

Project 1

<Blackjack>

CIS 5 -41366

Michael Guerrero

02/05/2021

Program Summary

This is my attempt at Blackjack with only using knowledge from Chapters 1-5. The user starts off at the main menu where they can check their current balance, win stats, and start new games.

Example:

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

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 **445 lines of code**, with around **19 variables**. Although I am sure with functions or arrays this number could probably drop a fair amount.

Project Shortcomings

During the creation of this project it started to suffer from scope creep. There were many things I wanted to include, though some were harder than others when it came to only using concepts from Chapters 1-5. Also setting up a menu system with a card naming system caused some bugs to appear so I had to spend a good

amount of time sorting those out. Examples were win declarations being given falsely, and being able to gamble more than your balance.

Project Results

The final product is a basic working of Blackjack, it does not have all of the features but the fundamentals are there. In total this project is not perfect but I believe I set myself up with a good foundation to build off of for Project 2. I hope for the next project I can implement a small database to store the players accounts and maybe even possibly add some kind of creative format to make the game look more appealing.

Pseudo-Code

Initialize Variables

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

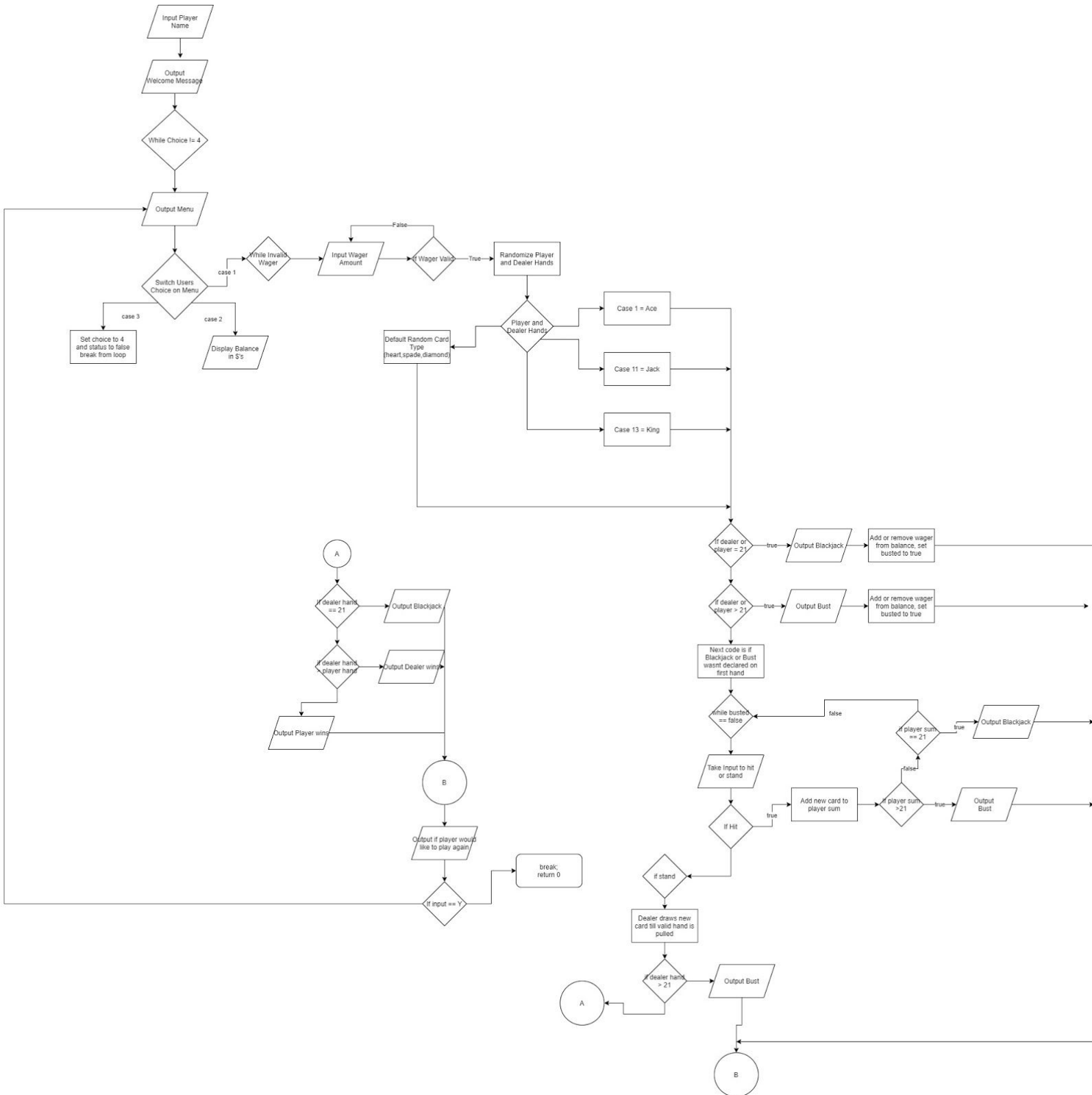
Ask player if they would like to play again

```

If yes return to main 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	8	iostream, iomanip, cmath, cstdlib, fstream, string, ctime
	4	variables/literals			No variables in global area, failed project!
	5	Identifiers			
	6	Integers	33	3	
	7	Characters	37	3	
	8	Strings	36	3	
	9	Floats No Doubles	35	3	Using doubles will fail the project, floats OK!
	10	Bools	38	4	
	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	5	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 repository
3	1	cin			
	2	Math Expression			
	3	Mixing data types ****			
	4	Overflow/Underflow ****			
	5	Type Casting	31	4	
	6	Multiple assignment *****			
	7	Formatting output	407	4	
	8	Strings	47	3	
	9	Math Library		4	All libraries included have to be used
	10	Hand tracing *****			
4	1	Relational Operators			
	2	if	384-390	4	Independent if
	4	If-else	396-404	4	
	5	Nesting	67-77	4	
	6	If-else-if	71-78	4	
	7	Flags *****			
	8	Logical operators	241-245	4	

	9	Math Library		4	All libraries included have to be used
	10	Hand tracing *****			
4	1	Relational Operators			
	2	if	384-390	4	Independent if
	4	If-else	396-404	4	
	5	Nesting	67-77	4	
	6	If-else-if	71-78	4	
	7	Flags *****			
	8	Logical operators	241-245	4	
	11	Validating user input	72-76	4	
	13	Conditional Operator		4	
	14	Switch	65	4	
5	1	Increment/Decrement	237	4	
	2	While	69	4	
	5	Do-while	84-425	4	
	6	For loop		4	
	11	Files input/output both		8	
	12	No breaks in loops *****			Failed Project if included
***** Not required to show			Total	100	