Black Jack

Version 7

Project 2

July 28, 2016

Mario Cabrera

CSC-5 45277

Table of Contents

Introduction 3

Rules 4

Card Values 4

Flow Chart 5

Code Statistics 6

Code 6

**Introduction**

Blackjack, also known as twenty-one, is the most widely played casino banking game in the world. 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.

Blackjack's precursor was twenty-one, a game of unknown origin. The first written reference is found in a book by the Spanish author Miguel de Cervantes, most famous for writing Don Quixote. Cervantes was a gambler, and the main characters of his tale "Rinconete y Cortadillo", from Novelas Ejemplares, are a couple of cheats working in Seville. They are proficient at cheating at ventiuna (Spanish for twenty-one), and state that the object of the game is to reach 21 points without going over and that the ace values 1 or 11. The game is played with the Spanish baraja deck, which lacks eights, nines and tens. This short story was written between 1601 and 1602, implying that ventiuna was played in Castilla since the beginning of the 17th century or earlier. Later references to this game are found in France and Spain.

When twenty-one was introduced in the United States, gambling houses offered bonus payouts to stimulate players' interest. One such bonus was a ten-to-one payout if the player's hand consisted of the ace of spades and a black jack (either the jack of clubs or the jack of spades). This hand was called a "blackjack", and the name stuck to the game even though the ten-to-one bonus was soon withdrawn. In the modern game, a blackjack refers to any hand of an ace plus a ten or face card regardless of suits or colours.

**Rules**

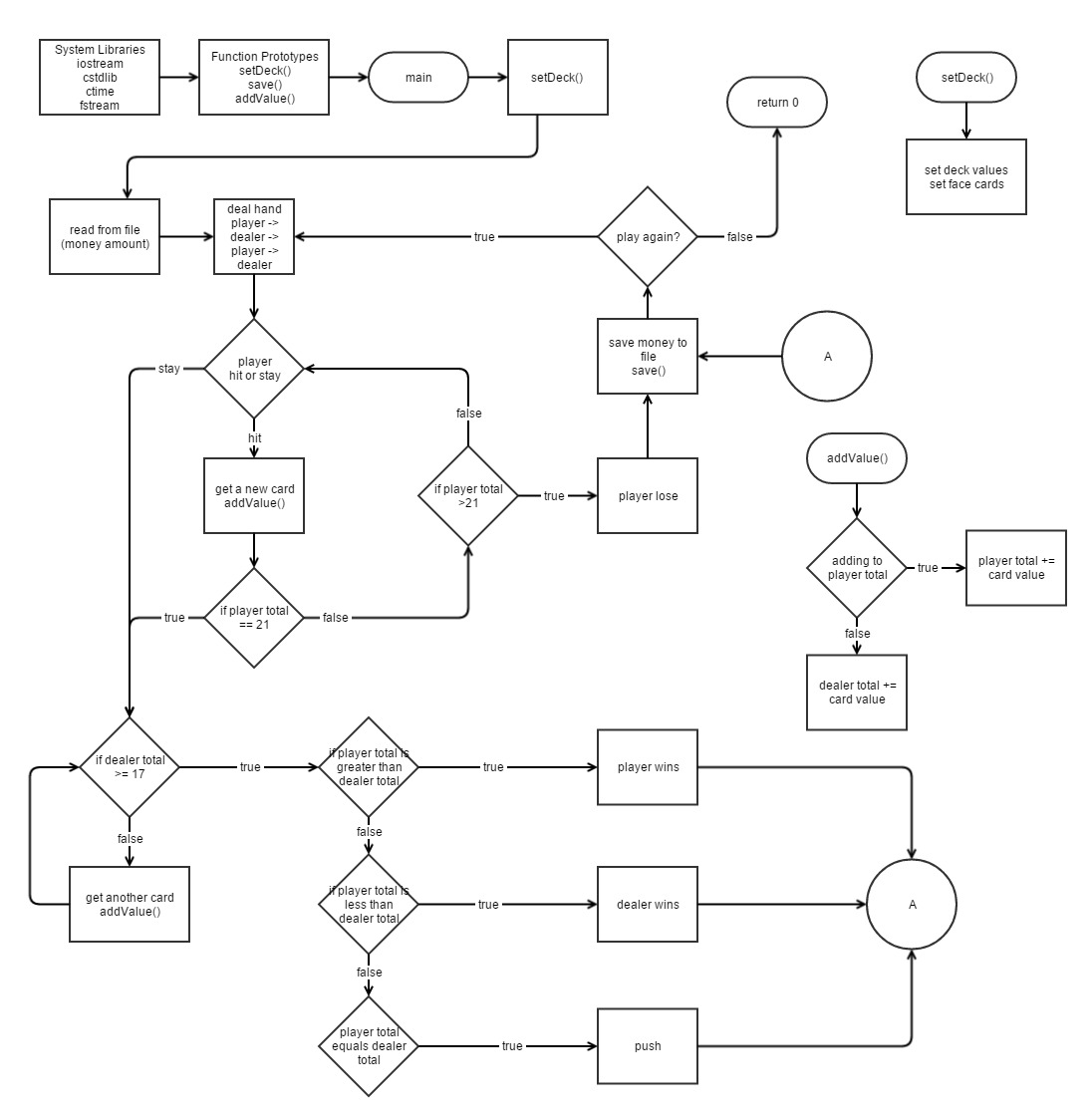
Upon running the program the player will be issued two playing cards and the dealer will have one visible card out of his two cards. Depending on the players total, the player must decide to accept more cards and get as close to 21 without going over (busting) or staying. Once the player feels comfortable with his/her choice and stay, the dealer then take cards. The dealer has to get as close to 21 as well without busting, but must keep taking cards until they have a 17 or more. Whoever is closest to 21 without going over, wins. If both the player and dealer have the same number, a tie, called a push, is issued and the game is reset.

**Card Values**

Card values are as follows:

|  |  |
| --- | --- |
| Face | Value |
| A | 1[[1]](#footnote-1) |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |
| 10 | 10 |
| J | 10 |
| Q | 10 |
| K | 10 |

**Flow Chart**

****

**Code Statistics**

|  |  |  |
| --- | --- | --- |
| **Element** | **#** | **Names** |
| Lines of code | 309 |  |
| Unique variables | 15 | SIZE, money, min, bet, pTotal, dTotal, choice, card, play, lose, valid, in, out |
| Arrays | 2 | cValues[] & deck[] |
| Libraries | 5 | iostream, cstdlib, ctime, iomanip, fstream |
| Functions | 5 | time( ), rand( ), srand( ), save( ), addValue( ) |
| Loops | 5 | do-while loop – line 81-273  do-while loop – line 90-119  while loop – line 151-195  while loop – 210-222  do-while loop – 261-272 |
| Switch Statements | 1 | lines – 157-194 |

**Code**

/\*

\* File: main.cpp

\* Author: Mario Cabrera

\* Created on July 26, 2016, 9:04 PM

\* Purpose: Final

\*/

//system libraries

#include <iostream> //Input / Output library

#include <cstdlib> //C standard library

#include <ctime> //C Time

#include <iomanip> //input / output manipulation

#include <fstream> //file stream library

using namespace std;

//user libraries

//global constants

//function prototypes

void save(int); //save money to file

void addValue(int &, int [],int);//add value of card to total

//execution begins here

int main(){

//random number seed

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

//declare and initialize variables

const int SIZE = 52; //const size of array (52 cards)

int cValues[SIZE] = {1,2,3,4,5,6,7,8,9,10,10,10,10,

1,2,3,4,5,6,7,8,9,10,10,10,10,

1,2,3,4,5,6,7,8,9,10,10,10,10,

1,2,3,4,5,6,7,8,9,10,10,10,10};

string deck[SIZE] = {"Ac","2c","3c","4c","5c","6c","7c","8c","9c","10c","Jc","Qc","Kc",

"As","2s","3s","4s","5s","6s","7s","8s","9s","10s","Js","Qs","Ks",

"Ad","2d","3d","4d","5d","6d","7d","8d","9d","10d","Jd","Qd","Kd",

"Ah","2h","3h","4h","5h","6h","7h","8h","9h","10h","Jh","Qh","Kh"};

int money; //total money player has

int min = 5; //minimum bet

int bet; //player's bet

int pTotal = 0;//total value of cards player has

int dTotal = 0;//total value of cards dealer has

char choice; //user input

int card; //card index

bool play; //play boolean

bool lose; //lose boolean

bool valid; //validation boolean

//retrieve money from the file;

ifstream in; //ifstream variable

in.open("money.txt");

in>>money;

in.close();

cout<<endl;

cout<<" Welcome to Blackjack!"<<endl;

cout<<endl;

cout<<"Starting with two card and adding accordingly,"<<endl;

cout<<"you must get closer to 21 than the dealer. But"<<endl;

cout<<"be warned, if you go over 21, you lose"<<endl;

cout<<"automatically and lose your bet. Don't worry,"<<endl;

cout<<"if you run out of funds, we will gift you more."<<endl;

cout<<endl;

cout<<" Suits"<<endl;

cout<<" h = hearts"<<endl;

cout<<" d = diamonds"<<endl;

cout<<" s = spades"<<endl;

cout<<" c = clubs"<<endl;

cout<<endl;

cout<<" Face Cards"<<endl;

cout<<" J = Jack"<<endl;

cout<<" Q = Queen"<<endl;

cout<<" K = King"<<endl;

cout<<" A = Ace"<<endl;

cout<<endl;

cout<<" Enjoy the game and have fun!"<<endl;

cout<<endl;

do{

cout<<endl;

//if player money is less than minimum bet, recharge $100

if(money==0 || money<min){

cout<<"You have been recharged $100 because you have funds below the minimum bet"<<endl;

money = 100;

}

do{

//set valid = true

valid=true;

cout<<"You have $"<<money<<". How much do you want to bet?"<<endl;

cout<<"Minimum bet is $"<<min<<"."<<endl;

//bet input

cin>>bet;

cout<<endl;

//check bet against minimum bet

if(bet<min){

cout<<"You entered a bet that was lower than the minimum bet"<<endl;

cout<<endl;

//if bet is less than minimum bet, set valid = false

valid=false;

}

//check bet against player money

if(bet>money){

cout<<"You entered a bet that was higher than the money you have"<<endl;

cout<<endl;

//if bet is more than player money, set valid = false

valid=false;

}

}while(valid==false);

play=true;

lose=false;

pTotal=0;

dTotal=0;

//get first cards

//player card

card = rand()%52;

cout<<"Player's first card: "<<deck[card]<<endl;

cout<<endl;

addValue(pTotal,cValues,card);

//dealer card

card = rand()%52;

cout<<"Dealer's first card: "<<deck[card]<<endl;

cout<<endl;

addValue(dTotal,cValues,card);

//player card

card = rand()%52;

cout<<"Player's next card: "<<deck[card]<<endl;

cout<<endl;

addValue(pTotal,cValues,card);

//output totals

cout<<fixed;

cout<<"Player's Total = "<<setw(2)<<pTotal<<endl;

cout<<"Dealer's Total = "<<setw(2)<<dTotal<<endl;

cout<<endl;

while(play==true){

//user input of h or s

cout<<"hit or stay? [h]/[s]"<<endl;

cin>>choice;

switch(choice){

case 'h':{

//player card

card = rand()%52;

cout<<"Player's next card: "<<deck[card]<<endl;

cout<<endl;

addValue(pTotal,cValues,card);

//output totals

cout<<fixed;

cout<<"Player's Total = "<<setw(2)<<pTotal<<endl;

cout<<"Dealer's Total = "<<setw(2)<<dTotal<<endl;

cout<<endl;

//if player's total is greater than or equal to 21, stop

if(pTotal==21 || pTotal>21){

play=false;

if(pTotal>21){

cout<<"Bust. You lose."<<endl;

//bet subtracted from total for loss

money-=bet;

lose=true;

}

}

break;

}

case 's':{

play=false;

break;

}

default:{

cout<<"You did not enter a valid choice"<<endl;

cout<<endl;

}

}

}

if(lose==false){

//dealer card

card = rand()%52;

cout<<"Dealer's next card: "<<deck[card]<<endl;

cout<<endl;

addValue(dTotal,cValues,card);

//output totals

cout<<fixed;

cout<<"Player's Total = "<<setw(2)<<pTotal<<endl;

cout<<"Dealer's Total = "<<setw(2)<<dTotal<<endl;

cout<<endl;

while(dTotal<17){

//dealer card

card = rand()%52;

cout<<"Dealer's next card: "<<deck[card]<<endl;

cout<<endl;

addValue(dTotal,cValues,card);

//output totals

cout<<fixed;

cout<<"Player's Total = "<<setw(2)<<pTotal<<endl;

cout<<"Dealer's Total = "<<setw(2)<<dTotal<<endl;

cout<<endl;

}

//if dealer total > 21, dealer bust

if(dTotal>21){

cout<<"Dealer Bust. You win!"<<endl;

cout<<endl;

//money added for win

money+=bet;

//if player total > dealer total, player win

}else if(pTotal>dTotal){

cout<<"You beat the dealer. You win!"<<endl;

cout<<endl;

//money added for win

money+=bet;

//if player total < dealer total, player lose

}else if(pTotal<dTotal){

cout<<"The dealer beat you. You lose"<<endl;

cout<<endl;

//bet subtracted from total for loss

money-=bet;

//if player total == dealer total, draw.

}else if(pTotal==dTotal){

cout<<"Push. Draw."<<endl;

cout<<endl;

}

}

//saving money to a file

cout<<"Saving..."<<endl;

cout<<endl;

save(money);

//play again choice with validation

do{

cout<<"Do you want to play again? [y]/[n]"<<endl;

cin>>choice;

cout<<endl;

//if choice is invalid, error message

if(choice!='y' && choice!='n'){

cout<<"You did not enter a valid choice"<<endl;

cout<<endl;

}

}while(choice!='y' && choice!='n');

}while(choice=='y');

cout<<"Thanks for playing. Until next time"<<endl;

//exit stage right!

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\*

\* addValue(int &, int [], int)

\*

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void addValue(int &total, int value[],int card){

total+=value[card];

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\*

\* save(int)

\*

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void save(int money){

ofstream out; //ofstream variable

out.open("money.txt"); //opening file

out<<money; //outputing money to file

out.close(); //closing file

}

1. Normally, an Ace carries the value of 1 or 11. But in this version, the Ace will remain a value of 1 [↑](#footnote-ref-1)