Analysis of heuristic functions for the game of isolation

The score of this analysis is to evaluate the heuristic functions that I have implemented for the Game agent. The purpose of these heuristics is to beat the baseline implementation, which is Alpha Beta with improved score.
I have three strategies that I will discuss:

1. Start out offensively and then switch to defense as the number of moves increases
2. Start out defensively and then switch to offense as the number of moves increases
3. Be defensive and try to minimize the opponents options for every move

The results of the heuristics is shown below and will be discussed

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

 Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                       Won | Lost    Won | Lost    Won | Lost    Won | Lost
    1      Random      47  |   3     46  |   4     48  |   2     46  |   4
    2      MM_Open     35  |  15     39  |  11     40  |  10     39  |  11
    3      MM_Center   39  |  11     44  |   6     42  |   8     43  |   7
    4      MM_Improved 39  |  11     34  |  16     37  |  13     39  |  11
    5      AB_Open     27  |  23     25  |  25     26  |  24     26  |  24
    6      AB_Center   29  |  21     33  |  17     26  |  24     25  |  25
    7      AB_Improved 30  |  20     25  |  25     24  |  26     24  |  26
--------------------------------------------------------------------------
           Win Rate:     70.3%         70.3%         69.4%         69.1%
```

The results show that there is not much of a difference between the heuristics. This is probably due to the fact that each algorithm is based on the same base formula, namely the number of legal moves for the player, minus the number of legal moves for the opponent. This result will added or subtracted a weight, based on the number of total moves in current game.

AB_Custom: start out offensively

This heuristic is formed based on the number of legal moves for the player minus the number of legal moves for the opponent. It then adds a move_count factor which consists of a weight, which is empirically set to 0.5 and then multiplied with the number of moves in the game.
In this case we have the heuristic function $h(g,p)$ where g is the game and p is the player. In the function we get the legal moves for the player p $m(p)$ and the legal moves for the opponent o $m(o)$.

We also see the weight γ that multiplies with the number of moves in the game gm so far. We have this function:

$$h(g,p) = m(p) - (m(o) + (\gamma * gm))$$

This will start out with a strategy that seeks a move that gives more options than the opponent, but besides this does not care too much about trying to block the opponent. This will change as the game progresses, so eventually the player will try to block more and more as the game progresses.

AB_Custom_2: start out defensively

This is the opposite of heuristic 1. In this function the player will start out blocking the opponent and then gradually become more offensive. Again, this is based on the number of legal moves for the player $m(p)$ minus the number of legal moves for the opponent $m(o)$. We have a start out blocking weight of 2 ε and an empirically tested decreasing value of 0.05 γ which is multiplied with the number of moves in the game gm so far.

$$h(g,p) = m(p) - (\varepsilon * m(o) - (\gamma * gm))$$

In this function we subtract the calculated increasing factor that is the result of the increasing moves in the game from the defensive value.

This heuristic seems to perform equally to

AB_Custom_3: block the opponent

In this function the player p will try to block the opponent o from the beginning. By setting ε to 2, we assure strictly blocking behaviour from the start till the end.

$$h(g,p) = m(p) - (\varepsilon * m(o))$$

The strategy is to just have more legal moves than the opponent and not if this move will search for the best result offensively.