Project 2

<Farkle Dice Game>

CIS5
Christian Beebe
20170601

Introduction

Title: Farkle Dice Game

Farkle is played with two or more players on a table with six die. On a player's turn they roll all six die at the same time.
After they roll a player will look at the value of all the die that they have just rolled. If the same number is rolled three or more times then a set value of points is given to the player. The players decide the value of the rolls before the game has started. In my game 3 of a kind is worth 500, 4 of as kind is worth 1000, 5 of a kind is worth 2000 and 6 of a kind is worth 3000. The values can also be determined by the value of the roll itself instead of a flat value. Another way to score is to get a straight, a roll of 1-6. The value of a straight is 1500 due to the fact that it is more common to get a straight than 6 or 5 of a kind. The player will Take turns rolling and adding up their points until one player reaches the target score.When the target score is reached by a player, the round is finished giving any remaining player one more shot at surpassing the current winner. The target score is determined by the players to decide the length of the game.

Summary

Project Size : About 250 Lines
Number of Variables : 6 Declared (Not including bools)

With this attempt at the game of Farkle my main focus was to make it actually feel as if you were taking part in the game. On my last attempt the player had no say in the outcome of the game. The winner was decided off of dice luck. This time around I have implemented a re-roll function into the game. Now the player may select which dice they would like to reroll after the first roll has been made.This is as far as I wish to take this game as it is not particularly complex.

The program took about 2 days to write with most time spent looking at different ways to implement a reroll feature.

The program also helped me strengthen my grasp on using searches and sorts, as well as implementing them into my programs as functions.

Description

The main point of this program is to show proficiency in the course material learned over this semester.

Pseudocode

Set RNG
Start SCreen
Select Target Score
    5000,10000,20000
Prompt the user to press any button
    Roll Dice
    Sort Dice
    Print Dice
    Prompt Re-roll
        If yes to reroll
        Reroll Dice
        Sort Dice
        Print Dice
    Award the player points
    Output the player's current score
Repeat the Program until the target score is reached

Link to Flowchart

https://www.gliffy.com/go/publish/12057959

Check-Off Sheet

Chapter	Section	Topic	Line Number
2	2	cout	55
	3	libraries	9-14
	4	variables/literals	38-44
	5	Identifiers	38-44
	6	Integers	40-42
	7	Characters	43-44
	8	Strings	48
	9	Floats	19
	10	Bools	121-128
	12	Variables 7 Characters or Less	Yes
	14	Arithmetic operators	135
	15	Comments 20%+	Yes
	16	Named Constants	19

3	1	cin	59
	2	Math Expressions	201

	5	Type Casting	201
	7	Formatting Output	197
	8	Strings	48
	9	Math Library	15

4	1	Relational Operators	100-105
	2	if	109
	4	if-else	138
	5	Nesting	275-286
	6	If-else-if	134-172
	8	Logical Operator	90
	11	Validating User Input	90
	13	Conditional Operator	189-190
	14	Switch	99

5	1	Increment/Decrement	108
	2	While	54-66
	5	Do-While	87-90
	6	For Loop	68
	11	Files Input/Output	42-43

6	3	Function Prototypes	23-26
	5	Passing by Value	296
	8	Returning values from functions	213-270
	10	No Global Variables	Yes
	11	Static Local	42-43

	12	Default Arguments	~
	13	Reference Parameters	213
	14	Overloading Functions	~

7	4	Array Initialization	42
	6	Processing Arrays	202-203
	7	Parallel Arrays	~
	8	Arrays as function arguments	293
	9	2-D Arrays	41

8	1	Linear and Binary Search	163-170
	3	Bubble and Selection Sort	273-283

```
/*
 * File:   main.cpp
 * Author: Christian Beebe
 * Created on 20170531
 * Purpose: Farkle Game Project
 */

//System Libraries
#include <iostream> //Input - Output Library
#include <ctime>    //Time for RNG
#include <cstdlib>  //Library for RNG
#include <fstream>  //File I/O
#include <string>   //Strings
#include <iomanip>  //In/Out Manipulation
```

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

//Global Constants

const float PERCENT=100.f; //Percent Conversion

//Function Prototypes

void score (int [],bool &,bool &,bool &,bool &,bool &,bool &,bool &,bool &);
int sort (int []);
int rollDie(int []);
void prntDie(int []);

//Execution begins here

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

   //Set RNG

   srand(static_cast<unsigned int>(time(0)));

   //Declare Variables

   ofstream out;
   unsigned short choice; //User input choice to choose game length
   int winScr;  //Winning Score both players are trying to reach
   int tempScr=0,scores[1][2]={}; //Temporary Score which is added to both scores
   int roll[6]; //Array of the rolls
   char start;  //char input to stop game from finishing too quickly
   char reRoll; //Option for reroll

   //Initialize Variables

   string outName="RollStats.dat"; //Name of Out file
   out.open(outName.c_str());  //Open out file

   //Starting Screen

   cout<<"!!! You are now playing Farkle !!!"<<endl;
   do{
      cout<<"Choose the number of the score you would like to play to."<<endl;
      cout<<"1 :  5000   (Short Game)"<<endl;
      cout<<"2 : 10000  (Medium Game)"<<endl;
```

```
cout<<"3 : 20000    (Long Game)"<<endl;
cin>>choice;
switch(choice){
   case 1: winScr=5000;break;
   case 2: winScr=10000;break;
   case 3: winScr=20000;break;
   default:cout<<"You did not input a correct choice, Try again"<<endl;
}
}while(choice<1||choice>3);
do{
for(int i=1;i<=2;i++){
   tempScr=0;
   cout<<"Player "<<i<<"'s Turn"<<endl;
   cout<<"Input any key to roll"<<endl;
   cin>>start;
   cout<<"You rolled "<<endl;
   rollDie(roll);

//Bubble sorting method to put the array in increasing order

sort(roll);

//Show the results of the Die

prntDie(roll);

//Re-roll

reRoll=0;
do{
   cout<<"Would you like to re-roll any die?    y/n"<<endl;
   cin>>reRoll;
}while(reRoll!='y'&&reRoll!='n');

if(reRoll=='y'){
   cout<<"Select which die you would like to re-roll.   1-6"<<endl;
   cout<<"Or you may press 0 to end your selection."<<endl;
   cout<<"*Press enter after each selection"<<endl;
   int choice;
   do{
   cin>>choice;
      switch(choice){
         case 6:roll[5]=0;break;
```

```
            case 5:roll[4]=0;break;
            case 4:roll[3]=0;break;
            case 3:roll[2]=0;break;
            case 2:roll[1]=0;break;
            case 1:roll[0]=0;break;
        }
    }while(choice!=0);
    for(int t=0;t<=5;t++){
        if(roll[t]==0){
            roll[t]=rand()%6+1;
        }
    }
}

//Print new dice results

sort(roll);
prntDie(roll);

//Scoring Check
bool straight=true;
bool sixsame=true;
bool fivsame=false;
bool forsame=false;
bool thrsame=false;
bool onefive=false;
bool thrpair=true;
bool twotrip=false;

score(roll,straight,sixsame,fivsame,forsame,thrsame,onefive,thrpair,twotrip);

//Award Points

if(straight==true){
    tempScr+=1500;
    cout<<"You rolled a straight.         +1500 Points"<<endl;
}
else if(sixsame==true){
    tempScr+=3000;
    cout<<"You rolled a 6 of a kind       +3000 Points"<<endl;
}
else if(twotrip==true){
    tempScr+=2500;
```

```
    cout<<"You rolled two triples         +2500 Points"<<endl;
}
else if(fivsame==true){
    tempScr+=2000;
    cout<<"You rolled a 5 of a kind       +2000 Points"<<endl;
}
else if(thrpair==true){
    tempScr+=1500;
    cout<<"You rolled three 2 pairs       +1500 Points"<<endl;
}
else if(forsame==true){
    tempScr+=1000;
    cout<<"You rolled a 4 of a kind       +1000 Points"<<endl;
}
else if(thrsame==true){
    tempScr+=500;
    cout<<"You rolled a 3 of a kind        +500 Points"<<endl;
}
else if(onefive==true){
        for(int n=0;n<6;n++){
            if(roll[n]==1){
                tempScr+=100;
        }
            else if(roll[n]==5){
                tempScr+=50;
        }
    }
    cout<<"You rolled a single 1's or 5's  +"<<tempScr<<" Points"<<endl;
}
//Show Current Scores of the Players

if(i%2==0){
    scores[0][1]+=tempScr;
    cout<<"Current Score is "<<scores[0][1]<<endl<<endl;
}
else{
    scores[0][0]+=tempScr;
    cout<<"Current Score is "<<scores[0][0]<<endl<<endl;
}

}
}while(scores[0][0]<winScr&&scores[0][1]<winScr);
```

```
    //Result Screen

    (scores[0][0]>scores[0][1])?cout<<"Player 1 Wins!    Congratulations!"<<endl:
        cout<<"Player 2 Wins!   Congratulations"<<endl;
    cout<<"Final Score was :"<<endl;
    cout<<"Player 1 : "<<scores[0][0]<<" points"<<endl;
    cout<<"Player 2 : "<<scores[0][1]<<" points"<<endl;

    //Output round results to a file

    out<<fixed<<setprecision(2)<<showpoint;
    out<<"Player 1 had a total score of "<<scores[0][0]<<" points."<<endl;
    out<<"Player 2 had a total score of "<<scores[0][1]<<" points."<<endl;
    out<<"There was a total of "<<scores[0][0]+scores[0][1]<<" points."<<endl;
    out<<"Player 1 had
"<<static_cast<float>(scores[0][0])/static_cast<float>(scores[0][0]+scores[0][1])*PERCENT<<"%
of the points"<<endl;
    out<<"Player 2 had
"<<static_cast<float>(scores[0][1])/static_cast<float>(scores[0][0]+scores[0][1])*PERCENT<<"%
of the points"<<endl;

    //Close Files

    out.close();

    //Exit stage right!
 return 0;
}


void score (int roll[],bool &straight,bool &sixsame,bool &fivsame,bool &forsame,bool &thrsame,
bool &onefive,bool &thrpair,bool &twotrip){
    //[1,2,3,4,5,6]
    for(int n=0;n<5;n++){
        if(roll[n]+1!=roll[n+1]){
            straight=false;
        }
    }

    //[x,x,x,x,x,x]
    for(int n=0;n<5;n++){
        if(roll[n]!=roll[n+1]){
            sixsame=false;
```

```
    }
}

//[x,x,x,x,x,z] or [z,x,x,x,x,x]
for(int n=0;n<2;n++){
    if(roll[n]==roll[n+1]&&roll[n+1]==roll[n+2]&&roll[n+2]==roll[n+3]&&roll[n+3]==roll[n+4]){
        fivsame=true;
    }
}

//[z,x,x,x,x,y] or [x,x,x,x,z,y] or [z,y,x,x,x,x]
for(int n=0;n<3;n++){
    if(roll[n]==roll[n+3]&&roll[n+1]==roll[n+2]&&roll[n+2]==roll[n+3]){
        forsame=true;
    }
}

//[z,x,x,x,w,y][x,x,x,z,w,y][z,w,x,x,x,y][z,y,w,x,x,x]
for(int n=0;n<4;n++){
    if(roll[n]==roll[n+1]&&roll[n+1]==roll[n+2]){
        thrsame=true;
    }
}

//Any 1's or 5's (1=100, 5=50)
for(int n=0;n<6;n++){
    if(roll[n]==1){
        onefive=true;
    }
    else if(roll[n]==5){
        onefive=true;
    }
}

//[x,x,y,y,z,z]
for(int n=0;n<5;n=n+2){
    if(roll[n]!=roll[n+1]){
        thrpair=false;
    }
}

//[x,x,x,y,y,y]
if(roll[0]==roll[2]&&roll[3]==roll[5]){
```

```
        twotrip=true;
    }
}


int sort (int r[]){
    int check=1;
    for(int x=1;x<=6&&check;x++){
        check=0;
        for(int t=0;t<5;t++){
            if(r[t+1]<r[t]){
                int holding=r[t];
                r[t]=r[t+1];
                r[t+1]=holding;
                check=1;
            }
        }
    }
}


int rollDie(int r[]){
    for(int d=0;d<6;d++){
        r[d]=rand()%6+1;
    }
}


void prntDie(int r[]){
    for(int y=0;y<=5;y++){
        cout<<r[y]<<" ";
    }
    cout<<endl;
}
```