**Roulette**

Project 2

CSC-5 Intro C++

Shen Jin

31 July 2014

**1 - Introduction**

**Rules and Gameplay:**

Project 2 is rather a modification of my Project 1. The game rules are exactly the same: ,players will have a menu to let them decide how they would like to place the bet. If they want to bet on a number in between 1 and 36, they will be paid 36 times their bet if they win. If they want to bet on a color—either black or red, or if they want to bet whether the number will be even or odd, they will be paid twice as much as the bet if they win. Other than those options, they can also choose to bet on a dozen of number sorted by range, and if they win, they will be paid 3 times as much as their bet.

**2 - Development**

**Approached Strategy:**

Since the most recent concepts we talked about are functions and arrays, the most important thing is to use them. I actually used one function with parameters in the Project 1, but at the time I barely understood what functions were in C++. This time I used functions in different ways and multiple times. To me, function is a concept that is once understood and used a lot by me. Sometimes the functions do not necessarily shorten the length of my entire code, however, they shorten the main function, which made it easier for me to go back and forth in between lines—it is more clear to see.

Array, on the other hand, gave me quite a bit of challenge. I heard many people in our class say that array is easy to them. That is just not the case to me. There are times I feel like I do not yet have enough knowledge to utilize arrays the way I prefer to. Then one day we learned vectors in class, which is awesome!

**Research:**

I suppose up to this day, I no longer need to call it a research even if vector is not required for this project, but I have to say it is very useful that I could not just leave it on the side. Vector helped me do my 2-dimensinal array in the way I wanted to display. For instance,

void display (vector<char>&, vector<float>&, vector<float>&);

for(int i=0;i<v1.size();i++){

cv1[i][0]=v1[i];

cv23[i][0]=v2[i];

cv23[i][1]=v3[i];

}

cout << "Game Type" << setw(10) << "Spin" << setw(12) << "Amoumt" << endl;

for(int i=0;i<v1.size();i++){

cout<< setw(5) << cv1[i][0]

<< setw(13) << static\_cast<int>(cv23[i][0])

<< setw(8) << "$" << cv23[i][1]<<endl;

}

**3 - Pseudo Code**

*This is a program of Roulette*

*Introduction of the game rules*

*Input game rules/ read file*

*Main function{*

*Declare Variables*

*Type of bet chosen by the player*

*How many times the player wants it to spin*

*The number bet*

*The actual number off the spins*

*The player bet on even or odd number*

*The color that gives off by the spin*

*The color of the number bet*

*The range of the number bet*

*The original total that the player has*

*The amount bet*

*The amount the player have after game*

*If the player wants to play again*

*For validation purpose*

*Vecotors/Arrays for 2-D output*

*Set the random number seed*

*Would you like to play again/Input Validation*

*Spin the wheel for times defined by user*

*Menu*

*Input validation for menu*

*Determine the random*

*Start of switch case*

*Case A*

*Call question function*

*Put the value into the vector(push\_back)*

*Get user input*

*Input Validation*

*Output the result*

*Call win/lose function for caseA*

*Put amount into vector*

*Case B*

*Call question function*

*Put the value into the vector(push\_back)*

*Get user input*

*Input Validation*

*Determine a random color 1=black, 0=red*

*Output the result*

*Call win/lose function for caseB*

*Put amount into vector*

*Case C*

*Call question function*

*Put the value into the vector(push\_back)*

*Get user input*

*Input Validation*

*Call win/lose function for caseC*

*Put amount into vector*

*Case D*

*Call question function*

*Put the value into the vector(push\_back)*

*Get user input*

*Input Validation*

*Output the result*

*Call win/lose function for caseD*

*Put amount into vector*

*End of switch case*

*End of for loop*

*End of do-while loop*

*Call display function*

*}*

*Void question function{*

*Get user input at the beginning of each game*

*Input validation*

*}*

*Void win/lose case A function{*

*Calculation for win*

*Calculation for lose*

*}*

*Void win/lose case B function{*

*Calculation for win*

*Calculation for lose*

*}*

*Void win/lose case C function{*

*Calculation for win*

*Calculation for lose*

*}*

*Void win/lose case D function{*

*Calculation for win*

*Calculation for lose*

*}*

*Void display function{*

*Set up rows and columns for 2-D array display*

*V2 + V3*

*Variables for swap purposes*

*Display and formatting*

*Static\_cast cv23[i][0] to display “spin” without decimal places*

*}*

**4 – Checklist**

1. Primitive Data Types: line 49. Declare variables

bool

char

int

float

string

unsigned short

2. System Level Libraries

<cstdlib>

<iostream>

<cmath>

<iomanip>

<string>

<fstream>

<vector>

3.Operators

+ , -, /, \*, %(for random and also for remainder), =,

&&, ||, <= , >=, ==, !=

++

4. Conditonals

do while loop : Line 71

for loop: Line 75

if: Line 96

else if: Line248

else: Line 248

5.Files: line 37

Read files for game rules

6.Menu: line 87

Switch cases

7. Functions: Line 217

8.Array and Vector: Line 315

**5 – Code**

*/\**

*\* File: Project\_2\_Roulette*

*\* Author: Shen Jin*

*\* Created on July 26, 2014, 4:02 PM*

*\*/*

*//System Level Libraries*

**#include <cstdlib>**

**#include <iostream>**

**#include <cmath>**

**#include <iomanip>**

**#include <string>**

**#include <fstream>**

**#include <vector>**

**using** **namespace** std;

*//User Defined Libraries*

*//Global Constants*

*//Function Prototypes*

**void** **question**(**float** **&**bank, **float** **&**bet, **bool** **&**invalid);

**void** **winLosA** (**int** numBet, **int** num, **float** **&**amnt, **float** **&**bank, **float** bet, vector**<float>&**);

**void** **winLosB** (**bool** co, string color, **float** **&**amnt, **float** **&**bank, **float** bet, vector**<float>&**);

**void** **winLosC** (string choice, **int** num, **float** **&**amnt, **float** **&**bank, **float** bet, vector**<float>&**);

**void** **winLosD** (string range, **int** num, **float** **&**amnt, **float** **&**bank, **float** bet, vector**<float>&**);

**void** **display** (vector**<char>&**, vector**<float>&**, vector**<float>&**);

*//Execution Begins Here!*

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

*//This is a program of Roulette*

*//Introduction of the game rules*

    cout **<<** "This program is a simplified roulette game."**<<** endl;

    cout **<<** "Game Rules:";

*//Input game rules/ read file*

    ifstream myFile;

    myFile.open("GameRules.txt");

    string line;

    string file\_contents;

**while**(getline(myFile, line)){

     file\_contents **+=** line;

     file\_contents.push\_back('\n');

     cout **<<** line **<<** endl;

    }

    cout **<<** endl **<<** endl;

    myFile.close();

*//Declare Variables*

**char** type; *//Type of bet chosen by the player*

**unsigned** **short** spin; *//How many times the player wants it to spin*

**unsigned** **short** numBet; *//The number bet*

**unsigned** **short** num; *//The actual number off the spins*

    string choice; *//The player bet on even or odd number*

**bool** co; *//The color that gives off by the spin*

    string color; *//The color of the number bet*

    string range; *//The range of the number bet*

**float** bank; *//The original total that the player has*

**float** bet; *//The amount bet*

**float** amnt; *//The amount the player have after game*

**char** again; *//If the player wants to play again*

**bool** invalid; *//For validation purpose*

    vector**<char>** v1; *//Vectors and arrays for 2-D output*

    vector**<float>** v2,v3;

**int** turn**=**1;

*//Set the random number seed*

    srand(**static\_cast<unsigned** **short>**(time(0)));

*//Determine the random number*

    num**=**rand()**%**36**+**1;

**do**{ *//Would you like to play again/Input Validation*

        cout **<<** "How many times would you like to spin?" **<<** endl;

        cin **>>** spin;

**for**(**int** i**=**0; i**<**spin; i**++**){ *//Spin the wheel for times defined by user*

**do**{

                invalid**=**false;

*//Menu*

                cout **<<** "Type A if you want to bet on a number. " **<<** endl;

                cout **<<** "Type B if you want to bet on a colors red or black." **<<** endl;

                cout **<<** "Type C if you want to bet whether the number is odd or even." **<<**endl;

                cout **<<** "Type D if you want to bet on a dozen of numbers by ranging."**<<** endl;

                cin **>>** type;

                v2.push\_back(**static\_cast<float>**(turn));

                turn**++**;

*//Start of switch case*

**switch**(type){

**case** 'A'**:**

**case** 'a'**:**{

                        question(bank, bet, invalid); *//Call the function*

                        v1.push\_back(65); *//Put into the vector*

**do**{

                            invalid**=**false;

                            cout **<<** "Which number would you like to bet from 1-36?" **<<** endl;

                            cin **>>** numBet; *//Get user input*

**if**(numBet**<**1 **||** numBet**>**36){ *//Input Validation*

                                invalid**=**true;

                            }

**if**(invalid**==**true){

                                cout **<<** "Invalid Entry." **<<** endl;

                            }

                        }**while**(invalid**==**true);

*//Output the result*

                        cout **<<** setw(8) **<<** "Number" **<<** setw(15) **<<** "Your Pick" **<<** endl;

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

                        cout **<<** setw(6) **<<** num **<<** setw(12) **<<** numBet **<<** endl;

                        cout **<<** fixed **<<** showpoint **<<** setprecision(2) **<<** endl;

                        winLosA(numBet, num, amnt, bank, bet, v3); *//Call the function*

                        v3.push\_back(amnt); *//Put amount into vector*

**break**;

                    }

**case** 'B'**:**

**case** 'b'**:**{

                        question(bank, bet, invalid); *//Call the function*

                        v1.push\_back(66);

**do**{

                            invalid**=**false;

                            cout **<<** "Would you like to bet on black(B) or red(R)?" **<<** endl;

                            cin **>>** color; *//Get user input*

**if**(color.length()**!=**1){ *//Input Validation*

                                invalid**=**true;

                            }

**if**(color[0]**!=** 'r' **&&** color[0]**!=** 'R' **&&** color[0]**!=** 'b' **&&** color[0]**!=** 'B'){

                                invalid**=**true;

                            }

**if**(invalid**==**true){

                                cout **<<** "Invalid Entry" **<<** endl;

                            }

                        }**while**(invalid **==** true);

*//Determine a random color 1=black, 0=red*

                        co**=**rand()**%**2;

*//Output the result*

                        cout **<<** setw(8) **<<** "Color" **<<** setw(15) **<<** "Your Pick" **<<** endl;

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

**if**(co**==**0){

                            cout **<<** setw(6) **<<** "red" **<<** setw(12) **<<** color **<<** endl;

                        }

**else**{

                            cout **<<** setw(6) **<<** "black" **<<** setw(12) **<<** color **<<** endl;

                        }

                        cout **<<** fixed **<<** showpoint **<<** setprecision(2) **<<** endl;

                        winLosB(co, color, amnt, bank, bet, v3); *//Call the function*

                        v3.push\_back(amnt); *//Put amount into vector*

**break**;

                    }

**case** 'C'**:**

**case** 'c'**:** {

                        question(bank, bet, invalid); *//Call the function*

                        v1.push\_back(67);

**do**{

                            invalid**=**false;

                        cout **<<** "Would you like to place your bet on E(even) "

                                "or O(odd)?" **<<** endl;

                        cin **>>** choice; *//Get user input*

**if**(choice.length()**!=**1){ *//Input Validation*

                            invalid**=**true;

                        }

**if**(choice[0]**!=**'E' **&&** choice[0]**!=**'e' **&&** choice[0]**!=** 'O' **&&** choice[0]**!=**'o'){

                            invalid**=**true;

                        }

**if**(invalid**==**true){

                            cout **<<** "Invalid Entry" **<<** endl;

                        }

                        }**while**(invalid**==**true);

*//Output the result*

                        cout **<<** setw(8) **<<** "Number" **<<** setw(15) **<<** "Your Pick" **<<** endl;

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

                        cout **<<** setw(6) **<<** num **<<** setw(12) **<<** choice **<<** endl;

                        cout **<<** fixed **<<** showpoint **<<** setprecision(2) **<<** endl;

                        winLosC (choice, num, amnt, bank, bet, v3); *//Call the function*

                        v3.push\_back(amnt); *//Put amount into vector*

**break**;

                    }

**case** 'D'**:**

**case** 'd'**:**{

                        question(bank, bet, invalid); *//Call the function*

                        v1.push\_back(68);

**do**{

                            invalid**=**false;

                            cout **<<** "Which dozen of number would you like to bet on?" **<<** endl;

                            cout **<<** "1:1-12; 2:13-24; 3: 25-36" **<<** endl;

                            cin **>>** range; *//Get user input*

**if**(range.length()**!=**1){ *//Input Validation*

                                invalid**=**true;

                            }

**if**(range[0]**<**'1' **||** range[0]**>**'3' ){

                                invalid**=**true;

                            }

**if**(invalid**==**true){

                                cout **<<** "Invalid Entry" **<<** endl;

                            }

                        }**while**(invalid **==** true);

*//Output the result*

                        cout **<<** setw(8) **<<** "Number" **<<** setw(15) **<<** "Your Pick" **<<** endl;

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

                        cout **<<** setw(6) **<<** num **<<** setw(12) **<<** range **<<** endl;

                        cout **<<** fixed **<<** showpoint **<<** setprecision(2) **<<** endl;

                        winLosD (range, num, amnt, bank, bet, v3); *//Call the function*

                        v3.push\_back(amnt); *//Put amount into vector*

**break**;

                    }

**default:**

                        cout **<<** "Please choose from A, B, C and D." **<<** endl **<<** endl;

                        invalid**=**true;

                } *// The end of switch case*

            }**while**(invalid**==**true); *// The end of do-while loop/ Input Validation for menu*

        }*//The end of for loop*

            cout **<<** "Would you like to play again? Type Y for yes "

                "or any other letter for no." **<<** endl;

            cin **>>** again;

    }**while**(again**==**'Y' **||** again**==**'y'); *//End of do-while loop*

    display(v1, v2, v3); *//Call the function*

**return** 0;

    }

**void** **question**(**float** **&**bank, **float** **&**bet, **bool** **&**invalid){

**do**{

        invalid**=**false;

        cout **<<** "How much total do you have? " **<<** endl; *//Get user input at the beginning of each game*

        cin **>>** bank;

**if**(bank**<=**0){ *//Input validation*

            invalid**=**true;

        }

**if**(invalid**==**true){

        cout **<<** "Invalid Entry. You need to have minimum 1 penny." **<<** endl;

        }

    }**while**(invalid**==**true);

**do**{

        invalid**=**false;

        cout **<<** "How much would you like to bet?" **<<** endl;

        cin **>>** bet;

**if**(bet**<=**0 **||** bet**>**bank){

            invalid**=**true;

        }

**if**(invalid**==**true){

            cout **<<** "Invalid Entry. You need to bet minimum 1 penny or " **<<** endl;

            cout **<<** "you cannot bet more than $" **<<** bank **<<** endl;

        }

    }**while**(invalid**==**true);

}

**void** **winLosA** (**int** numBet, **int** num, **float** **&**amnt, **float** **&**bank, **float** bet, vector**<float>** **&**v3){

**if**(numBet **==** num){

        amnt **=** bank **-** bet **+** bet **\*** 36; *//Calculation for win*

        cout **<<** "Congratulations! You won $" **<<** bet **\*** 36 **<<** endl;

        cout **<<** "Now you have $" **<<** amnt **<<** endl **<<** endl;

    }

**else**{

        amnt **=** bank **-** bet; *//Calculation for lose*

        cout **<<** "Sorry, you lost $" **<<** bet **<<** endl;

        cout **<<** "Now you have $" **<<** amnt **<<** endl;

        cout **<<** "Good luck on your next spin!" **<<** endl **<<** endl;

    }

}

**void** **winLosB** (**bool** co, string color, **float** **&**amnt, **float** **&**bank, **float** bet, vector**<float>** **&**v3){

**if**((color[0]**==**'b' **||** color[0] **==**'B') **&&** co**==**1){

        amnt **=** bank **-** bet **+** bet **\*** 2; *//Calculation for win*

        cout **<<** "Congratulations! You won $" **<<** bet **\*** 2 **<<** endl;

        cout **<<** "Now you have $" **<<** amnt**<<** endl;

    }

**else** **if**((color[0]**==**'r' **||** color[0] **==**'R') **&&** co**==**0){

        amnt **=** bank **-** bet **+** bet **\*** 2; *//Calculation for lose*

        cout **<<** "Congratulations! You won $" **<<** bet **\*** 2 **<<** endl;

        cout **<<** "Now you have $" **<<** amnt**<<** endl;

    }

**else**{

        cout **<<** "Sorry, you lost $" **<<** bet **<<** endl;

        cout **<<** "Now you have $" **<<** bank **-** bet **<<** endl;

        cout **<<** "Good luck on your next spin!" **<<** endl **<<** endl;

    }

}

**void** **winLosC** (string choice, **int** num, **float** **&**amnt, **float** **&**bank, **float** bet, vector**<float>** **&**v3){

**if**(num**%**2**==**0 **&&** (choice[0]**==**'E'**||**choice[0]**==** 'e')){

        amnt **=** bank **-** bet **+** bet **\*** 2; *//Calculation for win*

        cout **<<** "Congratulations! You won $" **<<** bet **\*** 2 **<<** endl;

        cout **<<** "Now you have $" **<<** amnt**<<** endl;

    }

**else** **if**(num**%**2**==**1 **&&** (choice[0]**==**'O'**||**choice[0]**==**'o')){

        amnt **=** bank **-** bet **+** bet **\*** 2; *//Calculation for lose*

        cout **<<** "Congratulations! You won $" **<<** bet **\*** 2 **<<** endl;

        cout **<<** "Now you have $" **<<** amnt**<<** endl **<<** endl;

    }

**else**{

        cout **<<** "Sorry, you lost $" **<<** bet **<<** endl;

        cout **<<** "Now you have $" **<<** bank **-** bet **<<** endl;

        cout **<<** "Good luck on your next spin!" **<<** endl **<<** endl;

    }

}

**void** **winLosD** (string range, **int** num, **float** **&**amnt, **float** **&**bank, **float** bet, vector**<float>** **&**v3){

**if**((num**>=**1 **&&** num**<=**12) **&&** range[0]**==**'1'){

        amnt **=** bank **-** bet **+** bet **\*** 3; *//Calculation for win*

        cout **<<** "Congratulations! You won $" **<<** bet **\*** 3 **<<** endl;

        cout **<<** "Now you have $" **<<** amnt**<<** endl **<<** endl;

    }

**else** **if**((num**>=**13 **&&** num**<=**24) **&&** range[0]**==**'2'){

        amnt **=** bank **-** bet **+** bet **\*** 3; *//Calculation for win*

        cout **<<** "Congratulations! You won $" **<<** bet **\*** 3 **<<** endl;

        cout **<<** "Now you have $" **<<** amnt**<<** endl **<<** endl;

    }

**else** **if**((num**>=**25 **&&** num**<=**36) **&&** range[0]**==**'3'){

        amnt **=** bank **-** bet **+** bet **\*** 3; *//Calculation for lose*

        cout **<<** "Congratulations! You won $" **<<** bet **\*** 3 **<<** endl;

        cout **<<** "Now you have $" **<<** amnt**<<** endl **<<** endl;

    }

**else**{

        cout **<<** "Sorry, you lost $" **<<** bet **<<** endl;

        cout **<<** "Now you have $" **<<** bank **-** bet **<<** endl;

        cout **<<** "Good luck on your next spin!" **<<** endl **<<** endl;

    }

}

**void** **display** (vector**<char>** **&**v1, vector**<float>** **&**v2, vector**<float>** **&**v3){

**const** **int** ROWS**=**100, COLS**=**2;

**char** cv1[ROWS][COLS]; *//Set up rows and columns for 2-D array*

**float** cv23[ROWS][COLS]; *//v2+v3*

**bool** swap; *//For swap purposes*

**char** ctemp;

**int** temp;

**for**(**int** i**=**0;i**<**v1.size();i**++**){

        cv1[i][0]**=**v1[i];

        cv23[i][0]**=**v2[i];

        cv23[i][1]**=**v3[i];

    }

**do**{

        swap**=**false;

**for**(**int** i**=**0;i**<**v1.size()**-**1;i**++**){

**if**(cv1[i][0]**>**cv1[i**+**1][0]){

                ctemp**=**cv1[i][0];

                cv1[i][0]**=**cv1[i**+**1][0];

                cv1[i**+**1][0]**=**ctemp;

**for**(**int** j**=**0;j**<**2;j**++**){

                    temp**=**cv23[i][j];

                    cv23[i][j]**=**cv23[i**+**1][j];

                    cv23[i**+**1][j]**=**temp;

                }

                swap**=**true;

            }

        }

    }**while**(swap);

**do**{

        swap**=**false;

**for**(**int** i**=**0;i**<**v1.size()**-**1;i**++**){

**if**(cv23[i][1]**<**cv23[i**+**1][1] **&&** cv1[i][0]**==**cv1[i**+**1][0]){

                ctemp**=**cv1[i][0];

                cv1[i][0]**=**cv1[i**+**1][0];

                cv1[i**+**1][0]**=**ctemp;

**for**(**int** j**=**0;j**<**2;j**++**){

                    temp**=**cv23[i][j];

                    cv23[i][j]**=**cv23[i**+**1][j];

                    cv23[i**+**1][j]**=**temp;

                }

                swap**=**true;

            }

        }

    }**while**(swap);

*//Display and formatting*

    cout **<<** "Game Type" **<<** setw(10) **<<** "Spin" **<<** setw(12) **<<** "Amoumt" **<<** endl;

**for**(**int** i**=**0;i**<**v1.size();i**++**){

        cout**<<** setw(5) **<<** cv1[i][0]

**<<** setw(13) **<<** **static\_cast<int>**(cv23[i][0]) *//static cast to display "spin" without decimal places*

**<<** setw(8) **<<** "$" **<<** cv23[i][1]**<<**endl;

    }

    cout**<<**"\n\n";

}