Hao Chen

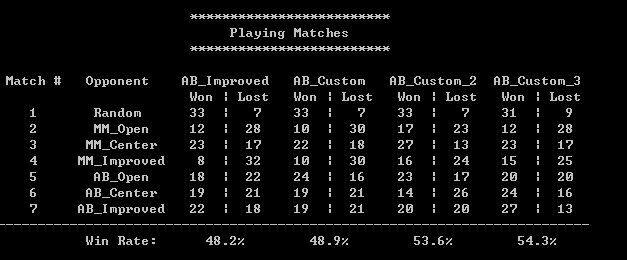
Udacity Artificial Intelligence

Heuristic Analysis

Custom\_score 1 is similar to the Improved\_score where it outputs a score equal to the difference in the number of moves available to the two players. However, if your available move is 2 more than your opponents available moves, you start to chase after your opponent, having a heuristic formula own\_moves-2\*opp\_moves.

Custom\_score 2 is also a score based implementation of heuristics. Since each corner moves are less desirable than the center moves, each player’s score is calculated by 2 times its available center moves plus 1 times its available corner move. Then we return the score difference between the two players.

Custom\_score 3 is a combination of custom\_score 1 + custom\_score 2. Each player’s score is calculated by 2 times its available center moves plus 1 times its available corner move. In addition, if the player’s available moves are 2 more than the opponent’s moves, then the player can start chasing after the opponent, returning a heuristic formula own\_moves-2\*opp\_moves.



By increasing the number of matches to 20 and running the tournaments.py, I concluded that heuristic 3 outperformed Improved\_score and worked the best. By comparing my result with ones on the forum, I see that my laptop’s CPU is also slower than the average, causing a ~50% win ratio for Improved\_score. Lastly, the only way to really know the best evaluation function is to try lots of variances in them and see which one works the best.