Jose Juan Velasquez

Jackson Burns

Project 1 Program Proposal: Blackjack in C++

Our program proposal consists of the game blackjack, a popular card game in which you’re given two cards from a 52-standard deck of cards and play a round based on what cards the dealer deals on the table. The game is simple, both you and the dealer (the house or the computer in this case) are given two initial cards and the goal is to stack up card numbers that add to 21 or less. If the total sum of the user is higher than the house, the user wins money based on the amount they bet and vice versa. You have the option to stop at a certain sum or risk it and keep adding cards to your hand. If your sum is greater than 21, you automatically lose.

Now that we know the rules, our program will consist of a simplified console version of blackjack, with an object that will represent the player and the deck of cards will be stored within a linked list structure. The player will always start with 100 coins (our currency used in the game), and they will have an option to see how many coins they want to bet. The game starts, deals two cards to the user and the user will have the option to either get more cards or stay as is. The computer will represent the house, but instead of programming individual cards for the computer, we will just generate a random number between 17-21 (since it is standard practice that the house stops adding to their sum once they reach a sum of 17 or higher), so that we can compare the computer’s card sums and with the user’s to see who wins. After the game ends, the cards will be reshuffled, and the game will repeat itself until the user wants to quit.

We plan to test this code by having another file that will work as a testing file, in which we can fill it with methods that will simulate a game and money handling. We will hardcode a set of cards and use that practice to test out different scenarios within our testing file, which will be composed of assertions to verify its validity.

We plan to divide our work in the following way. Jose Velasquez will work on the Game Initializer (deck creation), Printing cards, Main game (methods that controls the whole game, node creation, and the win/lose checker methods, while Jackson will work on the Shuffle cards, Deal cards, freeing memory, and money handler methods.

Our program will showcase the best of C++ since it combines the best features that would best for it from Java and C. It takes the polymorphism aspect from Java, allowing us to freely modify and utilize objects and classes. For example, having a global variable for our coins, it will be very easy to use getCoins() and setCoins() methods to change the currency within the player’s object depending on whether they win or not. Since we will be implementing a linked list for the deck of cards, utilizing the memory allocation aspect of C will make our game much faster and less resource intensive than a language that uses a garbage collector.

This program can be considered complex because it makes us organize an entire game with random scenarios which must be considered for this to work. We also need to consider what can be kept track of in Objects, as well as what is stored in the main program loop. Keeping track of the heap and main memory utilization is also important, as C++ does not have a garbage collector, so heap freeing is incredibly important.