The three attempted heuristics I will analyze are:

1. **square of moves difference** - This heuristic is an attempt to emphasize having more moves than the opponent. After repeated trials*, there isn't much difference between the improved heuristic and this square of moves difference (76.57% to 76.14% in favor of improved). This suggests that emphasizing big differences in number of moves isn't much better than treating the differences linearly.
2. **weight own moves** - This heuristic is an attempt to emphasize the value of having more moves available, not just relative to opponent moves. We are multiplying the player's moves by a factor greater than 1 before subtracting the opponent moves. This emphasizes keeping the player's options open versus making moves to eliminate opponent moves. I tried several values for the weight, eventually settling on 1.3. Although in developing this heuristic I observed it outperforming the improved heuristic by a couple of percentage points a few times, in my final run of repeated trials, it was significantly worse, 75.43% to 73.14%. I suspect some of this is just variance because of the 85 to 15 score for this heuristic vs. Random strategy. That rate is atypically low, and it may have been worthwhile to increase the number of games.
3. **weight own moves but penalize opponent moves at endgame** - This heuristic is designed to emphasize reducing the opponent's moves near the endgame. While I felt like heuristic #2 had some promise, I noticed it occasionally losing near the endgame by not cutting off the opponent when it had a chance. This heuristic flips the script near endgame by increasing the relative importance of reducing the number of opponent moves. It was the only heuristic I tried that showed real promise, beating improved 78.57% to 72.71% over repeated trials, an improvement of nearly 6%. In particular, you can see it significantly outperforming improved in Match 4 vs. MM_Improved (77 wins vs. 70 wins) and Match 7 vs. AB_Improved (69 wins vs. 59 wins). So this heuristic is winning more against reasonable opponents that approach the improved heuristic, only lacking iterative deepening.

Future work: While I tried several values for theta (the weight on the player's moves) and a couple of different values for when to switch heuristic 3 to endgame (settling on 25 moves with a switch of theta from 1.3 to 0.75), it would be interesting to try a search grid over some possible values to optimize the weight and the indication of endgame. It would also be interesting to perhaps change the weight as a continuous function of the number of moves rather than flipping the weight after an arbitrary number of moves.

*for repeated trials I arbitrarily decided to run 25 matches (100 games) against each opponent and take whatever result emerged.

The heuristic performances are shown on the following pages in order of performance.

Heuristic 3: increase_opponent_move_penalty
$ python tournament.py

This script evaluates the performance of the custom heuristic function by comparing the strength of an agent using iterative deepening (ID) search with alpha-beta pruning against the strength rating of agents using other heuristic functions. The `ID_Improved` agent provides a baseline by measuring the performance of a basic agent using Iterative Deepening and the "improved" heuristic (from lecture) on your hardware. The `Student` agent then measures the performance of Iterative Deepening and the custom heuristic against the same opponents.

```
*************************
 Evaluating: ID_Improved
*************************

Playing Matches:
----------
  Match 1: ID_Improved vs   Random          Result: 91 to 9
  Match 2: ID_Improved vs   MM_Null         Result: 81 to 19
  Match 3: ID_Improved vs   MM_Open         Result: 67 to 33
  Match 4: ID_Improved vs MM_Improved       Result: 70 to 30
  Match 5: ID_Improved vs   AB_Null         Result: 72 to 28
  Match 6: ID_Improved vs   AB_Open         Result: 69 to 31
  Match 7: ID_Improved vs AB_Improved       Result: 59 to 41

Results:
----------
ID_Improved         72.71%

*************************
   Evaluating: Student
*************************

Playing Matches:
----------
  Match 1:   Student   vs   Random     Result: 93 to 7
  Match 2:   Student   vs   MM_Null    Result: 87 to 13
  Match 3:   Student   vs   MM_Open          Result: 72 to 28
  Match 4:   Student   vs MM_Improved        Result: 77 to 23
  Match 5:   Student   vs   AB_Null     Result: 83 to 17
  Match 6:   Student   vs   AB_Open          Result: 69 to 31
  Match 7:   Student   vs AB_Improved        Result: 69 to 31


Results:
----------
Student             78.57%
```

Heuristic 1: square_move_diff
Brians-MBP:AIND-Isolation blcooley$ python tournament.py

This script evaluates the performance of the custom heuristic function by comparing the strength of an agent using iterative deepening (ID) search with alpha-beta pruning against the strength rating of agents using other heuristic functions. The `ID_Improved` agent provides a baseline by measuring the performance of a basic agent using Iterative Deepening and the "improved" heuristic (from lecture) on your hardware. The `Student` agent then measures the performance of Iterative Deepening and the custom heuristic against the same opponents.

```
*************************
 Evaluating: ID_Improved
*************************

Playing Matches:
----------
  Match 1: ID_Improved vs   Random          Result: 97 to 3
  Match 2: ID_Improved vs   MM_Null         Result: 84 to 16
  Match 3: ID_Improved vs   MM_Open         Result: 71 to 29
  Match 4: ID_Improved vs MM_Improved       Result: 67 to 33
  Match 5: ID_Improved vs   AB_Null         Result: 79 to 21
  Match 6: ID_Improved vs   AB_Open         Result: 68 to 32
  Match 7: ID_Improved vs AB_Improved       Result: 70 to 30


Results:
----------
ID_Improved        76.57%

*************************
   Evaluating: Student
*************************

Playing Matches:
----------
  Match 1:   Student   vs   Random     Result: 95 to 5
  Match 2:   Student   vs   MM_Null    Result: 85 to 15
  Match 3:   Student   vs   MM_Open          Result: 78 to 22
  Match 4:   Student   vs MM_Improved        Result: 64 to 36
  Match 5:   Student   vs   AB_Null     Result: 74 to 26
  Match 6:   Student   vs   AB_Open          Result: 69 to 31
  Match 7:   Student   vs AB_Improved        Result: 68 to 32


Results:
----------
Student            76.14%
```

Heuristic 2: increase own moves
Brians-MBP:AIND-Isolation blcooley$ python tournament.py

This script evaluates the performance of the custom heuristic function by comparing the strength of an agent using iterative deepening (ID) search with alpha-beta pruning against the strength rating of agents using other heuristic functions. The `ID_Improved` agent provides a baseline by measuring the performance of a basic agent using Iterative Deepening and the "improved" heuristic (from lecture) on your hardware. The `Student` agent then measures the performance of Iterative Deepening and the custom heuristic against the same opponents.

```
*************************
 Evaluating: ID_Improved
*************************

Playing Matches:
----------
  Match 1: ID_Improved vs   Random      Result: 95 to 5
  Match 2: ID_Improved vs   MM_Null     Result: 84 to 16
  Match 3: ID_Improved vs   MM_Open     Result: 74 to 26
  Match 4: ID_Improved vs MM_Improved   Result: 72 to 28
  Match 5: ID_Improved vs   AB_Null     Result: 76 to 24
  Match 6: ID_Improved vs   AB_Open     Result: 67 to 33
  Match 7: ID_Improved vs AB_Improved   Result: 60 to 40


Results:
----------
ID_Improved         75.43%

*************************
   Evaluating: Student
*************************

Playing Matches:
----------
  Match 1:   Student   vs   Random      Result: 85 to 15
  Match 2:   Student   vs   MM_Null     Result: 87 to 13
  Match 3:   Student   vs   MM_Open     Result: 66 to 34
  Match 4:   Student   vs MM_Improved   Result: 63 to 37
  Match 5:   Student   vs   AB_Null     Result: 79 to 21
  Match 6:   Student   vs   AB_Open     Result: 67 to 33
  Match 7:   Student   vs AB_Improved   Result: 65 to 35


Results:
----------
Student             73.14%
```