**Project 1:**

**Blackjack 21**

**CIS-17A-40651**

**Antonio Gines**

**04/22/18**

**Intro**

Most of my family members are able to play Blackjack 21 very well and as such I have been exposed to it for a very long time. From parties to hangouts out, Blackjack has provided me with countless fun (albeit, sometimes infuriating) experiences and this familiarity was the basis for my programming project

**Rules of the Game**

Game: Blackjack 21

Blackjack is a gambling card game in which the player starts by placing a bet valued between the minimum and maximum amount of money that the house allows. The player as well as the dealer are dealt two cards and of the two cards the dealer receives, one of them is revealed. The player, with the hand they are dealt can choose to stay (keep their current cards), hit (receive another card), double (add an additional bet equal to the first one and receive a singular extra card), or, if the first two cards are equal, split (add an additional bet and split the hand into two hands where both receive an additional card). If the ultimate value of the player’s hand is 21, greater than the dealer’s hand, or if the dealer busts, the player wins and receives double his/her gross bet. If the player and the dealer have the same hand without it being a bust it is a tie and there is no gain nor loss. If the player busts or if the dealer has a higher hand, the player loses the bet.

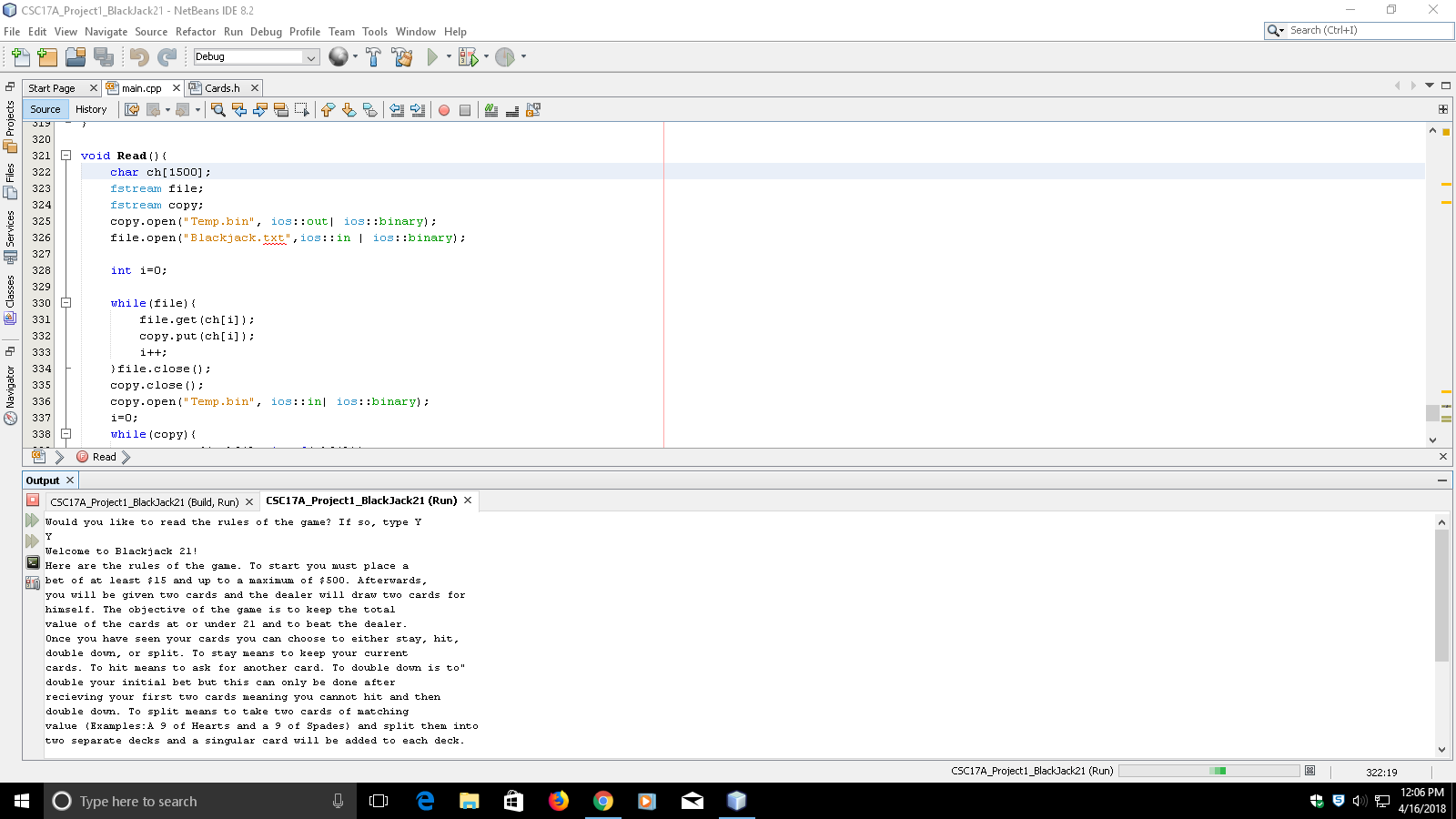
**Summary of Process**

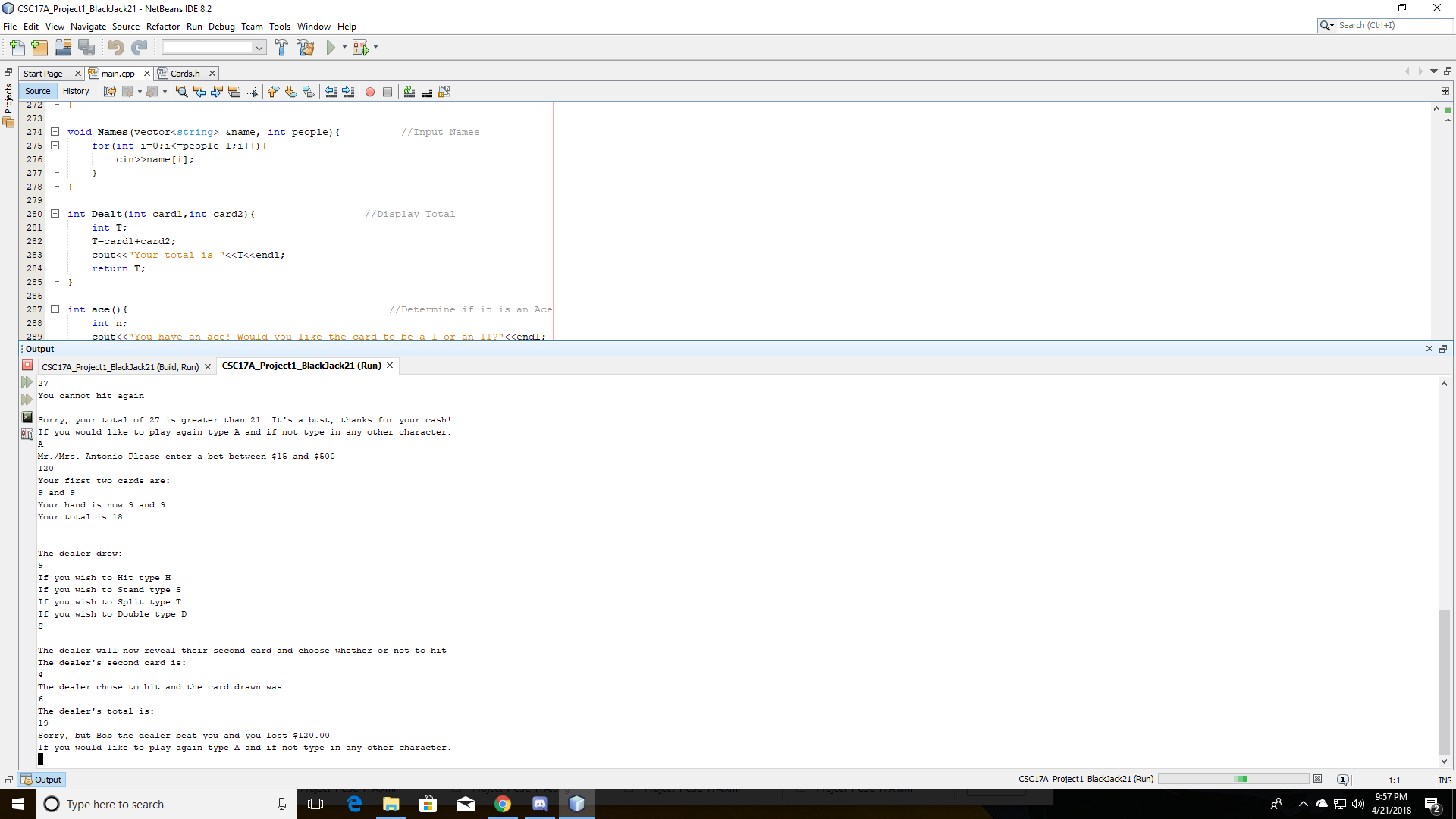
Size: 370 lines

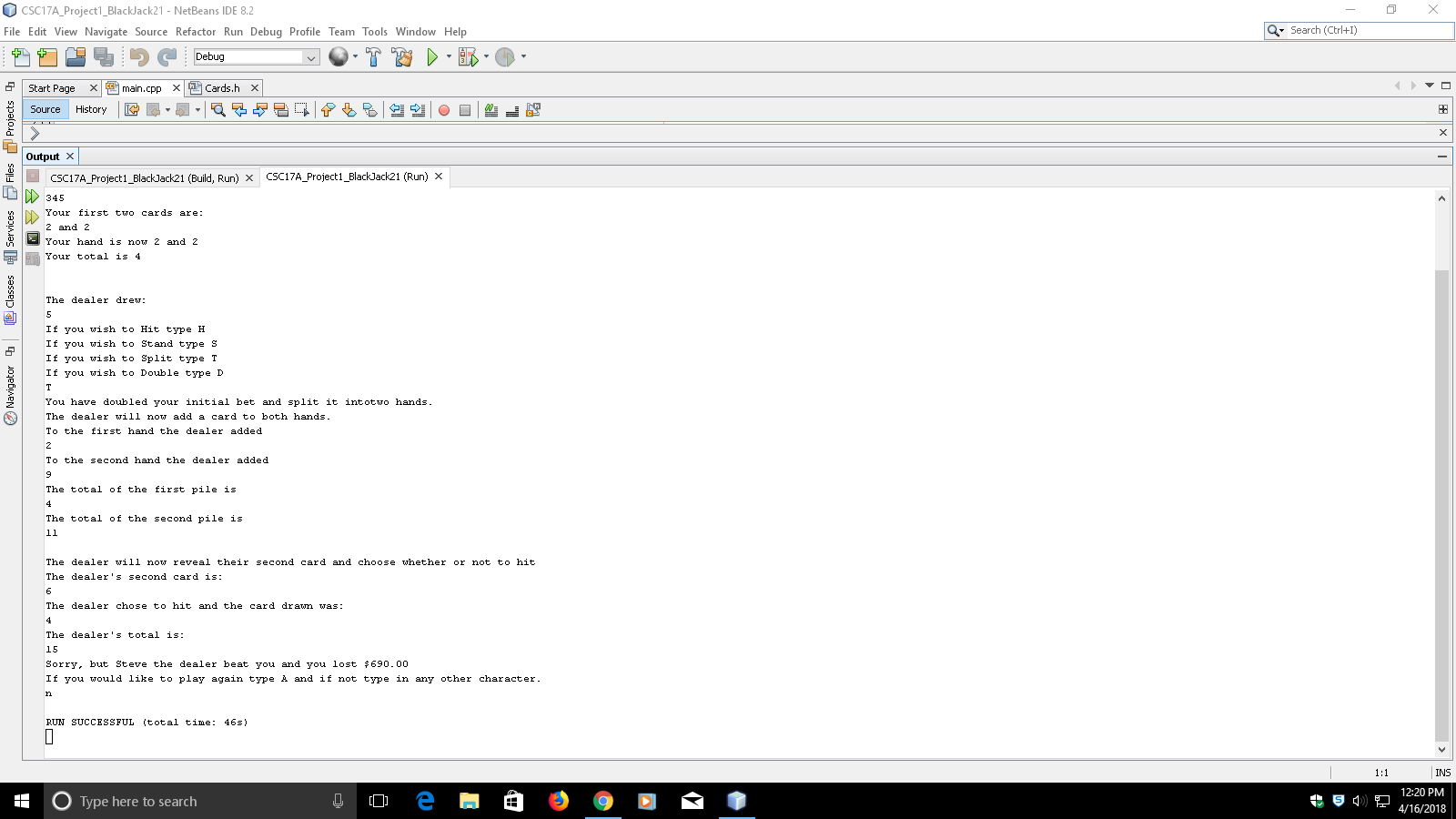
The foundation of this code came from a project from my previous semester in intro to computer science in which we also had to create a card game. It took about 5 days to implement most of the new concepts with a majority of the time being spent figuring out how to dynamically allocate an array of pointers within a structure. The rest of the time was devoted equally to creating character arrays, string objects, copying to and reading from binary files, etc..

**Proof of a Working Project**

In the case that the text is too small to see, all of these images will be included as JPEG’s in the project folder.







**Psuedo Code**

*Global Constants*

*Function Prototypes*

*Reading Instructions From A Binary File*

*Create Array of Structures and fill Pointers to Use as Cards*

*Destroy Dynamic Arrays*

*Determine Ace*

*Player’s total cards Dealt*

*Array Cards Dealt to Player and Dealer*

*Record Player’s and Dealer’s Names*

*Automatic Win if First hand is 21*

*Initalize*

*Explain the procedure of the game from file*

*Ask for name of player and dealer via Function*

*Do{*

*Call for Function to Create Array of Structures and Fill Pointers Randomly*

*While both cards are the same and equal to eleven, re-randomize*

*Ask for player’s bet while it is out of bounds*

*Give player first two cards from array*

*If one or both cards is equal to 1 or 11 got to function to determine value of ace*

*Record the value of the greatest card for later*

*If the cards are equal to 21*

*The player automatically wins*

*Else Continue*

*Display one of dealer’s cards from array*

*Ask player’s what they want to do based on their hand*

*If they don’t have a split*

*Ask if they want to hit*

*Ask if they want to stay*

*Ask if they want to double*

*If they do have a split*

*Ask if they want to hit*

*Ask if they want to stay*

*Ask if they want to double*

*Ask if they want to split*

*Switch based on choice*

*Switch hit*

*Do*

*Pull card 3 from array based on value count*

*Increment count*

*If card 3 is equal to 1 or 11 call for Ace function*

*Compare value of card 3 to first 2 and store*

*Compare value of card 3 to value potential*

*Card 3 equal to potential*

*Add to the total value of the cards*

*Display card 3*

*Display total value of cards*

*If total is less than 21*

*Ask if they want to hit again*

*Input decision to hit*

*Else they cannot hit again*

*While the total is less than 21 and if they want to hit again*

*Break out of switch*

*Switch stay*

*Break out of switch*

*Switch double*

*Pull card 3 from array*

*Increment the bet and display the bet*

*If card 3 is equal to 1 or 11 call for Ace function*

*Display card 3*

*Display total value of cards*

*If card 3 is greater than card 2*

*Store value of card 3 for later*

*Else store the value of card 1*

*Break out of the switch*

*Switch split*

*Set card 1 equal to hand 1*

*Set card 2 equal to hand 2*

*Increment the bet*

*Pull card 3 from array*

*Pull card 4 from array*

*Deal the third card to the first hand*

*Display card 3 from Array*

*If card 3 is equal to 1 or 11 call ace function*

*If card 3 is greater than card 2*

*If card 3 is greater than card 4*

*Store the value of card 3 for later*

*Deal the fourth card card to the second hand*

*Display card 4 from array*

*If card 4 is equal to 1 or 11 call ace function*

*If card 4 is greater than card 2*

*If card 4 is greater than card 3*

*Store the value of card 4 for later\*

*Total of the first hand*

*Total of the second hand*

*If the first hand is greater than the second hand*

*The total that will be used is the first hand*

*Else the total that will be used is the second hand*

*Break out of switch*

*If total is greater than 21 it is a bust and you lost*

*Else*

*The dealer’s total is calculated*

*The dealer’s second card is pulled and revealed from an array*

*While the dealer’s total is less than the player’s*

*The dealer’s hit card is pulled from array and displayed*

*If the dealer’s hand is greater than 21*

*The dealer busts and you win*

*Display the payout*

*Display the highest card in your hand*

*Else If your hand is greater than the dealer’s and less than or equal to 21*

*You beat the dealer*

*Display payout*

*Display highest card in your hand*

*Else If the dealer’s hand is higher than the player’s*

*The dealer beat you*

*Display the dealer’s largest card from function*

*Sort array with bubble sort*

*Display Sorted Cards*

*Sort array with select sort*

*Display sorted cards*

*Display bet lost*

*Else If the total of both the player’s and the dealer’s hand is equal*

*You tied, nothing was lost*

*Would you like to play again?*

*Input decision*

*}while the decision is yes*

*Function to determine Automatic Win based on First Hand*

*If number 1 + 2 = 21*

*val=true*

*Else value equal to false*

*Return Val*

*Function to Input Names*

*For i is equal to 0 to i is equal to PEOPLE*

*Input names*

*Function to Display Total*

*Total is equal to card 1 plus card 2*

*Return total*

*Function to Determine Value of an Ace*

*Input choice of either 1 or 11 for n*

*While n is not equal to 1 or 11*

*Re-enter value of n*

*Return n*

*Function to Create Arrays of Structures and FIll Pointers Randomly*

*Allocate Memory to pointers Dynamically*

*For j is equal to 0 to j is equal to the Quantity of People Playing*

*Set structure variable equal to quantity of people*

*Set structure variable equal to quantity of maximum possible size of hand*

*Dynamically Allocate Pointer in Structure*

*Int i=0;*

*For i is equal to 0 to i less than or equal to size of hand minus 1*

*Structure j/Hand i is equal to a random number between 1 and 11*

*Function to Read in the Rules from a binary file*

*Define Character Array*

*Define file stream for Blackjack.txt file*

*Define file stream for Temp.bin file*

*Open Temp.bin for output in binary mode*

*Open Blackjack.txt for input in binary mode*

*i=0*

*While Blackjack.txt is open*

*Get each character from the txt file and copy it into a character array at i*

*Copy Each character of the character array into the Binary File*

*Increment i by 1*

*Close Temp.bin*

*Close Blackjack.txt*

*Re-open Temp.bin for input in binary mode*

*i=0*

*While Temp.bin*

*Copy characters from the Temp.bin into the character array*

*Output the array at i*

*Increment i by 1*

*Close Temp.bin*

**Flowchart**

Due to its size it is in a separate file within the folder

**Code**

/\*

\* File: main.cpp

\* Author: Antonio Gines

\* Created on April, 12 , 2018, 12:37 PM

\* Purpose: Blackjack 21

\*/

//System Libraries

#include <iostream> //Used for Input/Output

#include <iomanip> //Used to format Output

#include <cstdlib> //Used for random generator

#include <ctime> //Used to seed random generator

#include <cmath> //Used to calculate largest card

#include <cstring>

#include <memory>

#include <string> //Used to input names

#include <fstream> //Used to read from file

#include <vector> //Used to contain names

using namespace std;

//User Libraries

#include "Cards.h"

//Global Constants - Math/Physics Constants, Conversions,

// 2-D Array Dimensions

//enum amount{earned};

//Function Prototypes

bool Read(fstream &, fstream &); //Input and Output Data via Binary Files

Cards \*DealHnd(int,int); //Allocate Memory and Create Cards

void Destroy(Cards \*,int); //Delete Arrays in Memory

int ace(); //Choose Whether an Ace=1 or 11

int Dealt(int ,int ); //Player's Total Cards Dealt

void Names(char [],char [],int); //Player's and Dealer's Name

bool AutoWin(int, int); //Determine if Automatic Win by 21

//Execution Begins Here

int main(int argc, char\*\* argv) {

//Declare Variables

int SIZE=10;

int Ppl=2,m=0;

int newbet,total,rng1,rng2,card1,card2,card3,card4,payout,ptntial;

int total1,total2,split1,split2,dealer1,dealer2,dealer3,dealtot;

char choice, hitagn, playagn,rule;

float bet;

bool win;

char you[20],deal[20];

string please=" Please enter a bet";

please+=" of at least $15 and at most $500";

fstream file;

fstream copy;

//Seed Random Number Generator

srand(static\_cast<int>(time(0)));

//Initialize Variables

ptntial=0; //Will be used later on as temp

//Rules of the Game

cout<<"Would you like to read the rules of the game? If so, type Y"<<endl;

cin>>rule;

if (rule=='Y'){

Read(file, copy);

}

//Begin Game

cout<<"Before we begin, what is your name and what is the "

<<"dealer's name?"<<endl;

Names(you, deal, Ppl); //Call FillAry

do{

Cards \*card;

card=DealHnd(Ppl,SIZE);

int count1=0;

newbet=0;

bet=0;

while(card[0].hand[0]==11&&card[0].hand[0]==card[0].hand[1]){

card[0].hand[1]=(rand()%11)+1;

}

do{ //Enter Player name

cout<<"Mr./Mrs. "<<you<<please<<endl;

cin>>bet;

}while(bet>500||bet<15);

card1=card[0].hand[0];

card2=card[0].hand[1];

cout<<"Your first two cards are:"<<endl;

cout<<card1<<" and "<<card2<<endl; //Deal The Cards

if(card1==1||card1==11){ //Choosing what happens if you get

card1=ace(); //1 or an 11

}if(card2==1||card2==11){

card2=ace();

}

cout<<"Your hand is now "<<to\_string(card1)<<" and "<<to\_string(card2)

<<endl;

total=Dealt(card1, card2);

//Record Largest Card's Value

if(card[0].hand[0]>card[0].hand[1]){

rng1=card1;

}else rng1=card2;

//Determine Auto Win

win=AutoWin(card1, card2);

win?cout<<"Congratulations! It's a Blackjack,you win"<<endl:cout<<endl;

if(win){ //If it is 21

cout<<"Your payout is:"<<endl;

cout<<"$"<<fixed<<setprecision(2)<<showpoint

<<bet<<endl;

}else cout<<endl;

if(win){

exit(0);

}

cout<<"The dealer drew:"<<endl; //Dealer's Turn

cout<<card[1].hand[0]<<endl;

if(card1==card2){ //Choices when you are able to split

cout<<"If you wish to Hit type H"<<endl;

cout<<"If you wish to Stand type S"<<endl;

cout<<"If you wish to Split type T"<<endl;

cout<<"If you wish to Double type D"<<endl;

}if(card1!=card2){ //Choices when you are unable to split

cout<<"If you wish to Hit type H"<<endl;

cout<<"If you wish to Stand type S"<<endl;

cout<<"If you wish to Double type D"<<endl;

}

cin>>choice; //Validating user input

while(choice!='H'&&choice!='S'&&choice!='T'&&choice!='D'){

cout<<"You have entered an invalid choice please try again"<<endl;

cin>>choice;

}if(choice=='T'&&card1!=card2){ //Inability to split when cards don't =

while(choice=='T')

cout<<"You cannot split because your cards are not the same. Please"

<<" enter a valid response"<<endl;

cin>>choice;

}

switch(choice){ //Begin switch based on input

case 'H':do{ //Choosing to hit

card3=card[0].hand[count1+2];

count1++;

cout<<"You have chosen to hit and your card is:"<<endl;

cout<<card3<<endl;

if(card3==1||card3==11){ //Choosing what happens

card3=ace(); //if you get 1 or an 11

}

total+=card3; //Incrementing total value of cards

if(card3>card2){

rng2=card3;

}else if(ptntial>card3&&ptntial>card2){

rng2=ptntial;

}else rng2=card1;

ptntial=card3;

cout<<"Bringing your total to:"<<endl;

cout<<total<<endl;

if(total<=21){ //Choosing to hit again

cout<<"Would you like to hit again? If yes type"

<<" H and if not type any other character"<<endl;

cin>>hitagn;

}else cout<<"You cannot hit again"<<endl;

}while(hitagn=='H'&&total<=21);

cout<<endl;break;

case 'S':cout<<endl;break; //Choosing to stay

case 'D':bet+=bet; //Choosing to double and incrementing bet

card3=card[0].hand[2];

cout<<"You have doubled your initial bet to"<<endl;

cout<<fixed<<setprecision(2)<<showpoint

<<bet<<endl;

cout<<"You will be given another card and cannot hit again"

<<endl;

cout<<"Your card is"<<endl;

cout<<card3<<endl;

if(card3==1||card3==11){ //Choosing what happens if you get

card3=ace(); //1 or an 11

}

cout<<"And your total is"<<endl;

total+=card3; //Incrementing total value of cards

if(card3>card2){ //Saving input to determine max value

rng2=card3; //of single card in a hand

}else rng2=card1;

bet;

cout<<total<<endl;break;

case 'T':split1=card1; //Splitting the hand

split2=card2;

cout<<"You have doubled your initial bet and split it into"

<<"two hands. "<<endl;

bet+=bet; //Incrementing the bet

bet;

card3=card[0].hand[2];

card4=card[0].hand[3];

cout<<"The dealer will now add a card to both hands."

<<endl;

cout<<"To the first hand the dealer added"<<endl;

cout<<card3<<endl; //Card added to the first hand

if(card3==1||card3==11){ //Choosing what happens if you get

card3=ace(); //1 or an 11

}

if(card3>card2){ //Saving input

if(card3>card4){

rng2=card3;

}

}else rng2=card1;

cout<<"To the second hand the dealer added"<<endl;

cout<<card4<<endl; //card added to second hand

if(card4==1||card4==11){ //Choosing what happens

card4=ace(); //if you get 1 or an 11

}

if(card4>card2){

if(card4>card3){ //Saving input

rng2=card4;

}

}else rng2=card1;

total1=card1+card3; //Total value of each hand

total2=card2+card4;

cout<<"The total of the first pile is"<<endl;

cout<<total1<<endl;

cout<<"The total of the second pile is"<<endl;

cout<<total2<<endl;

if(total1>total2){ //Determine which hand is

total=total1; //Favorable

}else total=total2;

cout<<endl;break;

}if(total>21){ //It's a bust if > 21

cout<<"Sorry, your total of "<<total<<" is greater than 21. It's a "

<<"bust, thanks for your cash!"<<endl;

}else{

dealtot=card[1].hand[0]+card[1].hand[1]; //If it's not an immediate bust

cout<<"The dealer will now reveal their second card and choose "

<<"whether or not to hit"<<endl;

cout<<"The dealer's second card is:"<<endl;

cout<<card[1].hand[1]<<endl;

int count=0;

while(dealtot<=total){ //Determine whether dealer hits

dealtot+=card[1].hand[count+2];

cout<<"The dealer chose to hit and the card drawn was:"<<endl;

cout<<card[1].hand[count+2]<<endl;

count++;

}if (count<=1){

count=1;

}

cout<<"The dealer's total is:"<<endl;

cout<<dealtot<<endl;

if(dealtot>21){ // If Dealer Busts

cout<<"Congratulations! "<<deal

<<" the dealer busts meaning you won!"<<endl;

cout<<"Your payout is $"<<fixed<<setprecision(2)<<showpoint

<<bet<<endl;

}

else if(total<=21&&total>dealtot){ //If you win

cout<<"Congratulations! You beat "<<deal<<" the dealer! "

<<endl;

cout<<"Your payout is $"

<<fixed<<setprecision(2)<<showpoint

<<bet<<endl;

}else if(dealtot>total){ //If the dealer busts

cout<<"Sorry, but "<<deal

<<" the dealer beat you and you lost $"<<fixed

<<setprecision(2)<<showpoint<<bet<<endl;

}else if(dealtot==total){ //If you tie

cout<<"You and the dealer tied so you get to keep your $"<<fixed

<<setprecision(2)<<showpoint<<bet<<endl;

}

}

cout<<"If you would like to play again type A and if not type in"

<<" any other character."<<endl;

cin>>playagn; //Would you like to replay?

Destroy(card,Ppl);

}while(playagn=='A'||playagn=='a'); //Choose to replay

//Exit stage right!

return 0;

}

bool AutoWin(int num1, int num2){ //If first hand is 21

bool Val;

if(num1+num2==21){

Val=true;

}else Val=false;

return Val;

}

void Names(char you[] ,char deal[], int people){ //Input Names

cout<<"Please enter your name"<<endl;

cin>>you;

if(islower(you[0])){

you[0]=toupper(you[0]);

}cout<<"Please enter the dealer's name"<<endl;

cin>>deal;

if(islower(deal[0])){

deal[0]=toupper(deal[0]);

}if(!strcmp(you,deal)){

cout<<"How coincidental, you and the dealer have the same name!"

<<" Maybe it's good luck!"<<endl;

}

}

int Dealt(int card1,int card2){ //Display Total

int T;

T=card1+card2;

cout<<"Your total is "<<T<<endl;

return T;

}

int ace(){ //Determine if it is an Ace

int n;

cout<<"You have an ace! Would you like the card to be a 1 or an 11?"<<endl;

cin>>n;

while(n!=1&&n!=11){

cout<<"You have not entered a valid number, please try again"<<endl;

cin>>n;

}

return n;

}

Cards \*DealHnd(int Ppl, int SIZE){

Cards \*card=new Cards[Ppl];

for(int j=0;j<=Ppl-1;j++){

card[j].person=Ppl;

card[j].num=SIZE;

card[j].hand=new int[card[j].num];

int i=0;

for(i=0;i<=card[j].num-1;i++){

card[j].hand[i]=rand()%10+1;//[1-11]

}

}

return card;

}

void Destroy(Cards \*c,int P){

delete[]c[0].hand;

delete[]c[1].hand;

delete[](c+0);

delete[](c+1);

}

bool Read(fstream &file, fstream &copy){

char ch[1500];

copy.open("Temp.bin", ios::out|ios::in | ios::binary);

file.open("Blackjack.txt",ios::in | ios::binary);

int i=0;

if(!file) return false;

else {

while(file){

file.get(ch[i]);

copy.put(ch[i]);

i++;

}

}

file.close();

copy.close();

copy.open("Temp.bin", ios::in| ios::binary);

i=0;

if(!copy)return false;

else{

while(copy){

copy.read(&ch[i],sizeof(ch[i]));

cout<<ch[i];

i++;

}cout<<endl;

cout<<endl;

copy.close();//Close the file

}return true;

}

**Check-off Sheet**

The check-off sheet will be included with the project folder