Project 1
Black-Jack

Danny Obeid

CIS-17C

4/26/2020

INTRODUCTION:

Title: Black-Jack

Rules:

The rules of Black-Jack are simple, it is you versus the dealer. The goal is beat the dealer's hand without going over 21. Face cards are worth 10 and Aces are worth 11 or 1, depending on your choice. Each player starts with two cards and can hit to add another until you're either satisfied with your hand, or you go over. If you go over 21, you lose. If the dealer goes over 21 you win! If you both get 21 it's a standoff. Good luck!

SUMMARY:

Lines: 530+

There are a lot of different ways to program the game of Black-Jack, but with the specific rules set it took quite a bit of adjusting and creativity to mold this into a successful program. Since the use of vectors was prohibited, I had to utilize other functionalities of the STL such as lists, maps, stacks etc. It took a bit of trial and error, but in the end, it worked out extremely well. The time frame of completion was about 8-9 days for about 3-4 hours a day. It was difficult in the beginning since I needed to get more familiar with the implementations of the STL containers since it was the first time I worked with a majority of them. Once that was taken care of, it was just mapping out how the game should be played, and in what order to program some of the functions. I had to reference some websites and seek help from tutors to fix some syntax issues I was having. This project is located on my public GitHub profile in a public Repo called Dobeid17/Project1_Black-Jack.

PSEUDO CODE:

Create struct and initialize data

Bring up main menu with options

Switch choice

Case 1: hear rules is selected display rules then ask play or quit

Case 2: call function playGame()

Case 3: quit

```
playGame()
While balance >= 5 && play == true
        Set balance and take in bet
        While bet < 5 && bet > balance
                Invalid bet, try again
        Call shuDeck() and assign to deck
        Draw cards to hand
        If you have an ace, ask to change it to an 11
        Show one card of dealers hand
        Ask to hit or stay
        While toHit == 1 && toPlay != 1
                Call hitCard() to add card
                Call isOver() check to see if it goes over 21
                        If over 21 you lose bet ask to play again or quit
                                If toPlay == 2 quit program
        If toPlay != 1
                display dealer hand
                Call compareHands() to check if dealer won before hitting
                If dealWon == true
                        You lose ask to play again
                        If toPlay == 2 quit program
                While isDealOver() == false && dealWon == false
                call hitCard()
                Call isDealOver() check if dealer hand goes over 21
                If dealer goes over 21 you win
                        Update balance ask to play again or quit
                If dealBust == false
                        If compareHands() == 1
```

You win update balance

Else if compareHands) == -1

You lose display balance

Else

It’s a draw return balance

Ask to play again or quit

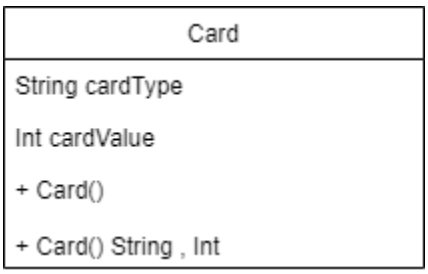
If toPlay == 2

return

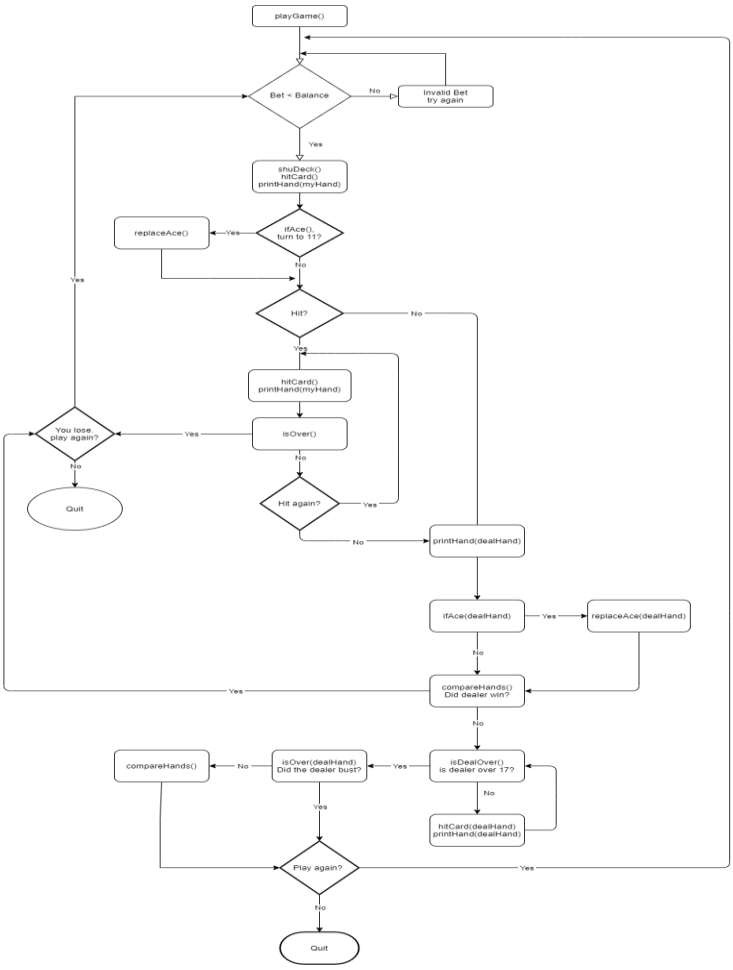
If balance < 5

Insufficient funds quit program

UML:



FLOWCHART:



FUNCTIONS:

void printHand(list<Card> hand);

- This function is used to display the hands of either the user or the dealer. It was called after the first initial draw, after the player or dealer hits, and when flipping the dealers hand.

void playGame();

- This is where the entire game is played. All other functions are called inside this function in order for the game to run. Where the player places their bets, collects their cards, hits cards, and also where the dealer is programmed to play. This is the root of the entire program.

stack<Card> shuDeck();

- This is where the deck we are going to play with is shuffled. Also, where the values of the cards are set along with their suits. Every face card must have a max value of 10, so

that is where it was set. Once it is verified that the player has enough the play, this function gets called.

Card hitCard(stack<Card>& deck);

- Whenever the player or the dealer adds a new card to their card, this function is called. If the player wants to hit after their initial pair cards this function is called. Also called if the dealer is under 17 and will continue to be called until it is over 17, or busts.

bool isOver(list<Card> hand);

- Function that checks if either the player or dealer is over the set target of 21. If either are over, they automatically lose. Called after every hit, for the player or dealer.

bool isDealOver(list<Card> dealHand);

- This function checks if the dealer is over the target limit of 17 to decide whether or not the dealer needs to hit again. As long as the dealer is under 17, hitCard() will continue to run.

bool is21(list<Card> hand);

- Function to check if the dealer reaches the target amount of 21 exactly. This is called after the final hit by the dealer.

bool ifAce(list<Card> hand);

- Once the first pair of cards are distributed, this function is called to see if either the player or user is carrying an ace. If an ace is identified it returns true and then runs replaceAce().

void replaceAce(list<Card> &hand)

- This function is called if ifAce() returns true. If true it swaps the 1 the player or dealer is carrying with an 11.

int compareHands(list<Card> myHand, list<Card> dealHand); wins if neither are 21

- If neither the player nor the dealer get exactly 21, this function Is called to compare their hands and see who's is higher. It is called after the dealer reveals their full hand to check if the dealer even needs to hit. After that, it called after it checks out that no one got 21, and no one busted. Decides who has the greater hand.

CHECK OFF LIST:

1. Container Class
    1. Sequences

- List: list<Card> myHand , and list<Card> dealHand were used to store the cards in both the players hand, and the dealers hand. These were referenced and iterated through multiple times throughout the code.

2. Associative Containers
    - Map: unordered_map<int, Card> = newDeck. An unordered map was used in the creation of the deck of cards to be used shuffled and distributed. It took in data of type int, and the struct object Card I created above.
3. Container Adapters:
    - Stack: stack<Card> doneDeck. This stack contained the shuffled deck we got from the unordered_map above. This was a crucial piece of the program because without a deck, the game cannot be played.

2. Iterators
    1. Forward Iterator:
        - list<Card>::iterator it = hand.begin(). The forward iterator was used many times in the code since it was what I used to advance through the list to access the data in it in order. Using advance(it, 1) inside of a loop I was able to go through the list one element at a time and sum its values to determine the overall hands value.
    2. Random Access Iterator:
        - unordered_map<int, Card>::iterator random_it = newDeck.begin(). This was my random access iterator and it was used in the shuffling of the deck.

3. Algorithms
    1. Mutating Algorithm
        - Emplace – Used in the shuDeck() function to put the new card on top of the unordered map.

REFERENCES:

Stackoverflow.com

Textbook

Gaddis Getting started with C++

Sample Output:

Hello and welcome to Black Jack!

What would you like to do?

1.) Hear rules.

2.) Play.

3.) Quit.

1

The rules of Black Jack are simple, it is you versus the dealer. The goal is beat the dealers hand without going over 21. Face cards are worth 10 and Aces are worth 11 or 1. Each player starts with two cards and can hit to add another. If you go over 21, you lose. If the dealer goes over 21 you win! If it is a standoff your bet gets returned. Good luck!

2.) Play.

3.) Quit.

2

Here is your current balance: $100

How much do you want to bet this hand?

Must be more than $5 and less than $ 100

15

Here is your hand: 8 Spades 4 Spades

Here is what the dealer is showing: 10 Diamonds

What are you going to do?

1.) Hit

2.) Stay

1

This is your new hand

8 Spades 4 Spades 4 Hearts

What are you going to do?

1.) Hit

2.) Stay

2

Here is the dealers hand.

7 Clubs 10 Diamonds

You Lose!

Here is your new balance: $ 85

What do you want to do?

1.) Play again.

2.) Quit

1

How much do you want to bet this hand?

Must be more than $5 and less than $ 85

15

Here is your hand: 4 Diamonds 1 Hearts

You pulled an Ace. Would you like to turn your Ace into an 11? (y/n)

y

Here is your new hand:

4 Diamonds 11 Hearts

Here is what the dealer is showing: 3 Diamonds

What are you going to do?

1.) Hit

2.) Stay

1

This is your new hand

4 Diamonds 11 Hearts 8 Diamonds

You're over 21! You Lose. Here is your balance: $70

Play again?

1.) Yes.

2.) Quit.

1

How much do you want to bet this hand?

Must be more than $5 and less than $ 70

30

Here is your hand: 10 Spades 5 Hearts

Here is what the dealer is showing: 7 Clubs

What are you going to do?

1.) Hit

2.) Stay

2

Here is the dealers hand.

4 Clubs 7 Clubs

The dealer hits. This is the dealers new hand.

4 Clubs 7 Clubs 2 Spades

The dealer hits. This is the dealers new hand.

4 Clubs 7 Clubs 2 Spades 1 Spades

The dealer hits. This is the dealers new hand.

4 Clubs 7 Clubs 2 Spades 1 Spades 10 Diamonds

The dealer busted! You Win!

Here is your balance: $100 continue playing?

What do you want to do?

1.) Play again.

2.) Quit