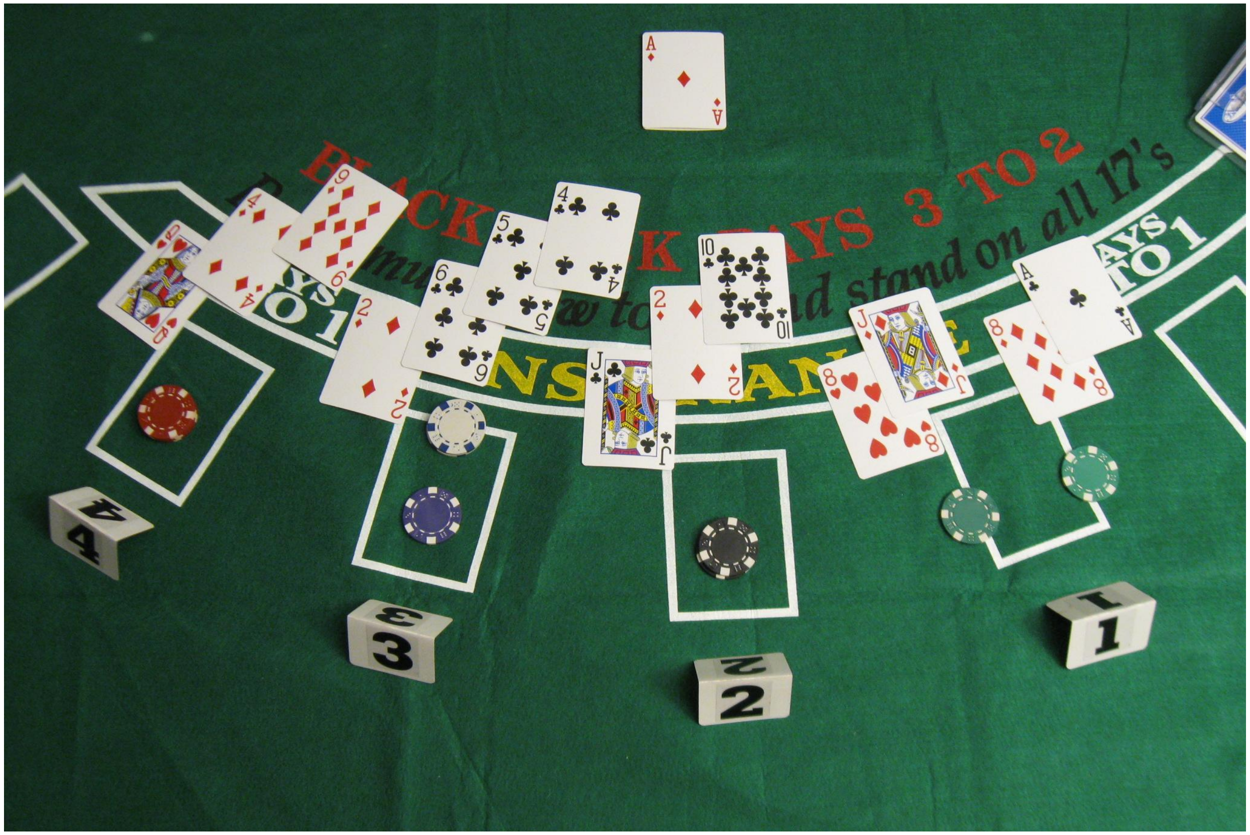

Project 1 - BlackJack

02.03.2022

Course: CIS-5

Author: Mike Ibrahim

Game Rules

- Every player is given an initial investment of $100.
- The dealer is given two (2) random cards, one of which is displayed to the user.
- The player is given two (2) random cards, both displayed to the user.
- The player can choose to hit or stand so long as his value is less than 21.
- If the player reaches $0, then the game is over.
- The value for aces for the dealer is always 11.
- The value for aces for the player is what the player decides (11 or 1).
- All face cards are worth 10 points (Jack, King, Queen).
- The dealer is forced to draw a card if his value is 17 or less.

Pseudo Code In-Text (Explained in English)

The code starts by declaring 22 variables. 5 of those variables are the controllers for the cards, so they are set to use the random seed initialized to give a random value between 1 and 11. Afterward, the rules are printed out, as well as the prize money. Then, we open the file where the user name and age are stored, and we make sure that it is not the EOF (End Of File), then we ask the user if this is his name, if so, we store his name and age to the proper variables and move on, if none of the users are him, then he declares his own new user. We then move on to checking age, if they are younger than 18, they cannot play, otherwise, we welcome them, and make sure to take note of their name and age in our file. The user is then prompted if he wants to play or not. If he chooses not to play, then the program exits and closes the opened files. If he chooses to play, the user is given the initial value of $100 and has to make a bet below $100 and greater than $0. If it is an invalid bet, exit the program. If the bet is valid, take away that value from money, then continue to get the face card, both dealer cards, both player cards, and if the user has an ace, prompt them with what value they want their ace to hold (11 or 1). We calculate the totals, and if the dealer has less than or equal to 17, he has to draw cards until that value is met or exceeded. We also give the user an option to hit or stand. Then we do the winnings of who won and who lost and make sure to deduct or add that value to money. And then we go through it all again.

Summary

- Lines: 255 (spaces included)
- Number of variables: 25

The project can be extended in the following project through the usage of functions and arrays. It includes many concepts that were covered through the lessons, and a couple of concepts not covered.

The project took me a decent bit of time, it was the tedious things that took a very long time. I do believe I could've done a better job, but there was such a strict time period, and during that period, I think I did a good job.

Flowchart

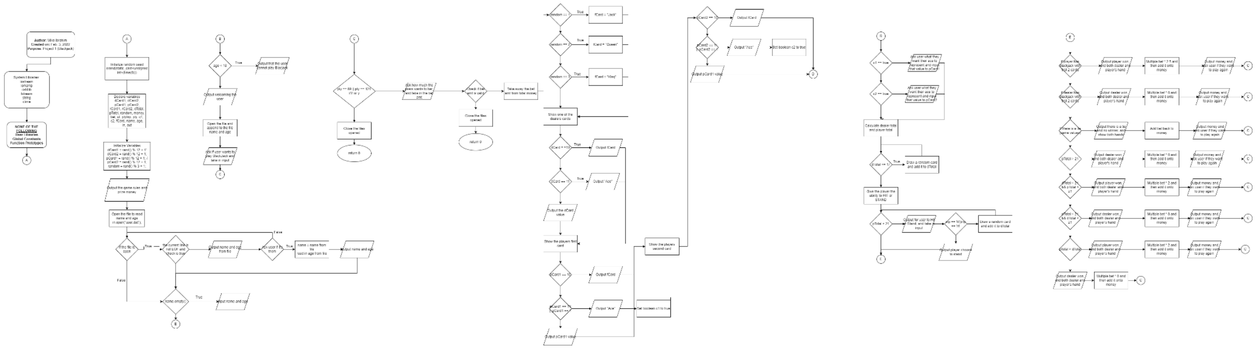
Author: Mike Ibrahim
Created on: Feb. 3, 2022
Purpose: Project 1 (Blackjack)
System Libraries
iostream
iomanip
cstdlib
fstream
string
ctime
NONE OF THE FOLLOWING
User Libraries
Global Constants
Function Prototypes
A
Initialize random seed srand(static_cast<unsigned int>(time(0)))
Declare variables dCard1, dCard2, pCard1, pCard2, rCard1, rCard2, dTotal, pTotal, random, money, bet, ut, plyInp, ply, c1, c2, fCard, name, age, in, out
Initialize Variables
dCard1 = rand() % 12 + 1;
dCard2 = rand() % 12 + 1;
pCard1 = rand() % 12 + 1;
pCard2 = rand() % 12 + 1;
random = rand() % 3 + 1;
Output the game rules and prize money
Open the file to read name and age in.open("user.dat");
If the file is open
True
the current line is not EOF and check is true
Output name and age from file
Ask user if it's them
True
name = name from file read in age from file
Output name and age
False
False
name.empty()
True
Input name and age
B
B
age < 18
True
Output that the user cannot play Blackjack
Output welcoming the user
Open the file and append to the file name and age
Ask if user wants to play Blackjack and take in input
C
C
ply == 'Y' || ply == 'y'
Ask how much the player wants to bet and take in the bet amt
Close the files opened
return 0
Check if bet amt is valid
Take away the bet amt from total money
Close the files opened
return 0
random == 1
True
fCard = "Jack"
random == 2
True
fCard = "Queen"
random == 3
True
fCard = "King"
Show one of the dealers cards
dCard1 == 10
True
Output fCard
dCard == 11
True
Output "Ace"
Output the dCard value
Show the players first card
pCard1 == 10
Output fCard
pCard1 == 11 || pCard1 == 1
Output "Ace"
Set boolean c1 to true
Output pCard1 value
Show the players second card
pCard2 == 10
Output fCard
pCard2 == 11 || pCard2 == 1
Output "Ace"
Set boolean c2 to true
Output pCard1 value
D
D
c1 == true
Ask user what they want their ace to represent and input that value to pCard1
c2 == true
Ask user what they want their ace to represent and input that value to pCard2
Calculate dealer total and player total
dTotal <= 17
True
Draw a random card and add it to dTotal
Give the player the ability to HIT or STAND
pTotal < 21
Output for user to Hit or Stand, and take in input
inp == 'h' || inp == 'H'
Draw a random card and add it to dTotal
Output player chose to stand
E
E
If player has blackjack with first 2 cards
Output player won, and both dealer and player's hand
Multiple bet * 2.5 and then add it onto money
Output money and ask user if they want to play again
C
If dealer has blackjack with first 2 cards
Output dealer won, and both dealer and player's hand
Multiple bet * 0 and then add it onto money
Output money and ask user if they want to play again
C
If there is a tie (same values)
Output there is a tie and no winner, and show both hands
Add bet back to money
Output money and ask user if they want to play again
C
pTotal > 21
Output dealer won, and both dealer and player's hand
Multiple bet * 0 and then add it onto money
Output money and ask user if they want to play again
C
dTotal > 21 && pTotal < 21
Output player won, and both dealer and player's hand
Multiple bet * 2 and then add it onto money
Output money and ask user if they want to play again
C
pTotal > 21 && dTotal < 21
Output dealer won, and both dealer and player's hand
Multiple bet * 0 and then add it onto money
Output money and ask user if they want to play again
C
pTotal > dTotal
Output player won, and both dealer and player's hand
Multiple bet * 2 and then add it onto money
Output money and ask user if they want to play again
C
Output dealer won, and both dealer and player's hand
Multiple bet * 0 and then add it onto money
C

Image with zoom accessibility can be accessed here

References

- Presentations offered by professor and Gaddis/Savitch
- Zoom lectures by professor
- Microsoft Documentation of <fstream> library

Program

```
//System Libraries
#include <iostream> //I/O Library
#include <iomanip>  //Formatting Library
#include <cmath>    //Math Functions
#include <cstdlib>  //Random Function Library
#include <fstream>  //File Library
#include <string>   //String Library
#include <ctime>    //Time Library
using namespace std;

//Function Prototypes

int main(int argc, char** argv) {
	//Initialize Random Seed

	srand(static_cast<unsigned int>(time(0)));


	//Declare Variables

	unsigned short dCard1, dCard2, pCard1, pCard2, rCard1, rCard2; //Variable for Dealer Cards, Player Cards, and 2
random cards that will be pulled out for dealer to draw, and for player to her

	unsigned short dTotal, pTotal;                                          //Calculates the total of all cards

	unsigned short random;                                                      //Random value to choose
which Face Card is presented

	float money = 100.0f;                                                   //Initial Money the user starts
with, in dollars ($)

	float bet;                                                                      //How much the
user wants to bet in dollars ($)

	int uI = 0;                                                                      //User input asking
player whether they want their ACE to represent a value of 11 or 1

	char plyInp, ply = 'n';                                                 //PlyInp is for player input, asking
if the users want to HIT or STAND. Ply is to check if the user wants to play BlackJack or No

	bool c1 = false, c2 = false;                                        //C1 is the check to see if the first card
for the user is an ACE, C2 is the check to see if their second card is an ACE

	string fCard = "";                                                          //The face card name
based on the random value

	string name = "";                                                           //Users name

	unsigned short age;                                                         //Users Age
```

```
fstream in, out;                                                        //File input and output

//Initialize the values
dCard1 = rand() % 12 + 1;     //Dealer Card 1
dCard2 = rand() % 12 + 1;     //Dealer Card 2
pCard1 = rand() % 12 + 1;     //Player Card 1
pCard2 = rand() % 12 + 1;     //Player Card 1
random = rand() % 3 + 1;      //Randomt to choose which 3 cards they have

//Start playing the game
cout << "*|***************************************************************************|*" << endl;
cout << "*|*************************Welcome to BlackJack************************|*" << endl;
cout << "*| Rules                                              |*" << endl;
cout << "*| 1: The value for Aces is what the player decides.                |*" << endl;
cout << "*| 2. The value for Aces for the Dealer is always 11                |*" << endl;
cout << "*| 3: All face cards are worth 10 points (Jack, King, Queen)           |*" << endl;
cout << "*| 4. The dealer is forced to draw a card if his value is 17 or less    |*" << endl;
cout << "*| Prize Money                                            |*" << endl;
cout << "*| The Player will start with $100.00.                         |*" << endl;
cout << "*| If the player gets a BlackJack and the dealer does not, then the ... |*" << endl;
cout << "*| ... player will receive 1.5x their initial bet.                 |*" << endl;
cout << "*| The game is over once the player has no money left.                |*" << endl;
cout << "*|*********************************************************************|*" << endl;

string data;    //To store in file information
char user;                  //For user to confirm if it's them or not
bool check = true; //To stop the while loop asking the user if it's them
int count = 0;    //Makes sure we are only checking the name section in the file, and not the age
in.open("user.dat"); //Open the file
if (in.is_open()) {          //Check if the file is open
    while (in.peek() != EOF && check) {  //While the next line is not "END OF FILE" and if the user hasn't confirmed it is them yet
        getline(in, data);                                    //Get the line and save it to data
        if (count % 2 == 0) {                            //Check if the line is on a name
            cout << "Is this you?" << endl;
            cout << data << endl;
            cin >> user;
            if (user == 'y' || user == 'Y') {//If the user confirms that this is them
                check = false;                                        //End the while loop
                name = data;                                          //Name is data which is used in the getline
                in >> age;                                                        //Age is read from the file
                cout << "Name: " << name << endl;
                cout << "Age:  " << age << endl;
            }
        }
        count++;
    }

    if (name.empty()) {     //If the user never confirmed their name, then ask them for name and age
        cout << "What is your name? " << endl;
        cin >> name;
        cout << "How old are you " << name << "?" << endl;
        cin >> age;
    }
}
if (age < 18) {  //Age restriction
    cout << "Unfortunately, you are too young to play BlackJack. You can play in " << 18 - age << " years." << endl;
    return 0;
} else { //Initial Welcoming
    cout << "Welcome " << name << " let's begin!" << endl;
    out.open("user.dat", ios::app); //Write the users name and age (new lines)
    out << name << "\n"<< age << "\n";
```

```
        cout << "Would you like to play BlackJack?" << endl;
        cin >> ply;
    }

    while (ply == 89 || ply == 121) //If the user input is capital Y or lowercase Y
    {
        cout << setprecision(2) << fixed; //Set precision to output money
        cout << "How much would you like to bet with your $" << setw(4) << money << endl;
        cin >> bet;
        if (bet <= 0 || bet > money) { //If bet is an invalid money, a negative number or more money than they have
            cout << "Invalid bet, you cannot play the game." << endl;
            return 0;       //exit
        } else {
            cout << "Bet of $" << setw(4) << bet << " is placed!" << endl;
            money -= bet; //Take away the bet from their money
        }

        switch (random) {    //Get Face Card
        case 1:
            fCard = "Jack";
            break;
```

```
case 2:
        fCard = "Queen";
        break;
case 3:
        fCard = "King";
        break;
}

cout << "The Dealer's card that is shown: " << endl;        //Show one dealers card (FACE-UP)
switch (dCard1) {
case 10: cout << fCard << endl;
        break;
case 11: cout << "Ace" << endl;
        break;
default:
        cout << dCard1 << endl;
}

cout << "The value of the players cards: " << endl;

if (pCard1 == 10) {                       //Show Player Card 1
```

```
        cout << fCard << " and ";
} else if (pCard1 == 11 || pCard1 == 1) {
        cout << "Ace and ";
        c1 = true;
} else {
        cout << pCard1 << " and ";
}

if (pCard2 == 10){                      //Show Player Card 2
        cout << fCard << endl;
} else if (pCard2 == 11 || pCard2 == 1) {
        cout << "Ace" << endl;
        c2 = true;
} else {
        cout << pCard2 << endl;
}

if (c1) {                               //If player's card 1 is an ACE, ask them what value they want for it
        cout << "What do you want your Ace for Card 1 to repesent? 11 or 1: ";
        cin >> ul;
        if (ul == 11 || ul == 1) {
```

```
        pCard1 = ul;
    }
}
if (c2) {                               //If player's card 2 is an ACE, ask them what value they want for it
    cout << "What do you want your Ace for Card 2 to repesent? 11 or 1: ";
    cin >> ul;
    pCard2 = (ul == 1 || ul == 11) ? ul : 0;
}

//Calculate the totals
dTotal = dCard1 + dCard2;
pTotal = pCard1 + pCard2;

if (dTotal <= 17) {             //If the dealer has a value 17 or less, then they are forced to draw a card
    do {
        cout << "Dealer drew a card" << endl;
        rCard1 = rand() % 12 + 1;     //Random Card for Dealer to pull if below 16
        dTotal += rCard1;
    } while (dTotal <= 17);
}
```

```
        if (pTotal < 21) {                  //Allow the user to choose between HIT or STAND
            do {
                cout << "HIT or STAND? H/S: " << endl;
                cin >> plyInp;
                if (plyInp == 72 || plyInp == 104) {   //Hit (H or h)
                    cout << "Player has choose to HIT:" << endl;
                    rCard2 = rand() % 12 + 1;       //Random Card for Player to pull
                    if (rCard2 == 10) {                         //If card is 10, pick a facecard
                        cout << fCard << endl;
                    } else if (rCard2 == 11 || rCard2 == 1) {       //If the value is an Ace then ask them what they want
                        cout << "Ace" << endl;
                        cout << "What do you want your Ace for the Card to repesent? 11 or 1: " << endl;
                        cin >> ul;
                        if (ul == 11 || ul == 1) {
                            rCard2 = ul;
                        }

                    } else {
                        cout << rCard2 << endl;
                    }
                    pTotal += rCard2;
                } else if (plyInp == 83 || plyInp == 115) {     //Stand (S or s)
                    cout << "Player choose to STAND!" << endl;
                }
            } while (pTotal < 21 && (plyInp == 72 || plyInp == 104));   //Keep going as long as they want to HIT and
their total is less than 21
        }

        cout << endl;
        if ((pCard1 == 10 && pCard2 == 11) || (pCard1 == 11 && pCard2 == 10)) {        //Check if player got BlackJack off
the bat
            cout << "BlackJack! You win!" << endl;
            cout << "Dealer's Hand: " << dTotal << endl;
            cout << "Player's Hand: " << pTotal << endl;
            bet *= 2.5;
        } else if ((dCard1 == 10 && dCard2 == 11) || (dCard1 == 11 && dCard2 == 10)) { //Check if dealer got BlackJack off
the bat
            cout << "BlackJack! You Lost. Dealer Won!" << endl;
            cout << "Dealer's Hand: " << dTotal << endl;
            cout << "Player's Hand: " << pTotal << endl;
            bet *= 0;
        } else if (dTotal == pTotal) {   //If both dealer and player have the same total (TIE) then it is a bust
```

```
            cout << "There is a tie. Both dealer and player have to push. No money is lost." << endl;
            cout << "Dealer's Hand: " << dTotal << endl;
            cout << "Player's Hand: " << pTotal << endl;
        } else {
            if (pTotal > 21) {      //If the player has more than 21, then they lost
                cout << "Player lost, value is: " << pTotal << endl;
                cout << "Dealer's Hand: " << dTotal << endl;
                bet *= 0;
            } else if (dTotal > 21 && pTotal < 21) {      //If the dealer is greater than 21 but player is less than 21
                cout << "Dealer lost, value is: " << dTotal << endl;
                cout << "Player's Hand: " << pTotal << endl;
                bet += bet;
            } else if (pTotal > 21 && dTotal > 21) {      //If both the dealer and player have greater than 21, then the dealer wins
                cout << "Player lost, value is: " << pTotal << endl;
                cout << "Dealer's Hand: " << dTotal << endl;
                bet *= 0;
            } else {
                if (pTotal > dTotal) {                    //If the player has more points than the dealer
                    cout << "Player Won! Value is: " << pTotal << endl;
                    cout << "Dealer's Hand: " << dTotal << endl;
                    bet += bet;
                } else {                                        //The dealer has more points than player
                    cout << "Dealer Won! Value is: " << dTotal << endl;
                    cout << "Player's Hand: " << pTotal << endl;
                    bet *= 0;
                }
            }
        }

        money += bet;

        cout << "This is your current balance " << name << " $" << setw(4) << money << endl;
        cout << "Play again? Y/N" << endl;
        cin >> ply;
    }

    in.close();
    out.close();

    return 0;
}
```