Project 1

<Farkle Dice Game>

CIS5
Christian Beebe
20170416

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 180 Lines
Number of Variables : About 23

With this project I tried to make the game as user friendly as possible. I spent time trying to give each roll its own value before deciding to read about arrays and learned that it is much easier to work with the value of each rolles when they are within the same variable. One thing I would like to come back and change when I have more experience would be to implement functions into the program to clean up the code and make it easier to read and also to allow the players to reroll in some instances.

The program took about 3 days to write with most time spent learning about arrays and how it include them into the program.

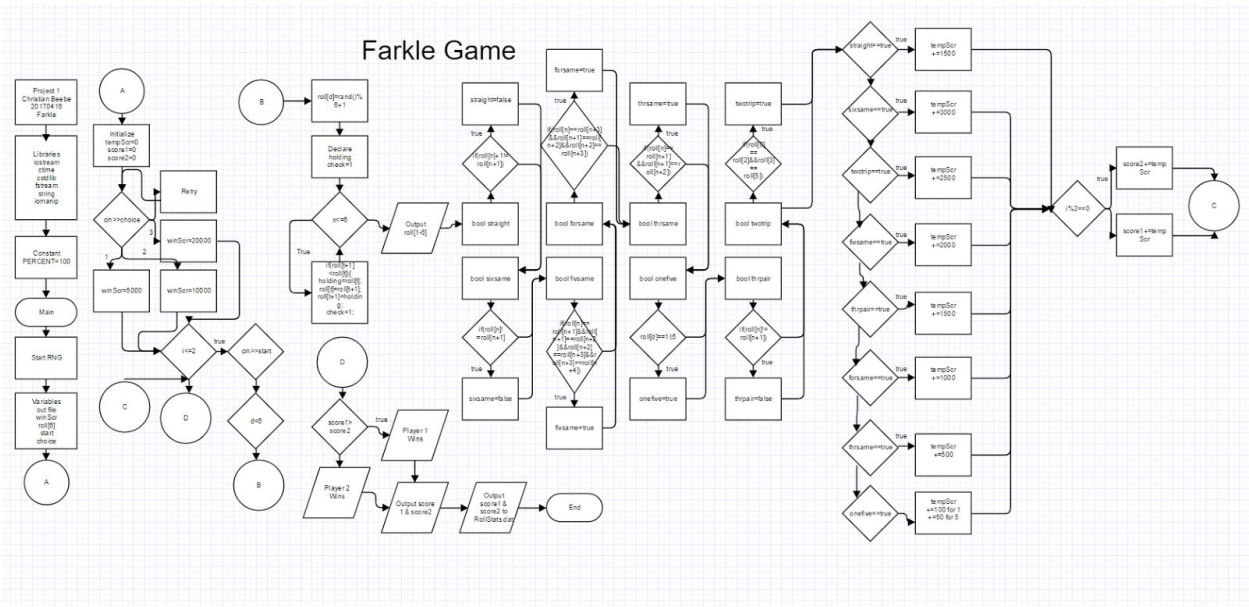
The program also helped me learn how to include do-while loops into my programs. My grasp on the command was not as strong as I wanted it to be before writing the program.

Description

The main point of this program is to show a grasp on already learned concepts while also learning about and including new concepts.

Pseudocode

Start Menu
Select Target Score
    5000,10000,20000
Prompt the user to press any button
    When a button is pressed begin the roll
    Format the Array to be in incremental order
Check the array for combinations of numbers
    Award the player points
Output the player's current score
    Repeat the Program until the target score is reached



Check-Off Sheet

Chapter	Section	Topic	Line Number
2	2	cout	36
	3	libraries	11-13
	4	variables/literals	29-33
	5	Identifiers	29-33
	6	Integers	30-32
	7	Characters	33
	8	Strings	42
	9	Floats	22
	10	Bools	83-135
	12	Variables 7 Characters or Less	29-33
	14	Arithmetic operators	140-174

	15	Comments 20%+	Throughout
	16	Named Constants	22

3	1	cin	42
	2	Math Expressions	211-216
	5	Type Casting	26
	7	Formatting Output	210
	8	Strings	42
	9	Math Library	~

4	1	Relational Operators	55-57
	2	if	83-137
	4	if-else	180-186
	5	Nesting	51-77
	6	If-else-if	140-176
	8	Logical Operator	49
	11	Validating User Input	47-48
	13	Conditional Operator	192-193
	14	Switch	43

5	1	Increment/Decrement	66
	2	While	
	5	Do-While	37-49
	6	For Loop	51
	11	Files Input/Output	42-43

```
/*
 * File:   main.cpp
 * Author: Christian Beebe
 * Created on 20170416
 * 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 - None

//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,score1=0,score2=0; //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

    //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;
      for(int d=0;d<6;d++){
         roll[d]=rand()%6+1;

         }

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

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

```
//Show the Results of the Die
cout<<roll[0]<<" "<<roll[1]<<" "<<roll[2]<<" "<<roll[3]<<" "<<roll[4]<<" ";
cout<<roll[5]<<endl;

//Scoring Check
bool straight=true;  //[1,2,3,4,5,6]
   for(int n=0;n<5;n++){
      if(roll[n]+1!=roll[n+1]){
         straight=false;
      }
   }

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

bool fivsame=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;
      }
   }

bool forsame=false; //[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;
      }
   }

bool thrsame=false; //[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;
      }
   }

bool onefive=false;  //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;
        }
    }

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

bool twotrip=false; //[x,x,x,y,y,y]
    if(roll[0]==roll[2]&&roll[3]==roll[5]){
        twotrip=true;
    }
//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){
           score2+=tempScr;
           cout<<"Current Score is "<<score2<<endl<<endl;
        }
        else{
           score1+=tempScr;
           cout<<"Current Score is "<<score1<<endl<<endl;
        }

        }
        }while(score1<winScr&&score2<winScr);
     //Result Screen
     (score1>score2)?cout<<"Player 1 Wins!    Congratulations!"<<endl:
        cout<<"Player 2 Wins!   Congratulations"<<endl;
     cout<<"Final Score was :"<<endl;
     cout<<"Player 1 : "<<score1<<" points"<<endl;
     cout<<"Player 2 : "<<score2<<" points"<<endl;

     //Output round results to a file
     out<<fixed<<setprecision(2)<<showpoint;
     out<<"Player 1 had a total score of "<<score1<<" points."<<endl;
     out<<"Player 2 had a total score of "<<score2<<" points."<<endl;
     out<<"There was a total of "<<score1+score2<<" points."<<endl;
     out<<"Player 1 had
"<<static_cast<float>(score1)/static_cast<float>(score1+score2)*PERCENT<<"% of the
points"<<endl;
```

```
    out<<"Player 2 had
"<<static_cast<float>(score2)/static_cast<float>(score1+score2)*PERCENT<<"% of the
points"<<endl;
    //Close Files
    out.close();
    //Exit stage right!
 return 0;
}
```