Alejandro Bravo

CSC-5-42520

Date: 05/30/19

project 2

**Introduction**

**Title: Street Craps**

Street Craps is a simplified variation of traditional casino Craps. Unlike its casino counterpart, this version is played without a banker or a table and offers only **pass/don’t pass** betting options.

The game begins when a player (the shooter) challenges an opponent and places a wager. The shooter then rolls the dice under the following conditions:

* **Automatic Win:** Rolling a **7 or 11** on the first attempt results in an immediate win.
* **Automatic Loss:** Rolling a **2, 3, or 12** on the first attempt results in an immediate loss.
* **Point System:** Any other number establishes a *point*. The shooter must roll the *point* again before rolling a **7** to win. If a **7** appears first, the shooter loses the bet. The dice continue rolling until one of these conditions is met.

Street Craps is a popular game, particularly in lower-income communities, where it is often played informally. Having grown up in such an environment, I frequently observed people engaging in this game. Even in high school, students played Street Craps despite the potential risk of fines. It remains a widely enjoyed game among friends and family, often serving as both a recreational activity and a means of quick earnings.

**Project Summary**

* **Total lines of code:** ~343
* **Number of variables used:** ~23
* **Programming concepts applied:** Chapters **1-5** and **6-8**

This project builds upon my initial implementation of Street Craps by incorporating more advanced programming concepts. While the game functions as intended, some bugs remain that require further refinement. Developing this project has significantly enhanced my understanding of complex programming principles. I have gained proficiency in utilizing **arrays, vectors, and functions** to improve program efficiency and structure.

Although **C++** initially posed a challenge, this experience has strengthened my confidence in solving fundamental programming problems and applying key software development concepts effectively.

**Flow chart**

A picture containing text, map

Description automatically generated

**pseudo code**

*Initialize*

*If player selects 1*

*The game begins*

*If player selects 2*

*The rule is display*

*The game begins*

*If player selects 3*

*Functions are call*

*The percent in winning is display*

*The game ends*

*Player enters the number of people playing*

*If player enter >2*

*Player enter the name of 2 people*

*Else player 1 is name Dealer and player enter their name*

*Player enter bet amount*

*If player enter bet <1.00*

*The message prompt player to enter a higher bet*

*Else bet must be integer*

*The game will loop back to enter bet*

*The dice is roll*

*If dice roll 7 or 11*

*Player 2 wins*

*If dice roll 2,3, or 12*

*Player 1 wins*

*Else the dice roll again*

*If dice roll point*

*Player 2 wins*

*If dice roll 7*

*Player 1 wins*

*Else dice continues to roll*

*Prompts to read what the dice roll if user choice*

*If answer yes, then the file is read*

*Else prompts if the user wants to play again*

*If user selects yes*

*The game loops back to make a bet*

*Else prompts the user with message and statistics of the game*

*Ends program*

**The project checklist**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Cross Reference for Project 1*** | | | | | |
|  |  |  |  |  |  |
| ***You are to fill-in with where located in code*** | | | | |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| ***Chapter*** | ***Section*** | ***Topic*** | ***Where Line #''s*** | ***Pts*** | ***Notes*** |
| *2* | *2* | *cout* | *59-77 and more* |  |  |
|  | *3* | *libraries* | *9 through 15* | *8* | *iostream, iomanip, cmath, cstdlib, fstream, string, ctime* |
|  | *4* | *variables/literals* | *31-36 and other lines* |  | *No variables in global area, failed project!* |
|  | *5* | *Identifiers* | *31-36 and other lines* |  |  |
|  | *6* | *Integers* | *31-36 and other lines* | *3* |  |
|  | *7* | *Characters* | *31-32 and other lines* | *3* |  |
|  | *8* | *Strings* | *33* | *3* |  |
|  | *9* | *Floats No Doubles* | *34* | *3* | *Using doubles will fail the project, floats OK!* |
|  | *10* | *Bools* | *125* | *4* |  |
|  | *11* | *Sizeof \*\*\*\*\** |  |  |  |
|  | *12* | *Variables 7 characters or less* | *33* |  | *All variables <= 7 characters* |
|  | *13* | *Scope \*\*\*\*\* No Global Variables* |  |  |  |
|  | *14* | *Arithmetic operators* | *can be find in most of project* |  |  |
|  | *15* | *Comments 20%+* | *use for most of project* | *5* | *Model as pseudo code* |
|  | *16* | *Named Constants* | *36* |  | *All Local, only Conversions/Physics/Math in Global area* |
|  | *17* | *Programming Style \*\*\*\*\* Emulate* |  |  | *Emulate style in book/in class repositiory* |
|  |  |  |  |  |  |
| *3* | *1* | *cin* | *49,84,85,90,148,157* |  |  |
|  | *2* | *Math Expression* | *in multiple lines* |  |  |
|  | *3* | *Mixing data types \*\*\*\** |  |  |  |
|  | *4* | *Overflow/Underflow \*\*\*\** |  |  |  |
|  | *5* | *Type Casting* | *26* | *4* |  |
|  | *6* | *Multiple assignment \*\*\*\*\** |  |  |  |
|  | *7* | *Formatting output* | *110,114,118,122,137,000* | *4* |  |
|  | *8* | *Strings* | *33* | *3* |  |
|  | *9* | *Math Library* | *15* | *4* | *All libraries included have to be used* |
|  | *10* | *Hand tracing \*\*\*\*\*\** |  |  |  |
|  |  |  |  |  |  |
| *4* | *1* | *Relational Operators* | *108 and many more* |  |  |
|  | *2* | *if* | *96,148,158* | *4* | *Independent if* |
|  | *4* | *If-else* | *114* | *4* |  |
|  | *5* | *Nesting* | *140* | *4* |  |
|  | *6* | *If-else-if* | *121* | *4* |  |
|  | *7* | *Flags \*\*\*\*\** |  |  |  |
|  | *8* | *Logical operators* | *100* | *4* |  |
|  | *11* | *Validating user input* | *48* | *4* |  |
|  | *13* | *Conditional Operator* | *114* | *4* |  |
|  | *14* | *Switch* | *56* | *4* |  |
|  |  |  |  |  |  |
| *5* | *1* | *Increment/Decrement* | *120* | *4* |  |
|  | *2* | *While* | *94* | *4* |  |
|  | *5* | *Do-while* | *132* | *4* |  |
|  | *6* | *For loop* | *228* | *4* |  |
|  | *11* | *Files input/output both* | *131,150* | *8* |  |
|  | *12* | *No breaks in loops \*\*\*\*\*\** |  |  | *Failed Project if included* |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| *\*\*\*\*\*\* Not required to show* |  |  | *Total* | *100* |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cross\_List\_For\_Proj2 |  |  |  |  |  |  |
| **Cross Reference for Project 2** | | |  |  |  |  |
| **You are to fill-in with where located in code** | | |  |  |  |  |
| **Chapter** | **Section** | **Topic** | **Where Line #''s** | **Pts** | **Notes** | |
| 6 |  | Functions |  |  |  | |
|  | 3 | Function Prototypes | 16-27 | 4 | Always use prototypes | |
|  | 5 | Pass by Value | 48 | 4 |  | |
|  | 8 | return | 305 | 4 | A value from a function | |
|  | 9 | returning boolean | 248,280 | 4 |  | |
|  | 10 | Global Variables |  | XXX | Do not use global variables -100 pts | |
|  | 11 | static variables | 50,305 | 4 |  | |
|  | 12 | defaulted arguments | 16,53,226,54 | 4 |  | |
|  | 13 | pass by reference | 307,315…. | 4 |  | |
|  | 14 | overloading | 226,334 | 5 |  | |
|  | 15 | exit() function | 88 | 4 |  | |
| 7 |  | Arrays |  |  |  | |
|  | 1 to 6 | Single Dimensioned Arrays | 48 | 3 |  | |
|  | 7 | Parallel Arrays | 49,48 | 2 |  | |
|  | 8 | Single Dimensioned as Function Arguments | 265,277,  288, 48.. | 2 |  | |
|  | 9 | 2 Dimensioned Arrays |  | 2 | Emulate style in book/in class repositiory | |
|  | 12 | STL Vectors | 52 | 2 |  | |
|  |  | Passing Arrays to and from Functions | 261,55,277… | 5 |  | |
|  |  | Passing Vectors to and from Functions | 52,309,84…. | 5 |  | |
|  |  |  |  |  |  | |
| 8 |  | Searching and Sorting Arrays |  |  |  | |
|  | 3 | Bubble Sort | 272 | 4 |  | |
|  | 3 | Selection Sort | 322 | 4 |  | |
|  | 1 | Linear or Binary Search | 282 | 4 |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
| \*\*\*\*\*\* Not r | equired to | show | Total | 70 | Other 30 points from Proj 1 first sheet tab | |

**Screen shoots of my Program**

Just in case if the program does not work in a different computer. Here are some of my screen shot of my program functioning.

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

**Program**

#include <iostream>//Input/Output Library

#include <string> // String Library

#include <fstream> // File IO Library

#include <cstdlib> // Random Number Generator Library

#include <ctime> // Time function

#include <iomanip> // format function

#include <cmath> // math libary

#include <stdlib.h> // exit function

#include <vector> // vector libary

using namespace std;

//Global Constants, no Global Variables are allowed

//Math/Physics/Conversions/Higher Dimensions - i.e. PI, e, etc...

//Function Prototypes

void dice(int r=0, int c=0, int cnt=0); // prints the dice at the start

bool onep ( char& ); // checks if the number of players is greater than 1

void math( int, int, int &); // get the number for array

void fillA (int [], int SIZE=0); // fill the array

void prntAry(int [],int,int); // print the array

bool linSrch(int [],int,int,int &,int &); // serach the array

void bublSrt(int [],int); // sort the array

void selSrt(vector<int> &,int,int &); // sort the vector

float perC(int [],int); // prints the percent of winning in the first row

void secR(vector<int> &, int, int &); // fill in the vector

void printV(vector<int> &,int, int &); //print the vector

float perC2(vector<int> &,int); // get the percent of wining second roll

//Execution Begins Here!

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

//seed random number generator

srand(static\_cast<unsigned int> (time(0))); //seed random number generator

//Declare Variables/Open Files

ifstream inFile("craps.txt");

ofstream outFile;

outFile.open("craps.txt");

inFile.open("craps.txt");

char ans, ans1; //the option for user to restart the game

char opt1; //the option for user at main menu

int opt; //the option for user at main menu

string P1, P2, line; //the names of players

float bet, total; //the bet for the game

int lucky = 7, lucky2 = 11, win, lost; // values

const float perCov = 100.0f; // convert decimal to percent

int row,column; // prints the dice on top

int die1,die2; // numbers for the dice row

int indx,indx1; // looks for the number in the array

const int SIZE=1000; // size of the array

int array[SIZE]; // the array to hold the first row dice sum

static int Total[SIZE]; // the arry to hold the total amount

int val=50;

static int sum;

vector <int> sumA(SIZE);

//The main menu of the game

dice(3,4);

fillA(array,1000);

cout << "🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟" << endl;

cout << setw(4) << " 🎲 Welcome to Bravo street craps.🎲" << endl;

cout << "🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟" << endl;

cout << "1. Play the Game" << endl;

cout << "2. Display the Rules and about" << endl;

cout << "3. For fun math"<<endl;

cin>>opt; //Gets to choose to show the rules or not

cout << endl;

while (opt < 1 || opt > 3 ) //Input validation. Can only enter 1 or 2.

{

cout << "Invalid Input. Please try again" << endl;

cin>>opt;

cout << endl;

}

cin.ignore(1000, '\n'); //Clears anything after player entered their value

switch (opt) { //Menu with the 2 choices

case 3:{

cout<<"The odds of winning in the first roll(out of a 1000 times play) "<<endl;

cout<<endl;

bublSrt(array,SIZE);

prntAry(array,SIZE,val);

if(linSrch(array,SIZE,val,indx,indx1)){

cout<<endl;

cout<<"7 is found at : "<<indx<<endl;

cout<<"12 is found at : "<<indx1<<endl;

cout<<"you won about "<< perC(array,SIZE)<<" percent of the time"<<endl;

} cout<<endl;

cout<<"your odds after the first row"<<endl;

secR(sumA,val,sum);

selSrt(sumA,val,sum);

printV(sumA,val,sum);

cout<<endl;

cout<<"you won about: "<<perC2(sumA,SIZE)<<" percent of the time"<<endl;

cout<<"The sum of the dice:"<<Total[SIZE]<<endl;

exit(0);

}

case 2: //Displays the rules if 2 is selected

{

cout << "--------------------------" << endl;

cout << "~ Rules of Street Craps. ~" << endl;

cout << "--------------------------" << endl;

cout << "1.The game is played with 2 six-sided dices" << endl;

cout << "2.The game is similar to casino craps but slight difference." << endl;

cout << "3.This version of craps does not have a banker and a table." << endl;

cout << "4.If the shooter rolls the dices and the sums 7 and 11 then they win bet."

<< endl;

cout << "5.If the shooter rolls the dices and the sums is 2,3,and 12 then shooter lose."

<< endl;

cout << "6.If any other number lands then it becomes point." << endl;

cout << "7.Point must be match in order for shooter to win" << endl;

cout << "8.if shooter lands on 7 after the first row then they lose" << endl;

cout << "9.the games continues until point is match or a seven is roll" << endl;

cout << "10.The most important rule is not getting caught by Police 😉" << endl;

cout << "-----------------------" << endl;

cout << "~ about Street Craps. ~" << endl;

cout << "-----------------------" << endl;

cout << "This version of Craps is popular in the Ghetto and is illegal to play in the street" << endl;

cout << "The game has different names like shooting dice and Ghetto Craps." << endl;

cout << "Its a fun game to play with friends and family 👍" << endl;

cout << "Press Enter to Play." << endl; //Allows player to read rules then press enter to play

cin.get();

}

default: //Will always end up here, no breaks before this. This is the actual game part

{

// The code for the game

cout<<"How many player are playing"<<endl;

cin>>opt1;

if( onep (opt1)){

cout<<endl;

cout<<"Enter your Name"<<endl;

P1="Dealer";

cin>>P2;

} else {

cout << "Enter player 1 and players 2 a.k.a the shooter Names" << endl;

cin>>P1; // the name of player 1

cin>>P2; // the name of player 2

cout << "\n"; // add a new line

}

do // start of the loop/restart game

{

cout << "shooter make bet" << endl;

cin>>bet; // the money that is being bet

while(cin.fail()) {

cout<< "please enter a integer" <<endl;

cin.clear();

cin.ignore(256,'\n');

cin>>bet;

}

while (cin && bet < 1.00f){ // checks if the money being bet is more than a dollar

cout << "The minimum bet is $1.00 Please place a higher bet." << endl;

}

// declare value

int die1 = rand() % 6 + 1; //[1,6] dice 1

int die2 = rand() % 6 + 1; //[1,6] dice 2

int sum = die1 + die2; // adds both dice to get sum

// write to the file

outFile << "\n";

outFile << "first roll Dices is: " << die1 << " second roll dices is: " << die2 << endl;

outFile << "the point was: " << sum << "\n";

// the logic of the game

if (sum == lucky || sum == lucky2) {

win++; // checks the sum of dice roll 7 or 11

cout << endl; // makes format better looking

cout << P2 << " wins $" << fixed << setprecision(2) << bet << endl;

} else if (sum == 2) {

lost++; // checks if the sum of the eyes roll 2

cout << endl; // makes format better looking

cout << P1 << " wins $" << fixed << setprecision(2) << bet << endl;

} else if (sum == 3) {

lost++; // checks if the sum of the eyes roll 3

cout << endl; // makes format better looking

cout << P1 << " wins $" << fixed << setprecision(2) << bet << endl;

} else if (sum == 12) {

lost++; // checks if the sum of the eyes roll 12

cout << endl; // makes format better looking

cout << P1 << " wins $" << fixed << setprecision(2) << bet << endl;

} else { // checks if the dice roll any other number

bool rollAgn; // make dice roll again if any other number is chosen

do {

die1 = rand() % 6 + 1; //[1,6] dice 1

die2 = rand() % 6 + 1; //[1,6] dice 2

int sumAgn = die1 + die2; // the sum of the dice being roll again

// write to the file

outFile << "\n";

outFile << "dice one roll: " << die1 << " dice two roll: " << die2 << endl;

outFile << "the sum of both dices is: " << sumAgn << endl;

if (sumAgn == 7) {

lost++; // checks if sum of dice if 7

rollAgn = false;

cout << endl; // makes format better looking

cout << P1 << " wins $" << fixed << setprecision(2) << bet << endl;

} else if (sum == sumAgn) {

win++; // checks if the sum and the roll again is the same

rollAgn = false;

cout << endl; // makes format better looking

cout << P2 << " wins $" << fixed << setprecision(2) << bet << endl;

} else rollAgn = true; // makes the dice roll again if the sum of the dice roll any other dice

} while (rollAgn);

}

inFile.close(); // close the file

cout << endl; // makes format better looking

cout << "Do you want to see what the dice roll? Y(es) or N(o)" << endl;

cin>>ans1;

if ((ans1 == 'y') || (ans1 == 'Y')) { // checks the user input

inFile.open("craps.txt"); // opens the file again

if (inFile.is\_open()){ // checks if the file is open

while (getline(inFile, line)) // reads what the file has

{

cout << line << '\n' << endl;

}

}

}

cout << "Another game? Y(es) or N(o)" << endl;

cin >> ans;

} while (ans == 'y' || ans == 'Y'); // restart the game

}

total= win + lost; // amounts of games play

cout << endl; // makes format better looking

cout << "Thanks for playing" << endl;

cout << "Total number of games = " << win + lost << endl;

cout << P2 << " won about " << round(perCov \* (win / total)) << "%" << endl;

cout << P1 << " won about " << round(perCov \* (lost / total)) << "%" << endl;

}

//Close files

outFile.close(); // close the outfile

inFile.close(); // close the infile

//Exit stage right or left!

return 0;

} void dice(int r,int c, int cnt){

//int cnt = 0;

for(r=0; r < 8; r++){

for(c=0; c < 16; c++){

if(r == 0 || r == 7)

cout << "\*";

else if (c == 0 || c == 7 || c == 15)

cout << "\*";

else if (cnt == 3 && (r == 3 && c == 3) )

cout << " \*\* ";

else if ( (r == 4 && c == 3) )

cout << " \*\* ";

else if (r == 4 && c == 10)

cout << " \*\* ";

else if (r == 3 && c == 10)

cout << " \*\* ";

else if (cnt != 3 && cnt !=4)

cout << " ";

}

cnt++;

cout << endl;

}

}bool onep (char &opt1 ){

cout<<opt1<<endl;

if (opt1=='1') {

return true;

} else

return false;

}void math(int die1, int die2, int &sum){

die1 = rand() % 6 + 1; //[1,6] dice 1

die2 = rand() % 6 + 1; // [1,6]

sum=die1+die2;

}

void fillA(int array[],int SIZE){

int die1,die2,sum;

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

math(die1,die2,sum);

array[i]= sum;

}

}void prntAry(int array[],int SIZE,int val){

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

cout<<setw(3)<<array[i]<<" ";//2 digit random number

if (i % val == (val - 1)) cout<<endl;

}

}void bublSrt(int array[], int SIZE) {

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

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

if (array[i] > array[i+1]) {

int temp = array[i];

array[i] = array[i+1];

array[i+1] = temp;

}

}

}

}bool linSrch(int array[],int SIZE ,int val, int &indx, int &indx1){

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

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

indx = i;

indx1= j;

if(array[i]==7 && array[j]==12){

return true;

}

}

}

}

float perC(int array[],int SIZE){

int num=0;

int num1=0;

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

if (array[j]==12){

num1++;

}

}

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

if (array[i]==7 ){

num++;

}

}

return ((num+num1) /(float)SIZE)\*100;

} void secR(vector<int> &sumA, int val, int &sum){

int die1,die2;

for (int j=0; j<sumA.size(); j++){

die1 = rand() % 6 + 1; //[1,6] dice 1

die2 = rand() % 6 + 1; // [1,6]

sum=die1+die2;

sumA[j]=sum;

}

} void printV(vector <int> &sumA,int val,int &sum){

for(int i=0;i<sumA.size();i++){

cout<<setw(3)<<sumA[i]<<" ";//2 digit random number

if (i % val == (val - 1)) cout<<endl;

}

}void selSrt(vector <int> &sumA, int val, int &sum){

for (int pos = 0; pos < sumA.size() - 1; pos++) {

for (int i = pos + 1; i < sumA.size(); i++) {

if (sumA[pos] > sumA[i]) {

int temp = sumA[i];

sumA[i] = sumA[pos];

sumA[pos] = temp;

}

}

}

}float perC2(vector<int> &sumA, int SIZE){

int num=0,die1,die2,sum;

int num1=0;

int Total[SIZE];

math(die1,die2,sum);

Total[SIZE]=sum;

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

math(die1,die2,sum);

if (sumA[j]==sum && sumA[j]!=7 && sumA[j]!=12){

num1++;

}

}return ((num1/(float)SIZE)\*100);

}