Project 2

<Blackjack>

CIS 5 -41366
Michael Guerrero
02/14/2021

Program Summary

This is my attempt at Blackjack with only using knowledge from the entire class The user starts off at the main menu where they can check their current balance, win stats, and start new games. There is also a login system that will remember the players username and password.

Example:

```
==========================================
==========================================
1)Play Game
2)Balance
3)Stats
4)Exit Game
==========================================
Select Choice :
```

```
Blackjack2_V1 (Build, Run) ×   Blackjack2_V1 (Run) ×   Blackjack2_V2 (Build, Run) ×   Blackjack2_V2 (Run) ×
Do you have an active user?(Y/N) : Y
Enter username : John
Enter the password : Doe
Invalid Username/Password, please try again.
Enter username :
```

Once assigned their hand the user can determine whether or not they would like to hit or stand.

Example:

```
Dealers Hand is 2 of hearts and a King bringing the total to 15
Jak's Hand is 2 of hearts and 13 of hearts bringing the total to 13
Would you like to Hit or Stand (H/S) :
```

Which will then display the correct win condition depending on the results.

Example:

```
Would you like to Hit or Stand (H/S) : H
Jak selected hit and drew 12 of spades bringing the total to 25
Jak has Busted!!!
Would you like to play again? (Y/N):
```

Project Size

This project is about ***409 lines of code***, with around ***24 variables***. The number of lines were able to compress quite a bit including functions and arrays into the program. Though the number of variables seemed to increase.

Project Shortcomings

During the creation and updating of the project from the first, I was having difficulty figuring out how I could include 2D Arrays or Vectors and in the end wasn't able to include that or a lot more. In hindsight I probably should have scrapped most of the program and started back again with the bare bones but trying to force everything to work ended up costing me half of the checklist.

Project Results

The final product is a basic working of Blackjack, unfortunately I was not able to add in the ability to double down or split but the core is there. In total this project is not the best but I had a lot of fun learning and implementing stuff we learned in class into a complete program.I gained a lot of insight on how version controls of programs really matter, as during my creation I ran into huge bugs where I would've had to reset entirely if not for having different checkpoints.

Pseudo-Code

Initialize Variables
Login to User Account/ Create New Account
Display Menu
Read in users input
Ask for wager
Randomize Dealer and Players hands as if cards were shuffled
Assign Ace,Jack,King and Queen to correct number values
If Player > 21 or equal to 21 set bust to true

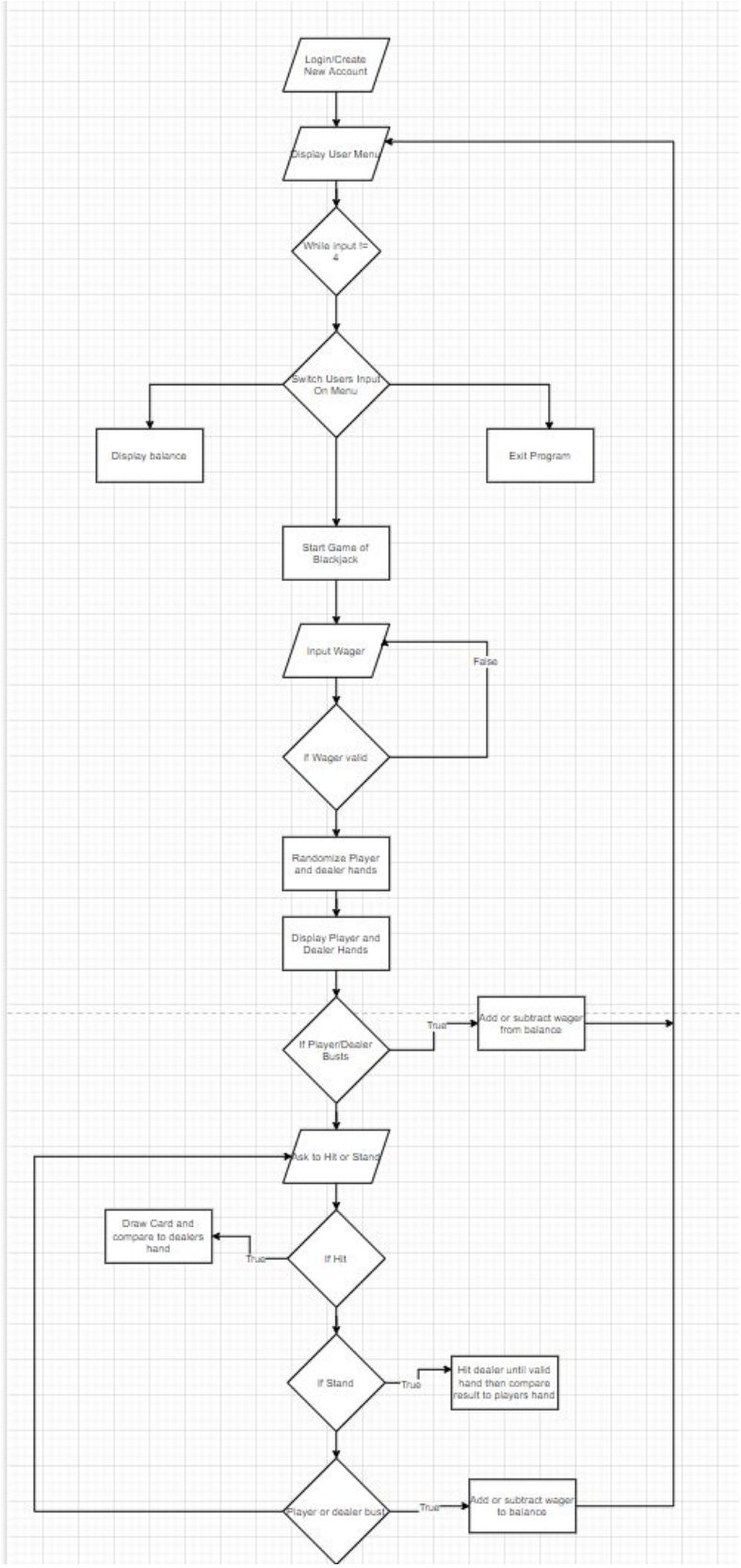
If dealer > 21 or equal to 21 set bust to true

If player and dealer are both less than 21 give player option to hit or stand

If hit assign player new card and add to total
If stand keep players balance and have dealer pull card until valid hand

Calculate the results
Add or subtract players wager from balance
Return player to Menu
Else end program

FlowChart (Hard to see but included in project file)



Project Check Off List

Chapter	Section	Topic	Where Line #"s	Pts	Notes
2	2	cout			
	3	libraries	14-21	5	iostream, iomanip, cmath, cstdlib, fstream, string, ctime
	4	variables/literals			No variables in global area, failed project!
	5	Identifiers			
	6	Integers	85	1	
	7	Characters	83	1	
	8	Strings	82	1	
	9	Floats, No Doubles	40	1	Using doubles will fail the project, floats OK!
	10	Bools	38	1	
	11	Sizeof *****			
	12	Variables 7 characters or less			All variables <= 7 characters
	13	Scope ***** No Global Variables			
	14	Arithmetic operators			
	15	Comments 20%+	1-413	2	Model as pseudo code
	16	Named Constants			All Local, only Conversions/Physics/Math in Global area
	17	Programming Style ***** Emulate			Emulate style in book/in class repositiory
3	1	cin			
	2	Math Expression			
	3	Mixing data types ****			
	4	Overflow/Underflow ****			
	5	Type Casting	36	1	
	6	Multiple assignment *****			
	7	Formatting output	63	1	
	8	Strings	82	1	
	9	Math Library		1	All libraries included have to be used
	10	Hand tracing ******			
4	1	Relational Operators			
	2	if	89	1	Independent if
	4	If-else	100-106	1	
	5	Nesting	153-166	1	
	6	If-else-if	157-161	1	
	7	Flags *****			
	8	Logical operators	46	1	
	11	Validating user input	89	1	
	13	Conditional Operator		1	
	14	Switch	56	1	
5	1	Increment/Decrement	109	1	
	2	While	46	1	
	5	Do-while	91-118	1	
	6	For loop		1	
	11	Files input/output both	132-136	2	
	12	No breaks in loops ******			Failed Project if included
****** Not required to show			Total	30	

Cross Reference for Project 2

You are to fill-in with where located in code

Chapter	Section	Topic	Where Line #"s	Pts	Notes
6		Functions			
	3	Function Prototypes	29-32	4	Always use prototypes
	5	Pass by Value	59	4	
	8	return	364	4	A value from a function
	9	returning boolean	140	4	
	10	Global Variables		XXX	Do not use global variables -100 pts
	11	static variables	325	4	
	12	defaulted arguments	146	4	
	13	pass by reference		4	
	14	overloading		5	
	15	exit() function	141	4	
7		Arrays			
	1 to 6	Single Dimensioned Arrays	324	3	
	7	Parallel Arrays		2	
	8	Single Dimensioned as Function Arguments		2	
	9	2 Dimensioned Arrays		2	Emulate style in book/in class repositiory
	12	STL Vectors		2	
		Passing Arrays to and from Functions		5	
		Passing Vectors to and from Functions		5	
8		Searching and Sorting Arrays			
	3	Bubble Sort		4	
	3	Selection Sort		4	
	1	Linear or Binary Search		4	
****** Not required to show			Total	70	Other 30 points from Proj 1 first sheet tab