Bradley Erwin

CSC 5 – 40488

Game Project:

A variation on “Between the Sheets”

**Introduction to the Game:**

This game is a variation of the betting game “Between the Sheets”. The program will output 2 random numbers between 1 and 12 (these will be called the limits). It will then give you the choice between playing the hand or passing for another hand. Whether you decide to play or not, after showing you 3 hands, then the computer player(s) will get a turn to play. They will each get 3 hands as well then it will return to you afterwards. There is a starting amount in the “pot”, which is “the house’s” money.

**Winning/Losing Hands:**

If you decide to play the hand, you will be asked how much that you want to bet. After that, another random number will be given. If that number is between the 2 limits, then you will win your money back as well as an addition amount equal to your bet. If the number is on the outside of the limits, then you lose your bet amount. Lastly if the number is equal to one of the limits, then you will lose twice the amount that you bet. Since the losses can be doubled, there is a possibility that you lose more money than you have. If that is the case, then a mobster will find you and collect. Any bet that is lost goes into the pot.

**Ending the Game:**

The game ends when there is no more money left in the pot or if the player and computers have all run out of money.

**Extra information:**

The computers will always play and will always bet a random number between 1 and 100.

**In/Out Files:**

The in file is “StartingMoney.dat” and it is used to determine the starting amounts of money for all of the players involved. The out file is “ResultsOfGames.dat” and it displays the results at the end of the game and includes the ending amounts of money of all the computer players and the player. Any computer that is not in the game will have a default amount of $0. The file will also have the win and loss percentage of just the player.

Code:

/\*

\* File: main.cpp

\* Author: Bradley Erwin

\* Created on January 29, 2017, 5:00 PM

\* Finished January 30, 2017

\* Purpose: Game Project

\*/

//System Libraries Here

#include <iostream>//Input/output library

#include <cstdlib> //Random number generator seed

#include <ctime> //Time used to set seed

#include <fstream> //File stream library

#include <iomanip> //Manipulation Library

#include <cmath> //Math Library

#include <vector> //Vector Library

using namespace std;

//User Libraries Here

//Global Constants Only, No Global Variables

//Like PI, e, Gravity, or conversions

const short PERCENT=100;//Percentage conversion

const int MILLION=1e6;//Million

const int HUNTHSD=1e5;//one hundred thousand

const int COLS=3;

//Function Prototypes Here

void gamePly(int [][COLS],vector<int> &,long int &,long int &,

int,int &,int &,int &);//Computer Play Function

void gamePly(int [],long int &,long int &,unsigned short &,

unsigned short &,unsigned short &);//Player Game Function

char gtChce(); //Get Choice for # computers

char gtChce2(); //Get Choice for viewing array

char gtChce3(); //Get Choice for Searching Array

char gtChce4(); //Get Choice for Search Array

bool winTest(char,char,char);

bool lssTest(char,char,char);

void prntAry(int [],int);

void srchAry(int [],int [],int);

void srtAry(int [],int);

//Program Execution Begins Here

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

//Set random number seed

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

//Declare all Variables Here

const int SIZE=500;

char choice=0, //Choice of amount of computers

choice2=0, //Choice for viewing array

choice3=0, //Choice for searching

choice4=0; //Choice for sorting

long potAmnt=500, //Default Starting amount in pot

c1Money=0,

c2Money=0,

c3Money=0,

mnyLeft=500;

unsigned short numGams=0,

wins=0,

losses=0;

int c1Wins=0,

c2Wins=0,

c3Wins=0,

c1Lss=0,

c2Lss=0,

c3Lss=0,

c1Gms=0,

c2Gms=0,

c3Gms=0,

val=0;

int cBets[SIZE][COLS]={};

int plrBets[SIZE]={};

int srchBet[SIZE]={};

vector<int> compGms(SIZE);

string plyrNme;

float winPct,

lossPct;

//Instantiate and open files

ifstream in;

ofstream out;

in.open("StartingMoney.dat");

while(in>>mnyLeft); //Read in starting amount for player

mnyLeft=mnyLeft>MILLION?HUNTHSD:mnyLeft;//Limit Starting amount

potAmnt=mnyLeft\*2; //Set pot = to starting money

in.close();

//Input or initialize values Here

cout<<"Please enter your name."<<endl;//Determine Player name

getline(cin,plyrNme);

choice=gtChce();

//Set Computer's starting money to equal Player's money

switch(choice){

case '3':{

c3Money=mnyLeft;

}

case '2':{

c2Money=mnyLeft;

}

case '1':{

c1Money=mnyLeft;

}

}

//Start game loop

do{

//Start player's game

gamePly(plrBets,potAmnt,mnyLeft,wins,losses,numGams);

//Switch to determine number of computers and to loop each round

switch(choice){

case '3':{ //For Computer #3

gamePly(cBets,compGms,potAmnt,c3Money,3,c3Wins,c3Lss,c3Gms);

}

case '2':{

gamePly(cBets,compGms,potAmnt,c2Money,2,c2Wins,c2Lss,c2Gms);

}

case '1':{

gamePly(cBets,compGms,potAmnt,c1Money,1,c1Wins,c1Lss,c1Gms);

}

}

}while(potAmnt>0&&(mnyLeft>0||c1Money>0||c2Money>0||c3Money>0));

//Print Array into file

out.open("CompBetsArray.dat");

out<<"Game: Bets:"<<endl;

out<<" C1: C2: C3:"<<endl;

out<<setw(3)<<compGms[0]<<" $"<<setw(3)<<cBets[0][0]<<" $"

<<setw(3)<<cBets[0][1]<<" $"<<setw(3)<<cBets[0][2]<<endl;

for(int i=1;i<SIZE;i++){

(cBets[i][0]!=0)?out<<setw(3)<<compGms[i]

<<" $"<<setw(3)<<cBets[i][0]<<" $"

<<setw(3)<<cBets[i][1]<<" $"<<setw(3)<<cBets[i][2]<<endl:

out<<"";

}

out.close();

//Calculations

winPct=wins\*1.0f/numGams\*PERCENT;

lossPct=losses\*1.0f/numGams\*(pow(10,2));

//Output Stats to file

out.open("ResultsOfGame.dat");

out<<setprecision(2)<<fixed<<endl;

out<<"Game Over! Final Results:"<<endl;

out<<plyrNme<<": $"<<mnyLeft<<endl;

out<<"Computer 1: $"<<c1Money<<endl;

out<<"Computer 2: $"<<c2Money<<endl;

out<<"Computer 3: $"<<c3Money<<endl;

out<<"The House: $"<<potAmnt<<endl;

out<<"Your Win Percentage = "<<winPct<<endl;

out<<"Your Loss Percentage = "<<lossPct<<endl;

out<<"Player Wins = "<<wins<<endl;

out<<"Player Losses = "<<losses<<endl;

out.close();

cout<<"Game Over! Please check ResultsOfGame.dat for results."<<endl;

choice2=gtChce2();

if(choice2='1')prntAry(plrBets,SIZE);

choice3=gtChce3();

if(choice3='1')srchAry(plrBets,srchBet,SIZE);

choice4=gtChce4();

if(choice4='1'){srtAry(plrBets,SIZE);

prntAry(plrBets,SIZE);}

//Exit

return 0;

}

void srtAry(int a[],int n){

for(int i=0;i<n-1;i++){

for(int j=i+1;j<n;j++){

if(a[i]>a[j]){

a[i]=a[i]^a[j];

a[j]=a[i]^a[j];

a[i]=a[i]^a[j];

}

}

}

}

void srchAry(int a[],int b[],int n){

int val=0;

cout<<"What amount do you want to find?"<<endl;

cin>>val;

cout<<endl;

cout<<val<<" was found at "<<endl;

for(int i=0,j=0;i<n;i++){

if(a[i]==val)b[j]=i,j++;

}

for(int i=0;i<n;i++){

(b[i]!=0)?cout<<b[i]<<" ":cout<<"";

}

cout<<endl;

}

void prntAry(int bets[],int n){

cout<<"Player Bets:"<<endl;

for(int i=0;i<n;i++){

if(bets[i]!=0)cout<<"$"<<bets[i]<<endl;

}

cout<<endl;

}

bool winTest(char d1,char d2,char d3){//Tests for win or loss

bool win;

if((d3>d1&&d3<d2)||(d3<d1&&d3>d2))win=true;

else win=false;

return win;

}

bool lssTest(char d1,char d2,char d3){//Test to see if loss is double

bool dble;//Double the loss?

if((d3==d1||d3==d2))dble=true;

else dble=false;

return dble;

}

char gtChce(){

char numComp;

cout<<"How many computers would you like to play with? "

"(between 1 and 3)"<<endl;

cin>>numComp;

//Validate choice for # of computers

while(numComp>'3'||numComp<'0'){

cout<<"Invalid choice, please choose between 1 and 3"<<endl;

cin>>numComp;

}return numComp;

}

char gtChce2(){

char numComp;

cout<<"Enter 1 to view the player bet array? "

<<endl;

cin>>numComp;

//Validate choice for # of computers

while(numComp>'2'||numComp<'1'){

cout<<"Invalid choice, please enter "

"1 to view array"<<endl;

cin>>numComp;

}return numComp;

}

char gtChce3(){

char numComp;

cout<<"Enter 1 to search the player bet array? "

<<endl;

cin>>numComp;

//Validate choice for # of computers

while(numComp>'2'||numComp<'1'){

cout<<"Invalid choice, please enter "

"1 to search array"<<endl;

cin>>numComp;

}return numComp;

}

char gtChce4(){

char numComp;

cout<<"Enter 1 to sort the player bet array? "

<<endl;

cin>>numComp;

//Validate choice for # of computers

while(numComp>'2'||numComp<'1'){

cout<<"Invalid choice, please enter "

"1 to sort array"<<endl;

cin>>numComp;

}return numComp;

}

void gamePly(int a[][COLS],vector<int> &v,long int &potAmnt,

long int &cMoney,int cNum,int &w,int &l,int &g){

//Initialize array

for(int game=0;(game<=2)&&(cMoney>0)&&(potAmnt>0);game++){

//Call random number generator for the dice

char die1=rand()%12+1; //Value from 1-12; Sets a Limit

char die2=rand()%12+1; //Value from 1-12; Sets a Limit

char die3=rand()%12+1; //Value from 1-12

int betAmt=rand()%100+1;//Betting Amount from 1-100

//Validate to not have same limits

while(die1==die2){

die2=rand()%12+1;//Reroll second die

}

//Display the 2 limits

cout<<"Will the next roll land between these 2 numbers?"<<endl;

cout<<static\_cast<int>(die1)<<" "<<static\_cast<int>(die2)<<endl;

cout<<endl;

while(potAmnt<betAmt){

betAmt-=2;

}

cout<<"Computer "<<cNum<<" bets "<<"$"<<betAmt<<endl;

cout<<endl;

cout<<"The next Roll is:"<<static\_cast<int>(die3)<<endl;

cout<<endl;

if(winTest(die1,die2,die3)){//Win?

w++;

cMoney+=betAmt; //C1's money increases

potAmnt-=betAmt; //Money pot decreases

cout<<"Computer "<<cNum<<" Won $"<<betAmt<<endl;}//Lose?

else if (lssTest(die1,die2,die3)){ //Did the roll land on a limit?

l++;

cMoney-=betAmt\*2; //If true, lose double the bet

potAmnt+=betAmt\*2;//Money pot increases

cout<<"Computer "<<cNum<<" Lost $"<<2\*betAmt<<endl;}

else(l++,

(cMoney-=betAmt), //Regular loss, just lose bet

(potAmnt+=betAmt), //Money pot increases

(cout<<"Computer "<<cNum<<" Lost $"<<betAmt<<endl));

cout<<"Computer "<<cNum<<" has $"<<cMoney<<" left."<<endl;

cout<<"There is $"<<potAmnt<<" left in the pot."<<endl;

cout<<endl;

a[g][cNum-1]=betAmt;

v[g]=g+1;

g++;

(potAmnt<1)?(cout<<"No money left "

"in the pot!"<<endl)&&(game+=3):' ';

(cMoney<1)?(cout<<"Computer "<<cNum<<" is out"

" of money!"<<endl)&&(game+=3):' ';

}

}

void gamePly(int a[],long int &potAmnt,long int &monyLft,unsigned short &wins,

unsigned short &losses,unsigned short &nGames){

//First Loop is for the player

for(int game=0;(game<=2)&&(monyLft>0)&&(potAmnt>0);game++){

//Call random number generator for the dice

char die1=rand()%12+1;//Value from 1-12; Sets a Limit

char die2=rand()%12+1;//Value from 1-12; Sets a Limit

char die3=rand()%12+1;//Value from 1-12

char decPlay;//Decision to Play?

int betAmnt=0;//Betting Amount

bool test1=(die3>die1&&die3<die2)||

(die3<die1&&die3>die2),//Did roll land within limits?

test2=(die3==die1||die3==die2);//Did roll land on a limit?

//Validate to not have same limits

while(die1==die2){

die2=rand()%12+1;//Reroll second die

}

//Show how much money is in player's pocket and in the pot

cout<<"You have $"<<monyLft<<" left."<<endl;

cout<<"There is $"<<potAmnt<<" left in the pot."<<endl;

//Display the 2 limits

cout<<"Will the next roll land between these 2 numbers?"<<endl;

cout<<static\_cast<int>(die1)<<" "<<static\_cast<int>(die2)<<endl;

cout<<"Press 1 to Play or 2 to Pass"<<endl;

cin>>decPlay;

cout<<endl;

while(decPlay<'1'||decPlay>'2'){ //Validate play decision

cout<<"That is not a valid input, please enter 1 to play "

"or 2 to pass"<<endl;

cin>>decPlay;

cout<<endl;

}

if(decPlay=='2'); //If pass, do nothing

else if(decPlay=='1'){ //Execute play

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

cin>>betAmnt; //Enter bet

while(potAmnt<betAmnt){

cout<<"Not enough funds in the pot, "

"please enter an amount that "

"is a max of $"<<potAmnt<<endl;

cin>>betAmnt;

}

cout<<endl;

cout<<"The next Roll is:"<<static\_cast<int>(die3)<<endl;

cout<<endl;

if(test1){

monyLft+=betAmnt;//Your money increases

wins++;

potAmnt-=betAmnt;//Money pot decreases

cout<<"You Won $"<<betAmnt<<endl;

}

else if(test2){

monyLft-=betAmnt\*2;//If true, lose double the bet

potAmnt+=betAmnt\*2;//Money pot increases

cout<<"You Lost $"<<2\*betAmnt<<endl;

losses++;

}

else if(!test2){

monyLft-=betAmnt;//Regular loss, just lose bet

potAmnt+=betAmnt;//Money pot increases

cout<<"You Lost $"<<betAmnt<<endl;

losses++;

}

else cout<<"Error"<<endl;

cout<<endl;

a[nGames]=betAmnt;

nGames++; //Increment Games Played

}

if (potAmnt<1){

cout<<"No money left in the pot!"<<endl;

game+=3;

}

if (monyLft<1){

cout<<"You are out of money!"<<endl;

game+=3;

}

}

}