Adam Nelson

CSCI 477, Prof. Mark Baldwin

4/30/19

**Design:**

I decided to base my Beta AI off of a general gambling strategy that’s akin to “buy low, sell high.” In poker it correlates to “fold when you’re losing, raise when you’re winning”. The idea is to minimize losses as early as possible, and to bet aggressively to intimidate your opponent.

The Beta AI adheres to the following logic:

1. If the bet2player == 0:
   1. If the Beta's hand is worth less than their opponents hand they check
   2. If their hand is worth more, they make a small bet of 5
2. If the bet2player > 0:
   1. If the Beta's hand is worth less than their opponents hand the fold
   2. If their hand is worth more, they raise by 7. If they cannot raise, they call
   3. If their hand is worth the same, they call

The Beta will always bet if they have the advantage, but always fold if they are at a disadvantage.

To give the Alpha AI a better chance, the Beta AI has a 20% chance of doing Check/Call instead of raising. This is effectively a 'handicap' for the Beta. This will slightly decrease the pot each round, thereby giving Alpha more chips after 20 rounds and a better chance of winning.

There were a few different factors that I considered when designing the Beta AI. The first was to have it raise the maximum amount whenever the logic calls for a raise. I decided against this because constantly making those large bets means larger losses when the Alpha wins a round. Aggressive betting does not mean blindly throwing all your chips in. Even Poker Pros reserve going all-in for only sure-fire winning hands.

The next factor was whether to include the option to fold, or always raise/call. Due to how the bidding round is designed, the Beta isn’t allowed to fold if it can check, even when their opponents hand is higher. Even if it was, it wouldn’t since it “Check” keeps a player in the game for free. If the Beta never folded, even when at a steep disadvantage, they would just keep wasting money hoping for a miracle turn of the cards. It’s much safer to cut your losses early.

The last factor was to have an option to raise, or to simply call the other player’s bet every time. Calling is much safer in the long term, but raising your opponent when you’re at an advantage gradually creates a “safety net” of chips to compensate for the hands you do lose.

**Performance:**

The Beta AI performed quite well. Here are the results after 10 Monte Carlo simulations:

|  |  |
| --- | --- |
| Round | Wins (Out of 100 games) |
| 1 | 65 |
| 2 | 66 |
| 3 | 63 |
| 4 | 62 |
| 5 | 62 |
| 6 | 63 |
| 7 | 52 |
| 8 | 65 |
| 9 | 49 |
| 10 | 63 |
| Total | **610/1000 games** |
| Ties | 7 |

The Beta AI consistently beats the Alpha AI. The Alpha raises by a random amount whenever possible. But since its decision to stay in the game is not attached to the opponent’s hand whatsoever, it doesn’t cut its losses or bet reasonable amounts to retain a “safety net”, two fatal mistakes the Beta AI avoids. Overall, the Beta AI performed very well, winning 61% of its games against the Alpha AI.