Project Two

Black Jack 21

Author: John Boutros

Date: 02/09/2019

CIS-5, 40107

Introduction

Title: BlackJack 21

The object of the game is to get a total point value of 21. The user is first asked if they want to play, then they can input their name using a file which was an example from the book. Although I would’ve liked to input a bet, to keep track of the players money, double downs was taking up too much time. The user sees their cards along with one of the dealers cards and then gets asked if they want another card. If the dealer has a card value equal to or less than 16, the dealer hits. The user continues to get asked if they want another card until the user says no and/or the dealer busts, loses or if the user busts, or has a higher cards value compared to the dealer.

Summary:

Project size: 438 lines of code.

Number of variables: ~27

Description:

The main point of this game is to tie all of the chapters together in one project. The main point of this project for myself was to neatly tie all the functions together while minimizing the main function. In and out files were a victory for personal reasons and being able to nicely format the table, hide the dealers card and sort the value of the cards were also a great personal accomplishment.

Pseudo Code:

Initialize

Ask if the user wants to play, enter one or two

If the user doesn’t enter one or two, do while continues to loop

Initialize more variables, set accumulators to zero, and strings to empty quotes.

For statement continues to loop until hitAgn=2 or someones point value exceeds 21.

If statement is executed when hit !=2

User and dealer get dealt cards through random number%10+2 and switch statement.

Players and dealers cards and point values get accumulated.

Displays dealers and players card and point values.

If the player and dealers point values are under 21, they have the option of obtaining another card.

Then there are the results, EX: “The dealer busts! You win!”

Which is based off of point values inside of an if statement

Once someone wins or loses, the user is asked if they would like to play again.

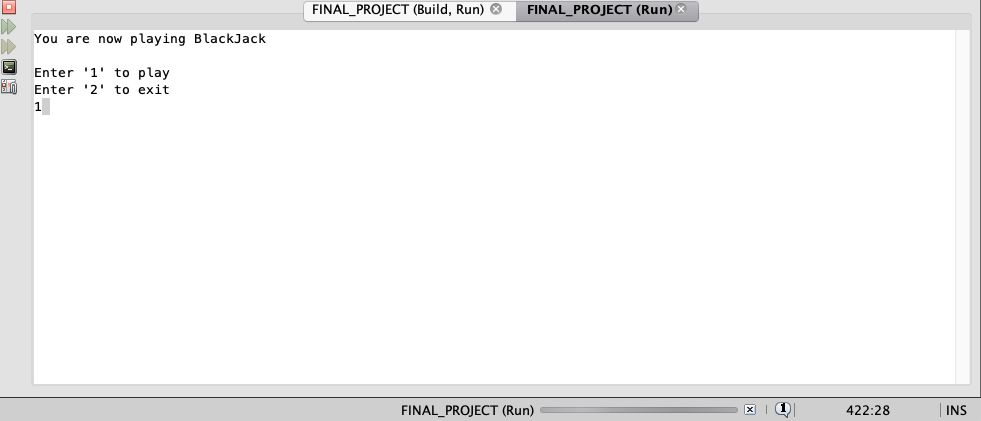
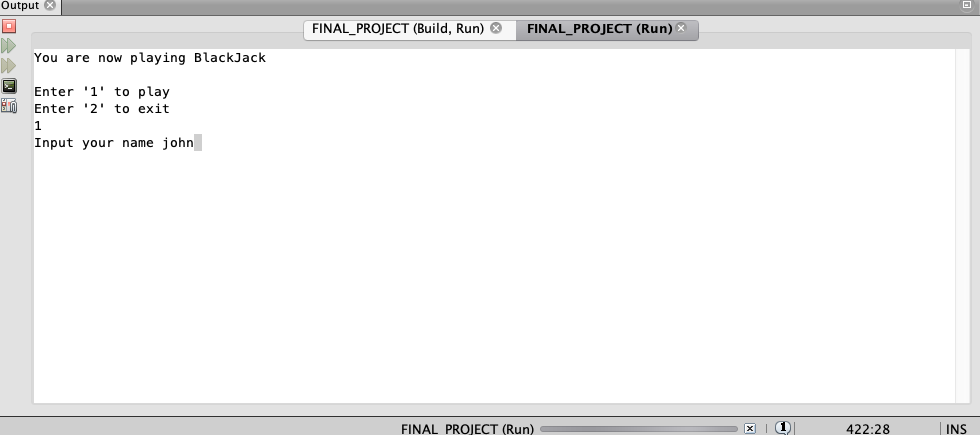
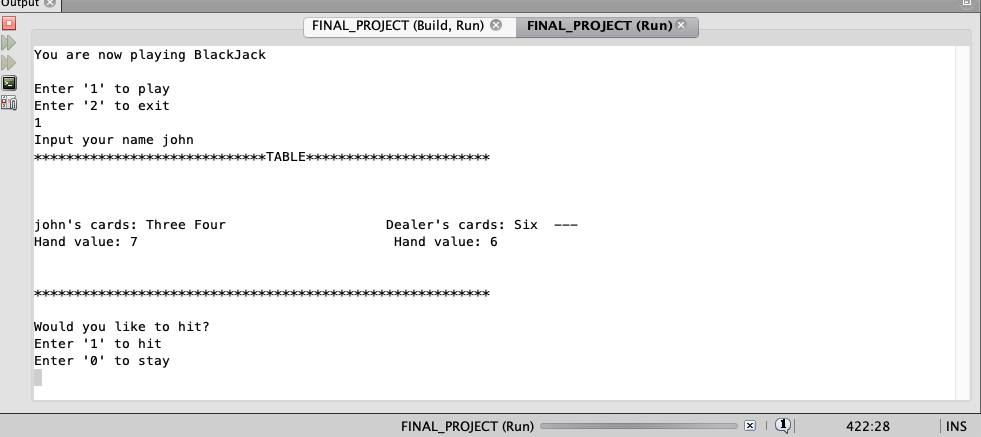
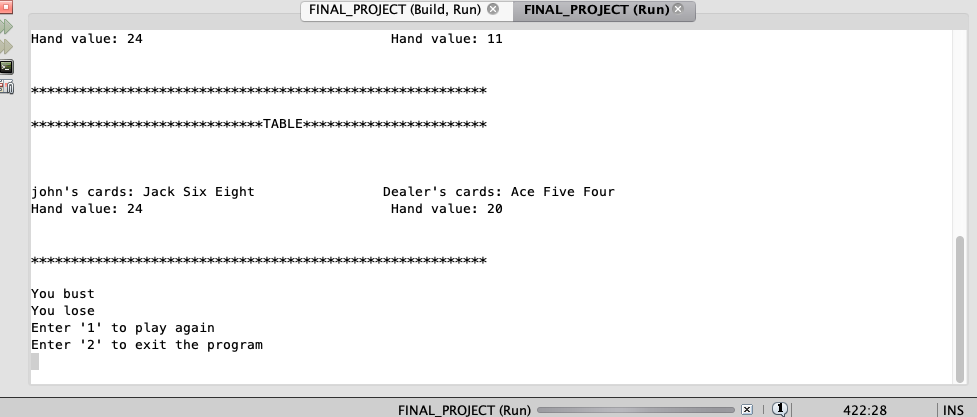
An output message states “Thanks for playing, make sure you collect your earnings from the RCC financial aid department!” and exits the game.

Flow chart: Will be turned in separately

Major Variables:

play, dlrCrd, dlrHnd, plyHand, plyCrd, plyShf, dlrShf, hit, plyAgn.

Program Pictures:



Program: I made the font size to be 3 on the program itself to save space, It starts on the next page.

/\*

\* File: main.cpp

\* Author: John Boutros

\* Created on: 02/07/2019

\* Purpose: Blackjack Project

\*/

//System Libraries Here

#include <iostream> //I/O Library

#include <iomanip> //Format Library

#include <cstdlib> //Contains srand

#include <ctime> //Time Library

#include <string> //String Library

#include <fstream> //File Manipulation Library

#include <cmath> //Math Library

#include <vector> //contains vector

using namespace std;

//User Libraries Here

//Global Constants Only, No Global Variables

//Like PI, e, Gravity, or conversions

const int SUIT=100;

//Function Prototypes Here

void crdSrc(unsigned short [],int,int &);

int max(unsigned short [],int);

void dlrSrc(vector<int> &,int);

void plySrc(unsigned short [],int);

void menu (string &);

int welcome();

void game(unsigned short,int &,string &,string [][SUIT],bool,bool,int,unsigned short [],int &,vector<int> &);

void dealer(string [][SUIT],unsigned short &,unsigned short &,int,unsigned short,unsigned short [],int &);

void dealer(string &,unsigned short &,unsigned short &,int,unsigned short,vector<int> &);

void ace(unsigned short &,unsigned short &,unsigned short &,unsigned short &);

void hideCrd(string &,string,unsigned short &,unsigned short ,int,unsigned short);

void table(string,string,string [][SUIT],unsigned short,unsigned short,int,int,unsigned short []);

void hit(unsigned short, unsigned short,unsigned short &,unsigned short,bool &,bool &,int);

void winLose(unsigned short,unsigned short,unsigned short,unsigned short,bool,bool,int &);

void dlrHndA(bool &,bool &,unsigned short &,unsigned short &,unsigned short,unsigned short,unsigned short &,string,string &);

//Program Execution Begins Here

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

//Seed Random Number

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

//Declare Arrays and Vectors

const int SIZE=100;

vector<int> dValCrd(SIZE);

string faceCrd[SIZE][SUIT];

unsigned short valCard[SIZE];

//Declare all Variables Here

bool dHit,pHit;

string name;

int nLoops,point,number;

unsigned short numGame;

menu(name);

game(numGame,nLoops,name,faceCrd,dHit,pHit,SIZE,valCard,number,dValCrd);

crdSrc(valCard,SIZE,point);

dlrSrc(dValCrd,SIZE);

plySrc(valCard,SIZE);

return 0;

}

//Reveal the dealers hand

void dlrHndA(bool &pHit,bool &dHit,unsigned short &check,unsigned short &numCard,unsigned short plyPnt,unsigned short dlrPnt,unsigned short &pInital,string dHand,string &cInital){

if(pHit==0&&dHit==0&&check==0){

pInital=dlrPnt;

cInital=dHand;

//Open the options for the win condition output

check++;

}else if((plyPnt>=21||dlrPnt>=21)&&check==0){

pInital=dlrPnt;

cInital=dHand;

//Open the branch for the win condition output

check++;

numCard++;

//Lock the card dealing if statement

dHit=0;

pHit=0;

}

}

//Test for win conditions

void winLose(unsigned short dlrPnt,unsigned short plyPnt,unsigned short numCard,unsigned short check,bool pHit,bool dHit,int &nLoops){

//Check the win conditions and output the result if any

if((check==1)&&(dlrPnt>=21||plyPnt>=21||(pHit==0&&dHit==0))){

if(dlrPnt>21&&plyPnt<=21){

cout<<"The house bust"<<endl;

cout<<"You win"<<endl;

}else

if(plyPnt>21&&dlrPnt<=21){

cout<<"You bust"<<endl;

cout<<"You lose"<<endl;

}else

if(plyPnt==dlrPnt){

cout<<"It's a tie, you win"<<endl;

}else

if(dlrPnt==21){

cout<<"Dealer got a BlackJack"<<endl;

if(plyPnt!=21){

cout<<"You lose"<<endl;}

}else

if(plyPnt==21){

cout<<"You got BlackJack"<<endl;

cout<<"You win"<<endl;

}else

if(dlrPnt<21&&dlrPnt>plyPnt){

cout<<"The dealer has a higher score"<<endl;

cout<<"You lose"<<endl;

}else

if(plyPnt<21&&plyPnt>dlrPnt){

cout<<"Your score is higher"<<endl;

cout<<"You win"<<endl;

}

}

}

//Hit the deck

void hit(unsigned short dlrPnt, unsigned short plyPnt,unsigned short &numCard,unsigned short check,bool &dHit,bool &pHit,int nLoops){

if(nLoops>0&&dlrPnt<21&&plyPnt<21&&check==0){

do{

//Ask if the player wants another card

cout<<"Would you like to hit?"<<endl;

cout<<"Enter '1' to hit"<<endl;

cout<<"Enter '0' to stay"<<endl;

cin>>pHit;

if(pHit==1){

if(dlrPnt<=16){

dHit=1;

cout<<"The dealers points is 16 or under, they also hit."<<endl;

numCard++;

}else{

cout<<"Dealer stays"<<endl;

dHit=0;

numCard++;

}

}else if(pHit==0){

if(dlrPnt<=16){

dHit=1;

cout<<"The dealer hits"<<endl;

numCard++;

}else{

dHit=0;

numCard++;

cout<<"Both pass"<<endl;

}

}

//User check, loop the question if user inp doesnt match the cout statement

}while(pHit!=0&&pHit!=1);

}

}

//Blackjack table

void table(string name,string cInital,string faceCrd[][SUIT],unsigned short pInital, unsigned short plyPnt,int nLoops,int size,unsigned short valCard[]){

string face;

unsigned short value;

for(int i=0;i<=nLoops;i++){

for(int j=0;j<=nLoops;j++){

face+=faceCrd[i][j];}

}

if(nLoops>=1){

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TABLE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";

cout<<endl;

cout<<endl;

cout<<name<<"'s cards: "<<left<<setw(30)<<face

<<"Dealer's cards: "<<cInital<<endl;

cout<<"Hand value: "<<left<<setw(33)<<plyPnt;

cout<<"Hand value: "<<pInital<<endl;

cout<<endl;

cout<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";

}

}

//Hide the value of the dealers hand

void hideCrd(string &cInital,string dHand,unsigned short &pInital,unsigned short dlrPnt,int nLoops,unsigned short check){

if(check==0){

if(nLoops==0){

cInital=dHand;

pInital=dlrPnt;

//Following loops hide additional cards and sum value of the dealers hand

}else if(nLoops>0){

cInital+=" --- ";

}

}

}

//Check if there is an Ace

void ace(unsigned short &dlrA, unsigned short &plyrA,unsigned short &dlrPnt,unsigned short &plyPnt){

if(plyrA>0||dlrA>0){

//If the Ace puts the hand value above 21 subtract 10

if(dlrPnt>21&&plyPnt>21){

dlrPnt=dlrPnt-(dlrA\*10);

plyPnt=plyPnt-(plyrA\*10);

//Decrement the ace counter

plyrA--;dlrA--;

}else if(dlrPnt>21){

dlrPnt=dlrPnt-(dlrA\*10);

dlrA--;

}else if(plyPnt>21){

plyPnt=plyPnt-(plyrA\*10);

plyrA--;

}

}

}

void dlrSrc(vector<int> &dValCrd,int size){

//this functions sorts the dealers hand by value

bool swap=true;

while(swap){

swap=false;

for(int i=0;i<dValCrd.size()-1;i++){

if(dValCrd[i]>dValCrd[i+1]){

dValCrd[i]+=dValCrd[i+1];

dValCrd[i+1]=dValCrd[i]-dValCrd[i+1];

dValCrd[i]-=dValCrd[i+1];

swap=true;

}

}

}

}

void plySrc(unsigned short valCard[],int size){

//This function sorts the players hand by value

int strScan, minInd, minVal;

for (strScan=0;strScan<(size-1);strScan++){minInd=strScan;minVal=valCard[strScan];

for(int index=strScan+1;index<size;index++){

if(valCard[index]<minVal){

minVal=valCard[index];minInd=index;

}

}

valCard[minInd]=valCard[strScan];

valCard[strScan]=minVal;

}

}

int max(unsigned short valCard[],int size){

//This function finds the largest value dealt to the player

float max=valCard[0];

for(int j=1;j<size;j++){

if(valCard[j]>max)max=valCard[j];

}

return max;

}

void crdSrc(unsigned short valCard[],int size,int &point){

int value=max(valCard,size);

int index=0;

point=-1;

bool found=false;

while(index<size&&!found){

if(valCard[index]==value){

found=true;

point=index+1;

}

index++;

}

}

//Deal the dealers Card

void dealer(string &dHand,unsigned short &dlrA, unsigned short &dlrPnt, int nLoops,unsigned short numCard,vector<int> &dValCrd){

//Declare Variables

string dlrCrd;

unsigned short dShuff,d10s;

dlrA=0;

//Randomly generate a number for the player between [2-11]

//This will represent the actual point value of the card

dShuff=rand()%10+2;

//Use the switch statement to allocate a string face value

//to the card associated with the number generated

switch(dShuff){

case 2:dlrCrd="Two ";break;

case 3:dlrCrd="Three ";break;

case 4:dlrCrd="Four ";break;

case 5:dlrCrd="Five ";break;

case 6:dlrCrd="Six ";break;

case 7:dlrCrd="Seven ";break;

case 8:dlrCrd="Eight ";break;

case 9:dlrCrd="Nine ";break;

case 10:d10s=rand()%4+1;

if(d10s==1){

dlrCrd="Ten ";break;

}else if(d10s==2){

dlrCrd="Jack ";break;

}else if(d10s==3){

dlrCrd="Queen ";break;

}else{

dlrCrd="King ";break;}

case 11:dlrCrd="Ace ";

dlrA++;break;

}

dlrPnt+=dShuff;

dValCrd[nLoops+1]=dShuff;

dHand+=dlrCrd;

}

//Players cards

void dealer(string faceCrd[][SUIT],unsigned short &plyrA, unsigned short &plyPnt,int nLoops,unsigned short numCard,int size, unsigned short valCard[],int &number){

string plyCrd,pSuit;

unsigned short pShuff,pEnd,p10s;

plyrA=0;

for(int i=nLoops;i<=numCard;i++){

pShuff=rand()%10+2;

pEnd=rand()%4+1;

switch(pShuff){

case 2:plyCrd="Two ";break;

case 3:plyCrd="Three ";break;

case 4:plyCrd="Four ";break;

case 5:plyCrd="Five ";break;

case 6:plyCrd="Six ";break;

case 7:plyCrd="Seven ";break;

case 8:plyCrd="Eight ";break;

case 9:plyCrd="Nine ";break;

//Randomly generate a number [1-4] and to allocate a

//string face value to the 10 point card

case 10:p10s=rand()%4+1;

if(p10s==1){

plyCrd="Ten ";break;

}else if(p10s==2){

plyCrd="Jack ";break;

}else if(p10s==3){

plyCrd="Queen ";break;

}else{plyCrd="King ";break;}

case 11:plyCrd="Ace ";

//Increment the ace counter

plyrA++;break;

}

}

static int cnt=0;

cnt++;

number=cnt;

plyPnt+=pShuff;

valCard[nLoops]=pShuff;

faceCrd[nLoops][nLoops]=plyCrd;

}

void game(unsigned short numCard, int &nLoops,string &name,string faceCrd[][SUIT],bool dHit,bool pHit,int size,unsigned short valCard[],int &number,vector<int> &dValCrd){

char play2;

string dHand,cInital;

unsigned short dlrPnt,plyPnt,plyrA,dlrA,pInital,check;

do{

play2=0;

check=0;

numCard=1;

pHit=1;

dHit=1;

dlrPnt=0;

plyPnt=0;

dHand="";

//loop until someone wins

for(nLoops=0;nLoops<=numCard;nLoops++){

if(pHit!=0){

dealer(faceCrd,plyrA,plyPnt,nLoops,numCard,size,valCard,number);

ace(dlrA,plyrA,dlrPnt,plyPnt);

}

if(dHit!=0){

dealer(dHand,dlrA,dlrPnt,nLoops,numCard,dValCrd);

ace(dlrA,plyrA,dlrPnt,plyPnt);

hideCrd(cInital,dHand,pInital,dlrPnt,nLoops,check);

}

table(name,cInital,faceCrd,pInital,plyPnt,nLoops,size,valCard);

hit(dlrPnt,plyPnt,numCard,check,dHit,pHit,nLoops);

winLose(dlrPnt,plyPnt,numCard,check,pHit,dHit,nLoops);

dlrHndA(pHit,dHit,check,numCard,plyPnt,dlrPnt,pInital,dHand,cInital);

}

if(play2==2){

cout<<"Thanks for playing"<<endl;

exit(0);}

//Ask if player wants to play again

cout<<"Enter '1' to play again"<<endl;

cout<<"Enter '2' to exit the program"<<endl;

cin>>play2;

if(play2==2){

exit(0);

}

}while(play2!=0);

}

int welcome(){

int play1;

//Intro menu for game

cout<<"You are now playing BlackJack\n\n";

cout<<"Enter '1' to play"<<endl;

cout<<"Enter '2' to exit"<<endl;

cin>>play1;

return play1;

}

//Output the welcome menu

void menu(string &name){

int play;

while(play!=1){

play=welcome();

//If user chooses to play a game prompt for their initials

if(play==1){

cout<<"Input your name ";

cin>>name;

//Upload these initials to the name file

ofstream input;

input.open ("name.txt");

input<<"Name: "<<name;

input.close();

}

//User chooses to not play

if(play==2){

cout<<"Weenie";

exit(0);

}

}

}