Udacity AI Project 3 - Jeff Rix

Results for CustomPlayer vs other players:

Opponent	# Matches	Win %
Final CustomPlayer against Random Opponent	200	97.2%
Final CustomPlayer against Greedy Opponent	200	76.1%
Final CustomPlayer against Self Opponent	200	50.0%
Final CustomPlayer against Minimax Opponent	200	84.0%

Option 1: Develop a custom heuristic

Results of building of Custom Player heuristic vs Baseline heuristic:

Opponent	**# Matches**	**Win %**
Baseline Heuristic against Minimax Opponent	200	52.9%
Opening Book with Baseline Heuristic against Minimax Opponent	200	57.9%
CustomPlayer - Offense to Defense Heuristic against Minimax Opponent	200	74.8%
Final: CustomPlayer - Defense to Offense Heuristic against Minimax Opponent	200	84.0%

Baseline Heuristic for Testing

For baseline testing I used an alpha-beta pruning algorithm with iterative deepening and the following scoring heuristic:

```
#my_moves - #opponent_moves
```

CustomPlayer Heuristic

I used an alpha-beta pruning algorithm with iterative deepening and an opening book based build from 5,000,000 random games.

For my heuristic I used a combination of two different scoring heuristics that would change strategies at the mid point in the game.

My customer player starts playing a defensive strategy where the player is trying to maximize available moves while minimizing the opponents moves :

```
(#my_moves * 2) - #opponent_moves
```

Halfway through the game it changes to playing an offensive strategy where the computer is trying to minimize the number of opponents moves:

```
#my_moves - (2 * #opponent_moves)
```

I tested the reverse heuristic (offense to defense) but it only performed with a 74.8% win rate.

What features of the game does your heuristic incorporate?
I used the features of style of play either offensive or defensive. My above finding shows that defensive to offensive performs better combined with an opening book, alpha-beta pruning and iterative deepening.

Why do you think those features matter in evaluating states during search?
The reason is in the beginning of the game the agent is trying to maximize the number of available space for moves. This aligns well with the shallow max depth of the search tree at the beginning of the game, the agent is not trying to find endgame moves but satisfying the defensive heuristic. As the game progresses the agent will have more available moves when it changes to an offensive strategy while using the expanded depth of the search tree to find the winning moves while it nears endgame.

I ran a test where I set the max depth limit to 4 (this is the closest I could achieve without time out). The results were the test agent only won 57.8% of matches. In the first half of the game, the test agent was achieving a comparable max depth as the iterative deepening version. In the second half of the game the test agent is greatly limited in tree depth and in its ability to find winning endgame moves.

Analyze the search depth your agent achieves using your custom heuristic.
At the beginning of the game the search depth averages around 7 and at the endgame it maxes out at 20. I increased the time limit to 300ms and 1000ms with no significant improvement in performance and only hit a max depth of 22. This lack of increase in search depth is due to the exponential increase in the amount of nodes at every level of the tree.

Option 2: Opening Book

Results of building of my_custom_player heuristic vs Baseline heuristic:

Opponent	# Matches	Win %
my_custom_player with Random opening moves against Minimax Opponent	200	74.5%
Final my_custom_player against Minimax Opponent	200	84.0%

Describe your process for collecting statistics to build your opening book. How did you choose states to sample?
I built the statistics by simulating 5 million random games. The performance of the opening book did not beat random opening moves until I passed 3 million simulated games. I also tried building the statistics with my final CustomPlayer but it took longer to run the simulations and at 200,000 simulated games the opening book was not beating random opening moves.

And how did you perform rollouts to determine a winner?
I used random simulations of the games.

What opening moves does your book suggest are most effective on an empty board for player 1?
(7,5) the lower middle of the board.

What is player 2's best reply?
(8,2)

References:
http://ajulio.com/assets/documents/Adversarial_Game.pdf