Final Project Write-Up

Blackjack

Phillip Pascual

CSC-5-40107

Introduction

Blackjack, also known as ‘21’, is a popular game of chance played in casinos worldwide. The object of the game is simple: to build a hand worth as much as 21 points without going over 21. The dealer, or ‘house’, does the same up, within a certain set of guidelines put forth by the house, and both hands are compared. If the player’s hand is greater than the dealer’s hand, or if the dealer goes over 21, the player wins. Alternatively, if the opposite is true, the dealer wins. Cards are worth their face value, except for face cards (Jack through King), which are worth 10, and Aces, which are worth 1 or 11. There are additional rules for more advanced players, but these are the basic rules of the game.

The house’s advantage comes from the setup of the game. While both of the player’s initial 2 cards are known, one card in the dealer’s hand is kept face-down, forcing the player to make a decision based on incomplete information. Additionally, any time the player exceeds 21 he or she automatically loses, in contrast to the dealer, who has a chance to ‘bust’ only after the player has been given an opportunity to do so. However, in spite of these advantages, the house advantage is still approximately 0.5%.

This program makes an initial effort in helping a blackjack player employ proper strategy in his or her play. By simulating a dealer’s base behavior, a player has the opportunity to play games of blackjack without any type of monetary risk.

Project Summary

Size: Approximately 460 lines

This project encompasses concepts of C++ covered and practiced in Chapter’s 1-8. It makes extensive use of functions in order to simulate the necessary steps to deal both the player’s and dealer’s blackjack hands, and uses functions to emulate the dealer’s artificial intelligence routines in order to deal according to the rules specified. Arrays and vectors were used in addition to the other primitive data types to store and display data.

All flowcharts, including the main flowcharts and flowcharts for each function, are included with this write-up. A checklist of concepts utilized is also included with this write-up.

/\*

File: main.cpp

Author: Phillip Pascual

Created on February 7, 2017, 12:30 PM

Purpose: Blackjack version 2 - Project 2

\*/

//System Libraries

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <iomanip>

#include <fstream>

#include <vector>

using namespace std;

//User Libraries

//Global Constants

//Such as PI, Vc, -> Math/Science values

//as well as conversions from system of units to

//another

const int SIZE=10;//Sets size of array

//Function Prototypes

int pDeal(string [],int &,int &,int &);

int dDeal(vector<string> &,int &);

void choice(string [],int &,int &,vector<string> &,int &);

int pHit(string [],int &,vector<string> &,int &);

void pCheck(int &,string[],vector<string> &,int &);

void pDouble(string [],int &,int &,vector<string> &,int &);

void pStand(vector<string> &,int &);

void dHit(vector<string> &,int&);

void compare(int,int,int,int &);

void sign(char [][10]);

//Executable code begins here!!!

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

//Set RNG seed

srand(static\_cast<unsigned int>(time(0)));//Sets RNG seed

//Instantiate and open files

ifstream in;

ofstream out;

in.open("blackjackstats.txt");

out.open("blackjackstats.txt",ios::app);

//Declare Variables

char welcome[3][10]={{' ','W','E','L','C','O','M','E',' ',' '},

{' ',' ',' ',' ','T','O',' ',' ',' ',' '},

{'B','L','A','C','K','J','A','C','K','!'}};

string name;

int stack=100,//Initial chip stack player has

bet;//Bet amount

int score;

int pTotal,dTotal;

char play='Y';

//Program description

sign(welcome);

cout<<"You will play a game of blackjack against a computer dealer. The "

"object"<<endl;

cout<<"of the game is to make a hand that beats the dealer's hand without "

"going"<<endl;

cout<<"over 21. The dealer will stand on any hand worth 17 or above."

<<endl;

cout<<endl;

cout<<"Please enter your name: ";

getline(cin,name);

//Display initial chip stack.

cout<<"Welcome "<<name<<". You have 100 chips to bet."<<endl;

do{

string pCards[SIZE]={};//,dCards[SIZE]={};

vector<string> dCards(SIZE);

pTotal=0;

dTotal=0;

pDeal(pCards,stack,bet=0,pTotal);//Get bet amount and deal player

//cards

cout<<endl;

dDeal(dCards,dTotal);//Deal initial dealer card

//Obtain player choice on if to hit, stand, double, or split

choice(pCards,pTotal,bet,dCards,dTotal);

compare(pTotal,dTotal,bet,stack);//Compare player and dealer hands

//As if player would like to play again

cout<<"Would you like to play again (Y or N)? ";

cin>>play;

while(play!='Y'&&play!='N'){

cout<<"Please enter Y or N: ";

cin>>play;

}

}while(play=='Y');

cout<<"Thanks for playing "<<name<<", the list of scores is available at "

"blackjackstats.txt."<<endl;

out<<stack<<" "<<name<<endl;

out<<endl;

cout<<endl;

cout<<"Here are some recent scores:"<<endl;

int i\_temp;

string s\_temp;

while(!in.eof()){

in>>i\_temp;

in>>s\_temp;

cout<<s\_temp<<" "<<i\_temp<<endl;

}

//Exit stage right!

in.close();

out.close();

return 0;

}

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

// Initial Player Deal Function

//Description: Deals and displays first two player cards and obtains bet amount

//Input: Bet amount

//Output: First two cards, score total, bet amount as reference variable

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

int pDeal(string pcard[],int &a,int &bet,int &pScore){

int deal;

int p1,pp1,p2,pp2;

cout<<"Please enter the amount that you would like to bet: ";

cin>>bet;

while (bet>a){

cout<<"Please enter a bet lower than your current stack: ";

cin>>bet;

}

//Generates first two cards for player

//Deal player cards

p1=pp1=rand()%13+2; //Generates random number between 2-14

//for card 1

//Corresponds random number 2-14 to player card 2-A

switch(p1){

case 2:pcard[0]+="2 ";break;

case 3:pcard[0]+="3 ";break;

case 4:pcard[0]+="4 ";break;

case 5:pcard[0]+="5 ";break;

case 6:pcard[0]+="6 ";break;

case 7:pcard[0]+="7 ";break;

case 8:pcard[0]+="8 ";break;

case 9:pcard[0]+="9 ";break;

//If RNG generates 10-K, score automatically set to add 10

case 10:{pcard[0]+="10 ";p1=10;}break;

case 11:{pcard[0]+="J ";p1=10;}break;

case 12:{pcard[0]+="Q ";p1=10;}break;

case 13:{pcard[0]+="K ";p1=10;}break;

//If RNG generates A, score automatically set to add A

case 14:{pcard[0]+="A ";p1=11;}break;

}

p2=pp2=rand()%13+2;//Generates random number between 2-14 for card 2

//Corresponds random number 2-14 to player card 2-A

switch(p2){

case 2:pcard[1]+="2 ";break;

case 3:pcard[1]+="3 ";break;

case 4:pcard[1]+="4 ";break;

case 5:pcard[1]+="5 ";break;

case 6:pcard[1]+="6 ";break;

case 7:pcard[1]+="7 ";break;

case 8:pcard[1]+="8 ";break;

case 9:pcard[1]+="9 ";break;

case 10:{pcard[1]+="10 ";p2=10;}break;

case 11:{pcard[1]+="J ";p2=10;}break;

case 12:{pcard[1]+="Q ";p2=10;}break;

case 13:{pcard[1]+="K ";p2=10;}break;

case 14:{pcard[1]+="A ";p2=11;}break;

}

//If two aces dealt, first ace switched from worth 11 to worth 1

if(pp1==14&&pp2==14){

p1=1;

}

pScore=p1+p2; //Adds first and second card to find initial score

cout<<"Your first two cards are: ";

for (int count=0;count<=1;count++){

cout<<pcard[count];

}

cout<<"= "<<static\_cast<int>(pScore);

if((p1+p2)==21){

cout<<"A natural blackjack! You win 3:2!"<<endl;

bet\*=1.5;

}

}

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

// Initial Dealer Function

//Description: Deals first dealer card and displays to player

//Input: None

//Output: First dealer card and total

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

int dDeal(vector<string> &dcard, int &dScore){

int d1,dd1;

//Deal first dealer card

d1=dd1=rand()%13+2; //RNG generates random number 2-14

//Corresponds random number 2-14 to dealer card 2-A

switch(d1){

case 2:dcard[0]+="2 ";break;

case 3:dcard[0]+="3 ";break;

case 4:dcard[0]+="4 ";break;

case 5:dcard[0]+="5 ";break;

case 6:dcard[0]+="6 ";break;

case 7:dcard[0]+="7 ";break;

case 8:dcard[0]+="8 ";break;

case 9:dcard[0]+="9 ";break;

case 10:{dcard[0]+="10 ";d1=10;}break;

case 11:{dcard[0]+="J ";d1=10;}break;

case 12:{dcard[0]+="Q ";d1=10;}break;

case 13:{dcard[0]+="K ";d1=10;}break;

case 14:{dcard[0]+="A ";d1=11;}break;

}

dScore=d1; //Stores initial dealer score

cout<<"The dealer's hand shows a "<<dcard[0]<<"= "<<setw(2)<<dScore

<<endl;

}

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

// Choice Function

//Description: Obtains choice to hit, stand, split or double down

//Input: Choice

//Output hit, stand, split or double down function

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

void choice(string cards[],int &totSc,int &bet,vector<string> &dCards,int &dealSc){

char ch;

cout<<"Would you like to [H]it or [S]tand or [D]ouble Down? ";

cin>>ch;

while(ch!='H'&&ch!='S'&&ch!='D'){

cout<<"Please enter a valid choice: ";

cin>>ch;

}

switch(ch){

case 'H':pHit(cards,totSc,dCards,dealSc);break;

case 'D':pDouble(cards,totSc,bet,dCards,dealSc);break;

case 'S':pStand(dCards,dealSc);break;

}

}

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

// Player Hit Function

//Description: Draws additional cards for player, displays additional cards and

// updated score. Asks player if wants to hit or stand.

//Input: Hit or Stand

//Output: Updated player hand and score

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

int pHit(string cards[],int &tot,vector<string> &dealer,int &dealSc){

int pAdd,pPlus;

int count=2;//Counter for array

bool c=true;

//do{

pPlus=pAdd=rand()%13+2;//Generates random number between 2-14

switch(pAdd){

case 2:cards[count]+="2 ";break;

case 3:cards[count]+="3 ";break;

case 4:cards[count]+="4 ";break;

case 5:cards[count]+="5 ";break;

case 6:cards[count]+="6 ";break;

case 7:cards[count]+="7 ";break;

case 8:cards[count]+="8 ";break;

case 9:cards[count]+="9 ";break;

case 10:{cards[count]+="10 ";pPlus=10;}break;

case 11:{cards[count]+="J ";pPlus=10;}break;

case 12:{cards[count]+="Q ";pPlus=10;}break;

case 13:{cards[count]+="K ";pPlus=10;}break;

case 14:{cards[count]+="A ";pPlus=11;}break;

}

count++;

tot+=pPlus;

for(int j=0;j<=count-1;j++){

if(cards[0]=="A "||cards[1]=="A "||cards[count-1]=="A "&&tot>=12){

tot-=10;

}

}

cout<<"The Dealer deals your next card. Your hand is now ";

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

cout<<cards[j];

}

cout<<"= "<<tot;

cout<<endl;

pCheck(tot,cards,dealer,dealSc);

//}while(c);

}

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

// Score Check Function

//Description: Checks score total and advises if bust or able to hit again

//Input: Score total

//Output: Bust, hit, or stand

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

void pCheck(int &score,string cards[],vector<string> &dealer,int &dScore){

char hitStnd;

if(score>=22){

cout<<"You bust!"<<endl;

}else{

cout<<"Would you like to [H]it or [S]tand? ";

cin>>hitStnd;

while(hitStnd!='H'&&hitStnd!='S'){

cout<<"Please enter H or S: ";

cin>>hitStnd;

}

if(hitStnd=='H'){

pHit(cards,score,dealer,dScore);

}else{

pStand(dealer,dScore);

}

}

}

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

// Double Down Function

//Description: Doubles the bet and proceeds through hit function

//Input: None

//Output: Double bet

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

void pDouble(string cards[],int &score,int &bet,vector<string> &dealer,int &dealSc){

bet\*=2;

cout<<"You double down, increasing your bet to "<<bet<<"."<<endl;

int pAdd,pPlus;

int count=2;//Counter for array

bool c=true;

//do{

pPlus=pAdd=rand()%13+2;//Generates random number between 2-14

switch(pAdd){

case 2:cards[count]+="2 ";break;

case 3:cards[count]+="3 ";break;

case 4:cards[count]+="4 ";break;

case 5:cards[count]+="5 ";break;

case 6:cards[count]+="6 ";break;

case 7:cards[count]+="7 ";break;

case 8:cards[count]+="8 ";break;

case 9:cards[count]+="9 ";break;

case 10:{cards[count]+="10 ";pPlus=10;}break;

case 11:{cards[count]+="J ";pPlus=10;}break;

case 12:{cards[count]+="Q ";pPlus=10;}break;

case 13:{cards[count]+="K ";pPlus=10;}break;

case 14:{cards[count]+="A ";pPlus=11;}break;

}

count++;

score+=pPlus;

for (int ace=2;ace<=count-1;ace++){

if(cards[ace]=="A "&&score>=12){

score-=10;

}

}

cout<<"The Dealer deals your next card. Your hand is now ";

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

cout<<cards[j];

}

cout<<"= "<<score<<endl;

if(score>22){

cout<<"You bust!"<<endl;

}

pStand(dealer,dealSc);

}

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

// Player Stand Function

//Description: Stops dealing player cards and starts dealing dealer cards until

// dealt over 17

//Input: None

//Output: Dealer hand and score

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

void pStand(vector<string> &dealer,int &dealSC){

int dCount=1;//Counter for array

int dAdd,dPlus;

dPlus=dAdd=rand()%13+2;//Generates random number between 2-14

switch(dAdd){

case 2:dealer[dCount]+="2 ";break;

case 3:dealer[dCount]+="3 ";break;

case 4:dealer[dCount]+="4 ";break;

case 5:dealer[dCount]+="5 ";break;

case 6:dealer[dCount]+="6 ";break;

case 7:dealer[dCount]+="7 ";break;

case 8:dealer[dCount]+="8 ";break;

case 9:dealer[dCount]+="9 ";break;

case 10:{dealer[dCount]+="10 ";dPlus=10;}break;

case 11:{dealer[dCount]+="J ";dPlus=10;}break;

case 12:{dealer[dCount]+="Q ";dPlus=10;}break;

case 13:{dealer[dCount]+="K ";dPlus=10;}break;

case 14:{dealer[dCount]+="A ";dPlus=11;}break;

}

dCount++;

dealSC+=dPlus;

cout<<"The Dealer flips his hole card. His hand is now ";

for (int j=0;j<=dCount-1;j++){

cout<<dealer[j];

}

cout<<"= "<<dealSC<<"."<<endl;

if(dealSC>21){

cout<<"The Dealer busts!";

}else{

dHit(dealer,dealSC);

}

}

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

// Dealer Hit Function

//Description: Deals additional dealer cards if dealer score under 17

//Input: None

//Output: Updated dealer hand and score

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

void dHit(vector<string> &dealer,int &dealSC){

int dCount=2;//Counter for array

int dAdd,dPlus;

dPlus=dAdd=rand()%13+2;//Generates random number between 2-14

switch(dAdd){

case 2:dealer[dCount]+="2 ";break;

case 3:dealer[dCount]+="3 ";break;

case 4:dealer[dCount]+="4 ";break;

case 5:dealer[dCount]+="5 ";break;

case 6:dealer[dCount]+="6 ";break;

case 7:dealer[dCount]+="7 ";break;

case 8:dealer[dCount]+="8 ";break;

case 9:dealer[dCount]+="9 ";break;

case 10:{dealer[dCount]+="10 ";dPlus=10;}break;

case 11:{dealer[dCount]+="J ";dPlus=10;}break;

case 12:{dealer[dCount]+="Q ";dPlus=10;}break;

case 13:{dealer[dCount]+="K ";dPlus=10;}break;

case 14:{dealer[dCount]+="A ";dPlus=11;}break;

}

dCount++;

dealSC+=dPlus;

cout<<"The Dealer deals his next card. His hand is now ";

for (int j=0;j<=dCount-1;j++){

cout<<dealer[j];

}

cout<<"= "<<dealSC<<"."<<endl;

if(dealSC>21){

cout<<"The Dealer busts!";

}else if(dealSC<17){

dHit(dealer,dealSC);

}

}

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

// Hand Compare Function

//Description: Compares dealer hand to player hand and determines who wins.

//Inputs: Dealer score, player score, bet amount, chip stack

//Output: Outcome of hand and updated chip stack.

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

void compare(int pTotal,int dTotal,int bet,int &stack){

cout<<endl;

if(pTotal>21&&dTotal<21){

stack-=bet;

cout<<"Sorry, you lose. Your stack is now "<<stack<<" chips."<<endl;

}else if(dTotal>21&&pTotal<21){

stack+=bet;

cout<<"You win! Your stack is now "<<stack<<" chips."<<endl;

}else if(pTotal>dTotal){

stack+=bet;

cout<<"You win! Your stack is now "<<stack<<" chips."<<endl;

}else if(pTotal<dTotal){

stack-=bet;

cout<<"Sorry, you lose. Your stack is now "<<stack<<" chips."<<endl;

}else{

cout<<"A push, nobody wins. Your stack remains "<<stack<<" chips."<<

endl;

}

}

void sign(char a[][10]){

for (int row=0;row<=2;row++){

for(int col=0;col<=9;col++){

cout<<a[row][col];

}

cout<<endl;

}

}