CSC – 5 Program Logic Using C++

Professor Mark Lehr

**Project #2**

MasterMind Game Project

By

Ryan Blanco

Winter 2018

Riverside City College

February 8, 2018

**Table of Contents**

***Introduction****..……………………………………………...………………………………………………..3*

***Summary*** ………..*..……………………..…………………………………..…………………………….3*

***Pseudo Code*** …………… *..……………………………………………………………………………...4*

***Flowchart*** *………………………………………………………..….…………….………………....…….5*

***Output Displays*** *……………………………………………………………………………………………8*

***References***  *..………………………………………………………………………………………………..9*

***Code*** ……. *..………………………………………………………………………………………………..9*

**Introduction:**

Title: MasterMind

MasterMind is typically played with two players, one being the codemaker while the other is the codebreaker. The board consists of four columns and ten rows. Each row represents a turn the codebreaker can make, and each column is where the codebreaker places their guess of the four-color code combination consisting of six varied colors (Blue, Green, Orange, Purple, Red, or Yellow). The codemaker can pick the code of their choosing and is hidden at the other end of the game board. Based on the codebreakers guess, the codemaker places pegs on the side that indicate a correct color in the correct location or a correct color in the incorrect location.

Ex) If the code is Red, Orange, Blue, Green and the codebreaker guesses Orange, Purple, Red, Green, then the codemaker would provide one peg indicating correct color and location, and two pegs indicating correct color but incorrect location.

If the codebreaker matches the code in under ten turns then that player wins, elsewise that player loses.

**Summary:**

Project size: 207 lines

Number of variables: 4

Number of methods: 5

This project demonstrates the use of the concepts learned thus far in the course.

The program runs with mostly proper output results, but some minor changes could be made.

One of the problems I had encountered when trying to emulate the game was the randomization of the codemaker code, but with some online research I was able to produce a random answer with every run.

The code utilizes arrays to randomize the color code. If not for the array, the randomization process would be more complicated.

**Pseudo Code:**

*Initialize*

*Random color code generation using integers*

*For integers 0 though 3*

*Switch six cases of colors*

*While turns one through ten*

*For integers 0 thought 3*

*If input color equals random color*

*Display correct color and correct location marker*

*If input color for integer 0 equals input color for integer 1, 2 or 3*

*Display correct color and incorrect location marker*

*If input color for integer 1 equals input color for integer 0, 2 or 3*

*Display correct color and incorrect location marker*

*If input color for integer 2 equals input color for integer 0, 1 or 3*

*Display correct color and incorrect location marker*

*If input color for integer 3 equals input color for integer 0, 1 or 2*

*Display correct color and incorrect location marker*

*If all input colors equal randomly generated colors*

*Display game won with correct answer*

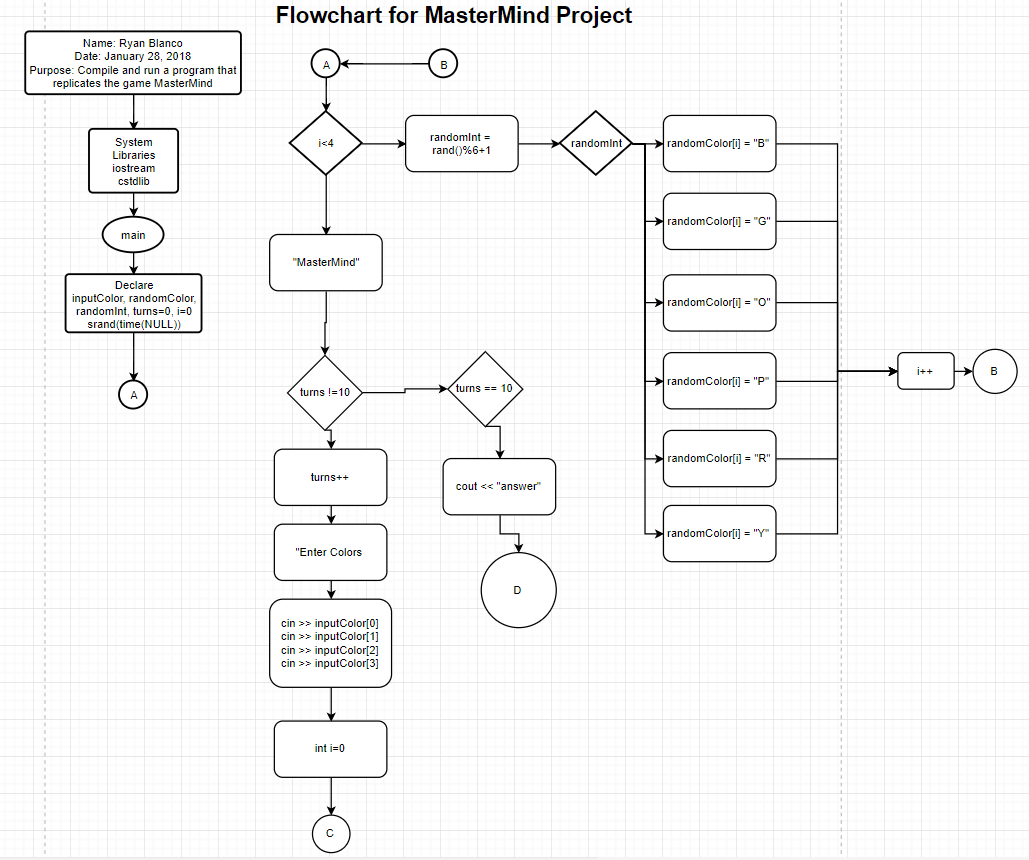
*Else*

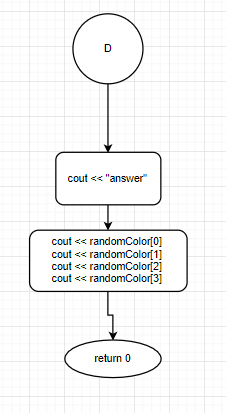
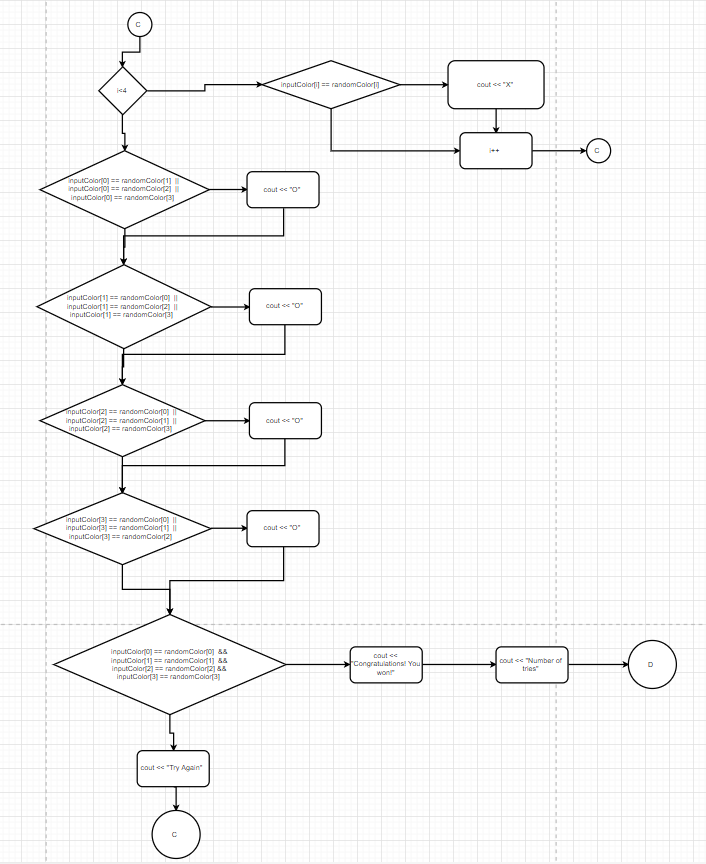
*Display try again*

*If turns are ten without correct answer*

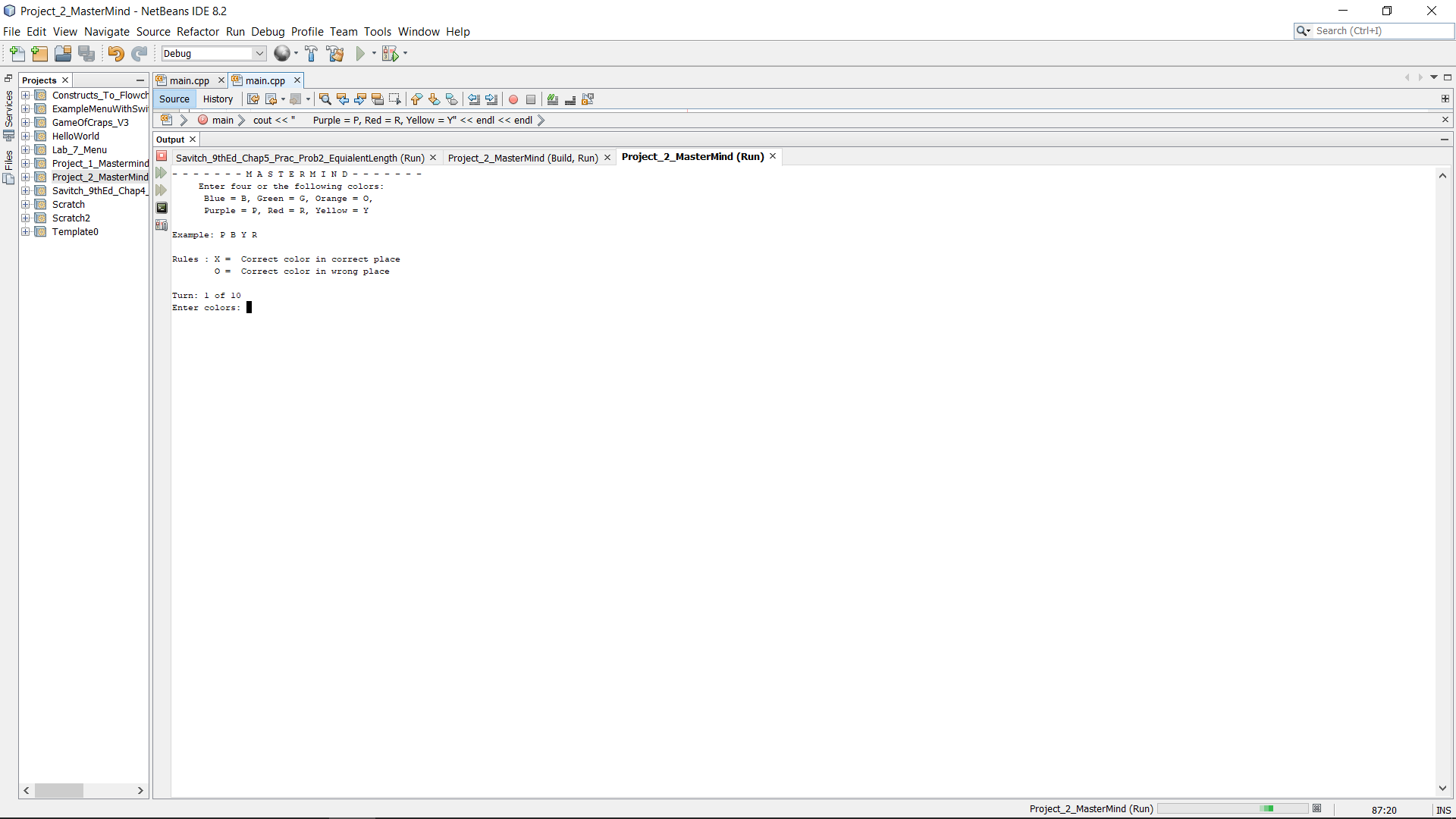
*Display game lost and show correct answer*

**Flowchart:**

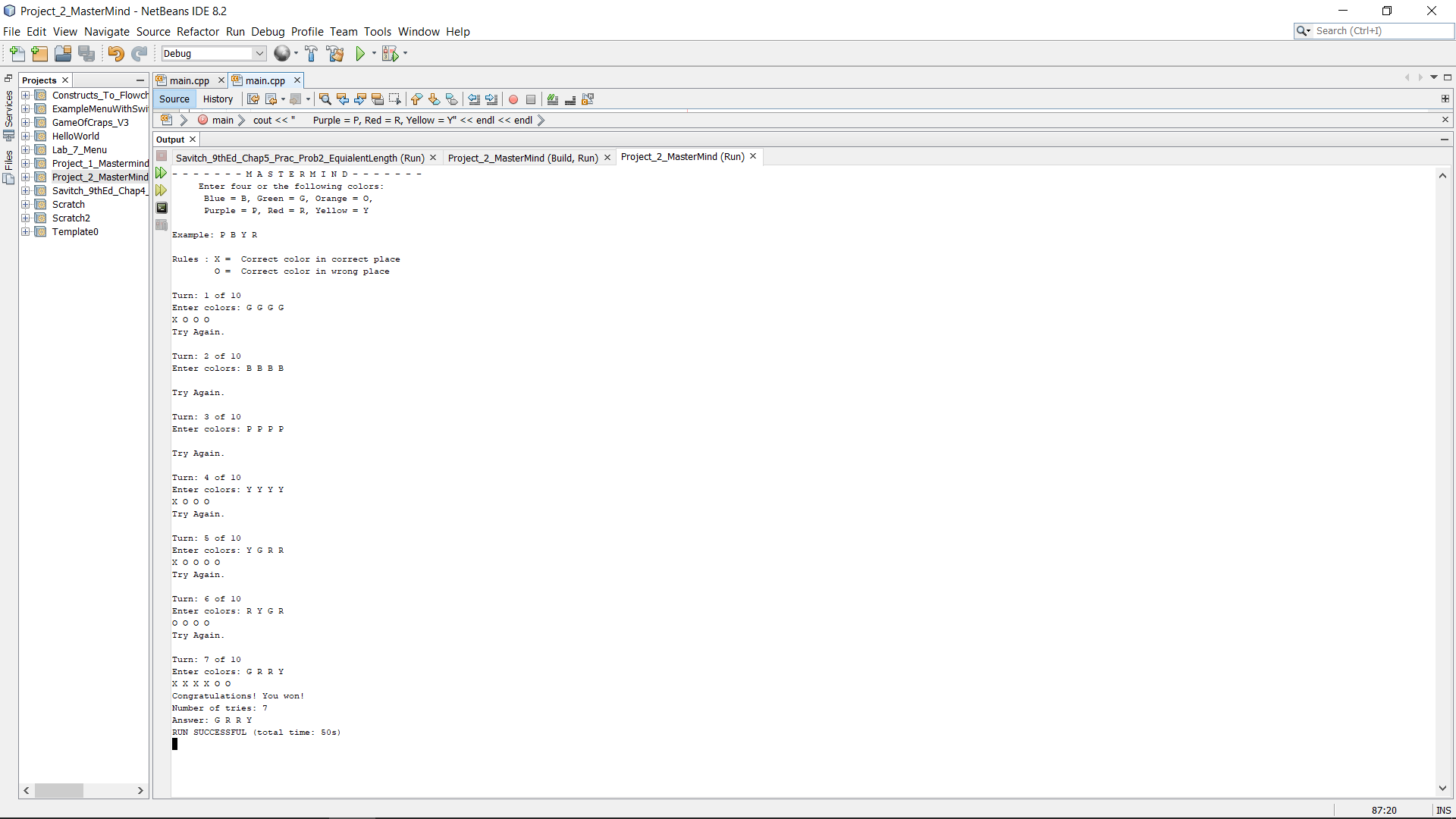
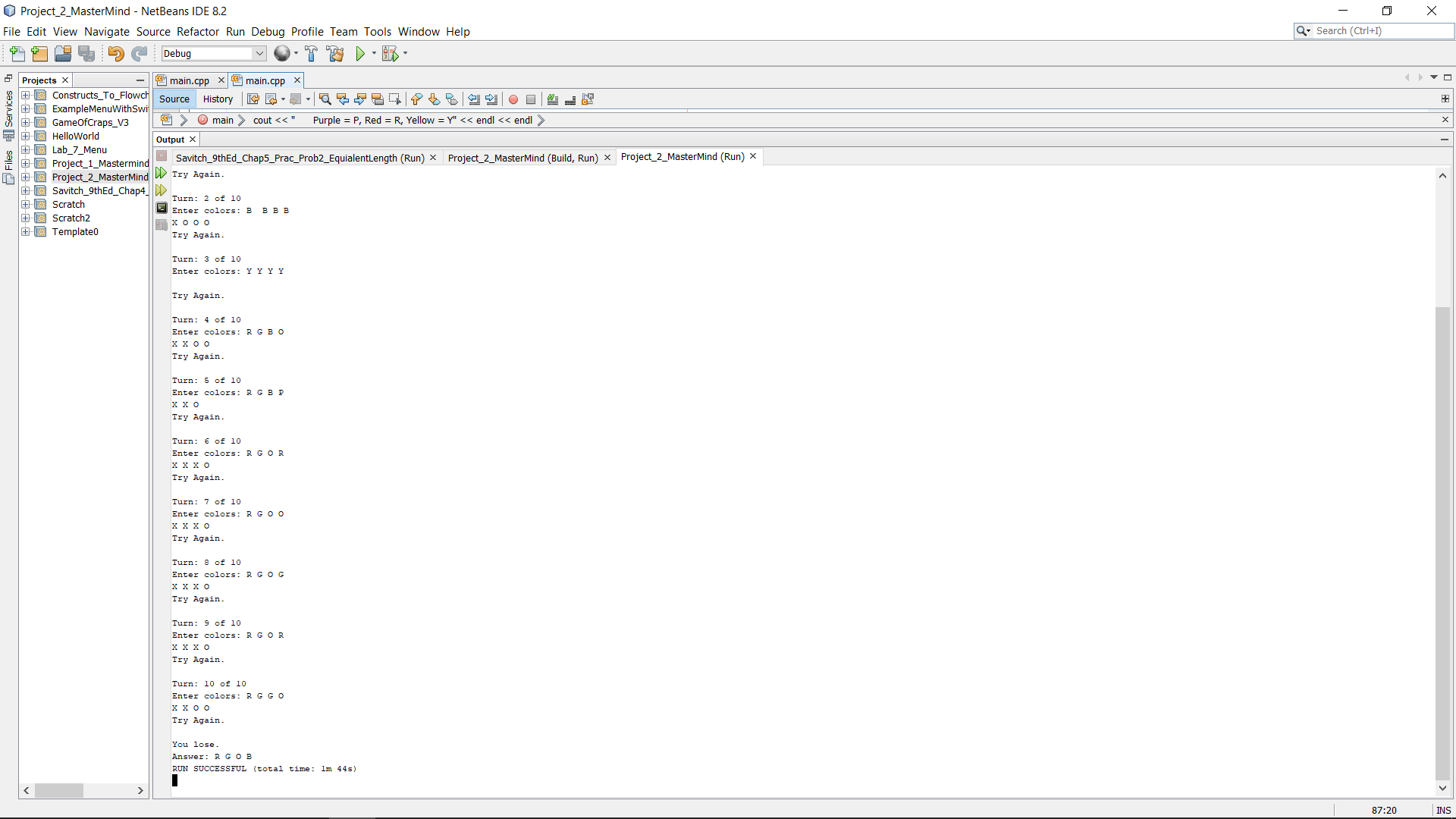




**Output Displays:**



*Figure 4: Start up output*

*Figure 5 – Response with win Figure 6 – Response with loss*

**References:**

1. Dr. Lehr’s Lecture and Lab
2. “Problem Solving with C++” Savitch, Walter, 9th Edition. (Textbook)

**Code:**

*/\**

*\* File: main.cpp*

*\* Author: Ryan Blanco*

*\* Created on January 28, 2018, 9:45 PM*

*\* Purpose: Compile and run a program that replicates the game "MasterMind"*

*\*/;*

*//System Libraries*

*#include <iostream> // Input - Output Stream Library*

*#include <iomanip> // Formatting Library*

*#include <ctime> // Unique Seed Value Library*

*#include <cstdlib> // Random Value Library*

*#include <string> // String Library*

*#include <fstream> // File I/O*

*#include <cmath> // Math Library*

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

*//User Libraries*

*//Global Constants*

*//Function Prototypes*

*// Program execution begins*

*int main(){*

*// Initialize Variables*

*char inputColor[4], randomColor[4];*

*int randomInt, turns = 0;*

*// Randomize Codemaker Pattern*

*// Seed for random number generation*

*srand(time(NULL));*

*// For statement to set integer to color character*

*for(int i=0;i<4;i++){*

*// Equation to randomize integers*

*randomInt = (rand()%6)+1;*

*// Switch statement to allocate random integers to characters*

*switch(randomInt){*

*// Set first case to equal blue "B" or "b"*

*case 1:*

*randomColor[i] = 'B';*

*randomColor[i] = 'b';*

*break;*

*// Set second case to equal green "G" or "g"*

*case 2:*

*randomColor[i] = 'G';*

*randomColor[i] = 'g';*

*break;*

*// Set third case to equal orange "O" or "o"*

*case 3:*

*randomColor[i] = 'O';*

*randomColor[i] = 'o';*

*break;*

*// Set fourth case to equal purple "P" or "p"*

*case 4:*

*randomColor[i] = 'P';*

*randomColor[i] = 'p';*

*break;*

*// Set fifth case to equal red "R" or "r"*

*case 5:*

*randomColor[i] = 'R';*

*randomColor[i] = 'r';*

*break;*

*// Set sixth case to equal yellow "Y" or "y"*

*case 6:*

*randomColor[i] = 'Y';*

*randomColor[i] = 'y';*

*break;*

*}*

*}*

*// Intro and Rules Prompt*

*//Game title text*

*cout << "- - - - - - - M A S T E R M I N D - - - - - - -" << endl;*

*//Prompt user input text*

*cout << " Enter four or the following colors:" << endl;*

*//Text display of input color options*

*cout << "Blue = B or b, Green = G or g, Orange = O or o," << endl;*

*cout << "Purple = P or p, Red = R or r, Yellow = Y or y" << endl << endl;*

*//Text display of example output*

*cout << "Example: P B Y R" << endl << endl;*

*//Text display of feedback key*

*cout << "Rules : X = Correct color in correct place" << endl;*

*cout << " O = Correct color in wrong place"<< endl << endl;*

*// Output Codebreaker Results from Input Values*

*// While statement for input response*

*while(turns != 10){*

*// Turn counter*

*turns++;*

*// Text display of current turn*

*cout << "Turn: " << turns << " of 10" << endl;*

*// Prompt user input*

*cout << "Enter colors: ";*

*// User input values*

*cin >> inputColor[0] >> inputColor[1];*

*cin >> inputColor[2] >> inputColor[3];*

*// For statement if any input matches random generation*

*for(int i=0;i<4;i++){*

*// If statement to determine match*

*if(inputColor[i] == randomColor[i])*

*// Output if input matches random random generation*

*cout << "X" << " ";*

*}*

*// If statement for first input value color match in wrong location*

*if(inputColor[0] == randomColor[1] ||*

*inputColor[0] == randomColor[2] ||*

*inputColor[0] == randomColor[3] ){*

*// Output if If statement is true*

*cout << "O" << " ";*

*}*

*// If statement for second input value color match in wrong location*

*if(inputColor[1] == randomColor[0] ||*

*inputColor[1] == randomColor[2] ||*

*inputColor[1] == randomColor[3]){*

*// Output if If statement is true*

*cout << "O" << " ";*

*}*

*// If statement for third input value color match in wrong location*

*if(inputColor[2] == randomColor[0] ||*

*inputColor[2] == randomColor[1] ||*

*inputColor[2] == randomColor[3]){*

*// Output if If statement is true*

*cout << "O" << " ";*

*}*

*// If statement for fourth input value color match in wrong location*

*if(inputColor[3] == randomColor[0] ||*

*inputColor[3] == randomColor[1] ||*

*inputColor[3] == randomColor[2]){*

*// Output if If statement is true*

*cout << "O" << " ";*

*}*

*cout << endl;*

*//If statement for input matching random generated sequence*

*if(inputColor[0] == randomColor[0] &&*

*inputColor[1] == randomColor[1] &&*

*inputColor[2] == randomColor[2] &&*

*inputColor[3] == randomColor[3]){*

*//Output text stating correct answer achieved*

*cout << "Congratulations! You won!" << endl;*

*//Output text to display number of turns used*

*cout << "Number of tries: " << turns << endl;*

*//Output text to display correct answer*

*cout << "Answer: ";*

*//Display of colors in correct order*

*cout << randomColor[0] << " " << randomColor[1] << " ";*

*cout << randomColor[2] << " " << randomColor[3];*

*return 0;*

*}*

*//Else statement if answer is not correct*

*else{*

*//Output statement for incorrect answer*

*cout << "Try Again." << endl << endl;*

*}*

*}*

*//If statement if all turns used without correct answer*

*if(turns == 10){*

*//Output indicating loss*

*cout << "You lose." << endl;*

*//Output text to display correct answer*

*cout << "Answer: ";*

*//Display of colors in correct order*

*cout << randomColor[0] << " " << randomColor[1] << " ";*

*cout << randomColor[2] << " " << randomColor[3];*

*}*

*//End of program*

*return 0;*

*}*