CIS-5

BlackJack Game

Created by: Matthew Bowers

Table of context

Introduction:

"Money won is twice as sweet as money earned"
-Paul Newman

Prompt:

Why I decided to make the blackjack game for a few reasons, the main reason why was that it has a simple enough rules that anyone can play, another reason why I picked the game was because it was my favorite card game growing up so it seemed like a perfect fit to make it in C++.

Rules:

Base Rules

The rules of the game are very simple, the goal of the game is to get to "21" or as close as possible.It becomes a challenge to player to not go over '21' and at the same time beat the "Dealer". When starting the player is given two cards at random from a deck of cards, the face value of the cards it what they total, (for example a 7 of clubs = 7), with some exceptions to this rule, all face cards equal 10 and the Ace can equal (11,10,or 1) depending on what the player wants to do with it.

When the game starts the player has an initial number that they get to work with, for example at the start of the game the player can have from 2 all the way to 21, from the start and this decided their next move.

Betting

Blackjack itself is a gambling game, it is played in casinos across the country. The game played with a chance of earning twice of what you bet has many risk involved. Since the first introduction of the project, I added a way to bet before the game has officially began to get the sense of playing for a risk added to the game.(The player has a $100 they can bet from and can choose how much they are willing to bet). If your the person that doesnt love betting imaginary money, well they have the option to bet $0.

After drawing two cards:

The player then has a choice to make, they can chose to draw another card or stay, this choice varys from game to game. The player may have a low hand and has to hit in order to get a higher hand as the reward is better than the risk.

The Game itself in itself is a risk and reward game, there are many risk taken when drawing a card but the reward may be winning when if otherwise would have lost.

My Approach to the Game:

Building a texted based card game from C plus plus was a challenge to begin with, and there were major obstacles to overcome.
Obstacles 1:
How to get a two cards drawn to a player at random every time they play the game (didn't make a deck of cards of “52” different cards, and the reason being is that in real blackjack the dealer has more than one deck so it is entirely possible to get the same card twice)
Obstacle 2:
How would I store the data from the cards drawn, especially for cards such as the Ace which can have 3 Different values depending on what's best for the situation.
Obstacle 3:
How would I have an opponent to play against as I didn't want to make a 2 player game, and in real blackjack there is a dealer that everyone is playing against .

Similarities to the game:

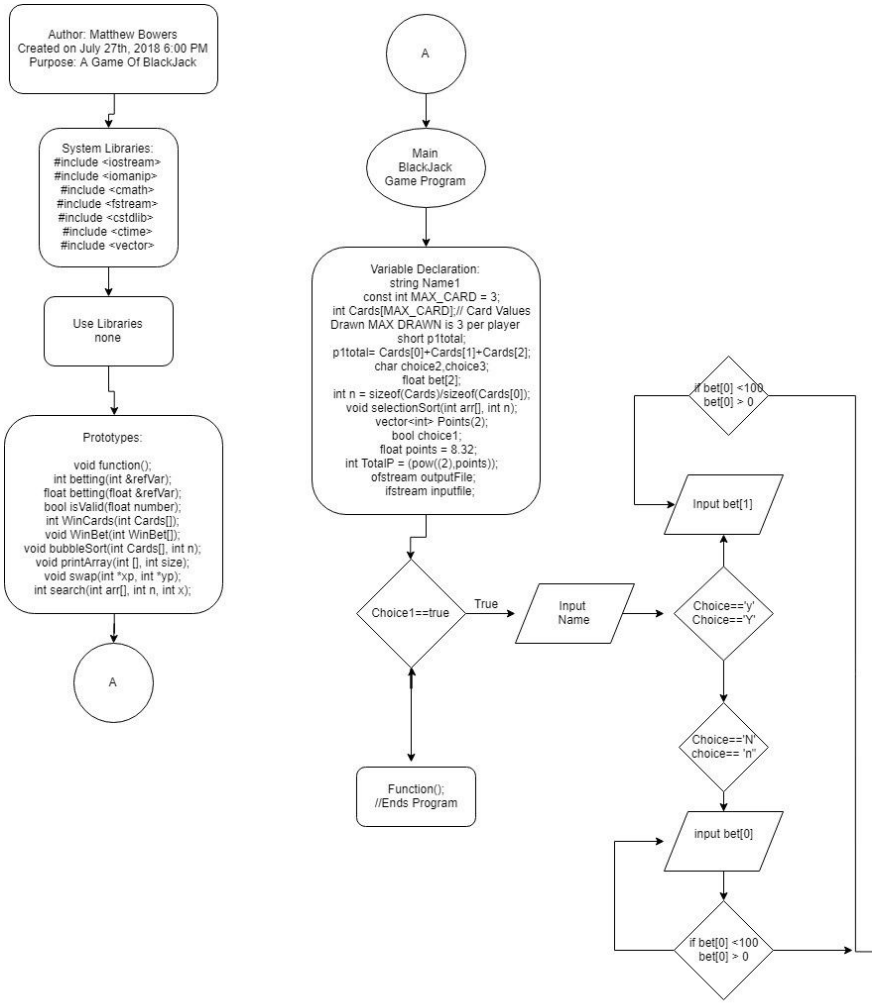
There a bunch of similarities I made to the game most of the core functions of the game are implemented such as a dealer, betting , being able to draw more cards.

Differences:

The key differences to the game compared to the program I made are the limitations of the game, and the randomness of drawing a card. I also limited the amount that the Player can draw based on the fact it made it more simple.

(CONTINUE)

The logic behind the code:

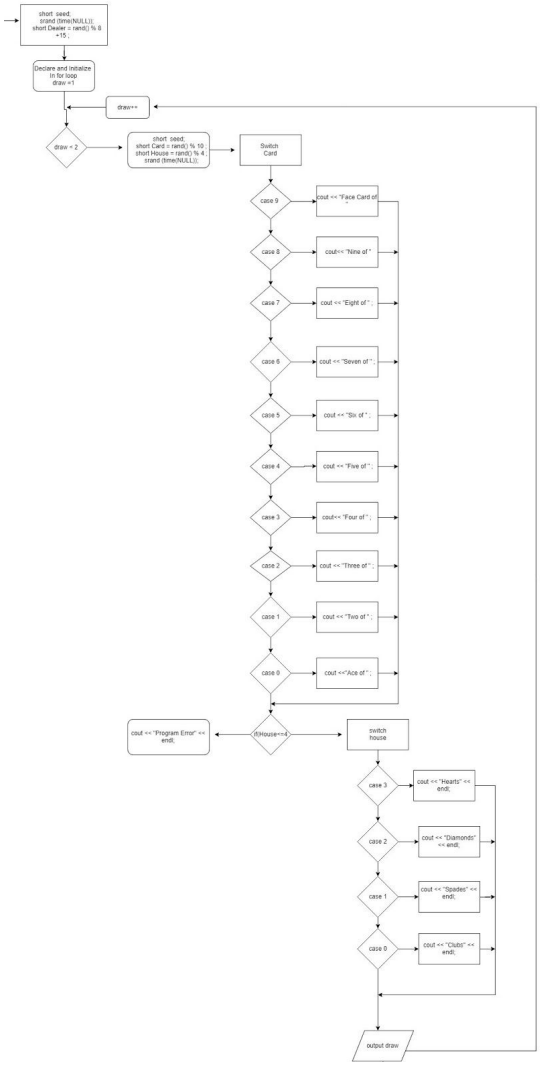


Opening Comments
Bring in system libraries
Set prototypes for project
Enter Main
Declare Variables that are used in the project

If user enters 1

They now input name
If user enters 0
The function runs to end the program

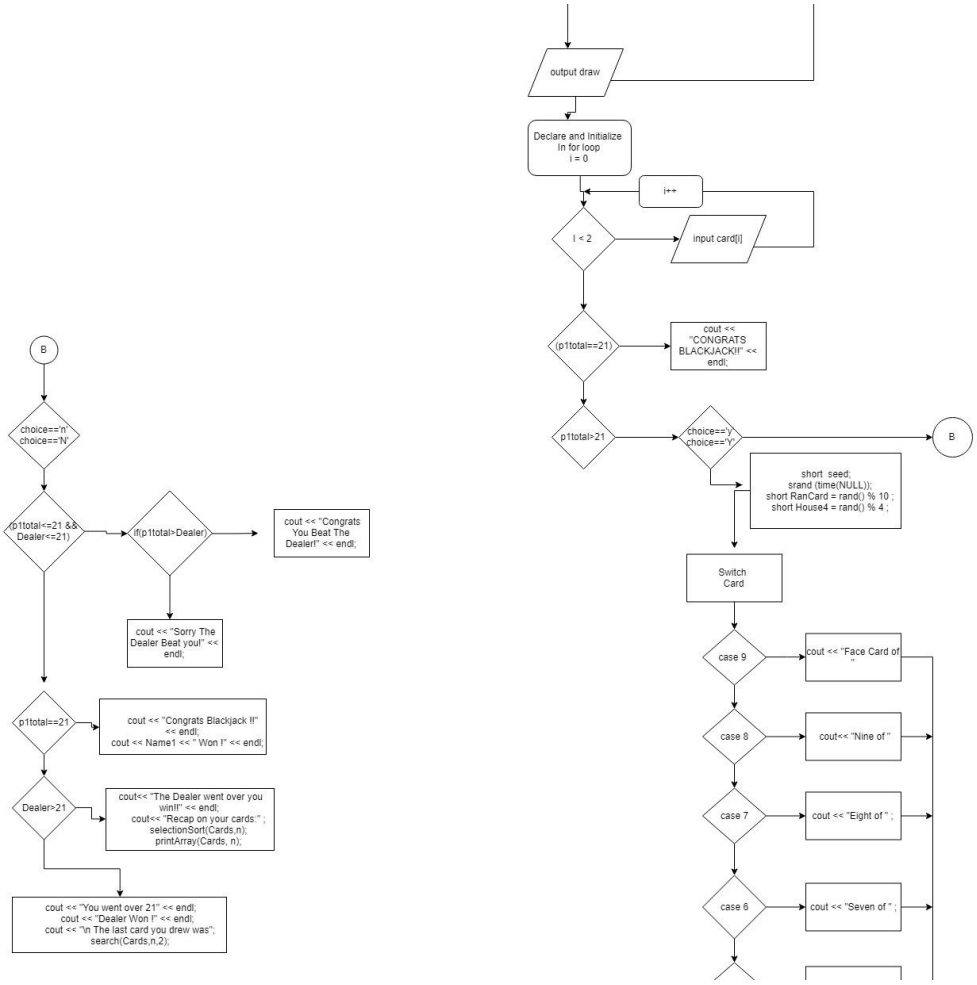
After the user has a choice between entering a bet with change or without
The program confirms its within the limits of 1-100 dollars



Then the program runs through a for statement that will run two times
Through a switch statement and a random number after this runs twice it

Will out put two random cards with the type of card they are

Ex: seven of hearts, five of clubs



After the program will run through a for statement twice, asking the user to input the values of cards they recieved (this is where they can choose what the value of Ace will be)

If they get a 21 its an automatic blackjack

If not the program continues, as its impossible to get over ‘21’ with two cards.

It prompts the user if they will like to draw again or stay

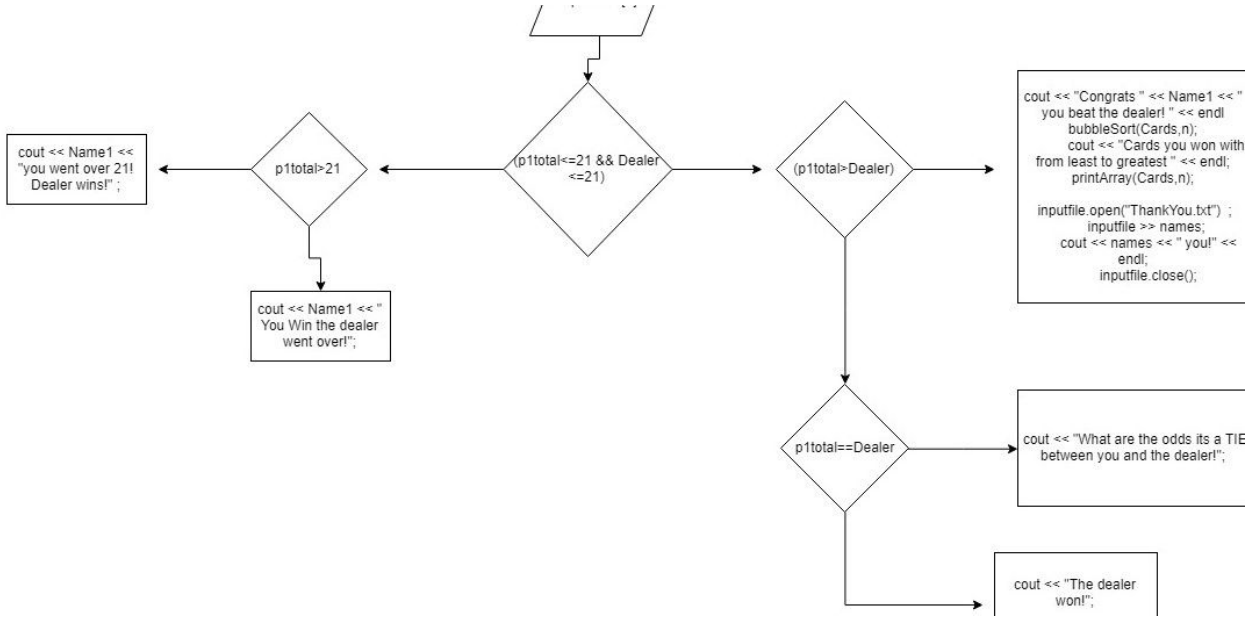
THEY DON'T DRAW AGAIN:
If the user uses to stay -> B

Then the program will then decided a winner based on different if statements.
IF STATEMENTS TO CHOOSE A WINNER:

If both hands are under ‘21’ the program will then decide who has a higher number and pick a winner.

Next the program will see if the user has a ‘21’ as there value and if they do you win with a blackjack.

Next is the dealer went over 21, its a automatic win for the user playing.



IF THEY CHOSE TO DRAW AGAIN:
The program runs through the same switch statement initially, to get a new card

The user then inputs the value of the new card and based on the results of drawing the new card there is a winner drawn.

If both hands are under ‘21’ the program will then decide who has a higher number and pick a winner.

Next the program will see if the user has a ‘21’ as there value and if they do you win with a blackjack.

Next is the dealer went over 21, its a automatic win for the user playing.

If the dealer's hand and the Users hand are equal then it is declared a tie.

OUTPUTS FOR LOSING AND WINNING:
Each route of either winning or losing has a different output, if won the function is used to declare how much money they win.
If they lose they are always getting $0 no function needed.

Also implemented are displays of the cards they won with from greatest to least , with the sort function used, there are different custom outputs based on how the user one or lost.

(Full Flow Chart will with the files of the program)

Version 1 Notes:

After Getting the switch statement to work in order for a random card to be drawn, there was difficulty in being able to store the random number as a variable to be used again, instead im having the user input the the value of cards with cin >> input. This solves the issue that I was having with Ace as it could be 3 different types of values and it changes if the player draws again. The user inputting the value they want to use with Ace fixes that problem.

Switch Statements used:

I made two random number generators the first is for the card itself and the second it for the type of house(as of yet this does not change the game outcome but it makes it feel more of a card game)
I used a For loop to generate 2 cards for play, as it outputs a different card each time, for example
Three of hearts

Four of clubs

Version 2 Notes:

Added another switch statement to be able to draw another card at random, this leads to many types of outcomes and used a lot of If and Else statements to get this done. How this was done was adding a Char ‘y’ or ‘no’ statement and if they chose to draw again would use the same code that was used to draw another card but just a single time. Alot of IF Else statements nested.

Dealer: Initially I wanted to make it a Two player game but for now I chose to make it a One player game against a “Dealer”, however the dealer can not make choices , the number for his hand is chosen at random at the start of the round, (I made it so the dealer can go over 21 as well)
Also no numbers less than 15 as in actual blackjack the dealer must hit on a 15 or lower so they could not have a hand that low.

Version 3 notes:

Added Text to the beginning of the game to make the game feel more game like, also inputted a true or false statement at the beginning of the game asking the user if they wanted to play the game. Also fixed issue where after drawing a card, there would be no winner chosen.

Future Changes to the game: I would like to implement where the user doesn't have to enter the cards they received but with my current knowledge of coding I couldn't seem to figure out, as well as the issue I was running it with the card ace. Also would like to add some type of point system into the game.

Version 4 Notes:

I implemented Betting and started to implement the new things learned from chapter 6 into the program such as exit(); function as included prototypes. In this version I changed the game around and added more things to the end of winning or losing outputs.

All of chapter 6s checklist are included in this build.

Version 5 Notes:

Implemented arrays and functions to the project as a way to store data and implemented them into functions ,as well as implemented search functions and sorting functions.

Finial build, there is now betting that displays how much money you won, as well as different functions that add to the program.

EXAMPLE OUTPUTS: will be included in the folders as well