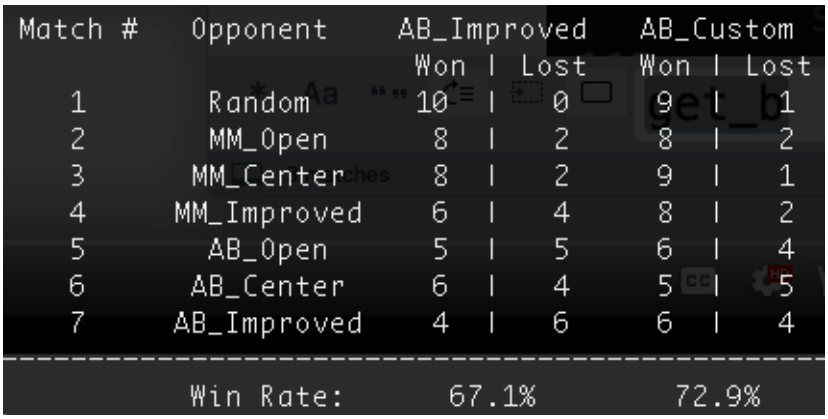
Heuristic 1

One of the lecture videos for this section discussed the effects of the formula, *my_moves - 2 * opp_moves* and how it caused the computer player to play more aggressively. I was intrigued by the idea of this and wondered what the results of a tag-like game style would be. Where one player would play aggressively while the other played defensively for half of the the game and then the two switched strategies. I tested both starting aggressively and defensively for the computer player to see which produced better results and it seemed that it was better to start off defensively and then finish the game aggressively (see Figure 1 and Figure 2).

Code:

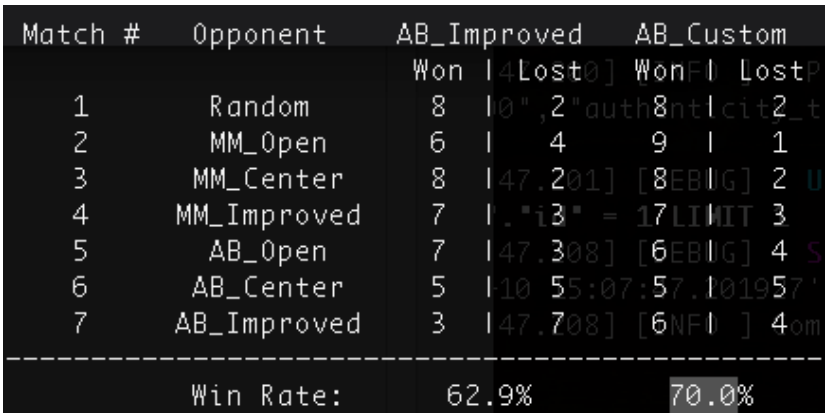
```
my_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
blanks_spaces = len(game.get_blank_spaces())
n = 3
m = 1

if blanks_spaces <= (game.height * game.width) / 2:
    m = 3
    n = 1

return (my_moves * n) - (opp_moves * m)
```

Figure 1 - Aggressive and then defensive - difference: 5.8%



Match #	Opponent	AB_Improved		AB_Custom	
		Won	Lost	Won	Lost
1	Random	10	0	9	1
2	MM_Open	8	2	8	2
3	MM_Center	8	2	9	1
4	MM_Improved	6	4	8	2
5	AB_Open	5	5	6	4
6	AB_Center	6	4	5	5
7	AB_Improved	4	6	6	4
	Win Rate:	67.1%		72.9%	

Figure 2 - Defensive and then aggressive - difference: 7.1%



Match #	Opponent	AB_Improved		AB_Custom	
		Won	Lost	Won	Lost
1	Random	8	2	8	2
2	MM_Open	6	4	9	1
3	MM_Center	8	2	8	2
4	MM_Improved	7	3	7	3
5	AB_Open	7	3	6	4
6	AB_Center	5	5	7	3
7	AB_Improved	3	7	6	4
	Win Rate:	62.9%		70.0%	

Heuristic 2

The main idea behind this heuristic was to penalize the computers moves that were located on the corners of the board and reward the opponent moves that were located on the corners. As previously stated, I noticed when during my trial runs of printing out the end result of game board, I noticed that corner moves often led to loses.

I tried two different techniques for this heuristic, one where I checked if the players current position was located on a corner, *game.get_player_location(player)*, and for the second I checked the number of future moves that were located on a corner. I had better results with the latter which is what you will see below.

Code:

```
my_moves = game.get_legal_moves(player)
opp_moves = game.get_legal_moves(game.get_opponent(player))

board_corners = [(0, 0), (0, (game.width - 1)), ((game.height - 1), 0), ((game.height - 1), (game.width - 1))]

my_in_corner = [move for move in my_moves if move in board_corners]
opps_in_corner = [move for move in opp_moves if move in board_corners]

return float(len(my_moves) - (2 * len(my_in_corner)) - len(opp_moves) + (2 * len(opps_in_corner)))
```

Heuristic 3

Here I am taking the difference between the computers available moves and the opponents available moves and then adding their difference in distance from the centre of the board. A board state where the computer has more moves available and is located closer to the centre of the board than the opponent will be rewarded v.s the alternative.

Code:

```
my_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))

w, h = game.width / 2., game.height / 2.
y, x = game.get_player_location(player)
opp_y, opp_x = game.get_player_location(game.get_opponent(player))

# Calculate the distance to centre for Max and Min player
distance_to_center = abs(w - y) + abs(h - x)
opp_distance_to_center = abs(w - opp_y) + abs(h - opp_x)

return float((my_moves - opp_moves) + (opp_distance_to_center - distance_to_center))
```

My recommendation

After running 100 matches, heuristic 3 seemed to be most effective, given that it beat AB_Improved by 2.9%. I think the reason it beats AB_Improved is because it takes two variables into its equation, the difference in the number of moves between the 2 players and the difference in distance from the centre for the 2 players. Both of these variables are vitally important to winning the game of isolation.

Final Results

	AB_Improved	AB_Custom	AB_Custom_2	AB_Custom_3
Round 1 - 20 matches	69.3%	71.4%	68.6%	71.4%
Round 2 - 20 matches	70.0%	72.9%	67.1%	70.0%
Round 3 - 20 matches	71.4%	70.7%	71.4%	72.9%
Round 4 - 20 matches	68.6%	70.7%	67.1%	75.0%
Round 5 - 20 matches	67.1%	64.3%	67.9%	71.4%
Average	69.3%	70.0%	68.4%	72.1%
Difference		0.7%	-0.9%	2.9%

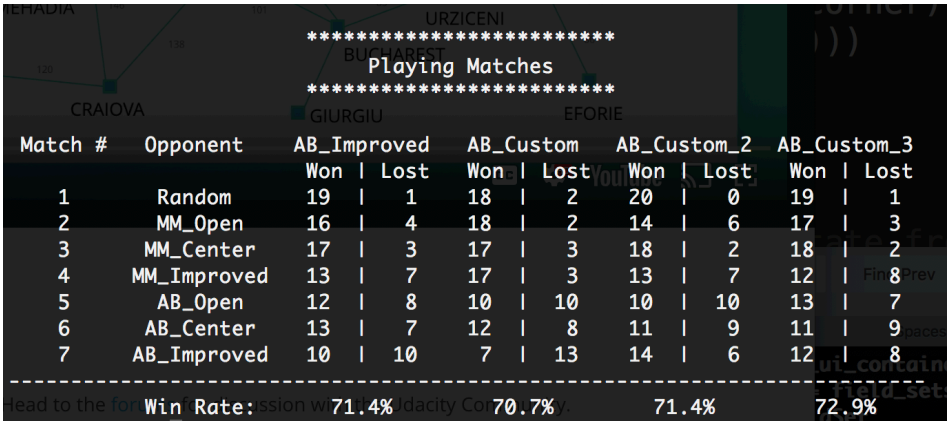
Round 1

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

 Match #    Opponent    AB_Improved    AB_Custom    AB_Custom_2    AB_Custom_3
                         Won | Lost    Won | Lost    Won | Lost    Won | Lost
    1         Random      19  |   1     19  |   1     19  |   1     20  |   0
    2        MM_Open      12  |   8     17  |   3     14  |   6     17  |   3
    3       MM_Center     14  |   6     17  |   3     19  |   1     20  |   0
    4     MM_Improved     17  |   3     15  |   5     13  |   7     12  |   8
    5        AB_Open      10  |  10     11  |   9      8  |  12      8  |  12
    6       AB_Center     13  |   7     13  |   7     12  |   8     12  |   8
    7     AB_Improved     12  |   8      8  |  12     11  |   9     11  |   9
----------------------------------------------------------------------------
          Win Rate:      69.3%        71.4%        68.6%        71.4%
```

Round 2

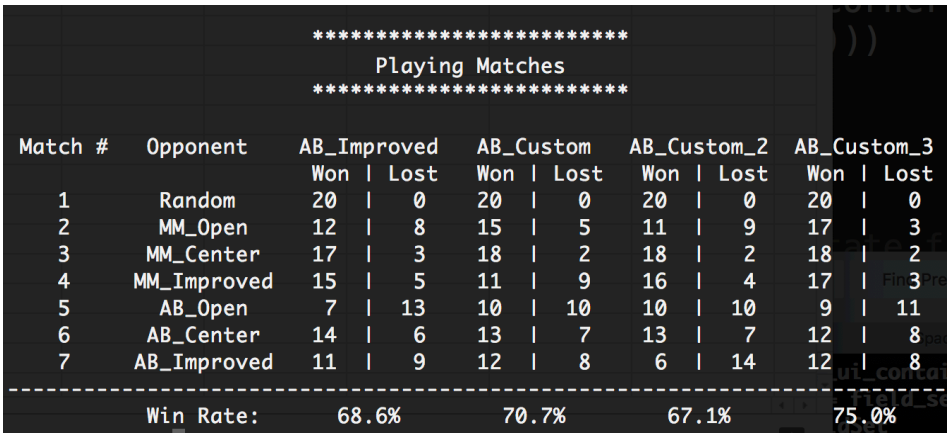
```
                    *************************
                        Playing Matches
                    *************************

 Match #    Opponent    AB_Improved    AB_Custom    AB_Custom_2    AB_Custom_3
                         Won | Lost    Won | Lost    Won | Lost    Won | Lost
    1         Random      20  |   0     20  |   0     20  |   0     20  |   0
    2        MM_Open      16  |   4     18  |   2     15  |   5     17  |   3
    3       MM_Center     17  |   3     17  |   3     16  |   4     18  |   2
    4     MM_Improved     14  |   6     16  |   4     15  |   5     14  |   6
    5        AB_Open       9  |  11     10  |  10      8  |  12     12  |   8
    6       AB_Center     12  |   8     11  |   9      9  |  11     10  |  10
    7     AB_Improved     10  |  10     10  |  10     11  |   9      7  |  13
----------------------------------------------------------------------------
          Win Rate:      70.0%        72.9%        67.1%        70.0%
```

Round 3



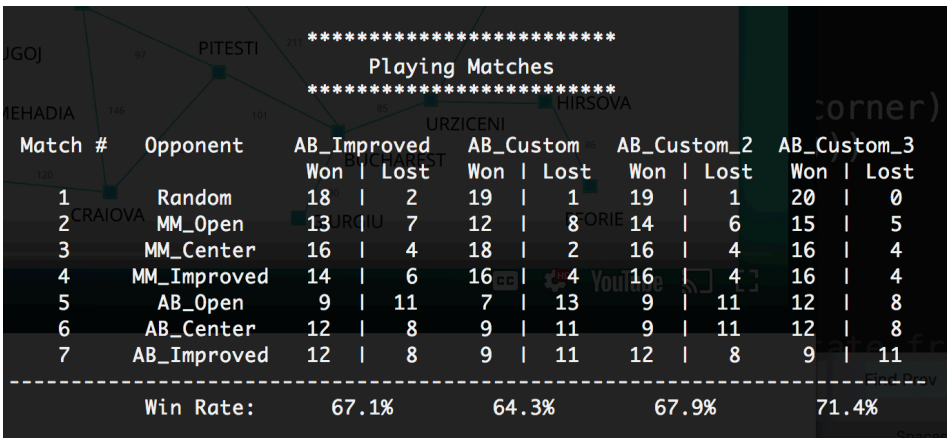
```
**************************
    Playing Matches
**************************
 Match #    Opponent    AB_Improved    AB_Custom    AB_Custom_2  AB_Custom_3
                        Won | Lost    Won | Lost    Won | Lost   Won | Lost
    1       Random      19  |   1     18  |   2     20  |   0    19  |   1
    2       MM_Open     16  |   4     18  |   2     14  |   6    17  |   3
    3      MM_Center    17  |   3     17  |   3     18  |   2    18  |   2
    4     MM_Improved   13  |   7     17  |   3     13  |   7    12  |   8
    5       AB_Open     12  |   8     10  |  10     10  |  10    13  |   7
    6      AB_Center    13  |   7     12  |   8     11  |   9    11  |   9
    7     AB_Improved   10  |  10      7  |  13     14  |   6    12  |   8
--------------------------------------------------------------------------------
          Win Rate:      71.4%        70.7%        71.4%        72.9%
```

Round 4



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

 Match #    Opponent    AB_Improved    AB_Custom    AB_Custom_2  AB_Custom_3
                        Won | Lost    Won | Lost    Won | Lost   Won | Lost
    1       Random      20  |   0     20  |   0     20  |   0    20  |   0
    2       MM_Open     12  |   8     15  |   5     11  |   9    17  |   3
    3      MM_Center    17  |   3     18  |   2     18  |   2    18  |   2
    4     MM_Improved   15  |   5     11  |   9     16  |   4    17  |   3
    5       AB_Open      7  |  13     10  |  10     10  |  10     9  |  11
    6      AB_Center    14  |   6     13  |   7     13  |   7    12  |   8
    7     AB_Improved   11  |   9     12  |   8      6  |  14    12  |   8
--------------------------------------------------------------------------------
          Win Rate:      68.6%        70.7%        67.1%        75.0%
```

Round 5



```
**************************
    Playing Matches
**************************
 Match #    Opponent    AB_Improved    AB_Custom    AB_Custom_2  AB_Custom_3
                        Won | Lost    Won | Lost    Won | Lost   Won | Lost
    1       Random      18  |   2     19  |   1     19  |   1    20  |   0
    2       MM_Open     13  |   7     12  |   8     14  |   6    15  |   5
    3      MM_Center    16  |   4     18  |   2     16  |   4    16  |   4
    4     MM_Improved   14  |   6     16  |   4     16  |   4    16  |   4
    5       AB_Open      9  |  11      7  |  13      9  |  11    12  |   8
    6      AB_Center    12  |   8      9  |  11      9  |  11    12  |   8
    7     AB_Improved   12  |   8      9  |  11     12  |   8     9  |  11
--------------------------------------------------------------------------------
          Win Rate:      67.1%        64.3%        67.9%        71.4%
```