David Roper

R. Vincent

Computer programming

420\_LCU\_05 sect. 00001

May 6th, 2019

Blackjack Card Game Program

(Épreuve Synthèse)

**User Manual:**

Welcome to Blackjack, the card game program. This manual will explain how the card game is played and how the program works. The card game Blackjack is played with 52-card deck (excluding Jokers) and the player plays against the dealer. The goal of the game is to have a hand of cards whose numerical summation is the closest, but not over 21. Face cards (Jacks, Kings, Queens) are equal to 10, also aces can count as a 1 or an 11 depending on what the player wants for their hand. In this program cards will be abbreviated, ex. King of Hearts is hK, 2 of clubs is c2.

After running the program, the player will be given the option to play the game of leave the program. If the player chooses to play, the deck is shuffled and the player is given the option to be “hit” (ask for a card) or to hold (refrain from taking any more cards). If the player asks to be hit, a card will be dealt to their hand, the dealer will also be given a card unless the dealer’s hand’s summation is more than or equal to 17, in which case the dealer will hold. After the cards are dealt, the program will survey the hands to see if any of them go over 21. If one of the hands does so, the program state that the other player is the winner and stops the round, bringing the player back to the first menu. If the player holds, the program will stop dealing cards to their hand, but will continue giving the dealer cards until they’re closer to 21. If the dealer end up getting the closer to 21 the player loses, but if the dealer goes over the player wins. In both scenarios the round finishes and the player is sent back to the main menu. If either the dealer or player is dealt an ace, its value (1 or 11) will be determine based of the hands current total and amount of aces. If the hand already has a total over ten or an ace, the next ace will automatically worth 1. If an ace is worth 11 but then the player draws a card making their total value go over 21, the ace is then converted to a 1.

**Design and programming techniques:**

The programming techniques I used very often were string splicing and list comprehension. The deck of cards is on a big list and to add the cards values to the total the program splices numerical values off the string and converts it to integer. The hands of the dealer and player are smaller lists, and the player hand is printed so the user knows which cards they were dealt. Function are defined such as val(x) and ace(x) to allow the program to convert a card into a numerical value to the player or dealers sum. The ace(x) function decides that when an ace is drawn, if its value is 1 or 11 through the use of if statements and for loops. First it scans the hand for existing aces using a for loop, if there is an ace in the hand already, the function returns the ace value as 1. The program then check if the hand’s current total is over 11, if it is it returns the ace value as 1, if not the value is 11. Finally the hands are scanned by the hand\_check(ls,tot) function, which changes an aces value from 11 to 1 if the next card dealt put the total over 21. While loops for the main menu and the player’s options are used to keep the program from stopping after one turn, breaks are used to stop the while loops after a round is finished or the user quits.

The program has a simplistic design that’s easy to understand coding wise and interface wise. The use of functions within the program was to simplify the amount of calculations done and the amount of if statements the program would have to read through. The program prints out the cards dealt and the current hand the player to make easier to keep track. Also explains how the user wins or loses making it easier to understand the game and program.