Joshua Waghorn

Dr. Lehr

CSC-5-7022

6 February 2020

Project 1

Introduction

**Title:** Blackjack Game

Blackjack is played on a poker table, typically with two to five or more players. It is also popularly known as Twenty-One. The game begins with a standard deck of fifty-two cards. Before cards are dealt, bets are placed and whoever wins the hand will get the pot. Next, the deck is shuffled and each player is dealt two cards, face-up, and the dealer is dealt one card face-up and one face-down. The object of the game is to get as close to or reach twenty-one with as few cards as possible. The dealer must hit until the arrive at a value of 17 or higher. Each player may hit until they reach their desired value or go over twenty-one. Cards are valued at their number value, face cards (King, Queen, and Jack) are worth ten and are sequentially higher, Aces are worth either one or eleven. A player may split their pair if they have two cards of the same denomination and they become two different hands. After another round of bets, the cards are evaluated to see if who wins the hand.

**Summary:**

Project size: Approximately 150 lines

The number of variables: About 15

I attempted to make a game of Blackjack but had major time constraints due to the Midterm, another class, and prior obligations as well as being sick. I created an option to enter the program with a simple yes or no answer and it brings the user to a menu where they can choose to read the rules, play the game, or quit the program. The first option shows the rules of the game according to The United States Playing Card Company’s website. The second option plays the game, which first asks for the player’s bet, then reveals their hand and the dealer’s hand. The program then goes back to the menu and the player can choose any of the options once more. The cards are randomized and unique from one to fifty-two and then assigned to their relevant number and suite in the deck, so that a value of one returns and Ace of Spades and a value of fifty-two returns a King of Diamonds. The final and third option is simply to quit the program and leaves a farewell message.

**Description:**

This program is designed to simply get the player’s bet and then draw cards and return their values to the player.

**Images:**

Asks if the user wants to play then displays the menu


Asks if the user wants to play then displays the menu

A screenshot of a cell phone

Description automatically generated

Asks for the user's bet

A screenshot of a cell phone

Description automatically generated

Displays the two hand of cards.

The program then goes back to the menu.

**Pseudo Code:**

Initialize

While the user’s answer does not equal y or Y

Ask if the user would like the play Blackjack

Get user input (yesno)

If input yesno is n or N

Output “Maybe next time!”

If input yesno is y or Y

Do

While choice does not equal 1, 2, or 3

Display the Menu

Get the User’s Menu choice (choice)

Switch statement(choice)

Case 1:

Display the rules of the game

Case 2:

Get the user’s bet amount

Display the user and the dealer’s hand

Case 3:

Output a thank you message

Quit the program

While the user’s answer is y or Y

End/Return 0

Void Function Menu

Output the three options

Void Function choice1

Outputs the rules

Void Function choice2

Card1=rand()%deck+1

Do

Card2=rand()%deck+1

While card1==card2

Do

Card3=rand()%deck+1

While card1==card3||card2==card3

Do

Card4=rand()%deck+1

While card1==card4||card2==card4||card3==card4

Calculate the player’s total

Calculate the dealer’s total

String cardIn

For(int card=1; card<=deck;card++)

If (card1==card) name1=cardIn

If (card2==card) name2=cardIn

If (card3==card) name3=cardIn

If (card4==card) name4=cardIn

Output

“How much would you like to bet?”

Input

bet

Output

“Your bet:

$”bet

“Your Hand:”<<card1<<” “<<card2<<endl

name1<<” “ <<name2<<endl

“Dealer’s Hand:<<card3<<” “<<card4<<endl

name3<<” “<<name4<<endl

**Flowchart:**A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

**Cross Reference for Project 1**

**You are to fill-in with where located in code**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Chapter** | **Section** | **Topic** | **Where Line #''s** | **Pts** | **Notes** |
| 2 | 2 | cout | 50, 52, 64, 77-79, 83-110, 163, 165-169 |  |  |
|  | 3 | libraries | 7-10 | 8 | iostream, iomanip, cmath, cstdlib, fstream, string, ctime |
|  | 4 | variables/literals | 31-35, 115-119 |  | No variables in global area, failed project! |
|  | 5 | Identifiers | 23-26, 31-37, 115-122 |  |  |
|  | 6 | Integers | 119 | 3 |  |
|  | 7 | Characters | 31, 32, 34, 118 | 3 |  |
|  | 8 | Strings | 37, 40, 121, 122, 125, 153-160 | 3 |  |
|  | 9 | Floats No Doubles | 35 | 3 | Using doubles will fail the project, floats OK! |
|  | 10 | Bools | N/A | 4 |  |
|  | 11 | Sizeof \*\*\*\*\* |  |  |  |
|  | 12 | Variables 7 characters or less | 31-35, 115-119 |  | All variables <= 7 characters |
|  | 13 | Scope \*\*\*\*\* No Global Variables |  |  |  |
|  | 14 | Arithmetic operators | 49-55, 68, 129-159 |  |  |
|  | 15 | Comments 20%+ |  | 5 | Model as pseudo code |
|  | 16 | Named Constants | 17 |  | All Local, only Conversions/Physics/Math in Global area |
|  | 17 | Programming Style \*\*\*\*\* Emulate |  |  | Emulate style in book/in class repositiory |
|  |  |  |  |  |  |
| 3 | 1 | cin | 51, 57, 164 |  |  |
|  | 2 | Math Expression | 130-144 |  |  |
|  | 3 | Mixing data types \*\*\*\* |  |  |  |
|  | 4 | Overflow/Underflow \*\*\*\* |  |  |  |
|  | 5 | Type Casting |  | 4 |  |
|  | 6 | Multiple assignment \*\*\*\*\* |  |  |  |
|  | 7 | Formatting output | 50, 52, 64, 77-79, 83-110, 163, 165-169 | 4 |  |
|  | 8 | Strings | 125, 126, 153-160 | 3 |  |
|  | 9 | Math Library | 7 | 4 | All libraries included have to be used |
|  | 10 | Hand tracing \*\*\*\*\*\* |  |  |  |
|  |  |  |  |  |  |
| 4 | 1 | Relational Operators | 49, 52, 53, 68, 156-159 |  |  |
|  | 2 | if | 52, 53, 156-159 | 4 | Independent if |
|  | 4 | If-else | N/A | 4 |  |
|  | 5 | Nesting | 49-70 | 4 |  |
|  | 6 | If-else-if | N/A | 4 |  |
|  | 7 | Flags \*\*\*\*\* |  |  |  |
|  | 8 | Logical operators | 49, 52, 53, 68, 130, 134, 139, 144 | 4 |  |
|  | 11 | Validating user input | 49, 52, 53, 68 | 4 |  |
|  | 13 | Conditional Operator | 52, 53, 156-159 | 4 |  |
|  | 14 | Switch | 58 | 4 |  |
|  |  |  |  |  |  |
| 5 | 1 | Increment/Decrement | 128, 134, 139, 144 | 4 |  |
|  | 2 | While | 49, 55, 68, 135, 140, 145 | 4 |  |
|  | 5 | Do-while | 54 | 4 |  |
|  | 6 | For loop | 154 | 4 |  |
|  | 11 | Files input/output both | 40, 126 | 8 |  |
|  | 12 | No breaks in loops \*\*\*\*\*\* |  |  | Failed Project if included |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| \*\*\*\*\*\* Not | required to | show | Total | 100 |  |