Project #2

BlackJack!

Joshua Duque

CIS 5-45548

28 July 2017

**Title: Blackjack!**

Game: The objective is to get as close to or score 21 points without going over. Going over is known as a bust and results in an automatic lost. The player is first dealt 2 cards and the sum of the two cards is the amount of points the player has. The player is then given the option to receive another card which is added up to the 2 initial cards the player received. If the player decides to stay, the house reveals its cards and whoever is closest to 21 wins. If the house and the player score the same points, the game results in a draw. For the player, all face cards have the value of 10, Ace cards have the value of 1 or 11, and any card with a number show its respective value. For the house, according to blackjack rules, if the first two cards have added value less than 16, they have to automatically pull another card and this continues until they go over 16.

**Summary**

Number of lines: 307

Number of variables: 30

This is a continuation of Project number 1. I included some arrays, and the missing concepts missed on the last projects. I couldn’t include 2d arrays and some last chapter concepts.

Pseudocode

*Include all Libraries*

*Declare all prototypes/functions*

*Enter main and set Random number seed generator*

*Declare Some Variables (Others were declared as they were needed)*

*Enter the start Menu by cin object and while loop. If “leave” is entered, user will not enter start menu*

*Display Menu (switch)*

*How to play*

*Play Game*

*End Game*

*Enter Valid Menu Option*

*If Entered How to play*

*Open Welcome Message through File*

*Close Welcome File*

*Display rules*

*Press enter to return to Menu*

*Display Menu*

*If Entered Leave Game*

*Goodbye is displayed and Program Will Terminate*

*If Entered Play Game*

*Open File*

*Enter UserName to be recorded into file “Stats”*

*Enter Do-While loop to Validate the amount of rounds the user desires (1round=3games)*

*If Number entered using cin object is positive,*

*Win, losses, draws are initialized to accumulate and record*

*Enter Nested For-Loop with another For-loop*

*Player total points and house total points are initialized for accumulation*

*Deal Cards 2 via function call deal(); x2*

*Values are added to initial total score*

*Give player option for another card using bool*

*If player wants another card Enter while- loop*

*Function call to deal() is used again x1*

*Values are added to accumulate*

*Give player option to get another card*

*If bool is true, enter while loop again*

*If bool is False*

*Display player total*

*Deal House Cards via function call dealh();*

*If Accumulated Value is less than 16,*

*Deal house until value is greater than 16 using while loop*

*Display House total*

*Outcome is Displayed using conditional operators and if and if-else statements*

*Compare player total to house total*

*Display Outcome*

*Increment wins by calling wi() function, Increment lost if lost, Increment draw if scores are tied*

*Loop Back till games within rounds are done x2 more*

*After 3 games, loop back till desired total rounds are reached*

*After all desired rounds are completed*

*Record Wins, lost, draw to file*

*Display “Stats are recorded to file”*

*Press Enter to continue*

*Loop Back to Menu*

*Display*

*How to play*

*Play Game*

*Leave Program*

*Function deal() (player)*

*Initialize Array and variables*

*Set random number generator to 0-13*

*Determine random number*

*Enter if statement to determine card value*

*Return Card value (Pass by Reference)*

*If Ace is return*

*Give option to choose card value 1 or 11*

*Return Ace Value*

*Determine Suit using String Array and random number generator 0-3*

*Return String based on number determined*

*Function dealh();*

*Set Random number generator 1-13*

*Return card Value*

*Set random number generator 1-4 to determine suit*

*Enter if statements*

*Print suit corresponding to number determine*

*Wi(); function*

*If called, increment win using static local variable*

*Return result*

|  |  |  |  |
| --- | --- | --- | --- |
| **Cross Reference for Project 2** | | | |
|  |  |  |  |
|  |  |  | **Used?** |
| **Chapter** | **Section** | **Topic** | **Line number** |
| 2 | 2 | cout | Yes |
|  | 3 | libraries | iostream, iomanip,, cstdlib, fstream, string, ctime |
|  | 4 | variables/literals | Yes |
|  | 5 | Identifiers | Yes |
|  | 6 | Integers | Yes |
|  | 7 | Characters | Yes |
|  | 8 | Strings | Yes |
|  | 9 | Floats No Doubles | Yes no doubles used |
|  | 10 | Bools | Yes |
|  | 11 | Sizeof \*\*\*\*\* |  |
|  | 12 | Variables 7 characters or less | Yes |
|  | 13 | Scope \*\*\*\*\* No Global Variables |  |
|  | 14 | Arithmetic operators | Yes |
|  | 15 | Comments 20%+ | Yes |
|  | 16 | Named Constants | No |
|  | 17 | Programming Style \*\*\*\*\* Emulate |  |
|  |  |  |  |
| 3 | 1 | cin | Yes |
|  | 2 | Math Expression | Yes |
|  | 3 | Mixing data types \*\*\*\* |  |
|  | 4 | Overflow/Underflow \*\*\*\* |  |
|  | 5 | Type Casting | Yes in random seed generator |
|  | 6 | Multiple assignment \*\*\*\*\* |  |
|  | 7 | Formatting output | Yes setprecision percentage |
|  | 8 | Strings | Yes |
|  | 9 | Math Library | No |
|  | 10 | Hand tracing \*\*\*\*\*\* |  |
|  |  |  |  |
| 4 | 1 | Relational Operators | Yes |
|  | 2 | if | Yes |
|  | 4 | If-else | Yes |
|  | 5 | Nesting | Yes |
|  | 6 | If-else-if | Yes |
|  | 7 | Flags \*\*\*\*\* |  |
|  | 8 | Logical operators | Yes |
|  | 11 | Validating user input | Yes |
|  | 13 | Conditional Operator | Yes |
|  | 14 | Switch | Yes |
|  |  |  |  |
| 5 | 1 | Increment/Decrement | Increment only |
|  | 2 | While | Yes |
|  | 5 | Do-while | Yes |
|  | 6 | For loop | Yes |
|  | 11 | Files input/output both | Yes |
|  | 12 | No breaks in loops \*\*\*\*\*\* |  |
|  |  |  |  |
| 6 | 3 | Function Prototypes | Yes |
|  | 5 | Passing by value | No |
|  | 8 | Returning values from functions | Yes |
|  | 9 | Returning a boolean \*\*\*\*\*\* |  |
|  | 10 | No Global Variables Allowed |  |
|  |  | Only Global Constants | No constants used… |
|  |  | Meaning Conversions,Physical Constants only |  |
|  | 11 | Static Local | Yes in wi() function |
|  | 12 | Default arguments | No |
|  | 13 | Reference Parameters | Yes |
|  | 14 | Overloading functions | No |
|  | 15 | Exit function \*\*\*\*\*\*\* |  |
|  |  |  |  |
| 7 | 4 | Array Initialization | Yes |
|  | 6 | Processing Arrays | Yes |
|  | 7 | Parallel Arrays | No |
|  | 8 | Arrays as function arguments | No |
|  | 9 | 2-D Arrays | No |
|  | 12 | STL Vector | No |
|  |  |  |  |
| 8 | 1 | Linear and Binary Search | No |
|  | 3 | Bubble and Selection Sort | No |
|  | 5 | Search/Sorting Vectors \*\*\*\*\*\* |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| \*\*\*\*\*\* Not required to show |  |  |  |

BlackJack Program

|  |
| --- |
| /\* |
|  | \* File: main.cpp |
|  | \* Author: Joshua Duque |
|  | \* Created on 7/27/17 |
|  | \* Purpose: Blackjack! with Arrays and sorting. |
|  | \*/ |
|  |  |
|  | //System Libraries |
|  | #include <cstdlib>//Random function srand |
|  | #include <iostream>//Standard input/output |
|  | #include <ctime>//time for random and program |
|  | #include <cstdlib> //srand |
|  | #include <string>// string |
|  | #include <fstream> // file |
|  | #include <cmath> //Math functions |
|  | #include <iomanip> //format |
|  | using namespace std; |
|  | //Function Prototypes |
|  | void deal(int&, int&, int&);// deal player function |
|  | void dealh(int&, int&, int&); //deal house function |
|  | float wi();//count win function |
|  | //Execution Starts Here |
|  | int main(int argc, char\*\* argv) { |
|  | //Random Seed and Variable Declaration |
|  | srand(static\_cast<unsigned int>(time(0))); |
|  | //Declare all variables here |
|  | char choice;//choice for menu |
|  | string enter, name, how,how2,how3;//enter program |
|  | //open an output file |
|  | cout<<"Type 'enter' to begin or type 'leave' to exit" <<endl; |
|  | cin>>enter; |
|  |  |
|  | while (enter != "leave" ){ |
|  | // Menu |
|  | cin.get(); |
|  | cout<<"---------------------------------------------------------"<<endl; |
|  | cout<<"BlackJack! Select an option"<<endl; |
|  | cout<<"1: Rules and How to"<<endl; |
|  | cout<<"2: Start!"<<endl; |
|  | cout<<"3: End Game"<<endl; |
|  | cout<<"---------------------------------------------------------"<<endl; |
|  | //Input your choice |
|  | cin>>choice; |
|  | //Solve the problem chosen |
|  | switch(choice){ |
|  | //summery |
|  | case '1':{ |
|  | ifstream input;// welcome input file |
|  | input.open("how.txt"); |
|  | input>>how;//welcome |
|  | cout<<how<<" "; |
|  | input>>how2; |
|  | cout<<how2<<" "; |
|  | input>>how3; |
|  | cout<<how3<<" "<<endl; |
|  | cout<<"Rules and How to play"<<endl; |
|  | cout<<"The objective is to score 21 points."<<endl; |
|  | cout<<"Cards 2-10 show its worth values."<<endl; |
|  | cout<<"Face cards are all worth 10."<<endl; |
|  | cout<<"Aces are worth 1 or 11."<<endl; |
|  | cout<<"Your initially dealt two cards and asked to"; |
|  | cout<<" Hit or Sit."<<endl; |
|  | cout<<"If you choose to hit, you will be given "; |
|  | cout <<"another card."<<endl; |
|  | cout<<"If you choose to sit, you will stay with "; |
|  | cout<<"the current cards you have"<<endl; |
|  | cout<<"The closest one to 21 without going over "; |
|  | cout<< "wins."<<endl; |
|  | cout<<"Going Over 21 is known as a bust"; |
|  | cout<<" and will result in a lost"<<endl; |
|  | cout<<"If a draw is to occur, no one wins"<<endl; |
|  | cout<<"Press Enter to continue"<<endl; |
|  | input.close(); |
|  | cin.get(); |
|  | } |
|  | break; |
|  |  |
|  | case '2': { |
|  | ofstream outputFile; |
|  | outputFile.open("Stats.txt");//open stats file |
|  | cout<<"Enter your first name!"<<endl; |
|  | cin>>name;//enter name to file |
|  | outputFile<<name<<endl;//record name to file |
|  | outputFile<<":"<<endl; |
|  | int rounds, games; // rpunds and games |
|  | cout<<"See if you can beat the house!"<<endl; |
|  | do{ //input validation |
|  | cout<<"How many rounds would you like to play? 1 round = 3 games"<<endl; |
|  | cout<<"Enter a positive number"<<endl; |
|  | cin>>rounds; |
|  | } while(rounds<0);//make sure rounds is positive |
|  | cout<<" "<<endl; |
|  | float win, lost,draw,wper; |
|  | win=0.0;//initialize accumulation for wins loss and draws |
|  | lost=0.0; |
|  | draw=0.0; |
|  | for(int count=0;count<rounds;count++) |
|  | { |
|  | for(int i=0;i<3;i++)//nested loops for number of games in rounds |
|  | { |
|  | //play the game |
|  |  |
|  | int card1,card2,value, suit, pT, total, hT; //card1 |
|  |  |
|  | //card2, value to pass to by reference, suit , player total |
|  | //points, total to pass by reference, house total |
|  | char yn; //yes or no |
|  | bool stay=true; //leave menu true or false |
|  | //initialize accumulator for running total |
|  | pT=0;// player total |
|  | hT=0;//house total |
|  |  |
|  | //Player's Initial Hand |
|  |  |
|  | cout<<"You have been dealt the following cards: "; |
|  |  |
|  | deal(card1, suit, total);// call fuction |
|  | cout<<" and "; |
|  | deal(card2, suit, total);// call deal fuction |
|  | pT=card1+card2;// add initial cards |
|  |  |
|  | cout<<"Your score is now: "; |
|  | cout<<pT<<endl;//display initial score |
|  | cout<<"Would you like another card?"<<endl; |
|  | cout<<"Enter '1' for yes and '0' for no: "; |
|  | cin>>stay; |
|  | //Allow Player to Hit or Sit |
|  |  |
|  | while (stay && pT<21) |
|  | { |
|  | cout<<"You have been dealt a "; |
|  | deal(value, suit, total); // deal player function call |
|  | pT+=value; //accumulate total player |
|  | cout<<"Your score is now: "; |
|  | cout<<pT; |
|  | cout<<" "<<endl; |
|  | cout<<"Would you like another card?"<<endl; |
|  | cout<<"Enter '1' for yes and '0' for no: "; |
|  | cin>>stay; |
|  |  |
|  | } |
|  | //House's Hand |
|  | cout<<" "<<endl; |
|  | cout<<"The house has been dealt: "; |
|  | dealh(value, suit, total); |
|  | hT+=value; |
|  | do{ |
|  | cout<<" and "; |
|  | dealh(value, suit, total); |
|  | hT+=value; |
|  | //if points are less than 16 house continues to deal |
|  | }while(hT<16); |
|  | cout<<" The house's score is: "; |
|  | cout<<hT; |
|  | cout<<" "<<endl; |
|  |  |
|  | //Outcome of game |
|  | if(pT==21 && (pT!=hT))// if player gets 21 and is not tied |
|  | cout<<"BlackJack! You win!"<<endl; |
|  | if(pT>hT&&pT<21) |
|  | { |
|  | cout<<"You win"<<endl; |
|  | wi(); |
|  | } |
|  | else if(hT>21&&pT<=21)//if house goes over 21 and player |
|  | { //is below 21 |
|  | cout<<"House went over 21. You win"<<endl; |
|  | wi(); |
|  | } |
|  | else if(pT>21)//if player gets over 21 |
|  | { |
|  | cout<<"Bust! You went over 21!"<<endl; |
|  | lost++; |
|  | } |
|  | else if(pT<hT&&hT<=21)//if house gets closer to 21 but |
|  | { //below 21 |
|  | cout<<"House Wins "<<endl; |
|  | lost++; |
|  | } |
|  | else if(pT==hT)//draw if player and house gets the same |
|  | { //score |
|  | cout<<"Draw!"<<endl; |
|  | draw++; |
|  |  |
|  | } |
|  |  |
|  | } |
|  | } |
|  | //record stats in file |
|  | win=wi()-1; |
|  | wper=(win/(rounds\*3))\*100; |
|  | outputFile<<" win-"<<endl; |
|  | outputFile<<win<<endl; |
|  | outputFile<<" lost-"<<endl; |
|  | outputFile<<lost<<endl; |
|  | outputFile<<" draw-"<<endl; |
|  | outputFile<<draw<<endl; |
|  | outputFile<<" win percentage: "; |
|  | outputFile<<"%"<<setprecision(3)<<wper; |
|  | outputFile.close();//close file |
|  |  |
|  | cout<<" "<<endl; |
|  | cout<<"Stats Updated to File!"<<endl; |
|  | cout<<"Press enter to continue"<<endl; |
|  | cin.get();//pause so its easier to see interface |
|  | break; |
|  |  |
|  | } |
|  | case '3': |
|  |  |
|  | cout<<"Goodbye!"<<endl; |
|  | exit(0);//exit menu and program |
|  |  |
|  | break; |
|  |  |
|  | default: cout<<"Please pick the given options"<<endl; |
|  | //input valididation using loop and default |
|  | } |
|  | } |
|  | cout<<"Goodbye"<<endl; |
|  |  |
|  | return 0;//exit |
|  | } |
|  |  |
|  | //Function Definition (Card Dealing Function) |
|  | //Pass by reference so we can change original argument and accumulate |
|  | void deal(int& crdV, int& crdS, int& Total){ //deal player |
|  | //Randomly selects card values |
|  | int r,siz=13; |
|  | r=rand()%13+0; |
|  | int card[siz]={1,2,3,4,5,6,7,8,9,10,11,12,13}; |
|  | if ((card[r]<=9) && (card[r]>=2)){//number card |
|  | crdV=card[r]; |
|  | cout<<crdV; |
|  | cout<<" of "; |
|  | } |
|  | if (card[r]==10){//jack |
|  | cout<<" Jack of "; |
|  | crdV=10; |
|  | } |
|  | if (card[r]==11){//queen |
|  | cout<<" Queen of "; |
|  | crdV=10; |
|  | } |
|  | if (card[r]==12){//king |
|  | cout<<" King of "; |
|  | crdV=10; |
|  | } |
|  | if (card[r]==1){ //Ace plus the option to pick if its 11 or 1 |
|  | cout<<" You got an Ace, "; |
|  | int ace; |
|  | cout<<"would you like the value to equal 1 or 11: "; |
|  | cin>>ace; |
|  | if (ace==1){ |
|  | crdV=1; |
|  | }else |
|  | crdV=11; |
|  | } |
|  | int size=4, rs; |
|  | string suit[size]={"Hearts","Diamonds","Spades","Clubs"}; |
|  | rs=rand()%4+0; |
|  | cout<<suit[rs]<<endl; |
|  | } |
|  | void dealh (int& crdV, int& crdS, int& Total){ //deal house |
|  | //Randomly selects card values |
|  | crdV=rand()%13+1; |
|  | if (crdV<=10&&crdV>=2){ |
|  | cout<<crdV; |
|  | cout<<" of "; |
|  | } |
|  | if (crdV==11){ |
|  | cout<<" Jack of "; |
|  | crdV=10; |
|  | } |
|  | if (crdV==12){ |
|  | cout<<" Queen of "; |
|  | crdV=10; |
|  | } |
|  | if (crdV==13){ |
|  | cout<<" King of "; |
|  | crdV=10; |
|  | } |
|  | if (crdV==1){ |
|  | cout<<" Ace of "; |
|  | if (Total<10){// Ace card value =11 if total house points is less than |
|  | crdV=11; //10 |
|  | }else |
|  | crdV=1;//if total is less than 10 the value of ace is 1 |
|  | } |
|  | crdS=rand()%4+1; // Suits |
|  | if(crdS==1){ |
|  | cout<<"Clubs "; |
|  | } |
|  | if(crdS==2){ |
|  | cout<<"Diamonds "; |
|  | } |
|  | if(crdS==3){ |
|  | cout<<"Spades "; |
|  | } |
|  | if (crdS==4){ |
|  | cout<<"Hearts "; |
|  | } |
|  | } |
|  | float wi(){// record wins function |
|  | static float win; |
|  | win++; |
|  | return win; |
|  |  |
|  | } |