# Farkle

# Dice Game

Adam Wayman

CIS-5

Fall 2017

49006

**Introduction:**
Farkle is a dice game where two or more players roll dice to attain a score. The goal is to get 10,000 points as a score overall to win.

# How to play the game

## Object of game:

## Reach 10,000 score

## Rules:

Farkle is a two+ game.

1.Each player rolls all 6 dice per turn and receives a score.

2.Keep rolling until a player gets 10,000.

3. There are certain combinations of dice rolls that earn points such as triplets, four of a kinds, six of a kind.
4. 5's = 50 point

   1's = 100 points

   1,1,1 = 300 points

   2,2,2 = 200 points

3,3,3 = 300 points

4,4,4 = 400 points

5,5,5 = 500 points

6,6,6 = 600 points

Four of a Kind = 1,000 points

Five of a Kind = 2,000 points

Six of a Kind = 3,000 points

A Straight of 1-6 = 1,500 points

Three Pairs = 1,500 points

Four of a Kind + a Pair = 1,500

Two sets of Three of a Kind = 2,500

## How I coded the game:

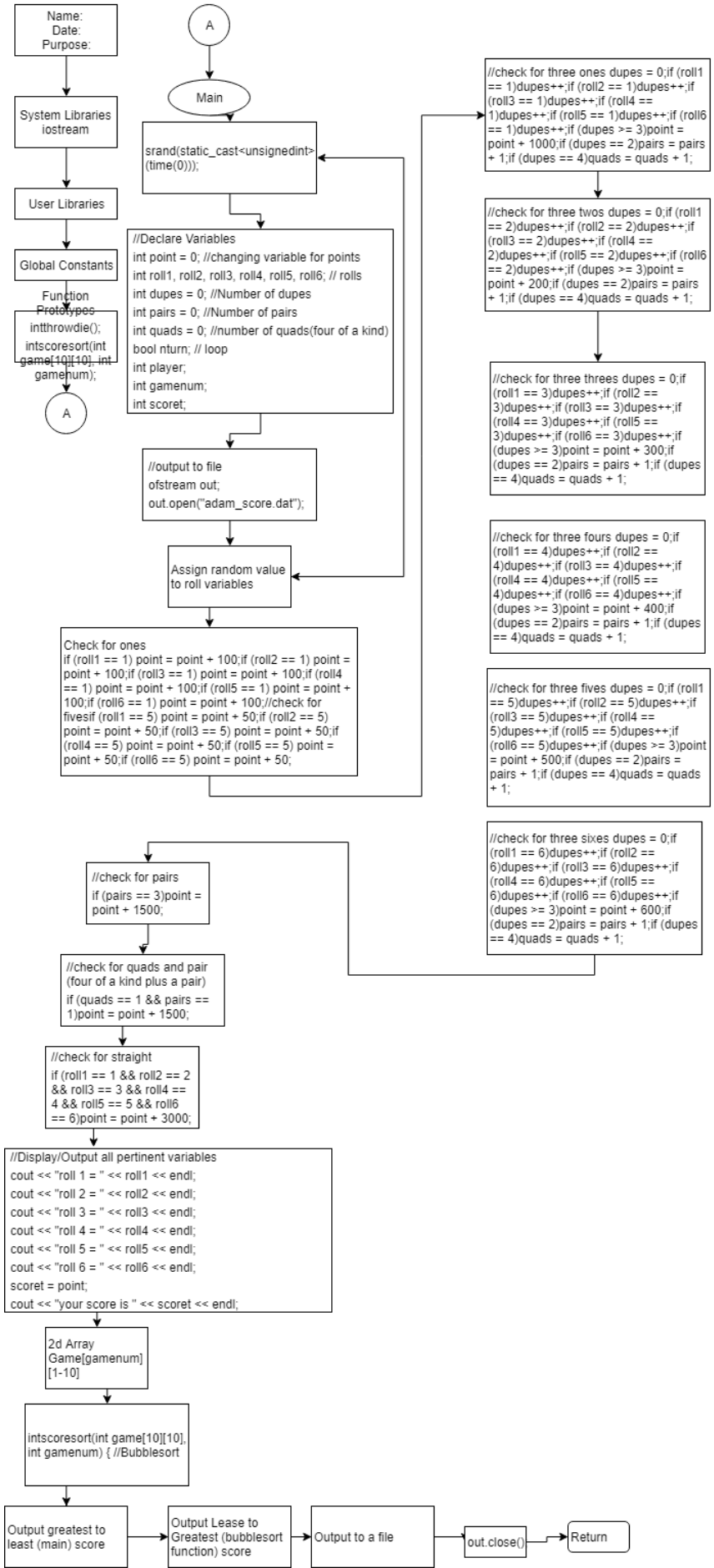
First I should say I'm not very confident in my ability to write or understand code/concepts.

First I needed to make a random number generator to simulate a dice roll. Then I needed a way to extract those dice rolls into if statements

to check if any of the combinations of dice rolls matched the combinations that earned the player points. Adding to the changing variable points was confusing to me when I started to overthink it bu t after some struggle and many hours I realized that it was best for me to make if statements and add to the points variable continuously until all the combinations were checked. I then demonstrated an array when I took all the rolls and score and put them into a function to be sorted.(Bubblesort)

## Flow Chart

# Farkle Flowchart



# References


1. Dr. Lehr's Lectures
2. "Starting Out with C++: From Control Structures through Objects" Gaddis, Tony. 8th Edition.
3. www.cplusplus.com


# Program

```
/*
    * File: main.cpp
    * Author: Adam Wayman
    * Created on Dec 9th, 2017, 11:48 AM
    * Purpose: Final Class Project Version 2 Bubble sort + Output to file
    */

    //System Libraries
    #include <iostream>//Input/Output Stream Library
    #include <cstdlib>//Random number
    #include <iomanip>
    #include <ctime>//time
    #include <fstream> //File I/O
    #include <cstring>

    using namespace std; //Standard Name-space under which System Libraries reside

    //User Libraries

    //Global Constants - Not variables only Math/Science/Conversion constants

    //Function Prototypes
    int throwdie();
    int scoresort(int game[10][10], int gamenum);
```

```
//Execution Begins Here!

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

//Declare Variables
int point = 0; //changing variable for points
int roll1, roll2, roll3, roll4, roll5, roll6; // rolls
int dupes = 0; //Number of dupes
int pairs = 0; //Number of pairs
int quads = 0; //number of quads(four of a kind)
bool nturn; // loop
int player;
int gamenum;
int scoret;
//output to file
ofstream out;
out.open("adam_score.dat");

//Array
int game[10][10];

//Initialize Variables/rolls
cout << "Which player number are you?" << endl;
cin>>player;
gamenum = 0;
while (point < 10000) {
//filling array
gamenum = gamenum + 1; //adds 1 to game number each loop
game[gamenum][0] = gamenum; // Game number
game[gamenum][1] = player; //player number

//rolls
srand(static_cast<unsigned int> (time(0)));
roll1 = rand() % 6 + 1; //[1,6]
roll2 = rand() % 6 + 1; //[1,6]
roll3 = rand() % 6 + 1; //[1,6]
roll4 = rand() % 6 + 1; //[1,6]
roll5 = rand() % 6 + 1; //[1,6]
roll6 = rand() % 6 + 1; //[1,6]
//Scoring the rolls
//check for ones
```

```
if (roll1 == 1)
point = point + 100;
if (roll2 == 1)
point = point + 100;
if (roll3 == 1)
point = point + 100;
if (roll4 == 1)
point = point + 100;
if (roll5 == 1)
point = point + 100;
if (roll6 == 1)
point = point + 100;
//check for fives
if (roll1 == 5)
point = point + 50;
if (roll2 == 5)
point = point + 50;
if (roll3 == 5)
point = point + 50;
if (roll4 == 5)
point = point + 50;
if (roll5 == 5)
point = point + 50;
if (roll6 == 5)
point = point + 50;
//check for three ones
dupes = 0;
if (roll1 == 1)dupes++;
if (roll2 == 1)dupes++;
if (roll3 == 1)dupes++;
if (roll4 == 1)dupes++;
if (roll5 == 1)dupes++;
if (roll6 == 1)dupes++;
if (dupes >= 3)point = point + 1000;
if (dupes == 2)pairs = pairs + 1;
if (dupes == 4)quads = quads + 1;
//check for three twos
dupes = 0;
```

```
if (roll1 == 2)dupes++;
if (roll2 == 2)dupes++;
if (roll3 == 2)dupes++;
if (roll4 == 2)dupes++;
if (roll5 == 2)dupes++;
if (roll6 == 2)dupes++;
if (dupes >= 3)point = point + 200;
if (dupes == 2)pairs = pairs + 1;
if (dupes == 4)quads = quads + 1;
//check for three threes
dupes = 0;
if (roll1 == 3)dupes++;
if (roll2 == 3)dupes++;
if (roll3 == 3)dupes++;
if (roll4 == 3)dupes++;
if (roll5 == 3)dupes++;
if (roll6 == 3)dupes++;
if (dupes >= 3)point = point + 300;
if (dupes == 2)pairs = pairs + 1;
if (dupes == 4)quads = quads + 1;
//check for three fours
dupes = 0;
if (roll1 == 4)dupes++;
if (roll2 == 4)dupes++;
if (roll3 == 4)dupes++;
if (roll4 == 4)dupes++;
if (roll5 == 4)dupes++;
if (roll6 == 4)dupes++;
if (dupes >= 3)point = point + 400;
if (dupes == 2)pairs = pairs + 1;
if (dupes == 4)quads = quads + 1;
//check for three fives
dupes = 0;
if (roll1 == 5)dupes++;
if (roll2 == 5)dupes++;
if (roll3 == 5)dupes++;
if (roll4 == 5)dupes++;
if (roll5 == 5)dupes++;
```

```
if (roll6 == 5)dupes++;
if (dupes >= 3)point = point + 500;
if (dupes == 2)pairs = pairs + 1;
if (dupes == 4)quads = quads + 1;
//check for three sixes
dupes = 0;
if (roll1 == 6)dupes++;
if (roll2 == 6)dupes++;
if (roll3 == 6)dupes++;
if (roll4 == 6)dupes++;
if (roll5 == 6)dupes++;
if (roll6 == 6)dupes++;
if (dupes >= 3)point = point + 600;
if (dupes == 2)pairs = pairs + 1;
if (dupes == 4)quads = quads + 1;
//check for pairs
if (pairs == 3)point = point + 1500;
//check for quads and pair (four of a kind plus a pair)
if (quads == 1 && pairs == 1)point = point + 1500;
//check for straight
if (roll1 == 1 && roll2 == 2 && roll3 == 3 && roll4 == 4 && roll5 == 5 && roll6 == 6)point =
point + 3000;
//Display/Output all pertinent variables
cout << "roll 1 = " << roll1 << endl;
cout << "roll 2 = " << roll2 << endl;
cout << "roll 3 = " << roll3 << endl;
cout << "roll 4 = " << roll4 << endl;
cout << "roll 5 = " << roll5 << endl;
cout << "roll 6 = " << roll6 << endl;
scoret = point;
cout << "your score is " << scoret << endl;
//array
game[gamenum][2] = roll1;
game[gamenum][3] = roll2;
game[gamenum][4] = roll3;
game[gamenum][5] = roll4;
game[gamenum][6] = roll5;
game[gamenum][7] = roll6;
```

```
game[gamenum][8] = scoret;
//output to file
out << roll1 << " " << roll2 << " " << roll3 << " " << roll4 << " " << roll5 << " " << roll6 <<
" " << scoret << endl;
cout << "Would you like to go again?(1 for yes, 0 for no) " << endl;
cin>>nturn;
if (nturn == false)break;
}
int i, j; //loop
for (i = 1; i <= gamenum; i++) {
cout << "Game Number " << game[i][0] << " ";
cout << "Player Number " << game[i][1] << " ";
cout << "Roll 1 Score = " << game[i][2] << " ";
cout << "Roll 2 Score = " << game[i][3] << " ";
cout << "Roll 3 Score = " << game[i][4] << " ";
cout << "Roll 4 Score = " << game[i][5] << " ";
cout << "Roll 5 Score = " << game[i][6] << " ";
cout << "Roll 6 Score = " << game[i][7] << " ";
cout << "Game Score = " << game[i][8] << " ";
cout << endl;
}
cout << endl;
out.close();
//sending array to sort
scoresort(game, gamenum);
//Exit the program
return 0;
}
int throwdie() {
srand(static_cast<unsigned int> (time(0)));
return (rand() % 6 + 1);
}
int scoresort(int game[10][10], int gamenum) { //Bubblesort
int i, j, k, m, n, o; //loop
int swap[10];
m = gamenum;
```

```
n = 8;
for (k = 1; o < m; o++) {
for (k = 1; k < m; k++) {
if (game[k][8] < game[k + 1][8]) {
for (i = 0; i <= n; i++)
swap[i] = game[k][i];
for (i = 0; i <= n; i++)
game[k][i] = game[k + 1][i];
for (i = 0; i <= n; i++)
game[k + 1][i] = swap[i];
}
}
}
cout << endl;
for (i = 1; i <= gamenum; i++) {
cout << "Game Number " << game[i][0] << " ";
cout << "Player Number " << game[i][1] << " ";
cout << "Roll 1 Score = " << game[i][2] << " ";
cout << "Roll 2 Score = " << game[i][3] << " ";
cout << "Roll 3 Score = " << game[i][4] << " ";
cout << "Roll 4 Score = " << game[i][5] << " ";
cout << "Roll 5 Score = " << game[i][6] << " ";
cout << "Roll 6 Score = " << game[i][7] << " ";
cout << "Game Score = " << game[i][8] << " ";
cout << endl;
}
//output to file
ofstream out;
out.open("adam_score.dat");
for (i = 1; i <= gamenum; i++) {
out << "Game Number " << game[i][0] << " ";
out << "Player Number " << game[i][1] << " ";
out << "Roll 1 Score = " << game[i][2] << " ";
out << "Roll 2 Score = " << game[i][3] << " ";
out << "Roll 3 Score = " << game[i][4] << " ";
out << "Roll 4 Score = " << game[i][5] << " ";
```

```
out << "Roll 5 Score = " << game[i][6] << " ";
out << "Roll 6 Score = " << game[i][7] << " ";
out << "Game Score = " << game[i][8] << " ";
out << endl;
}
out.close();
return gamenum;
}
```