Blackjack 21 (Natural)

Venkatesh Tavva

Project 1

CIS5 – 45548

July 21, 2017

**Introduction:**

Title: Blackjack 21 (Natural)

Natural 21 is a type of blackjack game, where the player is dealt 2 random cards and the dealer takes two cards. If the cards add up to 21, they win their original bet in addition to 1.5 times their bet. So, for example if a user bet 20 dollars, if they win they would get their original 20 dollars in addition to 30 dollars from the dealer. The player can get a face card (King, Queen, and Jack) which takes a number value of 10, in addition to an Ace, which carries a value of 11 in this game. This combination would give a player 21. However, they can achieve 21, through just a 10 card, not a face card necessarily and an Ace. If, however the dealer also gets 21, the player gets to “push”, which they get to keep their original bet. If a player does not get 21, then they forfeit their bet and loose.

**Summary:**

Project size: about 160 lines

Number of variables: 9

This project took me about a week to do and most of it was spent to actually thinking about executing the idea and experimenting with different ways of getting the end result. It is a very simple game, which involves many concepts we learned so far, though not all. It can be improved and enhanced in many different ways, one which involves maybe inputting a “hit me” option just like in regular 21. One of the major troubles I had was dealing with the program not being able to realize that it had 21, but yet kept displaying a loss instead of a win.

**List of Concepts:**

|  |  |  |  |
| --- | --- | --- | --- |
| Chapter | Section | Topic | Line Number |
| 2 | 2 | cout | 44,45,50,52,56,58,60,62,64,66,70,72,74,76,82,84,86,88,  90,92,109,111,113,115,117,119,122,125,128,131,134,  137, 140,143,146,149,152,154,155,158,162 |
|  | 3 | Libraries | 12,13,14,15 |
|  | 4 | Variables/literals | 28,29,30,31,32,33 |
|  | 5 | identifiers | 28,29,30,31,32,33 |
|  | 6 | integers | 28,30,31,32,33 |
|  | 7 | characters | 29 |
|  | 8 | strings |  |
|  | 9 | floats |  |
|  | 10 | bools |  |
|  | 11 | Size of |  |
|  | 12 | Variables 7 characters or less | 28,29,30,31,32,33 |
|  | 13 | Scope\*\*\*\*\* |  |
|  | 14 | Arithmetic operators | 35,36,37,38,39,40,41 |
|  | 15 | comments | 28,29,30,31,32,33,35,36,37,38,39,40,41,58,60,62,64,66,70,72,74,76,78,84,86,88,90,92,96,98,100,102,104,106,108,110,112,114,116,118,121,124,127,130,133,136, 139,142,145,148 |
|  | 16 | Named constants |  |
|  | 17 | Programming style |  |
|  |  |  |  |
| 3 | 1 | Cin | 45,50,157 |
|  | 2 | Math expression | 39,40,41 |
|  | 3 | Mixing data types |  |
|  | 4 | Overflow/underflow |  |
|  | 5 | Type casting | 27 |
|  | 6 | Multiple assignment |  |
|  | 7 | Formatting output |  |
|  | 8 | strings |  |
|  | 9 | Math library | 15,35,36,37,38 |
|  | 10 | Hand tracing |  |
|  |  |  |  |
| 4 | 1 | Relational operators | 48,108,110,112,114,116,118,121,124,127,130,133,136,139,142,145,151,159,160 |
|  | 2 | if | 47,108,151,160 |
|  | 4 | If-else | 108,148 |
|  | 5 | nesting |  |
|  | 6 | If-else-if | 108,110,112,114,116,118,121,124,127,130,133,136, 139,142,145,148 |
|  | 7 | flags |  |
|  | 8 | Logical operators | 108,110,112,114,116,118,121,124,127,130,133,136, 139,142,145 |
|  | 11 | Validating user input |  |
|  | 13 | Conditional operator |  |
|  | 14 | switch | 57,69,83,95 |
|  |  |  |  |
| 5 | 1 | Increment/decrement |  |
|  | 2 | While | 159 |
|  | 5 | Do-while | 43 |
|  | 6 | For loop |  |
|  | 11 | Files input/output both |  |
|  | 12 | No breaks in loops |  |
|  |  |  |  |

**Pseudo Code:**

*Initialize*

*If player is ready to deal*

*Display input for bet from player*

*Else if player selects ‘n’*

*Display “Good-bye” and exit*

*If player inputs bet*

*Player is dealt two random cards*

*Computer holds two random cards*

*If the two cards player has add to 21*

*Player wins original bet + (1.5\*bet)*

*Else if dealer’s two cards add to 21, but not player*

*Player loses and bet is lost*

*Else if dealer and player both get 21*

*Player gets to keep original bet*

*Else*

*Players loses and bet is lost*

*If player chooses to play again*

*Start the program from beginning*

*Else if player selects ‘n’*

*Display “Good-bye” and exit*

**Actual Code:**

/\*

\* File: main.cpp

\* Author: Venkatesh Tavva

\* Created on July 17, 2017, 11:45 AM

\* Purpose: Create a game of natural 21, a

\* blackjack game where user is dealt

\* random two cards and hopefully they are

\* equal to 21

\*/

//System Libraries

#include <iostream> //Input - Output Library

#include <ctime>

#include <cstdlib>

#include <cmath>

using namespace std; //Name-space under which system libraries exist

//User Libraries

//Global Constants

//Function Prototypes

//Execution begins here

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

//Declare variables

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

int bet;// money bet

char choice1, choice2;

int betwon; //a 3 to 2 amount won if player has a natural 21

int pc1,pc2;// 2 cards randomly assigned to player

int cc1,cc2;// 2 cards randomly held by computer

int sumpc, sumcc;// sum of player cards and sum of computer cards

//Initialize variables

pc1 = rand()%13+1;//random number [1,13]

pc2 = rand()%13+1;// random number [1,13]

cc1 = rand()%13+1;// random number [1,13]

cc2 = rand()%13+1;// random number [1,13]

betwon = (bet+(1.5\*bet));// calculate winnings based on a 3 to 2 ratio

sumpc = pc1+pc2;//sum of the two player cards

sumcc = cc1+cc2;//sum of the two computer cards

//Input data

do{

cout<<"Hello, Welcome to Blackjack."<<endl<<endl;

cout<<"Ready to Deal? (y or n) ";

cin>>choice1;

cout<<endl;

if (choice1 == 'y') {

cout<<"How much do you want to bet? $";

cin>>bet;

cout<<endl;

//Map inputs to outputs or process the data

//Output the transformed data

cout<<"You are holding: ";

switch (pc1){

case 1: cout<<"a Ace,";//value of 11

break;

case 11: cout<<"a king,";//value of 10

break;

case 12: cout<<"a Queen, ";//value of 10

break;

case 13: cout<<"a Jack, ";//value of 10

break;

default: cout<<pc1<<",";//randomly assigned number

break;

}

switch (pc2){

case 1: cout<<" a Ace";//value of 11

break;

case 11: cout<<" a King";//value of 10

break;

case 12: cout<<" a Queen";//value of 10

break;

case 13: cout<<" a Jack";//value of 10

break;

default: cout<<pc2;//randomly assigned number

break;

}

cout<<endl<<endl;

cout<<"The computer is holding: ";

switch (cc1){

case 1: cout<<"a Ace, ";//value of 11

break;

case 11: cout<<"a King, ";//value of 10

break;

case 12: cout<<"a Queen, ";// value of 10

break;

case 13: cout<<"a Jack, ";//value of 10

break;

default: cout<<cc1<<",";//randomly assigned number

break;

}

switch (cc2){

case 1: cout<<" a Ace";//value of 11

break;

case 11: cout<<" a King";//value of 10

break;

case 12: cout<<" a Queen";//value of 10

break;

case 13: cout<<" a Jack";//value of 10

break;

default: cout<<cc2;//randomly assigned number

break;

}

cout<<endl<<endl;

if ((pc1==1)&&(pc2==10||pc2==11||pc2==12||pc2==13))//a ace and a queen, jack, king, or a 10.

cout<<"Congrats, you have a natural 21"<<endl<<endl<<"WON: $"<<betwon;

else if (sumpc == 21)//sum of two cards equal 21 if not a face card

cout<<"Congrats, you have a natural 21"<<endl<<endl<<"WON: $"<<betwon;

else if ((pc2==1) && (pc1==10||pc1==11||pc1==12||pc1==13))//a ace and a queen, jack, king, or a 10.

cout<<"Congrats, you have a natural 21"<<endl<<endl<<"WON: $"<<betwon;

else if ((cc1==1)&&(cc2==10||cc2==11||cc2==12||cc2==13))//lost if computer has 21 as well and player doesn't

cout<<"Computer has a natural 21"<<endl<<endl<<"YOU LOST";

else if ((cc2==1) && (cc1==10||cc1==11||cc1==12||cc1==13))//lost if computer has 21 as well and player doesn't

cout<<"Computer has a natural 21"<<endl<<endl<<"YOU LOST";

else if (sumcc == 21)//lost if computer has two cards with sum of 21 and player doesn't

cout<<"Computer has a natural 21"<<endl<<endl<<"YOU KEEP YOUR ORIGINAL BET: $"

<<bet;

else if (((cc1==1)&&(cc2==10||cc2==11||cc2==12||cc2==13)) && ((pc1==1)&&(pc2==10||pc2==11||pc2==12||pc2==13))) //both player and computer have 21

cout<<"Computer has a natural 21"<<endl<<endl<<"YOU KEEP YOUR ORIGINAL BET: $"

<<bet;

else if (((cc1==1)&&(cc2==10||cc2==11||cc2==12||cc2==13)) && (sumpc == 21))//both player and computer have 21

cout<<"Computer has a natural 21"<<endl<<endl<<"YOU KEEP YOUR ORIGINAL BET: $"

<<bet;

else if (((cc1==1)&&(cc2==10||cc2==11||cc2==12||cc2==13)) && ((pc2==1) && (pc1==10||pc1==11||pc1==12||pc1==13)))//both player and computer have 21

cout<<"Computer has a natural 21"<<endl<<endl<<"YOU KEEP YOUR ORIGINAL BET: $"

<<bet;

else if (((cc2==1) && (cc1==10||cc1==11||cc1==12||cc1==13)) && ((pc1==1)&&(pc2==10||pc2==11||pc2==12||pc2==13)))//both player and computer have 21

cout<<"Computer has a natural 21"<<endl<<endl<<"YOU KEEP YOUR ORIGINAL BET: $"

<<bet;

else if (((cc2==1) && (cc1==10||cc1==11||cc1==12||cc1==13)) && (sumpc == 21))//both player and computer have 21

cout<<"Computer has a natural 21"<<endl<<endl<<"YOU KEEP YOUR ORIGINAL BET: $"

<<bet;

else if (((cc2==1) && (cc1==10||cc1==11||cc1==12||cc1==13)) && ((pc2==1) && (pc1==10||pc1==11||pc1==12||pc1==13)))//both player and computer have 21

cout<<"Computer has a natural 21"<<endl<<endl<<"YOU KEEP YOUR ORIGINAL BET: $"

<<bet;

else if ((sumcc == 21) && ((pc1==1)&&(pc2==10||pc2==11||pc2==12||pc2==13)))//both player and computer have 21

cout<<"Computer has a natural 21"<<endl<<endl<<"YOU KEEP YOUR ORIGINAL BET: $"

<<bet;

else if ((sumcc == 21) && (sumpc == 21))//both player and computer have 21

cout<<"Computer has a natural 21"<<endl<<endl<<"YOU KEEP YOUR ORIGINAL BET: $"

<<bet;

else if ((sumcc == 21) && ((pc2==1) && (pc1==10||pc1==11||pc1==12||pc1==13)))//both player and computer have 21

cout<<"Computer has a natural 21"<<endl<<endl<<"YOU KEEP YOUR ORIGINAL BET: $"

<<bet;

else//player and computer don't have 21

cout<<"You don't have a natural 21"<<endl<<endl<<"YOU LOST";

}

if (choice1=='n'){

cout<<"Goodbye";

}

cout<<endl;

cout<<"Do you want to play again? (y or n) ";

cin>>choice2;

cout<<endl;

}while(choice2=='y');

if (choice2=='n')

{

cout<<"Goodbye";

}

//Exit stage right!

return 0;

