**BLACKJACK**

**(PROJECT 2)**

**CSC-5 (40652)**

**Rodrigo Leyva**

**Date: 2/8/2018**

**Introduction**

No one knows the real origin of this world-famous card game. The earliest reference found of this game was in a book written by Miguel de Cervantes, who is famous for writing *Don Quixote.* It was a short story that was written in 1602 implying that the game might’ve been played before then. The game was nicknamed “21” when it gained popularity in Nevada in 1931 when they chose to become the first state to make gambling legal. Fun fact: the first bonus payout for getting blackjack “21” on the opening hand paid 10-to-1 odds. The highest bonus you’ll see today is x1.5.

**Rules of the game.**

In this version of blackjack, we will be playing with a 52-card deck and there will be no betting of currency. In a real-world setting, most Las Vegas tables use 4 or 6 deck sets. (Don’t ever play a single deck blackjack game, the odds are horrible).

The object of the game is to beat the dealer by getting as close to 21 as possible without going over 21. It might sound very simple, but this game has so much math and statistics involved if you want to excel at it.

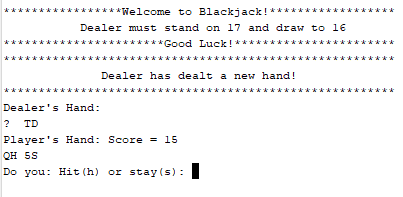
The dealer gives two cards to the player, and two cards to themselves. The player’s both cards are revealed but the dealer leaves one card face down. After receive the cards, the player decides if he wants to hit or to stay. To hit means you will get another card. To stay means you want no more cards and the dealer reveals their card.

After the dealer has revealed their card, if their total doesn’t equal at least 17, they must deal themselves cards until they reach at least 17. Once the dealer reaches 17, they must stand. At this point, if the dealer hasn’t gone over 21, whoever has the highest score wins. If at any point during the dealing of cards, if the player goes over 21, the dealer wins. If the dealer goes over 21 while they are dealing themselves cards, the player wins. If there is a tie, there is a push and no one wins and then a new hand is dealt.

**Our version of the game**

Unlike a real table game of Blackjack, this game doesn’t have betting of currency involved. Therefore, this means it also doesn’t pay out for blackjack bonus multipliers. Since there is no betting, we have also disabled the ability to double and to split cards. (side note: splitting isn’t just good for betting, it’s also a great way to split double 8s, since getting 16 in blackjack is horrible.

The game includes suits just to add realism to the game, but suits do not change the value nor change the value of a card. Blackjack experts can track how many of each card has been dealt by the use of suits, but since this is a single deck of cards it doesn’t really matter. The suit is labeled as a letter after the value of card. Example below:



H = Hearts

D = Diamonds

S = Spades

C = Clover

If you see a letter instead of a number, it means it’s a face card. A face card is worth ten. If you see an A, it’s an ace, which is worth 1 or 11. An ace is the best card in the deck since it can be worth two different values. As long as you don’t bust(go over 21, it can be worth 1 or 11, whichever benefits you more.

A = Ace

T = Ten

J = Jack

Q = Queen

K = King

**Pseudo code(also attached separately in notepad file):**

// BLACKJACK GAME

//random value library

//input and output stream

// unique seed library

// formatting library

//string library

//prototype functions

//displays card

//shows hand

//random card is dealt from the deck

//adds up hands

//adds up totals

// random generators

//our 52 card deck

//the total score of dealer's cards

//the hand dealt to dealer

//player's total hand

//player's dealt hand

// loop for the start of a hand

// shuffle deck to even the chance for each card

// deal cards to dealer and player, each receives two cards.

//first player card

//first dealer card (hidden)

//second player card

//second dealer card

// loop for the start of a hand

while (true) {

// shuffle deck to even the chance for each card

// deal cards to dealer and player, each receives two cards.

//first player card

//first dealer card (hidden)

//second player card

//second dealer card

// welcome sign with table rules and signals a new hand being dealt

//gives basic rule of dealer rule

//shows message to signal new hand

// calculate total score for each hand, including after each hit

// shoe dealer's and player's hand. only show one of the dealer's card per blackjac rules

//showing that the dealer has one face down card

//shows the total score of the player's cards

//show player cards

// ask player if they want to hit or stay with their dealt cards.

//player chooses between hit and stay

//player decided to stand

//inpout validation. makes sure that the player chose a valid input, otherwise loops again

// add player's card to update total and checks to make sure player hasn't gone over 21.

// check if player has busted after hits, if player has the dealer wins.

// player went over 21, dealer won.

// dealer has to reach 17 per blackjack rules. checks if dealer has 17, if not, dealer hits.

//activates if dealer goes over 21

// if no one busts, compare scores to see who has the higher score

// player and dealer have same score, ends in a "push/tie"

// "The Player wins" output message

// "The dealer wins" output message

//calculates score for face cards. "jack, queen, king, ace"

// calculations for each face card

// ace value

// calculates the suit of the card. Doesn't chance the value of the card in blackjack, but adds to realism of the game :)

//clover suit

//diamond suit

//hearts suit

// spade suit

//displays hand totals

//function for dealing out random cards

//randomizes when dealing out a card from a 52 card deck

//determine soft ace value, per blackjack rules.

//function that adds adds up the dealer and player's cards after a hit and displays

**Program:**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*

\* File: main.cpp

\* Author: Rodrigo

\*

\* Created on February 6, 2018, 10:37 PM

\*/

// BLACKJACK GAME

#include <cstdlib> //random value library

#include <iostream> //input and output stream

#include <ctime> // unique seed library

#include <iomanip> // formatting library

#include <string> //string library

void Shuffle(bool Deck[]);

void ShowCard(int Card);

void ShowHand(int Hand[], const int CardCount);

int DealCard(bool Deck[]);

int ScoreHand(int Hand[], const int CardCount);

void Score(int DealerHand[], const int DealerTotal, int PlayerHand[], const int PlayerTotal);

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

{

using namespace std;

// random generators

time\_t qTime;

time(&qTime);

srand(qTime);

bool Deck[52]; //our 52 card deck

int iDealerTotal = 0; //the total score of dealer's cards

int DealerHand[12]; //the hand dealt to dealer

int iPlayerTotal = 0; //player's total hand

int PlayerHand[12]; //player's dealt hand

// loop for the start of a hand

while (true) {

// shuffle deck to even the chance for each card

Shuffle(Deck);

// deal cards to dealer and player, each receives two cards.

PlayerHand[0]= DealCard(Deck); //first player card

DealerHand[0]= DealCard(Deck); //first dealer card (hidden)

PlayerHand[1] = DealCard(Deck); //second player card

DealerHand[1]= DealCard(Deck); //second dealer card

iDealerTotal = 2;

iPlayerTotal = 2;

// welcome sign with table rules and signals a new hand being dealt

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

cout << " Dealer must stand on 17 and draw to 16 " << endl; //gives basic rule of dealer rule

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Good Luck!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " Dealer has dealt a new hand! " << endl; //shows message to signal new hand

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

char PlayerChoice;

bool PlayerHits = true;

int PlayerScore = ScoreHand(PlayerHand, iPlayerTotal);

// calculate total score for each hand, including after each hit

do

{

// shoe dealer's and player's hand. only show one of the dealer's card per blackjac rules

cout << "Dealer's Hand: " << endl;

cout << "? "; //showing that the dealer has one face down card

ShowCard(DealerHand[1]);

cout << endl;

cout << "Player's Hand: Score = " << ScoreHand(PlayerHand, iPlayerTotal) << endl; //shows the total score of the player's cards

ShowHand(PlayerHand, iPlayerTotal); //show player cards

// ask player if they want to hit or stay with their dealt cards.

cout << "Do you: Hit(h) or stay(s): ";

cin >> PlayerChoice;

if (PlayerChoice == 'h') //player chooses between hit and stay

{

PlayerHand[iPlayerTotal] = DealCard(Deck);

++iPlayerTotal;

}

else if (PlayerChoice == 's') //player decided to stand

{

PlayerHits = false;

}

else //inpout validation. makes sure that the player chose a valid input, otherwise loops again

{

cout << "Error: Please pick a valid choice" << endl;

}

cout << endl;

// add player's card to update total and checks to make sure player hasn't gone over 21.

PlayerScore = ScoreHand(PlayerHand, iPlayerTotal);

}

while (PlayerHits && PlayerScore < 22);

// check if player has busted after hits, if player has the dealer wins.

if (PlayerScore > 21)

{

// player went over 21, dealer won.

cout << "// The Dealer Wins!!! \\" << endl;

Score(DealerHand, iDealerTotal, PlayerHand, iPlayerTotal);

}

else

{

// dealer has to reach 17 per blackjack rules. checks if dealer has 17, if not, dealer hits.

int iDealerScore = ScoreHand(DealerHand, iDealerTotal);

while (iDealerScore < 17) {

DealerHand[iDealerTotal] = DealCard(Deck);

++iDealerTotal;

iDealerScore = ScoreHand(DealerHand, iDealerTotal);

}

bool HouseBust = (iDealerScore > 21); //activates if dealer goes over 21

if (HouseBust)

{

// Dealer has gone over 21 and busted, player wins.

cout << "$$ The Player Wins!!! $$" << endl;

Score(DealerHand, iDealerTotal, PlayerHand, iPlayerTotal);

}

else

{

// if no one busts, compare scores to see who has the higher score

if (PlayerScore == iDealerScore) {

// player and dealer have same score, ends in a "push/tie"

cout << "\*\*\*\* TIE! \*\*\*\*" << endl;

Score(DealerHand, iDealerTotal, PlayerHand, iPlayerTotal);

} else if (PlayerScore > iDealerScore) {

// "The Player wins" output message

cout << "$$ The Player Wins!!! $$" << endl;

Score(DealerHand, iDealerTotal, PlayerHand, iPlayerTotal);

} else {

// "The dealer wins" output message

cout << "// The House Wins!!! \\" << endl;

Score(DealerHand, iDealerTotal, PlayerHand, iPlayerTotal);

}

}

}

}

return EXIT\_SUCCESS;

}

void Shuffle(bool Deck[]) { //shuffles deck

for (int iIndex = 0; iIndex < 52; ++iIndex)

{

Deck[iIndex] = false;

}

}

void ShowCard(int Card) { //calculates score for face cards. "jack, queen, king, ace"

using namespace std;

// calculations for each face card

const int CardRank = (Card % 13);

if (CardRank == 0)

{

cout << 'A'; // ace value

}

else if (CardRank < 9)

{

cout << (CardRank + 1);

}

else if (CardRank == 9)

{

cout << 'T';

}

else if (CardRank == 10)

{

cout << 'J'; // Jack value

}

else if (CardRank == 11)

{

cout << 'Q'; //queen value

}

else {

cout << 'K'; //king value

}

// calculates the suit of the card. Doesn't chance the value of the card in blackjack, but adds to realism of the game :)

const int Suit = (Card/13);

if (Suit == 0) {

cout << 'C'; //clover suit

} else if (Suit == 1) {

cout << 'D'; //diamond suit

} else if (Suit == 2) {

cout << 'H'; //hearts suit

} else {

cout << 'S'; // spade suit

}

}

void ShowHand(int Hand[], const int CardCount) { //displays hand totals

using namespace std;

for (int CardIndex = 0; CardIndex < CardCount; ++CardIndex) {

const int iDealCard = Hand[CardIndex];

ShowCard(iDealCard);

cout << " ";

}

cout << endl;

}

int DealCard(bool Deck[]) { //function for dealing out random cards

bool iCardDealt = true;

int iDealCard = -1;

do

{

iDealCard = (rand() % 52); //randomizes when dealing out a card from a 52 card deck

if (!Deck[iDealCard]) {

iCardDealt = false;

}

}

while (iCardDealt);

return iDealCard;

}

int ScoreHand(int Hand[], const int CardCount) { //determine soft ace value, per blackjack rules.

int AceScore = 0;

int iScore = 0;

for (int CardIndex = 0; CardIndex < CardCount; ++CardIndex) {

const int iDealCard = Hand[CardIndex];

const int CardRank = (iDealCard % 13);

if (CardRank == 0) {

++AceScore;

++iScore;

} else if (CardRank < 9) {

iScore = iScore + (CardRank + 1);

} else {

iScore = iScore + 10;

}

}

while (AceScore > 0 && iScore < 12) {

--AceScore;

iScore = iScore + 10;

}

return iScore;

}

void Score(int DealerHand[], const int DealerTotal, int PlayerHand[], const int PlayerTotal) { //function that adds adds up the dealer and player's cards after a hit and displays

using namespace std;

cout << "Dealer's Hand: Score = " << ScoreHand(DealerHand, DealerTotal) << endl;

ShowHand(DealerHand, DealerTotal);

cout << "Player's Hand: Score = " << ScoreHand(PlayerHand, PlayerTotal) << endl;

ShowHand(PlayerHand, PlayerTotal);

cout << endl;

}