

Final Project Proposal Blackjack

We will implement a version of Blackjack in Terminal. It will have a user go up against multiple bots (including the dealer), with 4 possible actions: Hit, Stand, Double, and Split.

- Hit: Draw a random card and add to your hand
- Stand: Keep the cards you have, no more drawing
- Double: Only possible at the beginning, can only hit once. Gives 2x the reward if won.
- Split: Split a hand into two and play each resulting hand consecutively.

The bots will have a threshold number where they will always Stand. Otherwise, we'll use `Math.random()` to determine if the bots will hit or double. We're planning on creating a String array with all the cards to represent the deck of cards.

We will utilize card counting (HI-LO) to assist the user, if they choose to use it. This keeps track of the number of high and low value cards that have already been dealt. This can assist the user in determining if they have an advantage on the next hand. An integer will be printed, which will represent the odds. If it's negative, you're likely to lose, while a positive number indicates winning.

We will have a Gambler superclass that contains all the methods that are shared by every participant (Hit, stand, double, split). Then we will have Player, Bot, and Dealer classes that extend this and have their own unique functionality. Player has no additional methods. Dealer will have additional preset conditions for implementing the same methods. For example, if the Dealer receives a hand that is at least 17 points in total after the setup method, then they will choose to stand. Bot will be the same as Human, except its decision for implementing a method is based on `Math.random()`.

We will have a total score in lieu of using real money. The score is multiplied by 1.5x for every match the user wins. The score will reset when the user loses. This provides a competitive side to our game, as you can compete with others to achieve the highest score.

Other methods besides the 4 actions:

- getHandValue: takes a player as an input, returns their current hand value
- blackjack: returns true if the player has a hand value of 21
- bust: returns true if the player has a hand value over 21
- outcome: has 3 print results: user win, user loss, push (if a tie occurs)
- aceCheck: set the value of an Ace in hand to 1 if hand value exceeds 21
- scoreChange: changes user balance by 1.5x or resets it to 0 depending on outcome()