Heuristic Analysis

The following, are the results obtained for the whole process including all the different cases being tested:

```
                      **************************
                           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     10 |   0
    2       MM_Open       10 |   0      8 |   2     9 |   1      9 |   1
    3      MM_Center      10 |   0     10 |   0    10 |   0     10 |   0
    4     MM_Improved      9 |   1      9 |   1     7 |   3      9 |   1
    5       AB_Open        5 |   5      3 |   7     4 |   6      4 |   6
    6      AB_Center       7 |   3      8 |   2     8 |   2      5 |   5
    7     AB_Improved      6 |   4      8 |   2     6 |   4      2 |   8
 ------------------------------------------------------------------------
           Win Rate:      81.4%        80.0%        75.7%        70.0%
```

Further detail regarding each case will be indicated below.

Case 1: *Custom score 1*

What I intended with this heuristic was to force the opponent to the limits, since that's where most of the losses take place. To do this, I built it over the AB_Improved heuristic. This case maximizes the distance from the center for the opponent and minimizes the distance from the center for the player. However, it is true that after a certain number of moves, it becomes rarer to carry out moves towards the center of the board and these moves may as well not be the optimum ones since they may limit more than at the beginning the movement of the player. For this reason, Hence, I am reducing the weight of this distance parameter as more moves are played. 0.7, which I found gave better results when playing against AB_Improved.

Results analysis:

This case performs adequately. Although the winning rate isn't as good as the winning rate for AB_Improved, it gets quite close when playing 40 games. It is quick to compute and involves additional information about the state of the board. The varying weight defines the importance given to the distance to the center during the game, being the constant used obtained after testing several cases iteratively against the AB_improved model. Given that the tested cases may not cover all combinations, this number could still be improved by further testing in order to obtain a higher accuracy.

Case 2: *Custom score 2*

What I intended with this case was to stay in the center and use Manhattan distance to process the distance to the middle instead of the Euclidean one. During the initial 10 % of the game, the program tries to capture the center positions. After this initial point, it tries to maximize the number of moves left.

Results analysis:

This case performs adequately, although it is clear that the result is worse than the AB_Improved. The Manhattan distance formula to calculate distance to the middle might be performing better than the Euclidean formula, given it is faster to process and it is more related to the player. The part of switching strategies depending on the status of the game may also be a good strategy, but the precise moment to switch strategy and the optimal strategy for each case may be more complex to define. Like before, the movement towards the center after the beginning of the game may not be the optimum strategy, and therefore most moves will be far from the middle of the board and towards the limits.

Case 3: *Custom score 3*

What I intended with this case was to penalize moves that are on the limits of the board and assess the quality of each move given the distance to the middle of the board for all future moves. The smaller the distance, the better the quality of each move. As it gets close to the end of the game, the program will try to minimize the number of moves left for the opponent.

Results analysis:

This case clearly performs worse than the AB_Improved one. Although this case takes multiple inputs to assess the quality of the board state, a sum may not be enough. This case also takes more processing power since it has to process future moves and the distance to the middle of the board for each move to be made during the game. Like before, switching strategies for the movement towards the center after the beginning of the game may not be the optimum strategy, and therefore most moves will be far from the middle of the board and towards the limits, although there is some difficulty in finding the best moment to switch strategy.

Overall comparison and results:

As seen from the results below:

```
                        **************************
                             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     10  |   0
    2       MM_Open       10  |   0      8  |   2     9  |   1      9  |   1
    3      MM_Center      10  |   0     10  |   0    10  |   0     10  |   0
    4     MM_Improved      9  |   1      9  |   1     7  |   3      9  |   1
    5       AB_Open        5  |   5      3  |   7     4  |   6      4  |   6
    6      AB_Center       7  |   3      8  |   2     8  |   2      5  |   5
    7     AB_Improved      6  |   4      8  |   2     6  |   4      2  |   8
-----------------------------------------------------------------------------
           Win Rate:      81.4%        80.0%        75.7%        70.0%
```

AB_Improved and AB_Custom perform very similarly, so if I had to choose from one of the three options, I would go for AB_Custom. It would be possible that after making some adjustments, AB_Custom could get even better results that AB_Improved.
Also, AB_Custom requires less processing and gets more information regarding the board than AB_Improved.
Even though AB_Custom performed significantly better than AB_Custom_2 and AB_Custom_3, these last 2 have indeed interesting ideas like switching strategies depending on the status of the game, in order to adapt to different circumstances and could be possible to improve after adjusting several parameters to a deeper extent, requiring as well a larger amount of time to perfect.