Project17A\_2

Mastermind

**CSC-17A**

**Taniguchi, Devin**

**6/8/2013**

**Introduction**

Title: Mastermind

Mastermind is a game where you have to guess the code of the computer. This code is a 5 color combination consisting of 5 different colors where duplicates of colors can occur. When the game begins the computer makes it code and the user must begin to guess this code with the 5 available colors. For each color guessed in the correct position the user is shown a white peg or W in the program and for each correct color guessed but in the wrong positions the user is shown a red peg.

For example: If the user entered R G G B G (red green green blue green) and the computer’s code was R B Y P P (red blue yellow purple purple) then he would be show one white and one red peg.

Additionally the user only has 10 tries to correctly guess the computers entire code otherwise its game over.

**Summary**

Project size: about 1250 lines with classes included

The number of variables: about 15

This project uses many concepts from the book like the different types of loops and functions, structures and its applications, and finally dynamically created arrays.

What I programmed was a one-sided version of the game, in addition to having a two player mode where the user can play with a friend and try to guess each other’s code. Later on it can be implemented where the computer tries to guess your code.

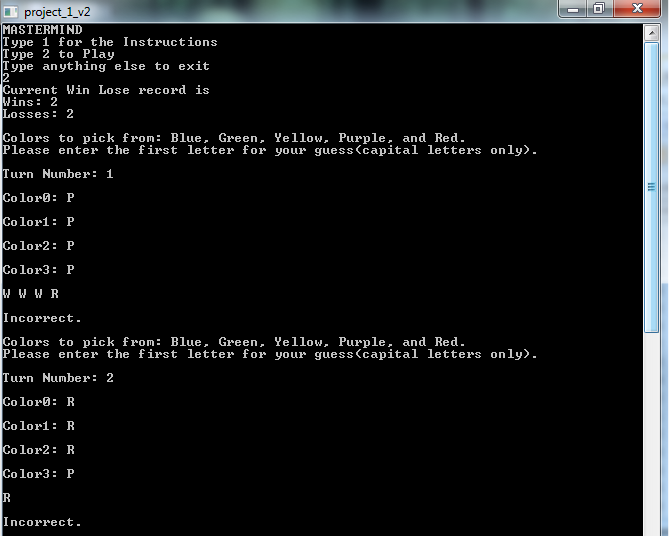
Added in new types of arrays and some more utilities.

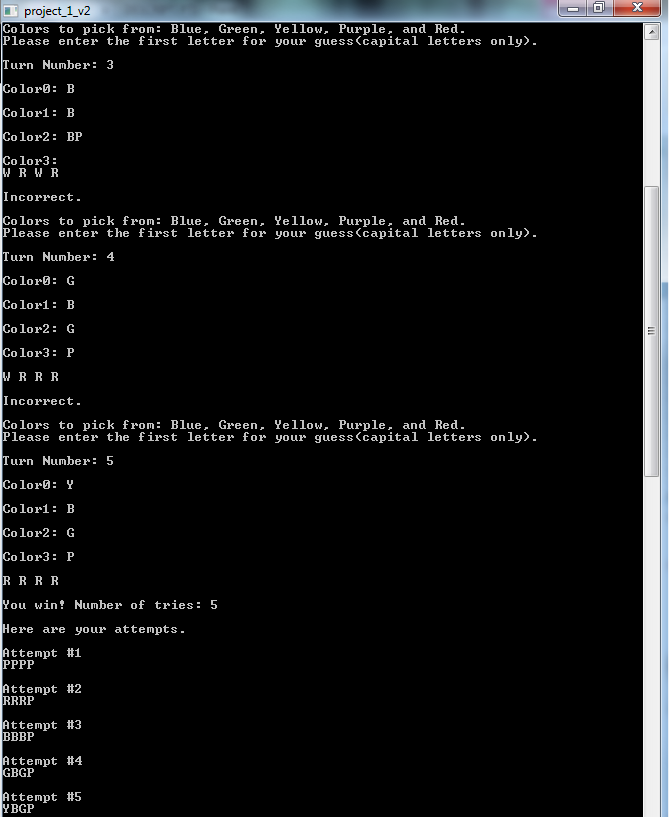
It took about 3 days to finish.

It was a little confusing implementing so many concepts from the book.

**Description**

The main point of the program is to guess the computers code. Mainly what’s in the introduction is what’s required to win. Which should look like this but with an additional color added to the code.





**Pseudo Code**

*Initialize*

*If 1 is pressed*

*Instructions appear*

*If 2 is pressed*

*Game will start*

*Prints win lose ratio*

*Obtain computer code*

*Input guesses for computer code*

*If user guesses a correct color/placement*

*Output an R*

*If user guesses a correct color*

*Output a W*

*If user guess all correct*

*User wins*

*Prints attempts*

*If turns exceed 10*

*User loses*

*Prints attempts*

*If 3 is pressed*

*Two player game will start*

*Players are asked for names and code*

*Player 1 guesses player 2’s code*

*If user guesses a correct color/placement*

*Output an R*

*If user guesses a correct color*

*Output a W*

*If user guess all correct*

*Player 1 wins his try*

*Prints attempts*

*If turns exceed 10*

*Player 1 loses his try*

*Prints attempts*

*Player 2 guesses player 1’s code*

*If user guesses a correct color/placement*

*Output an R*

*If user guesses a correct color*

*Output a W*

*If user guess all correct*

*Player 2 wins his try*

*Prints attempts*

*If turns exceed 10*

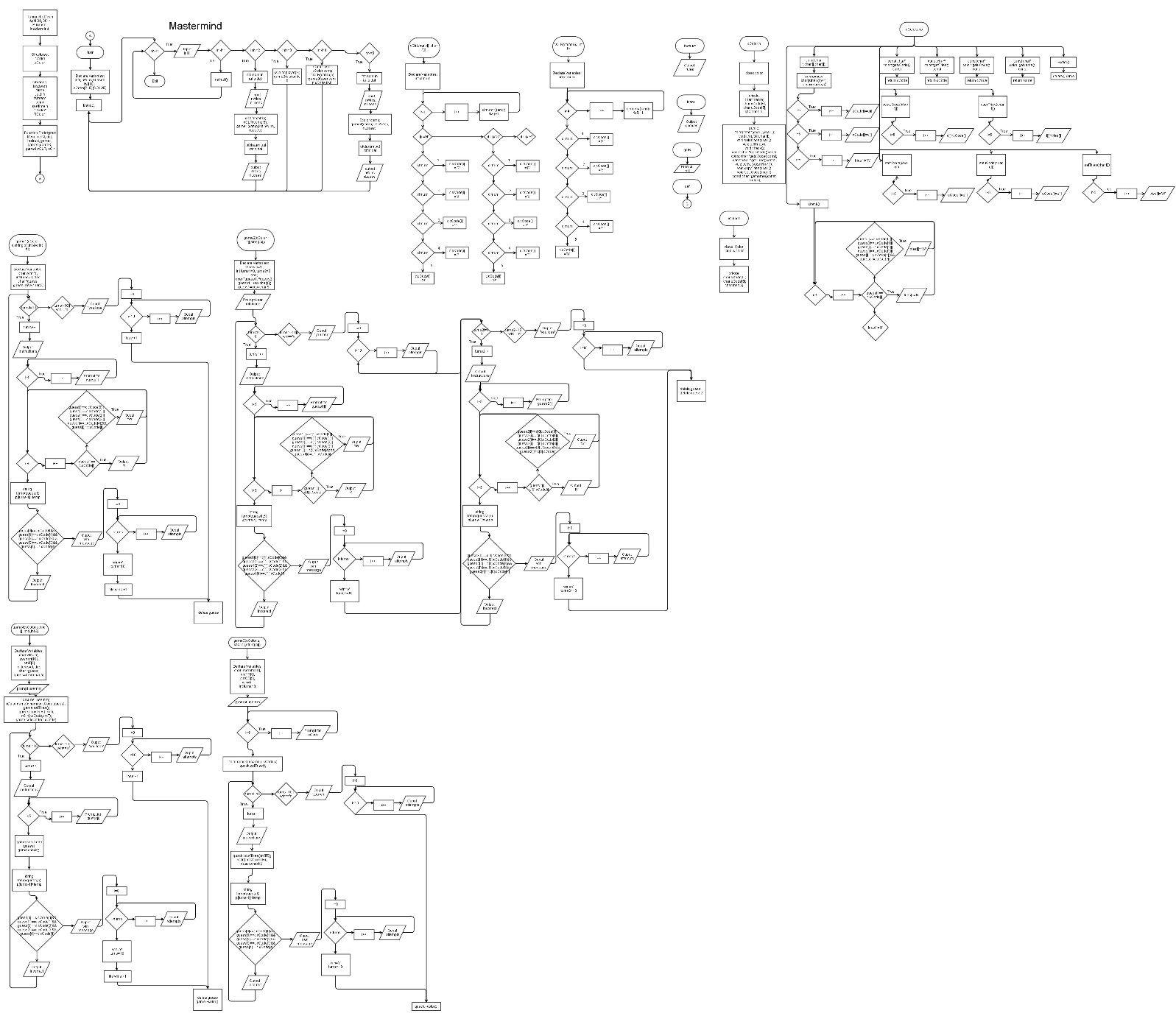
*Player 2 loses his try*

*Prints attempts*

*If anything above 3 is pressed*

*Exit Program*

**Flowchart**

****

**Major Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Name** | **Description** | **Location** |
| Integer | inN | Menu option | Main(), getN() |
|  | clrnum | Number to decide each of the computer’s colors | cClr() |
|  | nWins | Number of wins | Main() |
|  | nLoses | Number of losses | Main() |
|  | turns | Int determining turns | game1() |
|  | turns1 | Number of turns: Player1 | Game2() |
|  | Turns2 | Number of turns: Player2 | Game2() |
| Character | cCode | Computer/User’s inputted colors | Color structures and classes |
|  | uCode | User’s inputted colors | Color classes |
|  | Win | Determines if you have won or not | Game1(),game2 |
|  | Guess | Dynamic array for guessing | Game1() |
|  | Guess1 | Dynamic array for guessing:Player1 | Game2() |
|  | Guess2 | Dynamic array for guessing:Player2 | Game2() |
|  | tries | Array for holding results | cColor.h,iColor.h |
|  | cmCD | Array to hold cCode copy | Game3,game4 |
|  | cmTR | Array to hold tries copy | Game3,game4 |
| String | List | Stores user inputs | main() |
|  | name | Player’s name | Color structure |
| Structure | cColor | Holds name string and code char array | Beginning |
|  | sColor | Holds name pointer and code char array | Beginning |
| Class | color |  |  |
|  | iColor | Child of the color class |  |

**C++ Constructs**

|  |  |  |
| --- | --- | --- |
| **Chapter** | **New syntax or Keyword** | **Location** |
| 2 | Int data type | inN, clrnum, turns |
|  | Char data type | cCode, win, |
|  | Arithmetic Operators | clrnum=(rand()%5)+1 |
|  | Comments | //Random number 0 to 5 |
|  | String data type | string temp(c[1],4); |
| 3 | The cin Object | cin>>guess[i];, . . . |
| 4 | The if Statement | if(guess[i]==c.cCode[i]), . . . |
|  | The if/else Statement | if(guess[0]==c.cCode[0]&&. . .  else{} |
|  | Logical Operators | guess[0]== c.cCode[1] . . . |
|  | Menus | do{  Menu();  inN=getN();  //reads in file  //Tells Current win lose ratio    switch(inN){  case 1: instruct();break;  case 2:{  //Read file in  ifstream in;  in.open("ratio.dat");  in>>nWins;  in>>nLoses;  cout<<"Current Win Lose record is"<<endl;  cout<<"Wins: "<<nWins<<endl;  cout<<"Losses: "<<nLoses<<endl;  cout<<endl;  //Play Game  color comp;  cCLR(comp,5);  game1(comp,list,nWins,nLoses);  //Output to same file  ofstream out;  out.open("ratio.dat");  out<<nWins<<endl;  out<<nLoses<<endl;  //Close files  in.close();  out.close();  break;  }  case 3:{  color player[2];  game2(player,list);  }  default: def(inN);}  }while(inN<4);  return 0; |
|  | The switch Statement | switch(clrnum){//Changes numbers into cCode  case 1:{ . . . |
| 5 | The Increment and Decrement Operators | i++ |
|  | The while Loop | while(turns!=10) |
|  | The do-while Loop | do{ . . . }while(inN<4); |
|  | The for Loop | for(int i=0;i<5;i++){} |
| 6 | Function Prototypes | void instruct();  void game(); |
|  | Return statement | return inN |
|  | Passing by value | void cCLR(color &,int); |
|  | Passing by reference | void game1(color,string[],int &,int &) |
| 7 | Arrays | list[10] |
|  | Array as function arguments | void game2(color,string []) |
| 9 | Dynamic Memory Allocation | Char \*guess;  Guess=new char[5]; |
| 10 | C-strings | name=new char[strlen(n)+1];  strcpy(name,n); |
| 11 | Accessing Structure Members | c.cCode[i]='R'; |
|  | Initializing a Structure | struct color{  string name;  char cCode[5];  }; |
|  | Arrays of Structures | cColor player[2]; |
|  | Structures as Function Arguments | void cCLR(color &,int); |
| 13 | Introduction to Classes | class color{  ….  } |
|  | Defining an Instance of a Class | class color{  ….  } |
|  | Why Have Private Members? | private:  char \*name;  char cCode[5];  char uCode[5];  char tries[5]; |
|  | Constructors | color(char \*,char [],char []); |
|  | Destructors | ~color(); |
| 14 | Copy Constructors | void copycCode(char []); |
| 15 | What Is Inheritance? | class iColor : public color{ |
|  | Constructors and Destructors in Base and Derived Classes | iColor(char \*n,char u[],char c[]):color(n,u,c){} |
| 16 | Exceptions | try{  for(int i=0;i<5;i++){  cout<<"Color"<<i<<": ";  cin>>guess[i];  cout<<endl;  …….  }  else  {  string exeString="ERROR: Correct choice not entered. \n";  throw exeString;  }  }  }    catch(string exeString)  {  cout<<exeString;  break;  } |

**References**

1. Textbook
2. <http://en.wikipedia.org/wiki/Mastermind_(board_game)>

**Program**

/\*

\* File: main.cpp

\* Author: Devin Taniguchi

\* Created on April 27, 2014, 9:32 PM

\* Purpose: Project

\*/

//Libraries

#include <cstdlib>

#include <iostream>

#include <ctime>

#include <fstream>

#include <string>

#include <fstream>

#include <sys/time.h>

#include "cColor.h"

#include "iColor.h"

using namespace std;

//Structures

//For regular games

struct cColor{

string name;

char cCode[5];

};

//For class using games

struct sColor{

char \*name;

char cCode[5];

};

//No Global Constants

//Function Prototype

void Menu();

int getN();

void def(int);

void instruct(); //Instructions

void game1(cColor,string [],int &,int &); //Function to hold game

void game2(cColor [],string []); //Function for head to head

void game3(sColor,char [],string []); //Function for comp guessing

void game4(sColor,string [],int &,int &); //Function to hold game

void cCLR(cColor &,int); //RNG for code

void nClr(char [],char []); //Selective RNG

//Execution Starts here

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

//Declare variables

string list[10];

int inN,

nWins,

nLoses;

//Set seed

struct timeval time;

gettimeofday(&time,NULL);

srand((time.tv\_sec \* 100) + (time.tv\_usec / 100));

//Start menu

do{

Menu();

inN=getN();

//reads in file

//Tells Current win lose ratio

switch(inN){

case 1: instruct();break;

case 2:{

//Read file in

ifstream in;

in.open("ratio.dat");

in>>nWins;

in>>nLoses;

cout<<endl;

cout<<"Current Win Lose record is"<<endl;

cout<<"Wins: "<<nWins<<endl;

cout<<"Losses: "<<nLoses<<endl;

cout<<endl;

//Play Game

cColor comp;

cCLR(comp,5);

game1(comp,list,nWins,nLoses);

//Output to same file

ofstream out;

out.open("ratio.dat");

out<<nWins<<endl;

out<<nLoses<<endl;

//Close files

in.close();

out.close();

break;

}

case 3:{

//Structure for game two

cColor player[2];

//Start game 2

game2(player,list);

break;

}

case 4:{

//Strutures for game 3

sColor user;

cColor comp;

//RNG for initial comp code

cCLR(comp,5);

cout<<endl;

//Start game 3

game3(user,comp.cCode,list);

break;

}

case 5:{

//Read file in

ifstream in;

in.open("ratio.dat");

in>>nWins;

in>>nLoses;

cout<<endl;

cout<<"Current Win Lose record is"<<endl;

cout<<"Wins: "<<nWins<<endl;

cout<<"Losses: "<<nLoses<<endl;

cout<<endl;

//Play Game

sColor comp;

game4(comp,list,nWins,nLoses);

//Output to same file

ofstream out;

out.open("ratio.dat");

out<<nWins<<endl;

out<<nLoses<<endl;

//Close files

in.close();

out.close();

break;

}

default: def(inN);}

}while(inN<6);

return 0;

return 0;

}

void nClr(char c[],char t[]){

int clrnum;

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

//If its completely wrong

if(t[i]=='B'){

char wrong=c[i];

//Random number 0 to 5

clrnum=(rand()%5)+1;

//Changes numbers into colors

switch(clrnum){

case 1:{

//Red

c[i]='R';

break;

}

case 2:{

//Blue

c[i]='B';

break;

}

case 3:{

//Yellow

c[i]='Y';

break;

}

case 4:{

//Purple

c[i]='P';

break;

}

case 5:{

//Green

c[i]='G';

break;

}

}

}

//If it was the right color

//But not right place

if(t[i]=='W'){

//Random number 0 to 5

clrnum=(rand()%5)+1;

//Changes numbers into colors

switch(clrnum){

case 1:{

//Red

c[i]='R';

break;

}

case 2:{

//Blue

c[i]='B';

break;

}

case 3:{

//Yellow

c[i]='Y';

break;

}

case 4:{

//Purple

c[i]='P';

break;

}

case 5:{

//Green

c[i]='G';

break;

}

}

}

//If right color and place

if(t[i]=='R'){

}

}

}

void cCLR(cColor &c, int k){

int clrnum;

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

//Random number 0 to 5

clrnum=(rand()%5)+1;

//Changes numbers into colors

switch(clrnum){

case 1:{

//Red

c.cCode[i]='R';

break;

}

case 2:{

//Blue

c.cCode[i]='B';

break;

}

case 3:{

//Yellow

c.cCode[i]='Y';

break;

}

case 4:{

//Purple

c.cCode[i]='P';

break;

}

case 5:{

//Green

c.cCode[i]='G';

break;

}

}

}

}

void game1(cColor c,string p[],int &w,int &l){

char win='n';

int des,

turns=0;

char \*guess;

guess=new char[5];

//Turn limit is 10, stops there

while(turns!=10){

turns++;

//Rules

cout<<"Colors to pick from: Blue, Green,";

cout<<" Yellow, Purple, and Red."<<endl;

cout<<"Please enter the first letter ";

cout<<"for your guess(capital letters only)."<<endl;

cout<<endl;

cout<<"Turn Number: "<<turns<<endl;

cout<<endl;

//User enter 4 colors

try{

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

cout<<"Color"<<i<<": ";

cin>>guess[i];

cout<<endl;

if(guess[i]=='P'||

guess[i]=='B'||

guess[i]=='G'||

guess[i]=='R'||

guess[i]=='Y'){

}

else

{

string exeString="ERROR: Correct choice not entered. \n";

throw exeString;

}

}

}

catch(string exeString)

{

cout<<exeString;

break;

}

//Checks if any are right color and position

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

if(guess[i]==c.cCode[i]){

cout<<"R"<<" ";

}

//Checks if any are the right color

else{

if((guess[i]==c.cCode[0]||

guess[i]==c.cCode[1]||

guess[i]==c.cCode[2]||

guess[i]==c.cCode[3]||

guess[i]==c.cCode[4])&&

guess[i]!=c.cCode[i]){

cout<<"W"<<" ";

}

}

}

cout<<endl;

cout<<endl;

//Sends array to a string and puts string into an array

//temp=c[1][0]<<c[1][1]<<c[1][2]<<c[1][3];

//l[turns-1]=temp;

string temp(guess,5);

p[turns-1]=temp;

//Confirms if all of the colors are right

if(guess[0]==c.cCode[0]&&

guess[1]==c.cCode[1]&&

guess[2]==c.cCode[2]&&

guess[3]==c.cCode[3]&&

guess[4]==c.cCode[4]){

cout<<"You win! Number of tries: "<<turns<<endl;

cout<<endl;

//Outputs different tries

cout<<"Here are your attempts."<<endl;

cout<<endl;

//Prints out each letter then ends line

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

cout<<"Attempt #"<<i+1<<endl;

cout<<p[i]<<endl;

cout<<endl;

}

//Too make sure lose doesn't print

win='y';

turns=10;

//changing win lose ratio

w+=1;

}

else{

cout<<"Incorrect."<<endl;

cout<<endl;

}

}

//If turns exceed 10 you lose

if(turns==10&&win=='n'){

cout<<"You lost."<<endl;

cout<<endl;

//Outputs different tries

cout<<"Here are your attempts."<<endl;

cout<<endl;

//Prints out each letter then ends line

for(int i=0;i<10;i++){

cout<<"Attempt #"<<i+1<<endl;

cout<<p[i]<<endl;

cout<<endl;

}

//Changing Win Lose Ratio

l+=1;

}

cout<<endl;

delete [] guess;

}

void game2(cColor c[],string p[]){

//Declare variables

char win='n';

int des,

turns1=0,

turns2=0;

char \*guess1,\*guess2;

guess1=new char[5];

guess2=new char[5];

//Info prompt

cout<<"Player 1 please enter your name."<<endl;

cin>>c[0].name;

cin.ignore();

cout<<"Player 2 please enter your name."<<endl;

cin>>c[1].name;

cin.ignore();

cout<<c[0].name;

cout<<" please enter your color combo while ";

cout<<c[1].name;

cout<<" looks away."<<endl;

//User 1 enters colors

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

cout<<"Color"<<i<<": ";

cin>>c[0].cCode[i];

cout<<endl;

}

cout<<c[1].name;

cout<<" please enter your color combo while ";

cout<<c[0].name;

cout<<" looks away."<<endl;

//User 2 enters colors

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

cout<<"Color"<<i<<": ";

cin>>c[1].cCode[i];

cout<<endl;

}

//Turn limit is 10, stops there

cout<<"Now it's time for Player 1 to guess Player 2's combo."<<endl;

while(turns1!=10){

turns1++;

//Rules

cout<<"Colors to pick from: Blue, Green,";

cout<<" Yellow, Purple, and Red."<<endl;

cout<<"Please enter the first letter ";

cout<<"for your guess(capital letters only)."<<endl;

cout<<endl;

cout<<"Turn Number: "<<turns1<<endl;

cout<<endl;

//User enter 5 colors

try{

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

cout<<"Color"<<i<<": ";

cin>>guess1[i];

cout<<endl;

if(guess1[i]=='P'||

guess1[i]=='B'||

guess1[i]=='G'||

guess1[i]=='R'||

guess1[i]=='Y'){

}

else

{

//Error message

string exeString="ERROR: Correct choice not entered. \n";

throw exeString;

}

}

}

catch(string exeString)

{

//Exits current game

cout<<exeString;

break;

}

//Checks if any are right color and position

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

if(guess1[i]==c[1].cCode[i]){

cout<<"R"<<" ";

}

//Checks if any are the right color

else{

if((guess1[i]==c[1].cCode[0]||

guess1[i]==c[1].cCode[1]||

guess1[i]==c[1].cCode[2]||

guess1[i]==c[1].cCode[3]||

guess1[i]==c[1].cCode[4])&&

guess1[i]!=c[1].cCode[i]){

cout<<"W"<<" ";

}

}

}

cout<<endl;

cout<<endl;

//Sends array to a string and puts string into an array

//temp=c[1][0]<<c[1][1]<<c[1][2]<<c[1][3];

//l[turns-1]=temp;

string temp(guess1,5);

p[turns1-1]=temp;

//Confirms if all of the colors are right

if(guess1[0]==c[1].cCode[0]&&

guess1[1]==c[1].cCode[1]&&

guess1[2]==c[1].cCode[2]&&

guess1[3]==c[1].cCode[3]&&

guess1[4]==c[1].cCode[4]){

cout<<"You win! Number of tries: "<<turns1<<endl;

cout<<endl;

//Outputs different tries

cout<<"Here are your attempts."<<endl;

cout<<endl;

//Prints out each letter then ends line

for(int i=0;i<turns1;i++){

cout<<"Attempt #"<<i+1<<endl;

cout<<p[i]<<endl;

cout<<endl;

}

//Too make sure lose doesn't print

win='y';

turns1=10;

}

else{

cout<<"Incorrect."<<endl;

cout<<endl;

}

}

//If turns exceed 10 you lose

if(turns1==10&&win=='n'){

cout<<"You lost."<<endl;

cout<<endl;

//Outputs different tries

cout<<"Here are your attempts."<<endl;

cout<<endl;

//Prints out each letter then ends line

for(int i=0;i<10;i++){

cout<<"Attempt #"<<i+1<<endl;

cout<<p[i]<<endl;

cout<<endl;

}

}

//Turn limit is 10, stops there

cout<<"Now it's time for Player 2 to guess Player 1's combo."<<endl;

while(turns2!=10){

turns2++;

//Rules

cout<<"Colors to pick from: Blue, Green,";

cout<<" Yellow, Purple, and Red."<<endl;

cout<<"Please enter the first letter ";

cout<<"for your guess(capital letters only)."<<endl;

cout<<endl;

cout<<"Turn Number: "<<turns2<<endl;

cout<<endl;

//User enter 5 colors

try{

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

cout<<"Color"<<i<<": ";

cin>>guess2[i];

cout<<endl;

if(guess2[i]=='P'||

guess2[i]=='B'||

guess2[i]=='G'||

guess2[i]=='R'||

guess2[i]=='Y'){

}

else

{

//Error message

string exeString="ERROR: Correct choice not entered. \n";

throw exeString;

}

}

}

catch(string exeString)

{

//Exits current game

cout<<exeString;

break;

}

//Checks if any are right color and position

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

if(guess2[i]==c[0].cCode[i]){

cout<<"R"<<" ";

}

//Checks if any are the right color

else{

if((guess2[i]==c[0].cCode[0]||

guess2[i]==c[0].cCode[1]||

guess2[i]==c[0].cCode[2]||

guess2[i]==c[0].cCode[3]||

guess2[i]==c[0].cCode[4])&&

guess2[i]!=c[0].cCode[i]){

cout<<"W"<<" ";

}

}

}

cout<<endl;

cout<<endl;

//Sends array to a string and puts string into an array

//temp=c[1][0]<<c[1][1]<<c[1][2]<<c[1][3];

//l[turns-1]=temp;

string temp(guess1,4);

p[turns1-1]=temp;

//Confirms if all of the colors are right

if(guess2[0]==c[0].cCode[0]&&

guess2[1]==c[0].cCode[1]&&

guess2[2]==c[0].cCode[2]&&

guess2[3]==c[0].cCode[3]&&

guess2[4]==c[0].cCode[4]){

cout<<"You win! Number of tries: "<<turns2<<endl;

cout<<endl;

//Outputs different tries

cout<<"Here are your attempts."<<endl;

cout<<endl;

//Prints out each letter then ends line

for(int i=0;i<turns2;i++){

cout<<"Attempt #"<<i+1<<endl;

cout<<p[i]<<endl;

cout<<endl;

}

//To make sure lose doesn't print

win='y';

turns2=10;

}

else{

cout<<"Incorrect."<<endl;

cout<<endl;

}

}

//If turns exceed 10 you lose

if(turns2==10&&win=='n'){

cout<<"You lost."<<endl;

cout<<endl;

//Outputs different tries

cout<<"Here are your attempts."<<endl;

cout<<endl;

//Prints out each letter then ends line

for(int i=0;i<10;i++){

cout<<"Attempt #"<<i+1<<endl;

cout<<p[i]<<endl;

cout<<endl;

}

}

cout<<endl;

delete [] guess1;

delete [] guess2;

}

void game3(sColor u, char c[],string p[]){\

//Declare variable

char usernm[10],

cmTR[5],

cmCD[5],

win='n';

int turns=0;

//Enter name

cout<<"Please enter your name."<<endl;

cin>>usernm;

u.name=usernm;

//User enters colors

//Rules

cout<<"Colors to pick from: Blue, Green,";

cout<<" Yellow, Purple, and Red."<<endl;

cout<<"Please enter the letters ";

cout<<"for your guess(capital letters only)."<<endl;

cout<<endl;

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

cout<<"Color"<<i<<": ";

cin>>u.cCode[i];

cout<<endl;

}

//Initialize class

color quack(u.name,u.cCode,c);

//Declare tries

quack.setTries();

//Copy class code

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

cmCD[i]=c[i];

}

//Start game

while(turns!=10){

cout<<"The computer will now guess."<<endl;

turns++;

//Set tries

quack.copyTries(cmTR);

//Copy class code

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

cout<<cmTR[i];

}

cout<<endl;

//Set comp colors

nClr(cmCD,cmTR);

quack.setcCode(cmCD);

//quack.copycCode(cmCD);

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

cout<<cmCD[i];

}

cout<<endl;

//Checks if any are right color and position

quack.check();

cout<<endl;

//Sends array to a string and puts string into an array

//temp=c[1][0]<<c[1][1]<<c[1][2]<<c[1][3];

//l[turns-1]=temp;

string temp(cmCD,5);

p[turns-1]=temp;

//Confirms if all of the colors are right

if(cmCD[0]==u.cCode[0]&&

cmCD[1]==u.cCode[1]&&

cmCD[2]==u.cCode[2]&&

cmCD[3]==u.cCode[3]&&

cmCD[4]==u.cCode[4]){

cout<<"The computer could guessed your code."<<endl;

cout<<"You Lose."<<endl;

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

cout<<endl;

//Outputs different tries

cout<<"Here are the computers attempts."<<endl;

cout<<endl;

//Prints out each letter then ends line

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

cout<<"Attempt #"<<i+1<<endl;

cout<<p[i]<<endl;

cout<<endl;

}

//Too make sure lose doesn't print

win='y';

turns=10;

}

else{

cout<<"Incorrect."<<endl;

cout<<endl;

}

}

if(turns==10&&win=='n'){

cout<<"The computer could not guess your code."<<endl;

cout<<"You Win."<<endl;

cout<<endl;

//Outputs different tries

cout<<"Here are the computer's attempts."<<endl;

cout<<endl;

//Prints out each letter then ends line

for(int i=0;i<10;i++){

cout<<"Attempt #"<<i+1<<endl;

cout<<p[i]<<endl;

cout<<endl;

}

}

//Delete

quack.~color();

}

void game4(sColor c,string p[],int &w,int &l){

char usernm[10],

cmT[5],

win='n';

int des,

turns=0;

char \*guess;

guess=new char[5];

//Enter name

cout<<"Please enter your name."<<endl;

cin>>usernm;

c.name=usernm;

iColor game(c.name,c.cCode,guess);

game.setTries();

game.copyTries(cmT);

nClr(c.cCode,cmT);

game.setcCode(c.cCode);

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

cout<<c.cCode[i];

}

cout<<endl;

//Turn limit is 10, stops there

while(turns!=10){

turns++;

//Rules

cout<<"Colors to pick from: Blue, Green,";

cout<<" Yellow, Purple, and Red."<<endl;

cout<<"Please enter the first letter ";

cout<<"for your guess(capital letters only)."<<endl;

cout<<endl;

cout<<"Turn Number: "<<turns<<endl;

cout<<endl;

//User enter 4 colors

try{

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

cout<<"Color"<<i<<": ";

cin>>guess[i];

cout<<endl;

if(guess[i]=='P'||

guess[i]=='B'||

guess[i]=='G'||

guess[i]=='R'||

guess[i]=='Y'){

}

else

{

string exeString="ERROR: Correct choice not entered. \n";

throw exeString;

}

}

}

catch(string exeString)

{

cout<<exeString;

break;

}

game.setuCode(guess);

game.check();

cout<<game.getTries();

cout<<endl;

cout<<endl;

//Sends array to a string and puts string into an array

//temp=c[1][0]<<c[1][1]<<c[1][2]<<c[1][3];

//l[turns-1]=temp;

string temp(guess,5);

p[turns-1]=temp;

//Confirms if all of the colors are right

if(guess[0]==c.cCode[0]&&

guess[1]==c.cCode[1]&&

guess[2]==c.cCode[2]&&

guess[3]==c.cCode[3]&&

guess[4]==c.cCode[4]){

cout<<"You win! Number of tries: "<<turns<<endl;

cout<<endl;

//Outputs different tries

cout<<"Here are your attempts."<<endl;

cout<<endl;

//Prints out each letter then ends line

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

cout<<"Attempt #"<<i+1<<endl;

cout<<p[i]<<endl;

cout<<endl;

}

//Too make sure lose doesn't print

win='y';

turns=10;

//changing win lose ratio

w+=1;

}

else{

cout<<"Incorrect."<<endl;

cout<<endl;

}

}

//If turns exceed 10 you lose

if(turns==10&&win=='n'){

cout<<"You lost."<<endl;

cout<<endl;

//Outputs different tries

cout<<"Here are your attempts."<<endl;

cout<<endl;

//Prints out each letter then ends line

for(int i=0;i<10;i++){

cout<<"Attempt #"<<i+1<<endl;

cout<<p[i]<<endl;

cout<<endl;

}

//Changing Win Lose Ratio

l+=1;

}

cout<<endl;

delete [] guess;

game.~color();

}

void instruct(){

//Rules

cout<<"The main objective is to solve the computer's code."<<endl;

cout<<endl;

cout<<"With a 4 color combination for a code and 5 colors"<<endl;

cout<<" to choose from you have the guess the combo in 10 "<<endl;

cout<<"or you lose."<<endl;

cout<<endl;

cout<<"Additionally each time you guess you will be told "<<endl;

cout<<"Whether you have a right color in the right "<<endl;

cout<<"position or a right color in the wrong position "<<endl;

cout<<"by the Rs and Ws at the end of your code."<<endl;

cout<<"If there are no R or W in a certain place that means "<<endl;

cout<<"that that letter does not exist in the combination."<<endl;

cout<<"(Rs for right spot and color and Ws for only right color)"<<endl;

cout<<endl;

}

void Menu(){

cout<<"MASTERMIND"<<endl;

cout<<"Type 1 for the Instructions"<<endl;

cout<<"Type 2 to guess the computer's code"<<endl;

cout<<"Type 3 to Play against a friend on the same computer"<<endl;

cout<<"Type 4 to have the computer guess your code"<<endl;

cout<<"Type 5 for choice 2 but with class"<<endl;

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

}

int getN(){

int inN;

cin>>inN;

return inN;

}

void def(int inN){

cout<<"You typed "<<inN<<" to exit the program"<<endl;

}

**cColor.h**

/\*

\* File: color.h

\* Author: Devin Taniguchi

\* Created on June 2, 2014, 11:31 AM

\*/

#ifndef COLOR\_H

#define COLOR\_H

#include <string>

#include <cstring>

using namespace std;

class color{

private:

char \*name;

char cCode[5];

char uCode[5];

char tries[5];

public:

//Initial info

color(char \*,char [],char []);

//Computer color combo

void cmpClr(char []);

//User color combo

void setuCode(char []);

//tries construct

void setTries();

//Check computer combo

void check();

//Return computer colors

const char \*getcCode()const;

//Return user colors

const char \*getuCode()const;

//Return tries

const char \*getTries()const;

//Copy comp colors

void copycCode(char []);

//Copy tries

void copyTries(char []);

//Resets cCode

void setcCode(char []);

//Return name

const char \*getname()const;

//Delete

~color();

};

#endif /\* COLOR\_H \*/

**cColor.cpp**

/\*

\* File: color.h

\* Author: Devin Taniguchi

\* Created on June 2, 2014, 11:31 AM

\* cColor specification

\*/

#include "cColor.h"

#include <string>

#include <cstring>

#include <iostream>

#include <time.h>

using namespace std;

//Copy user info into class

color::color(char \*n,char u[],char c[]){

name=new char[strlen(n)+1];

strcpy(name,n);

//copy user code

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

uCode[i]=u[i];

}

//copy comp code

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

cCode[i]=c[i];

}

//initial tries

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

tries[i]='W';

}

}

//Returns computer Color combo

const char\* color::getcCode() const{

return cCode;

}

//Returns user Color combo

const char\* color::getuCode() const{

return uCode;

}

//Returns tries

const char\* color::getTries() const{

return tries;

}

//Returns name

const char\* color::getname() const{

return name;

}

//Copies user Color combo

void color::copycCode(char c[]){

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

c[i]=cCode[i];

}

}

//Copies tries results

void color::copyTries(char t[]){

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

t[i]=tries[i];

}

}

//Resets the comp color combo

void color::setcCode(char c[]){

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

cCode[i]=c[i];

}

}

//Sets USer combo

void color::setuCode(char u[]){

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

uCode[i]=u[i];

}

}

//Sets tries

void color::setTries(){

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

tries[i]='W';

}

}

//Check computer combo

void color::check(){

//Checks if any are right color and position

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

if(cCode[i]==uCode[i]){

tries[i]='R';

}

//Checks if any are the right color

else{

if((cCode[i]==uCode[0]||

cCode[i]==uCode[1]||

cCode[i]==uCode[2]||

cCode[i]==uCode[3]||

cCode[i]==uCode[4])&&

cCode[i]!=uCode[i]){

tries[i]='W';

}

//If none then bad

else{

tries[i]='B';

}

}

}

}

//Delete

color::~color(){

delete []name;

}

**iColor.h**

/\*

\* File: iColor.h

\* Author: Devin

\*

\* Created on June 8, 2014, 11:19 PM

\*/

#ifndef ICOLOR\_H

#define ICOLOR\_H

#include "cColor.h"

#include <string>

using namespace std;

class iColor : public color{

private:

char cCode[5];

char uCode[5];

char tries[5];

public:

//Constructor #2

iColor(char \*n,char u[],char c[]):color(n,u,c){}

//Resets cCode

void setcCode(char []);

//Return computer colors

const char \*getcCode()const;

//User color combo

void setuCode(char []);

//Return user colors

const char \*getuCode()const;

//Return tries

const char \*getTries()const;

//Check combo

void check();

//Initial info

};

#endif /\* ICOLOR\_H \*/

**iColor.cpp**

/\*

\* File: iColor.h

\* Author: Devin

\* Created on June 8, 2014, 11:19 PM

\* Specification for iColor class

\*/

#include "iColor.h"

#include "cColor.h"

#include <string>

#include <cstring>

#include <time.h>

#include <iostream>

using namespace std;

//Resets the comp color combo

void iColor::setcCode(char c[]){

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

cCode[i]=c[i];

}

}

//Returns computer Color combo

const char\* iColor::getcCode() const{

return cCode;

}

//Sets USer combo

void iColor::setuCode(char u[]){

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

uCode[i]=u[i];

}

}

//Returns user Color combo

const char\* iColor::getuCode() const{

return uCode;

}

//Returns tries

const char\* iColor::getTries() const{

return tries;

}

//Check combo

void iColor::check(){

//Checks if any are right color and position

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

if(uCode[i]==cCode[i]){

tries[i]='R';

}

//Checks if any are the right color

else{

if((uCode[i]==cCode[0]||

uCode[i]==cCode[1]||

uCode[i]==cCode[2]||

uCode[i]==cCode[3]||

uCode[i]==cCode[4])&&

uCode[i]!=cCode[i]){

tries[i]='W';

}

//If none then bad

else{

tries[i]='B';

}

}

}

}