Project 1

<Roulette Game>

CSC-5

Name: Kevin Rivera

Date: 11/05/17

Introduction

Title: Roulette

Roulette is a casino game where players choose to place bets on a single number, various grouping of numbers, the colors red or black, whether the number is odd or even, or if the numbers are high (19-36) or low (1-18). To determine a winning number and color, the wheel is spun on a spinning platform with wedges for the roulette ball to land in. The ball is to eventually land on a number of a certain color (black or red) to determine the outcome.

The betting system on this game is simple. If a player bets on a single number—also known as straight bet—the player, then has a chance for a 35/1 payoff. The other bets only add the amount they betted to the dealer. Say if the player bets $35 on odd numbers, but lands on even, you lose the bet. However, if the number were to land on an odd number, they earn $35. Vice versa for even numbers. The game ends when the player decides to step out.

Summary

Project size: ~120 lines

Number of variables: 5-10

This program consists of if and else-if statements to determine whether if the player has won the game of roulette. The program runs off a random number generator to find a winning number, or to check if a number is even or odd. The game is a simple concept, yet much can be added to make a more complete game of roulette. In this instance, the program does not have group of numbers, colors, or high or low numbers.

While making the program, I came across multiple programs having to do with reading the inputs and having the program run through the if and else-if statements correctly.

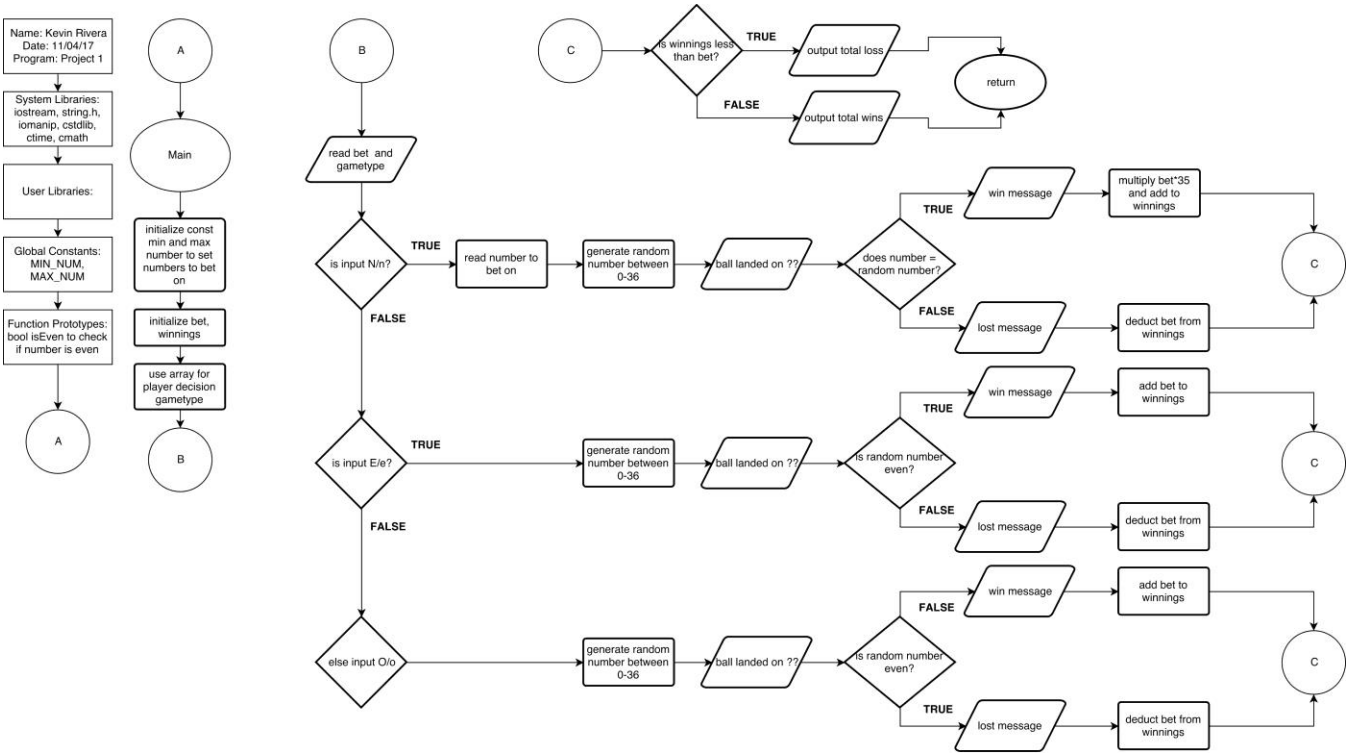
I had to reference to some concepts we did not cover in class such as:

!strcmp – this was to compare the char inputs as I did not use strings using the == operator.

Description

The purpose of the program is to provide a game and betting options for the player. This works by checking how much a player wants to bet and what type of bet they would like to place.

Flowchart



Pseudocode

Initialize

Input bet

Input bet type

    If player chooses on specific number

        If number generated equals number inputted

            Player wins 35/1 payoff

        Else

            Player loses bet

    If player chooses odd bet

        If number is odd

            Player wins bet

        Else

            Player loses bet

    If player chooses even bet

        If number is odd

            Player wins bet

        Else

            Player loses bet

Show results of earnings

Program end

Major Variables

Type	Variable Name	Description	Location
Integer	MIN_NUM	Lowest number to be generated	main()
	MAX_NUM	Highest number to be generated	main()
	num	Number input for number bet	main()
	rng	Randon number generated	main()
Float	bet	Bet inputted	main()
	wins	Total earned/lost	main()
Character	gamet	Game type chosen	main()
Bool	isEven	Check if number is even	main()

C++ Constructs

Chapter	Section	Topic	Line number
2	2	cout	30-31
	3	libraries	iostream, iomanip, cmath, cstdlib, fstream, string, ctime
	4	variables/literals	24
	5	Identifiers	9, 20, 21, 22, 24, 27
	6	Integers	20-22
	7	Characters	27
	8	Strings	38, 63,90 (!strcmp string compare)
	9	Floats No Doubles	24
	10	Bools	9
	12	Variables 7 characters or less	9, 20, 21, 22, 24, 27
	16	Named Constants	21
	17	Programming Style ***** Emulate	
3	1	cin	35, 39, 45,
	2	Math Expression	12, 49, 72, 96, 116
	8	Strings	38, 63,90 (!strcmp string compare)
	9	Math Library	6
4	1	Relational Operators	12, 54, 75, 115
	2	if	12, 42, 52, 68, 75, 78, 92, 100, 113
	4	If-else	14
	5	Nesting	12, 14,
	8	Logical operators	12, 54, 75, 115
5	1	Increment/Decrement	57, 63, 81, 87, 103, 109,

Program

```
#include <iostream>
#include <string.h>
#include <iomanip>
#include <cstdlib>
#include <ctime>
#include <cmath>
using namespace std;

// bool to check if number is even
bool isEven(int n)
{
   if (n%2==0)
      return true;
   else
      return false;
}

int main()
{
   // constant for the random numbers to be generated
   int const MIN_NUM = 0, MAX_NUM = 36;
   int num;    // number for input for number gametype
   int rng;    // random number to be generated

   float bet, wins = 0; // bet inputted, total earned/lost

   // array for player decisions (number, even, odd)
   char gamet[3];


   // display main menu
   cout << "Welcome to Roulette!\n\n";
                                          // balance //
   // player inputs their money for bet
```

```
cout << "How much would you like to bet?\n$";
cin >> bet;

// player chooses gametype
cout << "Betting on a specific number (N), odd numbers(O), or even numbers(E)? ";
cin >> gamet;

// run this if player chose inputted N or n
if(!strcmp(gamet,"N")||(!strcmp(gamet,"n")))
{
   cout << "What number would you like to bet on? ";   // what number to bet on
   cin >> num;

   // random number generator between 0-36
   srand(time(NULL));
   rng = rand() % (MAX_NUM - MIN_NUM + 1) + MIN_NUM;

   cout << "The ball landed on " << rng << "\n";

// loss
if(num != rng)
{
   cout << "You lose $" << bet << "\n";
   wins -= bet;    // add -bet to winnings
}
// win
else
{
   cout << "You win $" << 35*bet << endl;
   wins += 35*bet; // multiply bet times 35 and add to winnings
}
}
```

```
// inputs E or e for gametype
if((!strcmp(gamet,"E")||(!strcmp(gamet,"e"))))
{
   // random number generator between 1-36
   srand(time(NULL));
   rng = rand() % (MAX_NUM - MIN_NUM + 1) + MIN_NUM;
   cout << "The ball landed on " << rng << endl;

   if(gamet == "E"||"e")
   {
      // win
      if(isEven(rng))
      {
         cout << "You win $" << bet << endl;
         wins += bet; // add bet to winnings
      }
      // loss
      else
      {
         cout << "You lose $" << bet << endl;
         wins -= bet; // add -bet to winnings
      }
   }
}

// input O or o for gametype
if((!strcmp(gamet,"O")||(!strcmp(gamet,"o"))))
{
   // random number generator between 1-36
   srand(time(NULL));
   rng = rand() % (MAX_NUM - MIN_NUM + 1) + MIN_NUM;
   cout << "The ball landed on " << rng << endl;

      // loss
      if(isEven(rng)) // check if random number is even
```

```
        {   // if even, then lose because betted on odd
            cout << "You lost $" << bet << endl;
            wins -= bet;    // add bet to winnings
        }
        // win
        else
        {
            cout << "You win $" << bet << endl;
            wins += bet;    // add -bet to winnings
        }
    }


    // final results
    if(wins < bet){
        cout << "You lost a total of $" << abs(wins); // absolute value used
    }                                       // to not show negative
    else
        cout << "You won a total of $" << wins;
    return 0;
}
```