Kai Sanchez

CPSC-39

Ms. Kanemoto

12/5/2024

Three Card Poker and Algorithm Explanation

The three card poker game is a popular card game I like to play with my family, and I enjoyed the blackjack game we tinkered with earlier in the semester. So I made this, I actually got to look back at that code to get ideas and inspiration. The game offers a starting amount of money, and the player needs to place a bet before each round. The goal is to have a better hand then the “Dealer”. If you lose, the bet is subtracted, and if you fold, the amount you wagered is gone. The game provides entertainment. I used algorithms for shuffling the deck, calculating hand values, and managing the players bet history. I think these algorithms add a unique layer to my overall idea and premise of the game.

First I did a deck shuffling algorithm. It ensures the cards are shuffled randomly to make the game fair. It uses (collection.shuffle(deck);). This takes an initialized list of cards and shuffles them to randomize the order. This makes the game feel new every time you bet. The big O time of this is O(n) as n is the number of cards in the deck. This is because the shuffle happens once through the deck.

Then, I used a recursive hand value calculation algorithm. It calculates the total value of a players hand by recursively summing up the individual card values. The numbered cards are worth their numerical value, and then face cards are 10, and the ace is 11. This method recursively computes the total value of the hand. It breaks down the card’s rank and makes it a numerical value. The big O value is O(n) where n is the number of cards.

One more algorithm is the bet history stack management. The game uses a stack to keep track of the players betting over the course of the game. Each time a new round is played, a new record is pushed to the stack, which the player can view at any time they want. So, when a round is finished, the bet amount, hand, and dealers hand are stored as a record. The big O time is O(1) for pushing a new record into the stack.

Some of the data structures I used was the Stack, List, Arrays, and LinkedList.
The stack was chosen to store the bet history, making it easy for the player to see their progress ands bets throughout their gaming experience. This was important to me because, If it didn't track your history then you can't go back and see where you messed up. The ArrayList was chosen to represent the deck of cards because it makes it easy to remove cards from the deck during each round. The deck is initialized as an ArrayList and shuffled. During each round, cards are drawn from the deck and removed. The LinkedList was used because it is efficient in insertions and deletions at the beginning and end of the list, which is good for dealing cards to a player.

I faced a lot of roadblocks in this program as it was my first time doing something I actually wanted to do, so weighing out options between what sounds cool, and what will actually be realistic to implement with my current skills. Before, the game would run fine, but it would just show your hand, and then the dealer's hand all at once, so you would hit play, place the bet, then it would just show both sets of cards and say "You won" or "You lost". So, I added a fold option. Therefore, the player can now look at their hand before deciding if they want to play just yet. That implementation is what I am most proud of. It makes the poker game seem more realistic.

Something I would change or implement next time around would be maybe adding an option where you can have multiple players playing at once, so everyone gets starting money and can bet whatever they want, and decide if they want to play their hand.