Mastermind

Edwin Leon

CIS 5

Spring 2017

44188

**Introduction**

Title: Mastermind

Mastermind is a basic code-breaking game. The game was originally created for two players, but throughout time people have played Mastermind by themselves. This game can be played by people from ages eight and up. The game was created in the process of developing security applications. This game requires a lot of logic, and the level of difficulty is fairly decent. The game consists on breaking a color code in less than twelve attempts. The code is randomly generated and is capable of creating more than 1000 distinct codes. The way the player has to decipher the code requires a lot of thinking which inspired me to write the code of this game.

**How the Mastermind works**

**Objective of the Game**

Break the 5-color code in twelve attempts or less.

Also, breaking the 6-color code in fifteen attempts or less.

**Rules of the Game**

Mastermind can be a single player game or a multi-player game.

Single player rules:

1. Input a color code guess with respect to its shortcuts. (Ex.: for Red input ‘r’ or ‘R’).
2. Keep repeating the first step until you manage to decipher the code.
   1. Hints will be given every time you input a new color code guess.

Multi-player rules:

1. The first player has to input a color code guess with respect to its shortcuts. (Ex.: for Red input ‘r’ or ‘R’).
   1. Hints will be given every time you input a new 5-color code guess.
2. If the first player didn’t guess the code correctly, then the second player has the chance to guess the color code.
3. Keep repeating steps 1 and 2 until someone wins, or no one manages to decipher the code in twelve attempts (cumulative).

**Summary**

When I wrote this program, I tried to make it as simple as possible. My main goal was to make the code readable, and user friendly. At the beginning, I wrote the program as easy as possible. My first goal was to make it work, but when I finished it and had accomplished my goal the program looked very messy. I created many versions of Mastermind, and every time I tried implementing something new to the code. My second version of my Mastermind, I transformed most of the original code into functions. For my third version, I changed many variables and created various arrays. For my last version, I decided to let the user choose between a harder mode of Mastermind. This program took me 3-4 days to complete it. Even though, I believe that are a lot of things I can improve on. This program helped me learn to use arrays and to be creative with my coding. The program has approximately 430 lines of code.

**Pseudocode**

//System Libraries

//Input - Output Library

//Random Number Library

//Time Library

//String Library

//Formatting Library

//File I/O

//Name-space under which systems libraries exist

//User Libraries

//Global Constants

//Function Prototypes

//Menu prototype

//Random code prototype

//Random code prototype Hard Mode

//Test win prototype

//Test win prototype Hard Mode

//Player guess prototype

//Player guess prototype Hard Mode

//File display prototype

//Execution begins here

//Set the random number seed

//Declare variables

//Output file

//Player name

//Win or Lose

//Size of the array

//User selection

//Colors array

//Guess array

//Mastermind

//Start time

//Do-While loop starts here

//Menu

//User choice

//Mastermind Normal Mode

//Size of the array

//Colors array

//Guess array

//Code creator

//Random color code

//Open file

//Player Name

//Starting message

//Play Mastermind

//Set gameWon to false

//Number of tries

//While loop starts here

//Attempt counter

//Key for the colors

//Number of tries

//Player guess

//Verifies if guess matches the code

//If testWin is true

//Set gameWon to true

//Win message

//winLose set to “Win”

//End of If loop

//If tries = 12

//Lose message

//winLose set to “Lose”

//Code

//End of If loop

//End of While loop

//File display

//Hard Mode Mastermind

//Size of the array

//Colors array

//Guess array

//Code creator

//Random color code

//Open file

//Player Name

//Starting message

//Play Mastermind

//Set gameWon to false

//Number of tries

//While loop starts here

//Attempt counter

//Key for the colors

//Number of tries

//Player guess

//Verifies if guess matches the code

//If testWin is true

//Set gameWon to true

//Win message

//Win message to file

//End of If loop

//If tries = 12

//Lose message

//Lose message to file

//Code

//End of If loop

//End of While loop

//File display

//End of If-Else loop

//End of Do-While loop

//Ends time

//Outputs time

//Close file

//Exit Program

//Menu Function

//Code creator function Normal Mode

//Creates the random color code

//Random integer from [1,5]

//Switch the random integer to a color

//Red

//Blue

//Yellow

//Purple

//Green

//Code creator function Normal Mode

//Creates the random color code

//Random integer from [1,5]

//Switch the random integer to a color

//Red

//Blue

//Yellow

//Purple

//Green

//Orange

//Color Matching Function Normal Mode

//Declare Variables

//Counter for matching colors

//Temporal array

//Temporal copy of code

//Right color and right position

//Set answer[y] to 0

//Increments counter

//Output Correct

//Resets counter

//Right color and wrong position

//Set answer[y] to 0

//Increments counter

//Output partial correct

//Check if player won the game

//Win = false

//Count correct

//Counts how many colors match and are in the right position

//Returns to original code

//Test guess[y] = temp[y]

//Increment count

//Count<5 then win=false, count>=5 win=true

//Return win

//Color Matching Function Hard Mode

//Declare Variables

//Counter for matching colors

//Temporal array

//Temporal copy of code

//Right color and right position

//Set answer[y] to 0

//Increments counter

//Output Correct

//Resets counter

//Right color and wrong position

//Set answer[y] to 0

//Increments counter

//Output partial correct

//Check if player won the game

//Win = false

//Count correct

//Counts how many colors match and are in the right position

//Returns to original code

//Test guess[y] = temp[y]

//Increment count

//Count<6 then win=false, count>=6 win=true

//Return win

//Player Guess Function Normal Mode

//Color counter

//Input a color

//r or R for Red

//b or B for Blue

//p or P for Purple

//g or G for Green

//y or Y for Yellow

//Invalid Input

//Do-While invalid input

//Try again message

//Input a color

//r or R for Red

//b or B for Blue

//p or P for Purple

//g or G for Green

//y or Y for Yellow

//Do while invalid input

//Player Guess Function Normal Mode

//Color counter

//Input a color

//r or R for Red

//b or B for Blue

//p or P for Purple

//g or G for Green

//y or Y for Yellow

//o or O for Orange

//Invalid Input

//Do-While invalid input

//Try again message

//Input a color

//r or R for Red

//b or B for Blue

//p or P for Purple

//g or G for Green

//y or Y for Yellow

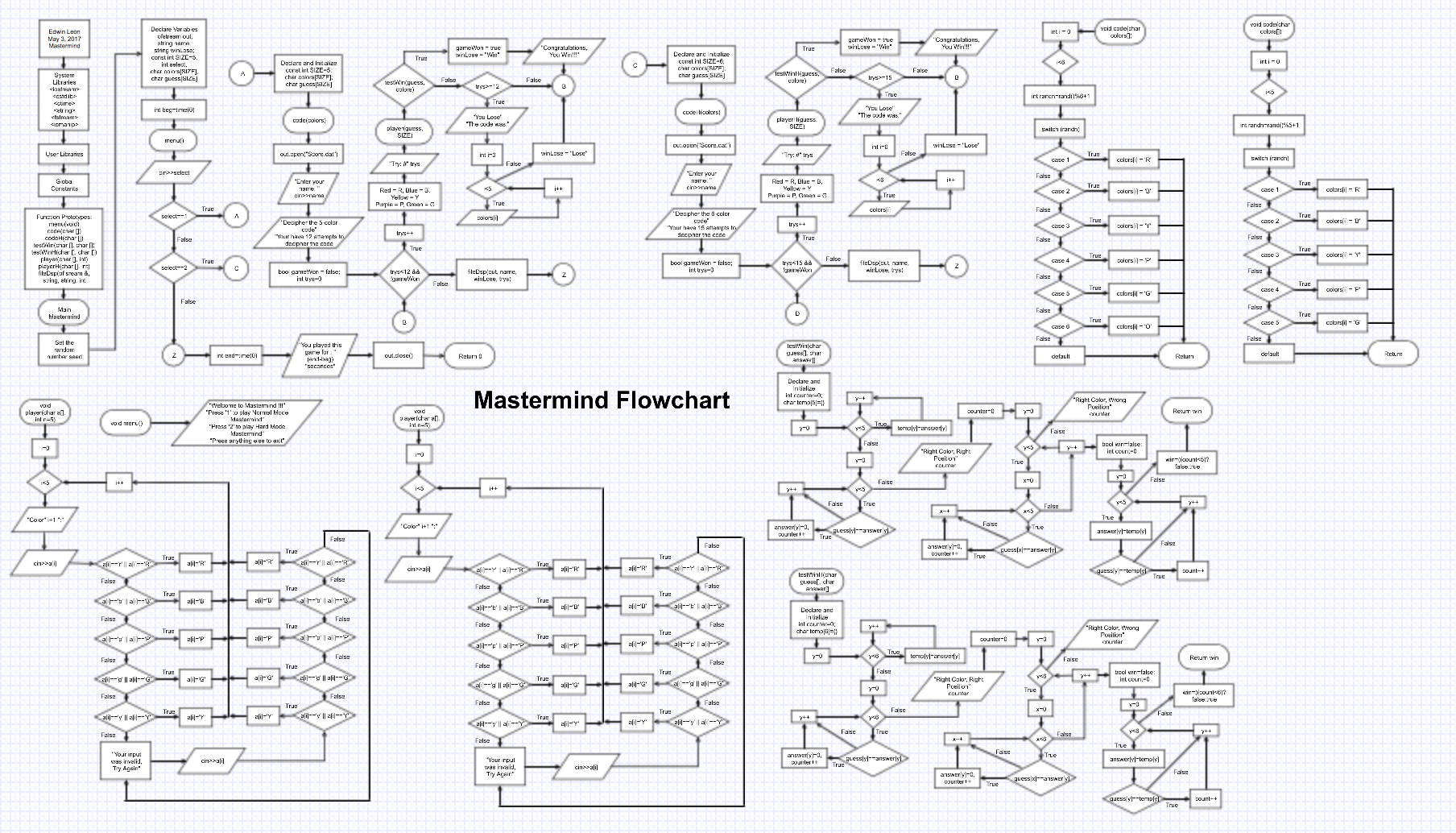
//o or O for Orange

//Do while invalid input

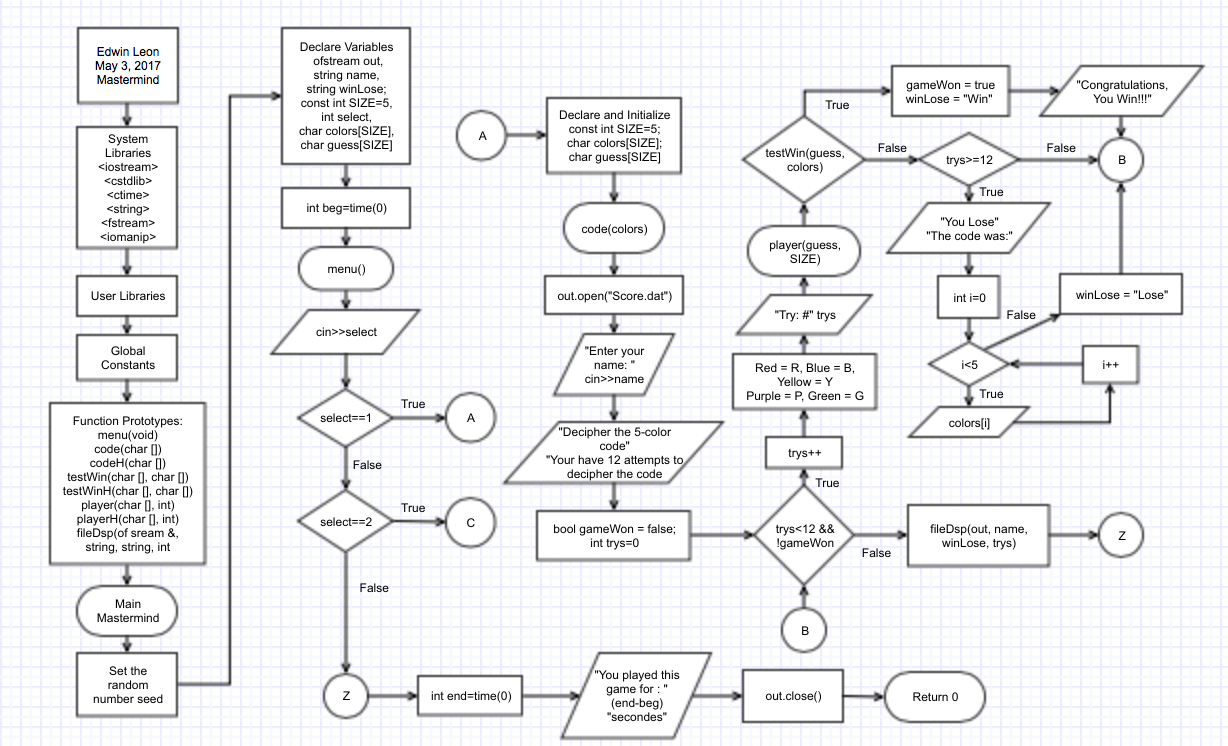
//File Display Function

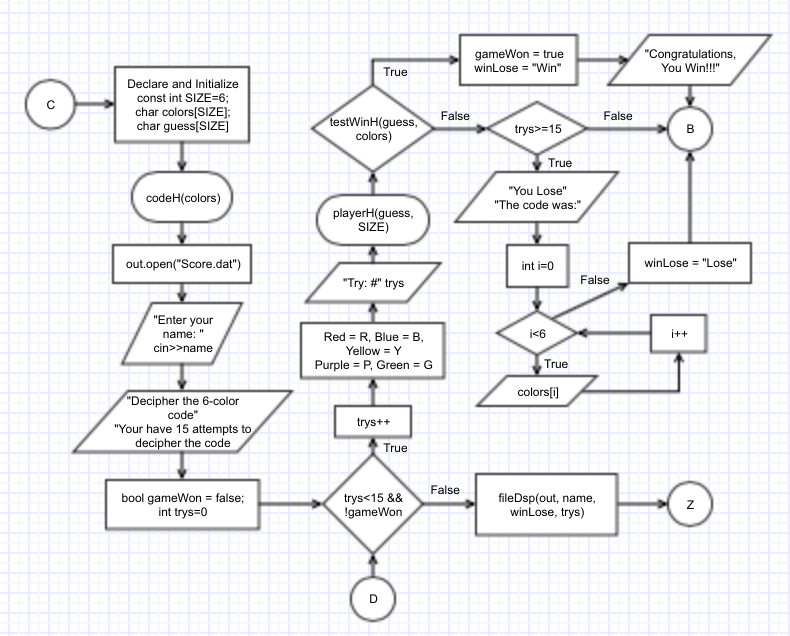
**Check Off Sheet**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **Where in Code** |
| **Chapter** | **Section** | **Topic** | **Line number** |
| 2 | 2 | cout | 58-68, 77-102, 117-174, 254-268, 302-316, 335, 354, 380, 402 |
|  | 3 | libraries | iostream, iomanip, cstdlib, fstream, string, ctime |
|  | 4 | variables/literals | 38-55, 71, 72, 108-110, 182, 210, 240, 241, 272, 273, 288, 289, etc. |
|  | 5 | Identifiers | 43, 44, 54-57, 81-105, 109-112, 136-160, etc. |
|  | 6 | Integers | 42, 47, 53, 72, 99, 108, 127, 154, 163, 181, 182, 209, 210, etc. |
|  | 7 | Characters | 24-29, 43, 44, 54, 55, 109, 110, 241, 289, etc. |
|  | 8 | Strings | 30, 39, 40, 429 |
|  | 9 | Floats No Doubles | 165 |
|  | 10 | Bools | 71, 126, 272, 320 |
|  | 11 | Size of \*\*\*\*\* |  |
|  | 12 | Variables 7 characters or less | 23-55, 108-110, etc. |
|  | 13 | Scope \*\*\*\*\* No Global Variables |  |
|  | 14 | Arithmetic operators | 73, 94, 128, 149, 162, etc. |
|  | 15 | Comments 20%+ | throughout the code |
|  | 16 | Named Constants | 53, 108 |
|  | 17 | Programming Style \*\*\*\*\* Emulate |  |
|  |  |  |  |
| 3 | 1 | cin | 63, 118, 337, 356, 382, 404 |
|  | 2 | Math Expression | throughout the code |
|  | 3 | Mixing data types \*\*\*\* |  |
|  | 4 | Overflow/Underflow \*\*\*\* |  |
|  | 5 | Type Casting | 165 |
|  | 6 | Multiple assignment \*\*\*\*\* |  |
|  | 7 | Formatting output | 431 |
|  | 8 | Strings | 30, 39, 40, 429 |
|  | 9 | Math Library |  |
|  | 10 | Hand tracing \*\*\*\*\*\* |  |
|  |  |  |  |
| 4 | 1 | Relational Operators | 73, 74, 94, etc. |
|  | 2 | if | 52, 87, 94, etc. |
|  | 4 | If-else | 357 |
|  | 5 | Nesting | 338-372, 383- 423, etc. |
|  | 6 | If-else-if | 338-350, 357-369, etc. |
|  | 7 | Flags \*\*\*\*\* | 71, 88, 126, 143, etc. |
|  | 8 | Logical operators | 52, 73, 107, 128, 162, etc. |
|  | 11 | Validating user input | 162, 372, 423 |
|  | 13 | Conditional Operator | 281, 329 |
|  | 14 | Switch | 184, 212 |
|  |  |  |  |
| 5 | 1 | Increment/Decrement | 74, 99, 129, etc. |
|  | 2 | While | 63, 128 |
|  | 5 | Do-while | 48, 354, 402 |
|  | 6 | For loop | 99, 154, 181, 209, 244, 249, etc. |
|  | 11 | Files input/output both | 59, 105, 114, 160, 429 |
|  | 12 | No breaks in loops \*\*\*\*\*\* |  |
|  |  |  |  |
| 6 | 3 | Function Prototypes | 23-30 |
|  | 5 | Passing by value | 84-139 |
|  | 8 | Returning values from functions | 282, 330 |
|  | 9 | Returning a Boolean \*\*\*\*\*\* |  |
|  | 10 | No Global Variables Allowed | None |
|  |  | Only Global Constants |  |
|  |  | Meaning Conversions, Physical Constants only |  |
|  | 11 | Static Local |  |
|  | 12 | Default arguments | 334, 379 |
|  | 13 | Reference Parameters |  |
|  | 14 | Overloading functions |  |
|  | 15 | Exit function \*\*\*\*\*\*\* |  |
|  |  |  |  |
| 7 | 4 | Array Initialization | 241 |
|  | 6 | Processing Arrays | throughout the code |
|  | 7 | Parallel Arrays | 250, 262, etc. |
|  | 8 | Arrays as function arguments | 24-29 |
|  | 9 | 2-D Arrays |  |
|  | 12 | STL Vector |  |
|  |  |  |  |
| 8 | 1 | Linear and Binary Search | 262, 310 |
|  | 3 | Bubble and Selection Sort |  |
|  | 5 | Search/Sorting Vectors \*\*\*\*\*\* |  |
| \*\*\*\*\*\* Not required to show |  |  |  |

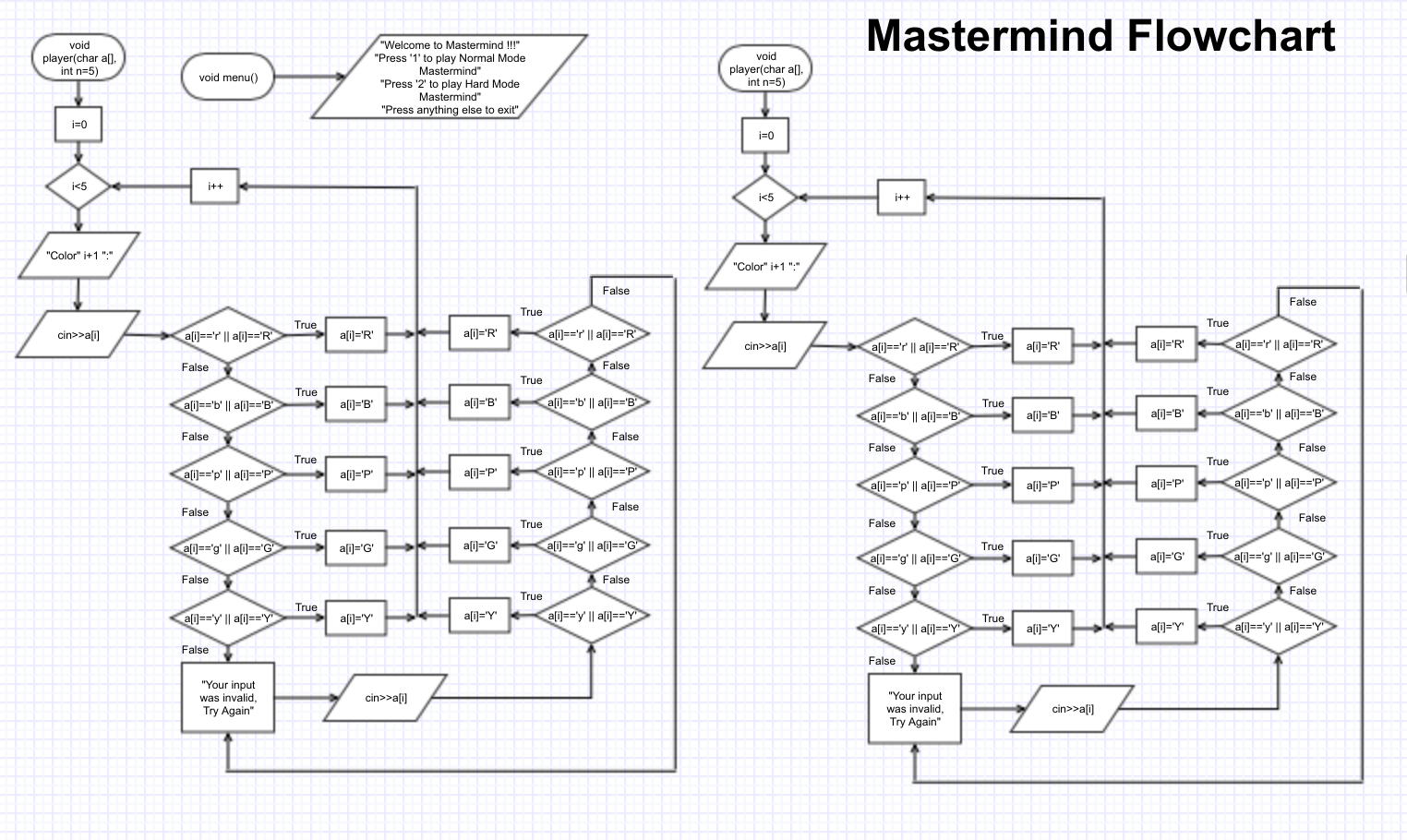
**Flowchart**

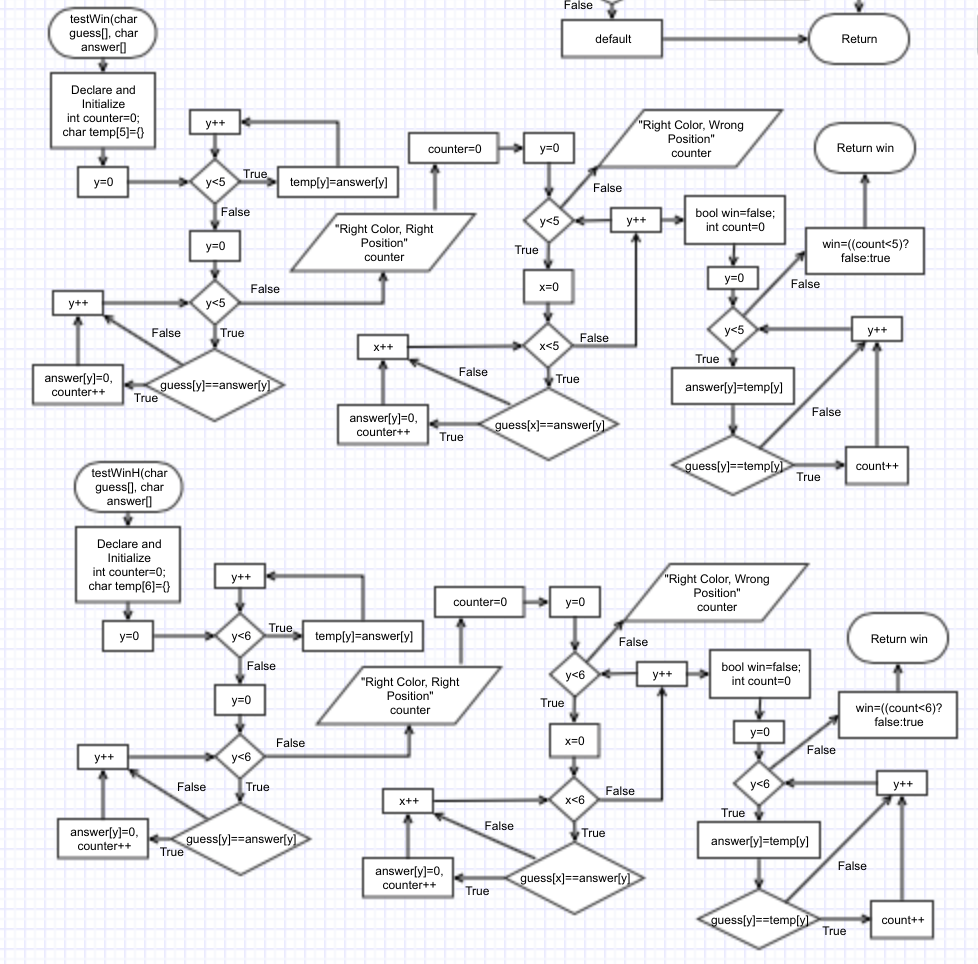
**Int main**

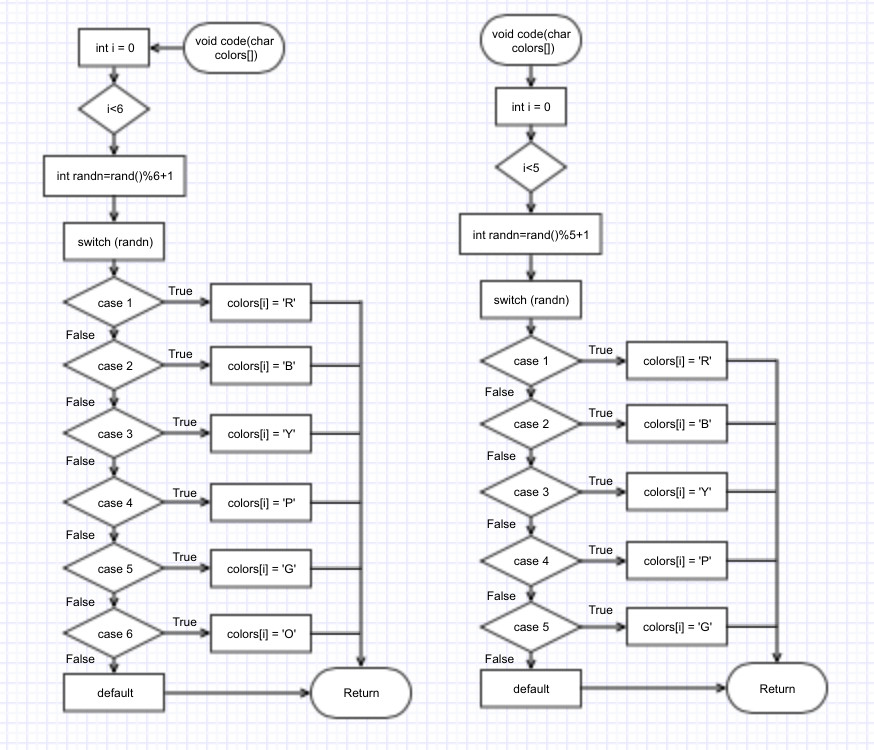
****

****

**Functions**

****

****

****

**Code**

/\*

\* File: main.cpp

\* Author: Edwin Leon

\* Created on June 02, 2017, 03:53 PM

\* Purpose: Mastermind V5 - Project 2

\*/

//System Libraries

#include <iostream> //Input - Output Library

#include <cstdlib> //Random Number Library

#include <ctime> //Time Library

#include <string> //String Library

#include <iomanip> //Formating Library

#include <fstream> //File I/O

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

//User Libraries

//Global Constants

//Function Prototypes

void menu(void); //Menu prototype

void code(char []); //Random code prototype

void codeH(char []); //Random code prototype Hard Mode

bool testWin(char [], char []); //Test win prototype

bool testWinH(char [], char []); //Test win prototype Hard Mode

void player(char[],int); //Player guess prototype

void playerH(char[],int); //Player guess prototype Hard Mode

void fileDsp(ofstream &,string,string,int); //File display prototype

//Execution begins here

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

//Set the random number seed

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

//Declare variables

ofstream out; //Output file

string name; //Player name

string winLose; //Win or Lose

const int SIZE=5; //Size of the array

int select; //User selection

char colors[SIZE]; //Colors array

char guess[SIZE]; //Guess array

//Mastermind

int beg=time(0); //Start time

do{

menu(); //Menu

cin>>select; //User choice

if(select==1){ //Mastermind Normal Mode

const int SIZE=5; //Size of the array

char colors[SIZE]; //Colors array

char guess[SIZE]; //Guess array

//Code creator

code(colors); //Random color code

cout<<endl;

out.open("Score.dat");

//Player Name

cout<<"Enter your name: "<<endl;

cin>>name;

//Starting message

cout<<"Decipher the 5-color code"<<endl;

cout<<"You have 12 attempts to decipher the code"<<endl;

cout<<endl;

//Play Mastermind

bool gameWon = false; //Set gameWon to false

int trys=0; //Number of tries

while(trys<12 && !gameWon){

trys++; //Attempt counter

//Key for the colors

cout<<"Red = R, Blue = B, Yellow = Y"<<endl;

cout<<"Purple = P, Green = G"<<endl;

//Number of tries

cout<<"Try: #"<<trys<<endl;

//Player guess

player(guess,SIZE);

//Verifies if guess matches the code

if(testWin(guess,colors)){ //If testWin is true

gameWon=true; //Set gameWon to true

cout<<endl;

cout<<"Congratulations, You Win!!!"<<endl;//Win message

winLose="Win";

cout<<endl;

}//End of If loop

if(trys>=12){ //If tries = 12

cout<<"You Lose"<<endl; //Lose message

winLose="Lose";

//Code

cout<<"The code was: ";

for(int i=0;i<5;i++){

cout<<colors[i];

}

cout<<endl<<endl;

}//End of If loop

}//End of While loop

fileDsp(out,name,winLose,trys);//File display

}

else if(select==2){ //Hard Mode Mastermind

const int SIZE=6; //Size of the array

char colors[SIZE]; //Colors array

char guess[SIZE]; //Guess array

//Code creator

codeH(colors); //Random color code

cout<<endl;

out.open("Score.dat"); //Open file

//Player Name

cout<<"Enter your name: "<<endl;

cin>>name;

//Starting message

cout<<"Decipher the 6-color code"<<endl;

cout<<"You have 15 attempts to decipher the code"<<endl;

cout<<endl;

//Play Mastermind

bool gameWon = false; //Set gameWon to false

int trys=0; //Number of tries

while(trys<15 && !gameWon){

trys++; //Attempt counter

//Key for the colors

cout<<"Red = R, "<<"Blue = B, "<<"Yellow = Y"<<endl;

cout<<"Purple = P, "<<"Green = G"<<"Orange = O"<<endl;

cout<<endl;

//Number of tries

cout<<"Try: #"<<trys<<endl;

//Player guess

playerH(guess,SIZE);

//Verifies if guess matches the code

if(testWinH(guess,colors)){ //If testWin is true

gameWon=true; //Set gameWon to true

cout<<endl;

cout<<"Congratulations, You Win!!!"<<endl;//Win message

winLose="Win"; //Win message to file

cout<<endl;

}//End of If loop

if(trys>=15){ //If tries = 12

cout<<"You Lose"<<endl; //Lose message

winLose="Lose";//Lose message to file

//Code

cout<<"The code was: ";

for(int i=0;i<6;i++){

cout<<colors[i];

}

cout<<endl<<endl;

}//End of If loop

}//End of While loop

fileDsp(out,name,winLose,trys);//File display

}

}while(select==1||select==2);

int end=time(0); //Ends time

cout<<"You played this game for: "<<

(static\_cast<float>(end-beg))<<" seconds"<<endl;

out.close(); //Close file

return 0;

}

void menu(){

cout<<"Welcome to Mastermind!!!"<<endl;

cout<<"Press '1' to play Normal Mode Mastermind"<<endl;

cout<<"Press '2' to play Hard Mode Mastermind"<<endl;

cout<<"Press anything else to exit"<<endl;

}

//Code creator function Normal Mode

void code(char colors[]){

//Creates the random color code

for(int i=0;i<5;i++){

int randn=rand()%5+1; //Random integer from [1,5]

//Switch the random integer to a color

switch(randn){

case 1:

colors[i]='R'; //Red

break;

case 2:

colors[i]='B'; //Blue

break;

case 3:

colors[i]='Y'; //Yellow

break;

case 4:

colors[i]='P'; //Purple

break;

case 5:

colors[i]='G'; //Green

break;

default:

break;

}

}

}

//Code creator function Normal Mode

void codeH(char colors[]){

//Creates the random color code

for(int i=0;i<6;i++){

int randn=rand()%6+1; //Random integer from [1,5]

//Switch the random integer to a color

switch(randn){

case 1:

colors[i]='R'; //Red

break;

case 2:

colors[i]='B'; //Blue

break;

case 3:

colors[i]='Y'; //Yellow

break;

case 4:

colors[i]='P'; //Purple

break;

case 5:

colors[i]='G'; //Green

break;

case 6:

colors[i]='O'; //Orange

break;

default:

break;

}

}

}

//Color Matching Function Normal Mode

bool testWin(char guess[],char answer[]){

//Declare Variables

int counter=0; //Counter for matching colors

char temp[5]={}; //Temporal array

//Temporal copy of code

for(int y=0;y<5;y++){

temp[y]=answer[y];

}

//Right color and right position

for(int y=0;y<5;y++){

if(guess[y]==answer[y]){

answer[y]=(0); //Set answer[y] to 0

counter++; //Increments counter

}

}

cout<<"Right Color, Right Position: "<<counter; //Output Correct

cout<<endl;

counter=0; //Resets counter

//Right color and wrong position

for(int y=0;y<5;y++){

for(int x=0;x<5;x++){

if(guess[x]==answer[y]){

answer[y]=(0); //Set answer[y] to 0

counter++; //Increments counter

}

}

}

cout<<"Right Color, Wrong Position: "<<counter<<" "; //Output partial correct

cout<<endl;

//Check if player won the game

bool win=false;//Win = false

int count=0; //Count correct

//Counts how many colors match and are in the right position

for(int y=0;y<5;y++){

answer[y]=temp[y]; //Returns to original code

if(guess[y]==temp[y]){ //Test guess[y] = temp[y]

count++; //Increment count

}

}

win=((count<5)?false:true); //Count<5 then win=false, count>=5 win=true

return win; //Return win

}

//Color Matching Function Hard Mode

bool testWinH(char guess[],char answer[]){

//Declare Variables

int counter=0; //Counter for matching colors

char temp[6]; //Temporal array

//Temporal copy of code

for(int y=0;y<6;y++){

temp[y]=answer[y];

}

//Right color and right position

for(int y=0;y<6;y++){

if(guess[y]==answer[y]){

answer[y]=(0); //Set answer[y] to 0

counter++; //Increments counter

}

}

cout<<"Right Color, Right Position: "<<counter<<" "; //Output Correct

cout<<endl;

counter=0; //Resets counter

//Right color and wrong position

for(int y=0;y<6;y++){

for(int x=0;x<6;x++){

if(guess[x]==answer[y]){

answer[y]=(0); //Set answer[y] to 0

counter++; //Increments counter

}

}

}

cout<<"Right Color, Wrong Position: "<<counter<<" "; //Output partial correct

cout<<endl;

//Check if player won the game

bool win=false;//Win = false

int count=0; //Count correct

//Counts how many colors match and are in the right position

for(int y=0;y<6;y++){

answer[y]=temp[y]; //Returns to original code

if(guess[y]==temp[y]){ //Test guess[y] = temp[y]

count++; //Increment count

}

}

win=((count<6)?false:true); //Count<5 then win=false, count>=5 win=true

return win; //Return win

}

//Player Guess Function Normal Mode

void player(char a[],int n=5){

for(int i=0;i<n;i++){

cout<<"Color "<<i+1<<": "; //Color counter

cin>>a[i]; //Input a color

if(a[i]=='r'||a[i]=='R'){ //r or R for Red

a[i]='R';

}

else if(a[i]=='b'||a[i]=='B'){ //b or B for Blue

a[i]='B';

}

else if(a[i]=='p'||a[i]=='P'){ //p or P for Purple

a[i]='P';

}

else if(a[i]=='g'||a[i]=='G'){ //g or G for Green

a[i]='G';

}

else if(a[i]=='y'||a[i]=='Y'){ //y or Y for Yellow

a[i]='Y';

}

else{ //Invalid Input

do{ //Do-While invalid input

cout<<"Your input was invalid, Try Again"<<endl; //Try again message

cin>>a[i]; //Input a color

if(a[i]=='r'||a[i]=='R'){ //r or R for Red

a[i]='R';

}

else if(a[i]=='b'||a[i]=='B'){ //b or B for Blue

a[i]='B';

}

else if(a[i]=='p'||a[i]=='P'){ //p or P for Purple

a[i]='P';

}

else if(a[i]=='g'||a[i]=='G'){ //g or G for Green

a[i]='G';

}

else if(a[i]=='y'||a[i]=='Y'){ //y or Y for Yellow

a[i]='Y';

}

}while(!((a[i]=='R')||(a[i]=='B')||(a[i]=='Y')||

(a[i]=='P')||(a[i]=='G')));

}

}

}

//Player Guess Function Hard Mode

void playerH(char a[],int n=6){

for(int i=0;i<n;i++){

cout<<"Color "<<i+1<<": "; //Color counter

cin>>a[i]; //Input a color

if(a[i]=='r'||a[i]=='R'){ //r or R for Red

a[i]='R';

}

else if(a[i]=='b'||a[i]=='B'){ //b or B for Blue

a[i]='B';

}

else if(a[i]=='p'||a[i]=='P'){ //p or P for Purple

a[i]='P';

}

else if(a[i]=='g'||a[i]=='G'){ //g or G for Green

a[i]='G';

}

else if(a[i]=='y'||a[i]=='Y'){ //y or Y for Yellow

a[i]='Y';

}

else if(a[i]=='o'||a[i]=='O'){ //o or O for Orange

a[i]='O';

}

else{ //Invalid Input

do{ //Do-While invalid input

cout<<"Your input was invalid, Try Again"<<endl; //Try again message

cin>>a[i]; //Input a color

if(a[i]=='r'||a[i]=='R'){ //r or R for Red

a[i]='R';

}

else if(a[i]=='b'||a[i]=='B'){ //b or B for Blue

a[i]='B';

}

else if(a[i]=='p'||a[i]=='P'){ //p or P for Purple

a[i]='P';

}

else if(a[i]=='g'||a[i]=='G'){ //g or G for Green

a[i]='G';

}

else if(a[i]=='y'||a[i]=='Y'){ //y or Y for Yellow

a[i]='Y';

}

else if(a[i]=='o'||a[i]=='O'){ //o or O for Orange

a[i]='O';

}

}while(!((a[i]=='R')||(a[i]=='B')||(a[i]=='Y')||

(a[i]=='P')||(a[i]=='G')||(a[i]=='O')));

}

}

}

//File Display Function

void fileDsp(ofstream &out,string n,string wl,int a){

out<<"Name Tries"<<endl;//Header of file

out<<n<<setw(8)<<a<<setw(8)<<wl<<endl; //Output= name,tries,and win or lose

}