My main heuristic is to find the distance of player from corner of the board. If the player is closer to corner than it is more likely he would have less moves to play, then the opposite player. We calculate it by calculating the distance of corner from centre of the board and subtracting it with the distance of the player from centre of the board. And obviously length of legal moves left for player, opposite player and common moves are quite important as it gives good idea on how much moves are left for game to finish.

Custom score 1:

```
return float(len(common_moves) + len(player_moves) + max(corner(game, m) for m in
common_moves) - len(opp_moves))
```

Length of the legal moves of player plus length of common moves of both player, plus maximum of all distance of the player from corner for each player legal moves subtract with legal moves of opposite player.

Custom score 2:

```
return float(len(player_moves) - len(opp_moves) + np.mean([corner(game, m) for m in
player_moves]) + len(common_moves))
```

Length of the legal moves of player plus length of common moves of both player plus mean of all distance of the player from corner for each player legal moves subtract with legal moves of opposite player

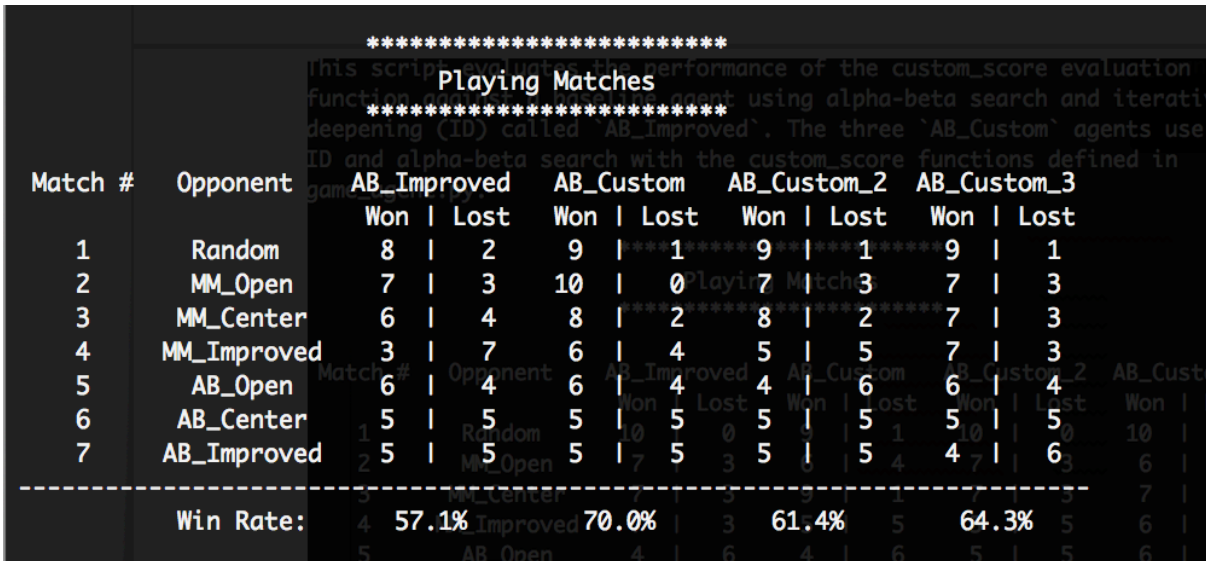
Custom score 3:

```
return float(len(player_moves) - len(opp_moves) + len(common_moves) + corner(game,
game.get_player_location(player)))
```

Length of the legal moves plus length of common moves plus distance from corner for each legal move subtract the moves of opposite player

Result:

Custom scores written in game_agent script most of the times performed better or same with the sample script heuristics. Though after running for tournament.py several times results always varied but custom_score mostly performed the best.



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

 Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                        Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1       Random       8  |   2     9  |   1     9  |   1     9  |   1
    2       MM_Open      7  |   3    10  |   0     7  |   3     7  |   3
    3      MM_Center     6  |   4     8  |   2     8  |   2     7  |   3
    4     MM_Improved    3  |   7     6  |   4     5  |   5     7  |   3
    5       AB_Open      6  |   4     6  |   4     4  |   6     6  |   4
    6      AB_Center     5  |   5     5  |   5     5  |   5     5  |   5
    7     AB_Improved    5  |   5     5  |   5     5  |   5     4  |   6
--------------------------------------------------------------------------
           Win Rate:      57.1%        70.0%        61.4%        64.3%
```

Explanation:

Heuristic 1 (custom_score) mostly performed better than other heuristics because it take into the factor that what was the maximum distance of player from corner of the board. Once a player is reached near centre it has more legal moves to play and will less likely to loose from player who is closer to corner.

I even tried using sum of corner distance for all legal moves but it seemed it was not better than min/max/mean value of all possibilities. Maybe because of too much variability in moves and location.

On first look I thought mean of all distances from corner should be good heuristics but it was not better than simple corner distance of each legal move proving that mean of both player corner distances are not much different resulting in not that a good heuristic.

I also tried simply using length of legal moves of player, opposite player and common moves but results were not better. The results did prove that corner distance is important factor in deciding winning position of the player.