# Matthew Sanchez

CSC5-47982

Project 2

Directions

Mastermind is the ultimate game of logic. The game starts with a computer generating four random colors. Begin by trying to guess the correct pattern. After each guess you will be hinted on the number of colors that were correct and in the right position. You will also be hinted on the number of colors that you entered that were correct but in the wrong spot. In this game you will be given 10 tries to get the correct sequence. Good luck!

Sample Input/Output
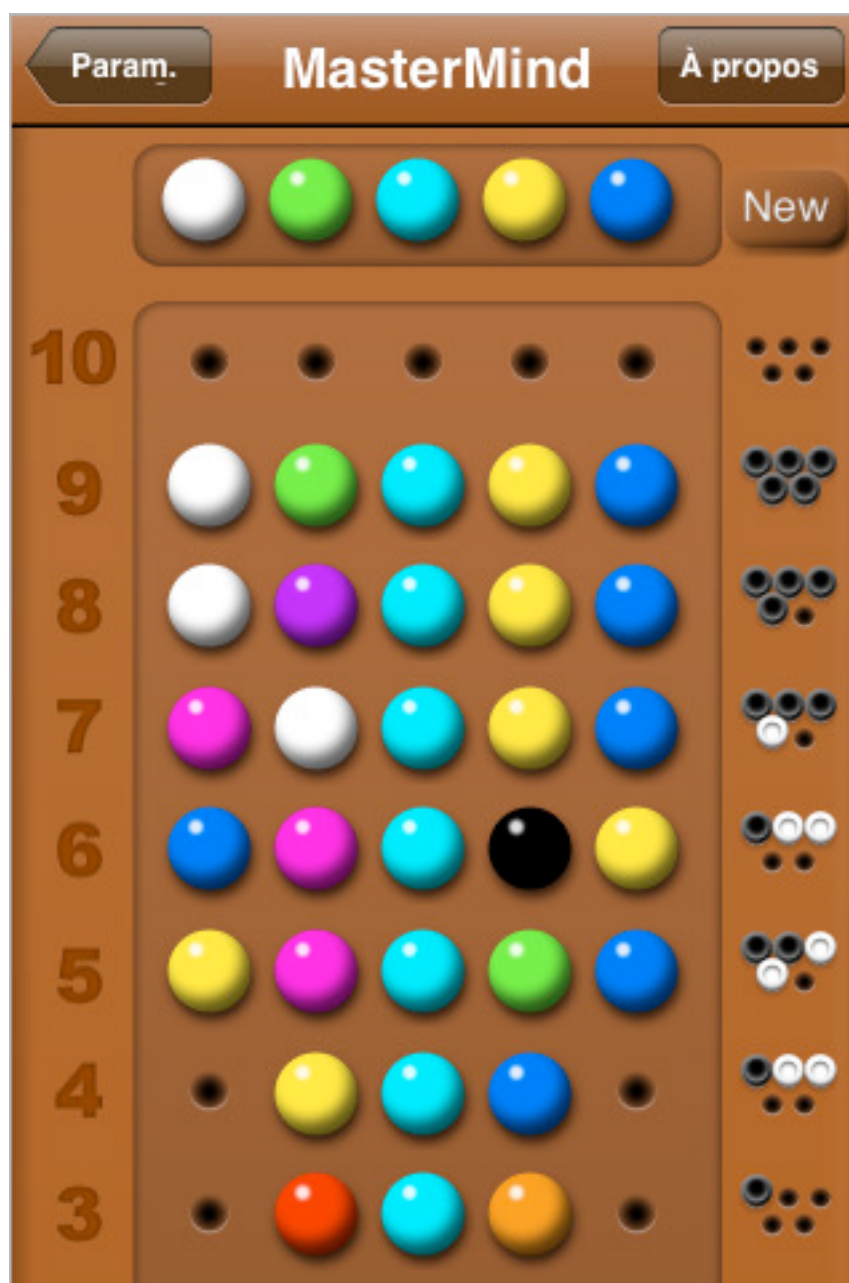
Pseudocode

Libraries: iostream, cstdlib, ctime

No global constants

Function Prototypes

1. Display Instructions

2. Get user input

3. Color conversion

4. Number correct and in right spot

5. Correct color but in wrong spot

Main

1. Declare Variables

2. Create 4 variables to convert random colors input to integer

3. Create an array variable for the random numbers/colors

   generated

4. Create an array variable for the users guesses

5. Create a counter variables for user guesses

6. Create a variable that inputs users decision to play again

7. Create a variable to count number of user guesses and initialize it

to zero

8. Create a variable to count number of users guesses left and initialize it to 10

9. Create a variable to output the number of colors the user got correct and in the correct position

10. Create a variable to outpu the number of colors the user got correct but were in the wrong spot

11. Create a two dimensional array to display the the history of the users guess after each input

12. Create a counter for this two dimensional array and initialize it to zero

13. Create another array to output the history of correct guesses

14. Create another array to output the history of the users correct color but in the wrong spot

15. Call function to display instructions

16. Call RNG function to generate the correct sequence

17. Range the random numbers 1-5

18. Assign each random number to the array "Correct"

19. Convert each random number to a character labeled as a color with the function

20.   Call function call to output for user to start guessing

21.   Input each user guesses into array "Guess"

22.   Call function determine how many are correct and display it

23.   Call function to determine how many are the right color but in wrong position and display it

24.   Display user history after each guess

25.   Break out if user wins

26.   Create a counter for user guesses

27.   Add 1 to counter after each user guess input

28.   Limit user guesses to 10

29.   Determine if user wins

30.   If user loses output correct sequence

31.   Prompt user to play again

32.   If user wants to play again initialize numGuesses to zero

33.   Also initialize guesses left counter to 10

34.   Exit Stage Right

Functions

1. Create a function to display instructions

2. Create a function to get user input

   a. Prompt user to enter guesses and convert lower case entries to uppercase

   b. Fill the 2 dimensional array with the users history and output after each guess

3. Create a function with a switch menu to convert inputs to characters

   a. Return each character

4. Create a function to check which guesses are correct color and in correct position

5. Create another function to determine if color is correct but in wrong position

   a. This function counts the number of colors in the guess and correct code and compares them

   b. Then it keeps the smaller of the 2 sums them then subtracts this sum from total right and in correct position

The code and flow chart were

submitted separately inside the

folder.