

Project 1

<Mastermind Game V2>

CIS-7
Miguel Galvez
10/15/21

Introduction

Title: Mastermind Game

Mastermind is a code-breaking game, in which the player has to deduce what the code might be. The code can be 4, 6, or 8 characters long.

It is up to the player to decide which length they would like the code to be, and whether or not duplicate characters are allowed.

The code consists of some combination of the numbers 1-9. The player is allowed a total of 10 attempts to figure out the correct code.

The game begins with the player configuring their settings (i.e. setting code-length and allowing/not allowing duplicates).

Once the settings are configured, a code is generated that is meant to be broken by the player. From there, the player then enters a combination of the numbers 1-9 until they either use up 10 tries or enter the correct code.

Along the way, the player is given three hints for each incorrect code they enter.

The program returns the number of correct codes in the correct spot entered, the number of correct codes but incorrect spot entered, and the number of incorrect codes entered.

Summary

Project Size: 333 Lines

The Number of variables: 15

My game is complete in terms of functionality.

I programmed using the concepts that I learned from CSC-5 review in CSC-17a, including structures, character arrays, dynamic memory and pointers.

The entire process took me about two weeks.

In the first week, I completed a version 1 of this game where duplicates are not considered.

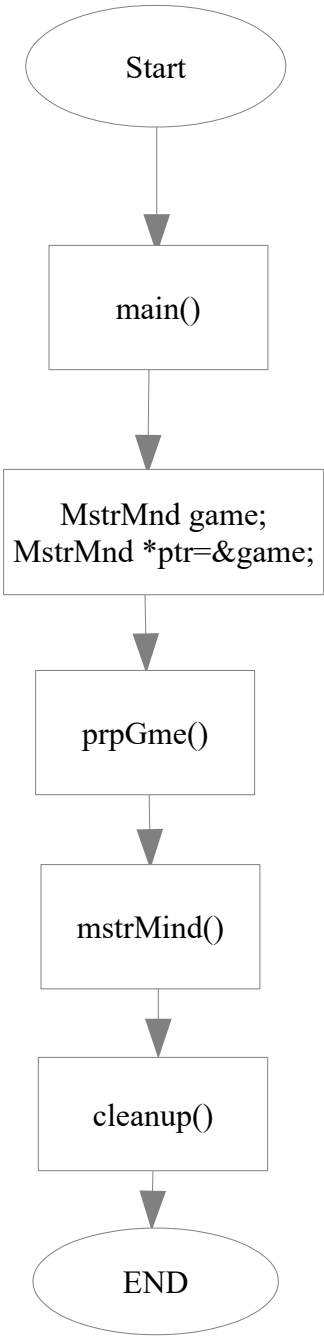
The process to find red, white, and empty pegs is more straight forward.

However, when it came to considering duplicates, I took another week to understand how to give the proper pegs.

Description

The objective for creating this program is to learn proper logical decisions. For instance, learning how to return the proper pegs such as red, white or empty.

Flowchart



Pseudo Code

Start

Prompt the player to enter 1 to allow duplicates or 0 to not allow duplicates.

If input is not 1 or not 0,

then prompt player again until a 1 or 0 is entered.

Prompt the player to enter 4,6, or 8 to set the code-length.

If input is not 4, or not 6, or not 8,

then prompt player again until a 4, or 6, or 8 is entered.

If duplicates are allowed

generate a random code that includes duplicates

Else

generate a random code that excludes duplicates.

ATMPS=10;

For (int tries=0;tries<ATMPS;tries++)

Prompt user to enter a code with possible characters being 1-9.

While the player entered an incorrect character,

prompt again to replace incorrect character entered.

If player entered a code that does not match the code generated,

provide a hint.

Red=0

white=0

empty

If one element from code matches the same element from generated code

red++

Create a copy of the key code

for(i=0;i<size;i++)

for(j=0;j<size;j++)

if copy[i]==usrCode[j]

white++

copy[i]=48

white-=red

empty =size-(red+white)

display red (),white(x), and empty(o) characters*

Else

congratulate player and end the game.

Major Variables

Type	Variable Name	Description	Location
MstrMnd	Game	Mastermind game structure	Main(),prpGme(),mstrMind(),hint(),clean(),whitePegs()
	*ptr	Pointer to mastermind game structure	Main(),prpGme(),mstrMind(),hint(),clean(),whitePegs()
Char	*usrCode	User-entered code	Myheader.h {Apart of MstrMnd structure}
	*key	Randomly generated code	Myheader.h {Apart of MstrMnd structure}
	rndNum	Random number	rndDgit()
	*ptr	Pointer to dynamically allocated character array	*dupLngth()
	cpy	Copy of key code	whtPegs()
Short	size	Code-length	Myheader.h {Apart of MstrMnd structure}, codLngth()
	y	Test variable for input validation	allwDup()
	Red	Total number of correct codes in correct spot	Hint(),dspHint()
	White	Total number of correct codes in wrong spot	Hint(),dspHint()
	empty	Total number of incorrect codes	Hint(),dspHint()
	ATMPTS	Total number of attempts allowed	mstrMind()
Bool	Dup	True to allow duplicates, false to not allow duplicates	Myheader.h {Apart of MstrMnd structure}

References

1. Gaddis Starting Out with C++ From Control Structures through Objects (2011) [7th Edition]
2. [Play Mastermind Online \(webgamesonline.com\)](http://webgamesonline.com)

Program

```
/*
1.      * File:  myheader.h
2.      * Author: Miguel Galvez
3.      * Created on October 6, 2021, 8:15 PM
4.      */
5.      #ifndef MYHEADER_H
6.      #define MYHEADER_H
7.      //Libraries
8.      #include <iostream>          //cout,cin
9.      #include <ctime>            //time()
10.     #include <cstring>          //strstr(),strcpy()
11.     using namespace std;
12.     //Public Structure
13.     struct MstrMnd{              //Mastermind Structure
14.         char *usrCode;          //User-entered code
15.         char *key;              //Auto-generated key code to be found by user
16.         short size;             //Code length
17.         bool dup;               //True to allow duplicates or false otherwise
18.     };
19.     //Function Prototypes
20.     void prpGme(MstrMnd *);      //Prepare Mastermind Game
21.     void allwDup(bool &);        //Prompt user to allow duplicates or not
22.     short codLngh();             //Set Code Length for Mastermind game
23.     char *dupLngh(short);        //Duplicate Code Length for User Input
24.     void genCode(char *,bool,short); //Generate Code to be Broken by User
25.     void mstrMind(MstrMnd *);    //Mastermind game
26.     void hint(MstrMnd *);        //Compute the appropriate hint
27.     void dspHint(short,short,short); //Displays Hint
28.     void clean(MstrMnd *);       //Cleanup Dynamically Allocated Memory
29.     char rndDgit();              //Returns a random digit
30.     bool chkDup(char *,int);     //If there is a duplicate, return true
31.     void whtPegs(MstrMnd *,      //Compute number of white+red pegs
32.         unsigned short &);
33.     #endif /* MYHEADER_H */
34.     /*
35.     * File:  main.cpp
36.     * Author: Miguel Galvez
37.     * Purpose: Mastermind Game
38.     * Created on September 25, 2021, 12:00 AM
39.     */
40.     #include "myheader.h" //Header file
41.     int main(int argc, char** argv){
42.         //Set Seed for Random number generator
43.         srand(static_cast<unsigned int>(time(0)));
44.         //Initialize Variables
```

```

45.         MstrMnd game;    //Mastermind game structure
46.         MstrMnd *ptr=&game; //Pointer to mastermind structure
47.         //Prepare Mastermind Game
48.         prpGme(ptr);
49.         //Start Mastermind Game
50.         mstrMind(ptr);
51.         //Cleanup
52.         clean(ptr);
53.         return 0;
54.     }
55.     #include "myheader.h"
56.     /*
57.      * File:  prpGme.cpp
58.      * Author: Miguel Galvez
59.      * Purpose: Prepare Mastermind Game
60.      * Created on September 25, 2021, 12:00 AM
61.      */
62.     //Prompt User for Mastermind Game Settings
63.     void prpGme(MstrMnd *ptr){
64.         //Prompt User to allow or not allow duplicates
65.         allwDup(ptr->dup);
66.         //Dynamically Allocate Code length
67.         ptr->size=codLngh();
68.         //Dynamically Allocate User Input Length
69.         ptr->usrCode=dupLngh(ptr->size);
70.         //Dynamically Allocate Key Code Length
71.         ptr->key=dupLngh(ptr->size);
72.         //Generate a Code for the game
73.         genCode(ptr->key,ptr->dup,ptr->size);
74.     }
75.     #include "myheader.h"
76.     /*
77.      * File:  allwDup.cpp
78.      * Author: Miguel Galvez
79.      * Purpose: Prompt user to allow or not allow
80.      *          duplicates for Mastermind Game
81.      * Created on September 25, 2021, 12:00 AM
82.      */
83.     //Prompt User to either allow or not allow duplicates
84.     void allwDup(bool &x){
85.         short y;    //Test input to check validation
86.         cout<<"Would you like to allow duplicates?\n"
87.             <<"Enter 1 for yes or 0 for no. ";
88.         cin>>y;
89.         //INPUT VALIDATION-Must be 0 or 1
90.         while(y!=0&&y!=1){
91.             cout<<"ERROR: Please enter either 1 for duplicates or 0 "
92.                 <<"for no duplicates. ";
93.             cin>>y;

```

```

94.         }
95.         //If input was acceptable, assign it to bool value
96.         x=y;
97.         cin.ignore(); //Clear the buffer
98.     }
99.     #include "myheader.h"
100.    /*
101.     * File:  codLngh.cpp
102.     * Author: Miguel Galvez
103.     * Purpose: Prompt player for code length in Mastermind Game
104.     * Created on September 25, 2021, 12:00 AM
105.     */
106.    //Determine Length of code
107.    short codLngh(){
108.        short size;
109.        //Prompt User for code length
110.        cout<<"Enter 4,6, or 8 for the code length. ";
111.        cin>>size;
112.        //INPUT VALIDATION-Make sure correct size was entered
113.        while(size!=4&&size!=6&&size!=8){
114.            cout<<"ERROR: Please enter 4, 6, or 8. ";
115.            cin>>size;
116.        }
117.        return size;
118.    }
119.    #include "myheader.h"
120.    /*
121.     * File:  dupLngh.cpp
122.     * Author: Miguel Galvez
123.     * Purpose: Mastermind Game
124.     * Created on September 25, 2021, 12:00 AM
125.     */
126.    //Duplicate character array as from codLngh
127.    char *dupLngh(short size){
128.        char *ptr;
129.        ptr=new char [size];
130.        return ptr;
131.    }
132.    #include "myheader.h"
133.    /*
134.     * File:  genCode.cpp
135.     * Author: Miguel Galvez
136.     * Purpose: Mastermind Game
137.     * Created on September 25, 2021, 12:00 AM
138.     */
139.    //Randomly Generates a Code to be broken
140.    void genCode(char *key, bool dup, short size){
141.        //Randomly Generate Code
142.        //If duplicates are allowed, generate a

```



```

143.         //random code with duplicates
144.         if(dup){
145.             for(int i=0;i<size;i++){
146.                 key[i]=rndDgit(); //Assign a random character
147.             }
148.             //If duplicates are not allowed, generate a code without duplicates
149.         }else{
150.             //Start with assigning the first element of key to a random digit 1-9
151.             key[0]=rndDgit();
152.             //Then assign a random digit to the next element of key
153.             for(int i=1;i<size;i++){
154.                 key[i]=rndDgit();
155.                 //While there is a repeated character in the code, try again
156.                 while(chkDup(key,i))
157.                     key[i]=rndDgit();
158.             }
159.         }
160.     }
161.     #include "myheader.h"
162.     /*
163.     * File:   rndDgit.cpp
164.     * Author: Miguel Galvez
165.     * Purpose: Generate a random digit from 1-9
166.     * Created on September 25, 2021, 12:00 AM
167.     */
168.     //Return a random digit
169.     char rndDgit(){
170.         char rndNum;        //Random number
171.         rndNum=rand()%10+48; //Ascii Code for numbers '0'=48 and '9'=57
172.         //If Random Number is 0, retry until random number is 1-9
173.         while(rndNum==48) rndNum=rand()%10+48;
174.         return rndNum;
175.     }
176.     #include "myheader.h"
177.     /*
178.     * File:   chkDup.cpp
179.     * Author: Miguel Galvez
180.     * Purpose: Determine if there is a duplicate value in array
181.     * Created on September 25, 2021, 12:00 AM
182.     */
183.     //Returns true if duplicate was detected
184.     bool chkDup(char *ptr,int n){
185.         //Not applicable if we're in the first entry
186.         if(n!=0){
187.             //Starting from the first entry, compare the
188.             //entry recently entered with each individual entry
189.             for(int i=0;i<n;i++){
190.                 if(ptr[n]==ptr[i])
191.                     return true; //If entry is the same, return true

```

```

192.     }
193. }
194.     return false;           //Otherwise return false
195. }
196. #include "myheader.h"
197. /*
198.  * File:  mstrMind.cpp
199.  * Author: Miguel Galvez
200.  * Purpose: Mastermind Game
201.  * Created on September 25, 2021, 12:00 AM
202.  */
203. //Mastermind game
204. void mstrMind(MstrMnd *x){
205.     const short ATMPS=10;    //Attempts allowed
206.     //Give instructions to User
207.     cout<<"Enter code in the format: ";
208.     for(int k=0;k<x->size;k++){
209.         cout<<"x";
210.     }
211.     cout<<" where x is a number 1-9\n";
212.     cout<<"Symbol for correct numbers in correct spot: *"<<endl;
213.     cout<<"Symbol for correct numbers in incorrect spot: x"<<endl;
214.     cout<<"Symbol for incorrect codes: o"<<endl;
215.     //Start Mastermind Game
216.     for(int trial=0;trial<ATMPS;trial++){
217.         cout<<"Attempt "<<trial+1<<"/"<<ATMPS<<endl;
218.         cout<<"enter code: ";
219.         for(int i=0;i<x->size;i++){
220.             cin>>x->usrCode[i];
221.             //INPUT VALIDATION-If incompatible code was entered
222.             while((x->usrCode[i])>57||(x->usrCode[i])<49){
223.                 cout<<"ERROR: Please enter a digit between 1-9. ";
224.                 cout<<"To replace: "<<x->usrCode[i]<<" \n";
225.                 cin>>x->usrCode[i];
226.             }
227.         }
228.         //If correct code was not entered, provide a hint
229.         if(strstr(x->key,x->usrCode)==NULL){
230.             hint(x);
231.             //Otherwise, user entered the correct code.
232.         }else{
233.             //Congratulate user
234.             cout<<"CONGRATULATIONS!!! YOU WON THE GAME!";
235.             //End iteration
236.             trial=ATMPS;
237.         }
238.     }
239. }
240. #include "myheader.h"

```

```

241.      /*
242.      * File:  hint.cpp
243.      * Author: Miguel Galvez
244.      * Purpose: Displays a hint to the user for mastermind game
245.      * Created on September 25, 2021, 12:00 AM
246.      */
247.      //Display hint depending on user code
248.      void hint(MstrMnd *ptr){
249.          unsigned short red=0, //Correct numbers in the right spot
250.                  white=0, //Correct number but incorrect spot
251.                  empty; //Incorrect Code
252.          //Compute the total number of "red" codes
253.          for(int i=0;i<ptr->size;i++){
254.              //If correct code was in correct spot, Increment total correct codes
255.              if(ptr->key[i]==ptr->usrCode[i]){
256.                  red++;
257.              }
258.          }
259.          //Return white=red+white
260.          whtPegs(ptr,white);
261.          //Subtract off reds from white to get white
262.          white-=red;
263.          //Compute total empty pegs
264.          empty=(ptr->size)-(red+white);
265.          //Give appropriate hints
266.          dspHint(red,white,empty);
267.      }
268.      #include "myheader.h"
269.      /*
270.      * File:  whtPegs.cpp
271.      * Author: Miguel Galvez
272.      * Purpose: Generate white pegs for Mastermind Game
273.      * Created on September 25, 2021, 12:00 AM
274.      */
275.      //Compute Red+White Pegs and Store it into white
276.      void whtPegs(MstrMnd *ptr,unsigned short &white){
277.          char cpy[ptr->size]; //Copy of the code
278.          //Copy code into separate array
279.          strcpy(cpy,ptr->key);
280.          //Check if there are any matches with the user code and key code copy
281.          for(int i=0;i<ptr->size;i++){
282.              for(int j=0;j<ptr->size;j++){
283.                  //If a match was found with corresponding key code, white it out
284.                  //so that it will not reoccur
285.                  if(cpy[i]==ptr->usrCode[j]){
286.                      white++; //Increment white to get Red+White
287.                      cpy[i]=48; //Ascii code for 0
288.                  }
289.              }

```

```

290.     }
291. }
292. #include "myheader.h"
293. /*
294.  * File:  dspHint.cpp
295.  * Author: Miguel Galvez
296.  * Purpose: Display a hint for Mastermind Game
297.  * Created on September 25, 2021, 12:00 AM
298.  */
299. //Displays the appropriate hint
300. void dspHint(short red,short white,short empty){
301.     cout<<"Hint:";
302.     //Display asterisk for red (correct) pegs
303.     if(red!=0){
304.         for(int i=0;i<red;i++){
305.             cout<<"*";
306.         }
307.     }
308.     //Display x for white (correct but in wrong spot) pegs
309.     if(white!=0){
310.         for(int i=0;i<white;i++){
311.             cout<<"x";
312.         }
313.     }
314.     //Display o for empty (incorrect) pegs
315.     if(empty!=0){
316.         for(int i=0;i<empty;i++){
317.             cout<<"o";
318.         }
319.     }
320.     cout<<endl;
321. }
322. #include "myheader.h"
323. /*
324.  * File:  clean.cpp
325.  * Author: Miguel Galvez
326.  * Purpose: Cleanup dynamic memory from Mastermind Game
327.  * Created on September 25, 2021, 12:00 AM
328.  */
329. //Delete Dynamically Allocated Arrays
330. void clean(MstrMnd *ptr){
331.     delete [] ptr->usrCode;
332.     delete [] ptr->key;
333. }

```