PROJECT 1

<BlackJack-Game>

Tejash Bhakta

CSC-5 48102

October 28, 2016

Introduction:

**Blackjack**, also known as **twenty-one**, is the most widely played casino banking game in the world.[[1]](https://en.wikipedia.org/wiki/Blackjack#cite_note-1) Blackjack is a comparing card game between a player and dealer, meaning players compete against the dealer but not against other players. It is played with one or more decks of 52 cards. The objective of the game is to beat the dealer in one of the following ways:

* Get 21 points on the player's first two cards (called a "blackjack" or "[natural](https://en.wikipedia.org/wiki/Natural_(gambling))"), without a dealer blackjack;
* Reach a final score higher than the dealer without exceeding 21; or
* Let the dealer draw additional cards until his or her hand exceeds 21.

**OBJECT OF THE GAME**

Each participant attempts to beat the dealer by getting a count as close to 21 as possible, without going over 21.

**CARD VALUES/SCORING**

It is up to each individual player if an ace is worth 1 or 11. Face cards are 10 and any other card is its pip value.

#### SETTLEMENT

A bet once paid and collected is never returned. Thus, one key advantage to the dealer is that the player goes first. If the player goes bust, he has already lost his wager, even if the dealer goes bust as well. If the dealer goes over 21, he pays each player who has stood the amount of that player's bet. If the dealer stands at 21 or less, he pays the bet of any player having a higher total (not exceeding 21) and collects the bet of any player having a lower total. If there is a stand-off (a player having the same total as the dealer), no chips are paid out or collected.

NOTE: Unlike original BlackJack, my game does not provide features like insurance if the dealer hits ace on first card, and splitting of identical cards.

**Summary:**

Project Size: approx. 400 lines (including the functions)

Number of variables: around 25 variables

Functions: 7

This project includes almost all concepts that were taught in class as well as from the textbooks.

This was a challenging project for me. The dealing with card value and the total when Ace is present gave me a hard time figuring out. I tested many possibilities to debug my code, but still I think the code is not completely debugged.

**Pseudo Code:**

On startup, menu pops up 🡪1. Start BlackJack

🡪2. Player Logs

* On choosing 1, game starts:

The first house card will be generated, then the two player’s cards and total will be displayed.

* While player hits, one more card is generated until player stands or is busted
* Else if player stands, the house cards are generated and displayed until house has a total of 17 or more—the player total and house total are compared and determines the winner
* Else if player doubles down then only one player card is generated and then the house plays rest of the deal

After this, the amount total is updated depending on the winning or losing of the player

Ask to deal again or exit

* If deal again, then repeat
* Else exit and update the file in the player logs
* On Choosing 2, the file leader board opens and displays the players name and their score against their name.

**Major Variables**

**Variable used throughout main**

|  |  |  |
| --- | --- | --- |
| **Type** | **Variable Name** | **Description** |
| Float | totAmt=100 | Holds the player’s total(initial value is 100) |
| betPlcd | Holds the bet placed by the player |
| Unsigned short | cmd | Holds the menu number |
| Char | deal | Hold the character to deal or exit the game |
| check | Hold the character when player wins, loses or busted |
| Fstream | in | Read and write file |

**Variables used and declared in the game when user input 1 from menu**

|  |  |  |
| --- | --- | --- |
| **Type** | **Variable Name** | **Description** |
| Char | hitstd | Holds the char when hit or stand or double down |
| Integer | num | Holds the player card value |
| com | Holds the house card value |
| pnum | Holds the player random number generated |
| cnum | Holds the house random number generated |
| pTotal | Holds the player’s card Total |
| cTotal | Holds the house’s card Total |
| String | cardface | Holds the card face for player |
| hosFace | Holds the house card face |
| Bool | isAce | True of false if ace present in the deal |

**Variable declared and when user input 2 from menu**

|  |  |  |
| --- | --- | --- |
| **Type** | **Variable Name** | **Description** |
| String | line | Stores the line obtained from the file ledrbrd.dat |

**FlowChart:**

<https://www.lucidchart.com/documents/view/ac82c334-d435-4ecb-9a75-48ce897e30e9>

or \*file attached in the folder

**Reference:**

1. **Textbooks (Savitch and Gaddis)**
2. **cplusplus.com**
3. **Github.com: marklehr/48101\_48102 repository**

**Main Program:**

//System Libraries

#include <iostream>

#include <iomanip>

#include <fstream>

#include <cTime>

#include <cstdlib>

using namespace std;

//Function Prototypes

int cardVal(int num);

string cardName(int num);

void display();

void seprtr();

int stand(int num1,string cardface);

char status(int total);

char status(int pTot, int cTot);

//Main Program starts here

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

//Random number generator

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

//Declaration of variable

float totAmt=100, betPlcd;

unsigned short cmd;

char deal, check;

string name;

fstream in;

//Display Introduction

display();

//Start the game

//Opening Commands

do{

seprtr();

cout<<"Choose from the menu"<<endl;

cout<<"1. Start BlackJack <$100 included>"<<endl;

cout<<"2. Player Logs"<<endl;

cin>>cmd;

switch(cmd){

case 1:{

deal='d';

cout<<"Enter Player Name"<<endl;

cin>>name;

while(deal=='d'){

do{

seprtr();

cout<<fixed<<setprecision(2);

cout<<"YOUR TOTAL right now is $"<<totAmt<<endl;

cout<<"Bet to be placed (min-10 && max-50)"<<endl;

cin>>betPlcd;

}while(betPlcd<10 || betPlcd>50 || betPlcd>totAmt);

//Declaration of variables for game insight

char hitstd;

int num=0, com=0,

pnum=0, cnum=0,

aceTot=0,

pTotal=0, cTotal=0;

string cardface, hosFace;

bool isAce=false;

//Display the House First Card

seprtr();

cout<<"House's Card"<<endl;

cnum=rand()%14+1;

com=cardVal(cnum);

hosFace=cardName(cnum);

cout<<hosFace<<"???"<<endl;

/\*if(com==1 || com==11)cout<<"Total would be "<<com-10<<" or"<<endl;

cout<<"Total would be "<<com<<endl;\*/

//Fetching the first two cards of the player

seprtr();

cout<<name<<"'s Cards"<<endl;

for(int open=1; open<=2; open++){

pnum=rand()%14+1;

num=cardVal(pnum);

if(num==1 || num==11){isAce=true;aceTot++;}

cardface+=cardName(pnum);

pTotal+=num;

if(aceTot==2)pTotal-=10;

}

//Display Player's First two Cards

cout<<cardface<<endl;

if(isAce==true && pTotal!=21)cout<<"Total would be "<<pTotal-10<<" or"<<endl;

cout<<"Total would be "<<pTotal<<endl;

seprtr();

//Condition Check if player hits BlackJack and ready for another deal

if(pTotal==21){

cout<<"\*\*\*BlackJack\*\*\*";

totAmt+=betPlcd\*1.5; //bet \*1.5;

}

//Player option to hit, stand or double down if not blackJack

else{

do{

cout<<"Hit 'h' or Stand 's' or Double down 'd'"<<endl;

cin>>hitstd;

}while(hitstd!='h' && hitstd!='s' && hitstd!='d');

}

//Generating next cards if the player hits

while(hitstd=='h'){

seprtr();

cout<<name<<"'s Card"<<endl;

pnum=rand()%14+1;

num=cardVal(pnum);

if(num==1 || num==11){isAce=true;aceTot++;}

cardface+=cardName(pnum);

pTotal+=num;

cout<<cardface;

//if any card is an ace, total reduces to -10 if total greater than 21

if(isAce==true && aceTot!=0){

if(pTotal>21){pTotal-=10;aceTot--;isAce=false;}

else cout<<endl<<"Total would be "<<pTotal-10<<" or";

}

cout<<endl<<"Total would be "<<pTotal<<endl;

seprtr();

//Check the players total and returns won or lost

check=status(pTotal);

if(pTotal==21){cTotal=stand(com,hosFace);status(pTotal,cTotal);break;}

else if(check!=0 && pTotal!=21)break;

//Next deal

cout<<"Hit or Stand"<<endl;

cin>>hitstd;

seprtr();

}

//If player stands the deal

if(hitstd=='s'){

seprtr();

//calls the function stand with card 1 as an input and returns the cards as well as the total

cTotal=stand(com,hosFace);

//checks the house total and returns who won

check=status(pTotal,cTotal);

}

//If the player double downs

else if(hitstd=='d'){

betPlcd\*=2;

pnum=rand()%14+1;

num=cardVal(pnum);

cardface+=cardName(pnum);

pTotal+=num;

cout<<cardface;

if(isAce==true && aceTot!=0){

if(pTotal>22){pTotal-=10;aceTot--;}

}

cout<<endl<<"Total would be "<<pTotal<<endl;

seprtr();

cTotal=stand(com,hosFace);

check=status(pTotal,cTotal);

}

//Check the winner and updates the total of the player's stack

switch(check){

case 'w':{ //Winner

cout<<"You win"<<endl;

totAmt+=betPlcd;

break;

}

case 'b':{

cout<<"You Busted"<<endl;

totAmt-=betPlcd;

break;

}

case 'l':{

cout<<"You Lose"<<endl;

totAmt-=betPlcd;

break;

}

case 'p':

cout<<"PUSH"<<endl;cout<<"Next Deal"<<endl;

}

seprtr();

cout<<fixed<<setprecision(2);

cout<<"Your Total now $"<<totAmt<<endl;

if(totAmt<=0){cout<<"Not enough Money"<<endl;break;}

do{cout<<"Next Deal 'd' or exit 'x' "<<endl;

cin>>deal;

seprtr();

}while(deal!='d' && deal!='x');

}

in.open("ldrbrd.dat",ios::app);

in<<"\n"<<name<<" "<<totAmt;

in.close();

break;

}

case 2:{

string line;

//Input values

in.open("ldrbrd.dat");

if(in.is\_open()){

while(getline(in,line)){

cout<<line<<"\n";

}

in.close();

}

break;

}

}

}while(cmd>0 && cmd<=2);

//Exit Program

return 0;

}

**Functions:**

* **void display()**

void display(){

cout<<"Welcome to the game of BlackJack"<<endl;

cout<<"• Each participant attempts to beat the dealer by getting a"<<endl;

cout<<" count as close to 21 as possible, without going over 21."<<endl;

cout<<"Let's get Started"<<endl;

}

* **void seprtr()**

void seprtr(){

cout<<"-------------------------------"<<endl;

}

* **char status(int total)**

char status(int total){

if(total>21)return'b';

else if(total==21)return 'w';

else return 0;

}

* **char status(int pTot, int cTot)**

char status(int pTot, int cTot){

char stat;

if(pTot>21)stat='b';

else if(cTot>pTot){

if(cTot>21){

cout<<"House Busted"<<endl;

stat='w';

}

else if(cTot==21){

cout<<"Its 21..House Wins"<<endl;

stat='l';

}

else {

cout<<"House Total Greater"<<endl;

stat='l';

}

}

else if(cTot==pTot)

stat='p';

else if(pTot>cTot){

if(pTot>21)stat='b';

else if (pTot<=21)stat='w';

else stat='l';

}

return stat;

}

* **int stand(int num1, string cardFace)**

int stand(int num1, string cardFace){

int com=0,cnum=0;

int cTotal=num1, aceTot=0, n=0;

char check;

bool isAce=false;

string cardface=cardFace;

if(com==11)isAce=true;

cout<<"House Cards"<<endl;

do{

cnum=rand()%14+1;

com=cardVal(cnum);

if(com==1 || com==11){isAce=true;aceTot++;}

cardface=cardface+cardName(com);

cTotal=cTotal+com;

if(isAce==true && aceTot!=0){

if(cTotal>=22){cTotal-=10;aceTot--;isAce=false;}

n++;

}

}while(cTotal<17);

//House on soft 17 draw one more card

if(cTotal==17 && n!=0){

cnum=rand()%14+1;

com=cardVal(cnum);

cardface=cardface+cardName(com);

cTotal=cTotal+com;

}

cout<<cardface;

if(n!=0 && cTotal<17) cout<<endl<<"Total would be "<<cTotal-10<<" or";

cout<<endl<<"Total would be "<<cTotal<<endl;

seprtr();

return cTotal;

}

* **string cardName(int card)**

string cardName(int card){

string cardface;

//Outputs the Name of the card

switch(card){

case 1:

case 11:cardface="An Ace|";break;

case 2: cardface="Two|"; break;

case 3: cardface="Three|";break;

case 4: cardface="Four|";break;

case 5: cardface="Five|";break;

case 6: cardface="Six|";break;

case 7: cardface="Seven|";break;

case 8: cardface="Eight|";break;

case 9: cardface="Nine|";break;

case 10:cardface="Ten|";break;

case 12:cardface="Jack|";break;

case 13:cardface="Queen|";break;

case 14:cardface="King|";break;

return cardface;

}

}

* **int cardVal(int card)**

int cardVal(int card){

int value=0;

//Returns the value on the card

switch(card){

case 1:

case 11:value=11;break;

case 2:value=2;break;

case 3:value=3;break;

case 4:value=4;break;

case 5:value=5;break;

case 6:value=6;break;

case 7:value=7;break;

case 8:value=8;break;

case 9:value=9;break;

case 10:

case 12:

case 13:

case 14:value=10;break;

}

return value;