

I try length of player's move - 1.3 * length of opponent move, length of player's move - length of opponent move, length of player's move - 1.5 * length of opponent move as the evaluation function. I choose the second one because it has the best behavior in the tournament.

The screen cut of tournament:

```

AIND-Isolation — -bash — 80x24
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       5 | 5         6 | 4         5 | 5         3 | 7
2   MM_Open      1 | 9         2 | 8         2 | 8         2 | 8
3   MM_Center    3 | 7         1 | 9         1 | 9         2 | 8
4   MM_Improved  1 | 9         2 | 8         1 | 9         1 | 9
5   AB_Open      8 | 2         7 | 3         6 | 4         6 | 4
6   AB_Center    2 | 8         4 | 6         5 | 5         4 | 6
7   AB_Improved  3 | 7         7 | 3         4 | 6         2 | 8
-----
Win Rate:   32.9%      41.4%      34.3%      28.6%

There were 2.0 timeouts during the tournament -- make sure your agent handles se
arch timeout correctly, and consider increasing the timeout margin for your agen
t.

(aind) D-10-18-218-212:AIND-Isolation David$

```