**Project Two**

Title

Higher or Lower V.6

Course

CSC-5

Section

47108

Due Date

June 9, 2023

Author

Haylee Ferguson

1. Introduction

My game, Higher or Lower, is a simple card game in which one can choose a starting balance that will be used to place bets, place bets, and gamble on whether the next card drawn will be of a higher, or lower value than the previous one. This game utilizes a 52 card deck, except for the Joker cards, that are shuffled at random and one card is pulled per round. This game is non-sequential, meaning every round is completely random.

1. Rules of the Game

When the player begins the game, they will select a starting balance, this balance does not have a limit. Once they have selected their starting amount, they will be prompted to choose a bet amount at the beginning of each round. A random card will then be drawn, and the user must decide if the next card will be of a higher, or lower value than the previous one. If the player chooses correctly, the amount they placed as a bet will be added to their balance. If the player chooses incorrectly, their bet amount will be deducted from their balance. The rounds will continue to go on until either the player decides to end the game, or they run out of funds. The player’s name and finishing balance amount will be saved to a player data text file that can be viewed at any point as it will be saved on the user’s computer.

1. Game Play

When the program begins, the user will be prompted with a welcome message and enter their name, which will be saved to a player data file as well as the money they finish with, to view at any point. The player can then select whether or not they would like to read the rules of the game by entering either “Y” or “N”. If the user chooses to have the rules read to them, the rules will be printed from a text file, ReadRules.txt, and then they may decide if they would like to read the rules again or begin the game. If the user decides to not read the rules, the game will begin. Upon starting the game, the user will enter “S” to signify they are ready to begin.

Once ready, the user will be asked how much money they would like to start out with; This amount does not have a limitation. Once that amount has been decided, the user will then select the amount they would like to bet on the round they are playing. The player may continue to place bets as long as they have a sufficient balance. If the player does not have enough funds, they will receive a message informing them they can no longer bet. A random card will be selected, and every round is a random card and is not sequential, and the user has to decide: Will the next card be higher, or lower in value? Once the player makes their guess, they will input either “H” for higher or “L” for lower. If the player guesses correctly, they will be congratulated and the amount they placed as a bet will be added to their funds. If the player guesses incorrectly, the amount they chose to bet will be deducted from their funds.

The game will continue in rounds until either the player runs out of money, in which they will be told so, or if they choose to exit the game at the beginning of any round by entering “Y” for yes or “N” for no to continue. Once the game has been completed, the user will get a thank you message and their player data will be saved.

1. Development Summary

As it stands, there are four versions of this game, not including a work in progress. The very first version did not include a read in text file for the rules, and instead output them. It was initially a switch case menu so that the user was prompted to either start the game or to read the rules. The beginning and ending messages were very simple.

Version two eliminated the menu and included the ReadRules.txt file to utilize a file input/output. It also included the PlayerData.txt file to save player data such as their name and finishing balance. There were still some technical difficulties that had not yet been corrected as well as some formatting issues.

Version three had very insignificant changes.

Version four was the complete version and included all project requirements except for the type casting element. And not many other adjustments were made.

I was working on adding in a string function that would read in the cards for every turn, but I had a difficult time with it. I started by creating one function, but realized that a string function can only return one output, so I knew that wasn’t going to work. Unfortunately I could not finish despite the guidance I received from the online lab aides. Changing the function I was planning on using meant I would have to rewrite most of my code, and I already had a difficult time at that.

As for the other criteria, I was running low on ideas on how to incorporate them as my game was very simple to begin with. It is clear now that I lack creativity with coding and that became apparent to me as I worked on this project. I have learned my weaknesses and will ensure that I am caught up before my next class in the fall.

(P.S. you were right about falling behind Professor!)

1. Specifications

5.1

**Sample User Input/Output**

When the game begins, the user will be prompted to enter their name, select a starting balance, and start the game.

cout<<”What is your name?”<<endl;

cin>>name;

cout<<”How much money would you like to start with “<<name<<”?”<<endl;

cin>>balance;

cout<<”Enter ‘S’ to start!”<<endl;

cin>>x;

The user’s starting balance will then be displayed on the screen:

cout<<”Your current balance is $ “<<balance<<endl;

And then they will be asked to choose how much they would like to bet:

cout<<”How much would you like to bet” <<name<<”?”<<endl;

If the bet the user places is more than the amount of funds they currently have available, the user will be told:

cout<<”I’m sorry”<<name<<” you don’t have enough to place that bet.”<<endl;

cout<<”Please place a new bet.”<<endl;

After the user places their bet, a card will be drawn to start the round:

cout<<”You drew the “<<card<<” of “<<suit<<”.”<<endl;

cout<<”Will the next card be higher (H) or lower (L)?”<<endl;

cin>>guess;

If the player guesses correctly, they will receive a congratulatory message and the addition of funds to their balance. If the player guesses incorrectly, the player will be told so and the bet amount will be deducted from their funds.

cout<<”Congratulations, “<<name<<”! You guessed correctly!”<<endl;

Or

cout<<”Sorry, “<<name<<” you guessed incorrectly.”<<endl;

At the end of every turn, the player will be asked if they would like to continue, if they would, they enter “Y”, if not they would enter “N”.

cout<<”Would you like to continue playing?(y/n)”<<endl;

cin>>choice;

At the end of the game, the player will be thanked and their data will be saved.

5.2

**Pseudocode**

| Line 23 | Diamonds, spades, clubs, and hearts |
| --- | --- |
| Line 24 | Card values 1-13, including king, queen, and jack \*\*comparing face value |
| Line 33 | Beginning of game |
| Line 46 | Open the “Rules.txt” file for reading |
| Line 53 | Read and output rules from the file |
| Line 59 | Close the file after reading |
| Line 60 | Set the flag to false to exit the loop |
| Line 63 | False to exit loop |
| Line 93 | Random card and suit |
| Line 105 | Cards and their values |
| Line 119 | cout<<”Your current balance is $”<<balance<<endl; |
| Line 120 | cout<<”How much would you like to bet, “<<name<<”?”<<endl; |
| Line 121 | cin>>bet; |
| Line 128 | Random card and suit |
| Line 140 | Cards and their values |
| Line 165 | Lose bet amount from balance if guessed incorrectly |
| Line 180 | Open a new file for writing |
| Line 187 | Write player’s name and money to the file |
| Line 191 | Close the file after writing |

5.3

**Variables**

| Char | x | answer | guess | choice |
| --- | --- | --- | --- | --- |
| String | name | pSuit | suit | card |
| Bool | drawCard | vBet |  |  |
| Float | balance |  |  |  |

6.Program

/\*

\* File: main.cpp

\* Author: Haylee Ferguson

\* Created on

\* Purpose: Game of Cards: Higher or Lower V5

\*/

//System Libraries

#include <iostream>

#include <fstream>

#include <ctime>

#include <iomanip>

#include <string>

#include <cstdlib>

#include <cmath>

using namespace std;

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

// Set Random Number seed

//Declare Variables

char x, answer, guess = 0, choice; // Diamonds, Spades, Clubs, and Hearts

int bet, cardVal, pCard = 0, suitI; //Card value 1-13, including King, Queen, and Jack \*\*comparing face value

bool drawCard, vBet;

string suits[] = { "Spades", "Hearts", "Diamonds", "Clubs" };

string pSuit, suit, card, name;

float balance;

// Initialize Inputs

cout<<setw(15)<<right<<"Welcome!"<<endl;

cout<<"Let's play a card game!"<<endl<<endl; //Beginning of game

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

cout<<setw(20)<<right<<"HIGHER OR LOWER!"<<endl;

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

bool readRules = true;

for (int i = 0; i < 1 && readRules; i++) {

cout<<"Do you want to read the rules? (y/n) "<<endl;

cin>>choice;

if (choice == 'y' || choice == 'Y') {

ifstream file("Rules.txt"); // Open the "Rules.txt" file for reading

if (!file) {

cout << "Failed to open file.\n";

return 1;

}

// Read and output the rules from the file

string line;

while (getline(file, line)) {

cout << line << endl;

}

file.close(); // Close the file after reading

readRules = false; // Set the flag to false to exit the loop

}

else if (choice == 'n' || choice == 'N') {

readRules = false; // false to exits the loop

}

else {

cout << "Invalid choice. Please enter 'y' or 'n'.\n";

}

}

cout<<"\nLet's play!"<<endl<<endl;

cout<<"What is your name?"<<endl;

cin>>name;

cout<<"How much money would you like to start with, "<<name<<"?"<<endl;

cin>>balance;

cout<<"Enter 'S' to start!";

cin>>x;

if (x == 's' || x == 'S')

cout<<"\n\nLet's begin!"<<endl<<endl;

do {

cout<<"Your current balance is $"<<balance<<endl;

cout<<"How much would you like to bet, "<<name<<"?"<<endl;

cin>>bet;

if (bet > balance) {

cout<<"I'm sorry "<<name<<" you don't have enough to place that bet."<<endl;

cout<<"Please place a new bet."<<endl;

continue;

}

srand(time(0)); //random card and suit

suitI = rand() % 4;

suit = suits[suitI];

cardVal = rand() % 13 + 2;

if (cardVal >= 2 && cardVal <= 10) {

card = to\_string(cardVal);

}

else {

switch (cardVal) {

case 11:

card = "Jack"; //Cards and their values

break;

case 12:

card = "Queen";

break;

case 13:

card = "King";

break;

case 14:

card = "Ace";

break;

}

}

//cout << "Your current balance is $" << balance << endl;

//cout << "How much would you like to bet, " << name << "?" << endl;

//cin >> bet;

cout<<"You drew the "<<card<<" of "<<suit<<"."<<endl;

cout<<"Will the next card be higher (H) or lower (L)?"<<endl;

cin>>guess;

cout<<endl;

srand(time(0)); //random card and suit

suitI = rand() % 4;

suit = suits[suitI];

pCard = rand() % 13 + 2;

if (pCard >= 2 && pCard <= 10) {

card = to\_string(pCard);

}

else {

switch (pCard) {

case 11:

card = "Jack"; //Cards and their values

break;

case 12:

card = "Queen";

break;

case 13:

card = "King";

break;

case 14:

card = "Ace";

break;

}

}

cout<<"The next card is the "<<card<<" of "<<suit<<"."<<endl<<endl;

if ((guess == 'H' || guess == 'h' && cardVal < pCard) ||

(guess == 'L' || guess == 'l' && cardVal > pCard)) {

cout << "Congratulations, "<<name<<"! You guessed correctly!"<<endl<<endl;

balance += bet;

cout<<"Would you like to continue playing? (y/n)"<<endl;

cin>>choice;

}

else {

cout<<"Sorry, "<<name<<", you guessed incorrectly."<<endl<<endl;

balance -= bet; //Lose bet amount from balance if guessed incorrectly

cout<<"Would you like to continue playing? (y/n)"<<endl;

cin>>choice;

}

if (balance <= 0)

{

cout<<"Sorry, "<<name<<" You've run out of funds! Game over! Thanks for playing :)"<<endl;

}

} while (balance > 0 && choice == 'Y' || choice == 'y');

ofstream outputFile("PlayerData.txt"); //Open a new file for writing

if (!outputFile) {

cout<<"Failed to create file.\n";

return 1;

}

// Write player's name and money to the file

outputFile<<"Player: "<<name<<endl;

outputFile<<"Money: $"<<balance<<endl;

outputFile.close(); // Close the file after writing

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

cout<<"You finished with $"<<balance<< ", "<<name<<"."<<endl;

cout<<"Thank you for playing!"<<endl;

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

return 0;

}