BlackJack v2.0

12.13.2016

Tejash Bhakta

Submitted To: Dr. Mark Lehr
CSC 5 - 48102
Riverside City College

Table of Contents

Overview

Blackjack, also known as **twenty-one**, is the most widely played casino banking game in the world.[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"), 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 Value and 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.

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.

My Approach

While thinking about how I was going to program this game, a couple questions arose:

- "Since the card game has four suites, meaning four of each card, how do I tell the computer that I want to limit the number of times a random number is chosen?"
- "How will a player win or lose the game?"
- How to store suit and card value in a single variable?

After a couple of hours of planning my program and toiling with the above questions, I did some research and found that most (if not all) of the questions I was asking myself had a common answer: arrays. I used 2d array for the shuffling the deck, and parallel arrays to store the suit and card value in two different arrays but at the same position

Pseudo Code:

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

Flowchart

https://www.gliffy.com/go/publish/11245951

CONSTRUCTS & CONCEPTS UTILIZED

Ch	sec	Topic			Check	Line
2	1	Parts of C Program			♣	
	2	cout			♣	throughout
	3	#include			♣	7,8,9,10
	4	variables and literals			♣	throughout
	5	identifiers			♣	throughout
	6	Integer Data Types			♣	throughout
	7	char			♣	35
	8	strings			♣	36
	9	floating point types			♣	34
	10	booleans				
	11	sizeof				
	12	variable assignments			♣	throughout
	13	scope			♣	throughout
	14	operators			♣	throughout
	15	comments			♣	throughout
	16	constants			♣	15

3	1	cin			♣	throughout
	2	mathematical expressions			♣	throughout
	3	mixing data types				
	4	underflow/overflow				
	5	type casting			♣	32
	6	multiple assignments			♣	68
	7	iomanip			♣	54,161,162
	8	char and string objects				
	9	more math			♣	throughout
	10	hand tracing			♣	

4	1	relational operators			♣	throughout
	2	independent if single line			♣	124,169,187, 190,230
	3	independent if multiple line			♣	139,146,244, 245
	4	dependent if			♣	95,249,254,2 58
	5	nested constructs			♣	250
	6	dependent if			♣	250
	7	flags				
	8	logical operators			♣	134,179,259, 254,258
	9	number ranges with logical operators			♣	249,254,258
	10	menu			♣	
	11	validating user input			♣	52,104,131,1 76

	12	comparing char/string strcmp			♣	
	13	conditional operator			♣	
	14	switch			♣	
	15	block and scope				

5	1	pre/post increment/decrement			♣	throughout
	2	while loop			♣	
	3	while loop validation				
	4	counters			♣	
	5	do while			♣	
	6	for loop			♣	
	7	running total			♣	
	8	sentinels				
	9	which loop			♣	
	10	nesting loops			♣	
	11	data files				
	12	break/continuation			♣	

6	3	Prototypes			♣	18-29
	4&5	Pass by value			♣	18-29
	6&7&8	multiple returns/void			♣	18-19-20-23-28
	11	Static local				
	12	Defaulted Arguments				
	13	Pass by reference			♣	18-19-20-23-28
	14	Overloading			♣	24-25

	15	exit			♣	182
7	1 to 6	1 Dim Arrays			♣	
	7	Parallel Arrays			♣	22
	8	Function Arguments			♣	18-23
	9	2 Dim Arrays			♣	
	10	Func Args with 2Dim			♣	18,19,21
	12	Vectors				
8	1	Search(linear search)			♣	184
	2	Sort				
	3	Vectors Search/Sort				

References

1. Dr. Lehr's Lectures & Lab

2. "Starting Out with C++: From Control Structures through Objects" Gaddis, Tony. 8th Edition. (Textbook)

Program

```
/*
 * File:   main.cpp
 * Author: Tejash Bhakta
 * Purpose: Final Project BlackJack v2.0
 * Created on December 11, 2016, 5:17 PM
 */
#include <iostream>   //Input/Output objects
#include <iomanip>    //Foramatting
#include <cTime>      //Time functoin for setting the seed of random numbers
#include <cstdlib>    //C standard Library for random numbers

using namespace std;

//Global Constants
const int COLS=13;

//Function Prototypes;
void draw(bool [][COLS],string [],string [],int,int &);
void stand(bool [][COLS],string [],string [],int &,int);
void cardFace(string[],string[],int,int,int,int &);
void shuffle(bool[][COLS],int);
void show(string[],string [],int);
void isAce(string[],int &,int);
char check(int);
char check(int,int);
void seprtr();
void pause(int);
void result(char,float &,float,int &);
void winPer(int,int);
int main(int argc, char** argv) {
          //Random number generator
    srand(static_cast<unsigned int>(time(0)));
          //Declaration of variable
```

```
    float            totAmt=100, betPlcd;  //Total Winnings, Bet Placed
    char             nxt, chk;          //Deal next game, stores the status value
    string           name;                //Name of Player
    int won=0,game=0;

    //Display Introduction
//    display();
    //Start the game
    //Opening Commands
        seprtr();

        //Start the Game
        nxt='d';
        cout<<"Enter Player Name"<<endl;
        cin>>name;

        while(nxt=='d'){
            game++;
            do{
               seprtr();
               cout<<endl<<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);

                //In game declaration of variables
                const int SIZE=10;
                const int SZE=4;
                bool deck[SZE][COLS];
                string playV[SIZE],dealerV[SIZE];
                string playS[SIZE],dealerS[SIZE];
                int deal=0;
                int playTot,dealerTot;
                playTot=dealerTot=0;
                char cmd,chk;
```

```
//Shuffle the deck
shuffle(deck,SZE);

//Draw the first card for dealer
draw(deck,dealerV,dealerS,deal+1,dealerTot);

//Drawing the first two cards for player
for(int i=0;i<2;i++){
    ++deal;
    draw(deck,playV,playS,deal,playTot);
}

//Show cards and its total
//Dealer's card
cout<<"     Dealer's Cards"<<endl;
show(dealerV,dealerS,1);
isAce(dealerV,dealerTot,1);
cout<<"Total is "<<dealerTot<<endl;
seprtr();
//Player's card
cout<<"     "<<name<<"'s Cards"<<endl;
show(playV,playS,deal);
isAce(playV,playTot,deal);
cout<<"Total is "<<playTot<<endl;

//Close the deals if player hits blackJack
//If BlackJack
if(playTot==21){
    cout<<"****BlackJack****"<<endl;
    totAmt+=betPlcd*2.5;
    won++;
}
else{
    do{
        seprtr();
```

```
        cout<<"Do you want to hit(h), stand(s), or double down(d)?"<<endl;
        cin>>cmd;
    }while(cmd!='h'&& cmd!='s'&& cmd!='d');
}
seprtr();

//Following commands
while(cmd=='h'){
    ++deal;
    draw(deck,playV,playS,deal,playTot);
    cout<<"     "<<name<<"'s Cards"<<endl;
    show(playV,playS,deal);                  //Display the Cards
    isAce(playV,playTot,deal);
    cout<<"Total is "<<playTot<<endl;
    seprtr();

    //Checks if player hits blackjack or gets busted and will break respectively
    chk=check(playTot);
    if(chk=='b' || chk=='w'){
    stand(deck,dealerV,dealerS,dealerTot,2);
    chk=check(playTot,dealerTot);
    break;
    }

    //Continue drawing?
    do{
        cout<<"Do you want to hit or stand?"<<endl;
        cin>>cmd;
    }while(cmd!='h' && cmd!='s');
    seprtr();
}

//If player stands
if(cmd=='s'){
    deal=2;
    stand(deck,dealerV,dealerS,dealerTot,deal);
```

```
        chk=check(playTot,dealerTot);
    }

    //If player double downs
    if(cmd=='d'){
        ++deal;
        betPlcd*=4;
        //Draw just 1 card
        draw(deck,playV,playS,deal,playTot);
        cout<<"     "<<name<<"'s cards"<<endl;
        show(playV,playS,deal);
        isAce(playV,playTot,deal);
        cout<<"Total is "<<playTot<<endl;
        seprtr();
        stand(deck,dealerV,dealerS,dealerTot,2);
        chk=check(playTot,dealerTot);
    }

//Show the results according to check code
//and update total
cout<<name<<" total: "<<setw(6)<<playTot<<endl;
cout<<"Dealer total: "<<setw(6)<<dealerTot<<endl;
result(chk,totAmt,betPlcd,won);
seprtr();

//Display Total
 cout<<"The total Amount right now is "<<totAmt<<endl;
 if(totAmt<10){
     cout<<"Sorry.Not Enough cash to carry on"<<endl;
     break;
 }
 winPer(game,won);
//Continue if enough cash
  do{
    cout<<"Next Deal 'd' or 'x' key to exit"<<endl;
    cin>>nxt;
```

```
                sepртr();
              }while(nxt!='d' && nxt!='x');
        }
        cout<<"Thanks for playing"<<endl;
    exit(0);
}
void isAce(string ary[],int &tot,int cards){
    int ace=0;
    for(int i=0;i<cards;i++){
        if(ary[i]=="Ace")ace++;
    }
    for(int i=1;i<=ace;i++){
        if(tot>21)tot-=10;
    }
}
void cardFace(string hand[],string suit[],int row,int col,int pos,int &tot){
    int value;
    switch(row){
        case 0:suit[pos]="Spade";break;
        case 1:suit[pos]="Hearts";break;
        case 2:suit[pos]="Diamonds";break;
        case 3:suit[pos]="Clubs";break;
    }
    switch(col){
        case 0: hand[pos]="Two";value=2;break;
        case 1: hand[pos]="Three";value=3;break;
        case 2: hand[pos]="Four";value=4;break;
        case 3: hand[pos]="Five";value=5;break;
        case 4: hand[pos]="Six";value=6;break;
        case 5: hand[pos]="Seven";value=7;break;
        case 6: hand[pos]="Eight";value=8;break;
        case 7: hand[pos]="Nine";value=9;break;
        case 8: hand[pos]="Ten";value=10;break;
        case 9: hand[pos]="Jack";value=10;break;
        case 10:hand[pos]="Queen";value=10;break;
        case 11:hand[pos]="King";value=10;break;
```

```
        case 12:hand[pos]="Ace";value=11;break;
    }
    tot+=value;
}
void shuffle(bool deck[][COLS],int n){
    for(int i=0;i<n;i++){
        for(int j=0;j<COLS;j++){
            deck[i][j]=false;
        }
    }
}
void draw(bool deck[][COLS],string hand[],string face[],int deal,int &tot){
    int row,col;
    do{
        row=rand()%4;
        col=rand()%13;
        if(deck[row][col]==false){
            cout<<"Selected "<<row<<" "<<col<<endl;
            cardFace(hand,face,row,col,deal-1,tot);
            deck[row][col]=true;
        }
    }while(deck[row][col]==false);
}
void show(string value[],string face[],int n){
    for(int i=0;i<n;i++){
        cout<<value[i]<<" "<<face[i]<<" | ";
    }
    cout<<endl;
}
char check(int pTotal){
    if(pTotal>21)return 'b';
    if(pTotal==21)return 'w';

}
char check(int pTotal,int dTotal){
    if(dTotal<=21 && pTotal<=21){
```

```
        if(pTotal>dTotal)return 'w';
        else if(dTotal>pTotal)return 'l';
        else if(dTotal==pTotal)return 'p';
    }
    else if(dTotal>21 && pTotal<21){
        cout<<"Dealer Busted"<<endl;
        return 'w';
    }
    else if(dTotal>21 && pTotal>21){
        cout<<"Dealer as well as player Busted"<<endl;
        cout<<"NEXT DEAL"<<endl;
    }
    else return 'b';
}
void seprtr(){
    cout<<"--------------------------------"<<endl;
}
void stand(bool deck[][COLS],string val[],string suit[],int &tot,int deal){
    do{
        draw(deck,val,suit,deal,tot);
        cout<<"     Dealer's Cards"<<endl;
        pause(1);
        show(val,suit,deal);
        isAce(val,tot,deal);
        cout<<"Total is "<<tot<<endl;
        ++deal;
        seprtr();
    }while(tot<17);
}
void pause(int n){
    int beg=static_cast<unsigned int>(time(0));
    int end,elapse;
    do{
        end=static_cast<unsigned int>(time(0));
        elapse=end-beg;
    }while(elapse<n);
```

```
}
void result(char code,float &tot,float bPlcd,int &won){
    switch(code){
        case 'w':{ //Winner
            cout<<"You win"<<endl;
            tot+=bPlcd*2;        //Add the winnings
            won++;
        }break;
        case 'b':{ //Busted ie player lost
            cout<<"You Busted"<<endl;
            tot-=bPlcd;       //Deduct the bet
        }break;
        case 'l':{ //House total greater than player
            cout<<"You Lose"<<endl;
            tot-=bPlcd;       //Deduct the bet
        }break;
        case 'p': //No deduction in the bet
            cout<<"PUSH"<<endl;
            cout<<"Next Deal"<<endl;
    }
}

void winPer(int totGame,int won){
    cout<<won<<" out of "<<totGame<<" won"<<endl;
    cout<<"win percentage: "<<won*100/totGame<<"%"<<endl;
}
```