

Project 2:Mastermind
Class: CSC-17A 42824
Name: Kayla Rodriguez
Date: 06/09/2016

Introduction

Title: Mastermind

Mastermind is a code breaking game for users to try and guess four different colors the computer picked by random. The user uses hints given by the computer telling them whether their color is in the right spot, or how many of the colors are correct. The user inputs how many attempts they are willing to play, and to win they must guess the correct colors before 10 tries.

Project Description and Checklist:

This project demonstrates the different constructs and learned in class. It utilizes concepts from chapters nine through 12 but does include a few concepts from chapter 13.

Total Line Count: 1029 lines

Concepts Utilized For Project From Ch 9-16:

- Structures
- Pointers with arrays
- Arrays of structures
- Structure types passed into functions
- String objects
- String class
- Reading a binary file
- Opening a binary file
- Writing a binary file
- Allocated memory
- Functions that return structure object variables
- Inputting a structure variable
- Outputting a structure variable
- Classes
- Using overloaded operators in a class
- Static variables
- Inheritance
- Polymorphism
- Templates
- Exceptions

Additional Concepts Utilized

- Classes
- Accessor and Mutator Functions within a class
- Separate cpp files and header files for the class
- Private member variables
- Public member functions
- Vectors
- Push_back() function
- Tables
- Character arrays

- Returning a char pointer within a function
- Converting character arrays to strings
- Random function generator
- File functions

Variables Used in Main

Constants:

Type	Variable Name	Description
const int	SIZE=4	Size variable of four used for for loops, for example, to input the four colors, to generate the computer's colors
char	gmelmt=10	This is the game limit. You lose if you cannot guess in 10 tries, but if the user cannot guess in 10 tries they can increment this constant in order to give them another turn to guess (even though they lost) this is demonstrated in the overloaded operators section of the program

File Variables:

fstream	infile	This variable holds the file containing the instructions for the game
string	instr	This is the string that the file instructions are placed into to be read
fstream	out	This is used for the results of the game file or just the output file
fstream	leader	This is used for the leaderboard file containing the winners and their scores
string	name	This variable holds the contents of the leader file to be outputted

Menu Variables:

int	inN	What the user inputs to input a menu option
bool	ReDsply=true	Variable allows the user to redisplay the menu or exit the game menu

Other Main Variables

int	nTrys=0	Counter to count how many tries the user has used up
UserColor	*clrPick	Array to store all the user color picks and turn picks stored in a class type variable
char	*userChar	Array to store all the colors the user picked in character form
char	*comChar	Array to store all the computer's color picks in character form
ComColor	data	Player data stored in the computer class type

Variables Used in the Game Function:

Type	Variable Name	Description
string	order[SIZE]	Contains an array of the string names of order
string	options[8]	Array which contains the only possible string options the computer or the user can pick
char	optChar[8]	Array which contains the only character options to convert the strings to characters for the user and computer
vector<string>	list	Vector which converts characters to one condensed string to be placed onto a table
ComColor	*cColor	Pointer that stores all the computer's colors in the ComColor class

char	choice	Choice gives the user the option to increment the number of games they can play when they lose (this is used in the demonstration of the overloaded operator)
UserColor	limit	Object used to be incremented (overloaded operator)
char	repeat	Character choice to allow the player to play again if they want
int	win=0	Win for the player, variable holds if the player has won
int	lose=0	Lose for the player, variable holds if the player has lost
int	games=0	Number of games for the player, variables holds how many games the user has played
bool	gmeOutcome	Variable flags the game outcome as either a win or a lose
string	username	Holds the player username

Function Prototypes

Functions Used In the Game:

Type	Variable	Description
void	game	This function plays the Mastermind game
char*	compic	Returns a random character for computer generated pick, returns the character representation of the computer's pick
UserColor*	input	User inputs their colors in this function and the spots they want the colors to be in. returns an array of these picks of type UserColor

char*	input2	Function returns the user's pick in character form to be outputted to a table
void	switchH	Hints function, displays hints after each guess
void	reppic	Representation of previous picks, displays table, and attempt number
void	results	Displays win or loss message to the user
string	aryToStr	Function converts a character array to a string

Functions Used for Writing and Reading Files in the Game:

void	writeFile	This Function writes the output file containing the table of choices, the user's attempt number and whether they won or lost
void	readFile	Reads the contents of the instruction file to show the user the instructions

Functions Used for the Leader Board:

void	readLdr	Function reads the contents of the leaderboard file
void	lder	Function allows for the user to enter their name to be sorted onto a leader board, writes their name and score onto the leaderboard
void	lderOutput	Function outputs the name and score of the player on the leader board file

Functions Used to Keep Track of the Player's Data:

void	plyerInfo	Sets the username for the player
void	findPlyr	Finds any given player
void	filePlyr	Reads and outputs contents of the player stats
void	read	Reads the player stats file, separate read function
void	writeData	Writes the player data to the player stats file
void	parce	Separates the player name from the player score (parces the file)

Functions Used at the end of the Game:

void	endGame	Displays the end game message
void	message	Function Displays the Mastermind Logo if the player wants

Menu Functions:

void	Menu	Function Outputs the menu choices
void	def	Displays what the user pressed to exit the game
void	getN	Function that takes in the userchoice of which number on the menu to pick

Concepts Learned and Location in the Code

Concept Learned	Chapter (Gaddis)	Location
Structures	Chapter 11	Used to return a character to generate computer generated picks within the compic function
Pointers with arrays	Chapter 9	Used in many places

		of the code to store color picks, char pointers used, string pointers, and structure pointers used
Arrays of structures	Chapter 11	Used to store the user's color pick and spot number - This was changed to a class and these concepts were used in project 1
Structure types passed into functions	Chapter 11	Used in majority of functions to use the contents of the UserColor structure and access which colors the user picked- This was changed to a class and these concepts were used in project 1
String objects/classes	Chapter 10	Used in multiple places in char arrays, converting characters to strings, array types, variable names
Reading a binary file	Chapter 12	Used to read in the instructions file
Opening a binary file	Chapter 12	Used to open the instructions file and the results file(ex: line 181)
Writing a binary file	Chapter 12	Writes the results of the game to a binary file
Allocated memory	Chapter 9	Used to return computer's choices char array, used to return user's choices char array, used to create a new array of type UserColor, and used to create a new

		each pick array. All four used allocated memory and all of the memory used was deleted at the end of the program (“cleaning after yourself”)
Functions that return structure object variables	Chapter 11	Input function contains all the user input and returns the UserPick array An object variable is still returned except the structure is changed to a class
Inputting /Outputting a structure variable	Chapter 11	This is used to have the user input their choices, and at the end to output the contents of this array - This concept was still used except it was used with a class and not a structure
Classes	Chapter 13	Classes were used for the UserColor, ComColor, and base class named Color
Static Variables	Chapter 14	This was used in the base class Color in the .h file (line 15,16) and this was used in the Color.cpp file(line12,13 in .cpp) to initialize the static members
Overloaded Operators	Chapter 14	This was used in the UserColor class to increment the number of turns the player is allowed to have. It is located in UserColor.h line 18 and in the UserColor.cpp lines

		13-17
Inheritance	Chapter 15	This is shown through the Color, UserColor, and ComColor classes. Both the UserColor and the ComColor classes are derived from the base class Color
Polymorphism	Chapter 15	This is located in the UserColor class lines 24-29 where the variable color (from the base class) is used in a setColor function in the UserColor.h file
Templates	Chapter 16	Both the Mastermind_Mge and ComMge classes are templates and this is located in their .h files. They print out strings of mastermind messages
Exceptions	Chapter 16	This is located within the *input function (lines 366-377 in main) it is used to validate the color inputted by the user

Pseudo Code

Declare/Initialize Variables

Initialize counter

Introduction to Game

Modify limit of attempted games

Computer Generated Pick Function

Returns a primitive data type char

Displays The Introduction to the game that was written in a file, explains how the game works

Input function

User inputs their first color choices, if user picks an option invalid, displays message again

Representation of pick function

Displays table of what user picked so far, the number attempt they are on
 If guessed correctly, program displays win message
 SwitchH function (shows hints after each play)
 User presses one to see hints for all of their picks (how many are in the right spot and how many are correct but not in the right spot).
 Default: No hints displayed
 Hints displayed after each new guess
 Number of tries Counter
 Increments the number of tries after the switch function displays
 Outputs Result
 Reveals to the user the computer's choice
 If the user can guess in below 10 tries, they won and output displays message that they won and the percentage of the board they took up.
 Else displays lose message
 Write the output file the user's results
 End

Sample Outputs:

```
Project 2

Mastermind Problem
Enter your username
This username is for the game's reference to track your stats
kaylajr

Game Menu for Mastermind
Type 1 for the instructions
Type 2 for to play the game
Type 3 to see the contents of the leader board
Type 4 to see the Mastermind Picture
Type 5 if you want to see your stats
Type anything else to exit

5
All Player Stats
Username: kayla
Games Played: 1
Wins: 1
Losses: 0
Player Stats:
Username: kaylajr
Games Played: 0
Wins: 0
Losses: 0
```

When the game first starts, the game asks for a username and this username is used to store the player's data when they are finished with the game. By pressing option 5, the

user can see different users' stats that are all stored in the file, as well as their own current stats. In this sample, the user has not yet played the game so all the stats are set to zero.

Each Number will produce a different menu option for the game. The contents of 4 is the mastermind logo (because every good game has its own logo displayed on the game!)

The contents of 3 produces all the players that have made it on the leaderboard along with how many tries it took them (their score)

```
Game Menu for Mastermind
Type 1 for the instructions
Type 2 for to play the game
Type 3 to see the contents of the leader board
Type 4 to see the Mastermind Picture
Type 5 if you want to see your stats
Type anything else to exit

3
LeaderBoard

Username      Score
lolab 2
x 1
```

Number 2 is the most important part, the actual game but these other features are added as extras for the user.

When playing the game, the user types the color they want as a string and they type out which spot they want to place this color. These colors are then displayed in a table represented as characters with a hint under it.

```

One color is in the correct spot.
Three colors are correct, but they are not in the correct spot.
Pick your first color
white
What spot would you like this color in?
1
Pick your second color
white
What spot would you like this color in?
2
Pick your third color
white
What spot would you like this color in?
3
Pick your fourth color
white
What spot would you like this color in?
4
Your colors are W W W W
Color Choices   Turn Number
-----
RRRR           1
RRRR           2
WWWW           3
None of your colors are correct or in the correct spot.
- - - - -

```

If the user ever wants to go back and see the instructions after each game, they can. They just need to press 1 on the menu and the instructions will be outputted. The instructions will be outputted like this:

```

1
Welcome to a game of Mastermind!

Mastermind is a code breaking game for users to try and guess four different
colors the computer picked by random. The user uses hints given by the computer
telling them whether their color is in the right spot, or how many of the colors
are correct. The user inputs how any attempts they are willing to play, and to win
they must guess the correct colors before
10 tries.

Please pick four colors and press enter to continue. You must write
out the full color name to pick that color. After you write out the
color the computer will ask you for the spot number. This means what
spot do you want your color to go in(first,second,third,fourth)
you will input this as either one of the four numbers (1,2,3,4).
The available colors are red, green, blue, yellow, brown, orange,
black, and white.The computer might does not pick any color more than
once. If you can guess in 10 tries or less, you win.
Your colors will be displayed in a table as single characters as:
red= r, green=g, blue=b, brown=n, black=k,
yellow=y, orange=o, and white = w.

```


