

CS 116

Programming Assignment 2

This is a short assignment to make sure that you understand the basics behind the chapters covered so far. The important thing, before you start this assignment, is to understand conditionals and iterations really well. The estimated time for completion is about three hours.

The program given to you is a small game (**ColorsAndCombinations**). All the code is done except for one function ("**GetStatistics**" and "**GetCombinations**") that you will have to implement. These functions consists of looping (one or multiple times) through the board to get color statistics and find the game's combinations (will explain in more details later in this document).

You need to use iterations (for-loops, while-loops, for-each ...) in this assignment. Your choice which one to use.

Copyright Notice

Copyright © 2010 DigiPen (USA) Corp. and its owners. All rights reserved.

No parts of this publication may be copied or distributed, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language without the express written permission of DigiPen (USA) Corp., 9931 Willows Road NE, Redmond, WA 98052

Trademarks

DigiPen® is a registered trademark of DigiPen (USA) Corp.

All other product names mentioned in this booklet are trademarks or registered trademarks of their respective companies and are hereby acknowledged.

Details

Gameplay Details:

The algorithms we are implementing are fairly simple; we want to see how many times each color is repeated in the board, then highlight combinations of three or more consecutive squares with the same color. Let me show you a graphical example.



Taking this board as an example, you can see that we have 3 red squares, 4 blue, 1 pink and 4 green. The text on the right of the board should reflect that.

Also you can see that we have 3 consecutive red squares, 4 consecutive green squares and 3 consecutive blue squares that need to be highlighted.

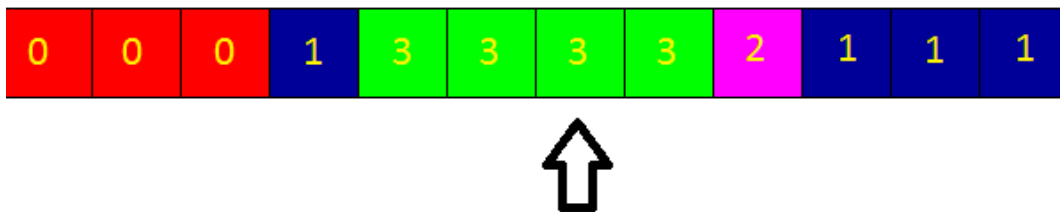
Code Details:

Multiple arrays are given to you in this assignment.

gaBoard: This is an array that contains 12 slots representing the board. In order to access any slot on the board you will need to use the array's name and an index to which slot you want to access. For example, `gaBoard[6]` will be the 7th slot in the array (green square, check the board picture below). `gaBoard` will contain an integer value representing the different colors.

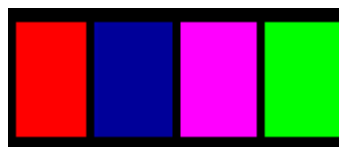
- 0 = Red
- 1 = Blue
- 2 = Pink
- 3 = Green

The following is the board representation:



gaBoardStatistics: This is an array that contains 4 slots. Every slot is an integer that will be used to store one color's statistics. For example, `gaBoardStatistics[1]` will have, at the end, how many blue squares we have in the board.

Below is a graphical representation for `gaBoardStatistics`



Now, in the **GetStatistics** function, you have to check the values inside **gaBoard** and change the values inside **gaBoardStatistics** accordingly.

PrintStatistics, is a function that will be called after **GetStatistics** and will just print on the screen the values it finds inside the **gaBoardStatistics** (don't worry I'm calling it, you don't have to do that).

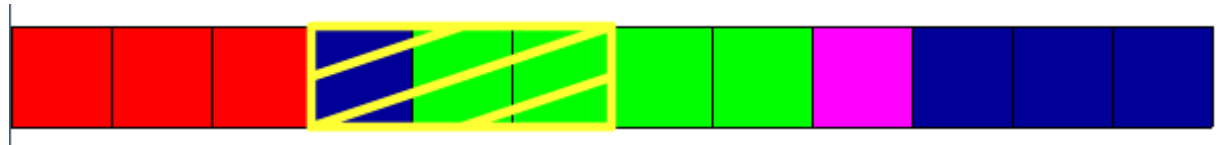
Originally, **GetStatistics** is empty, so when running the game, without any modification to the function, all texts on the screen should show 0.

As for the **GetCombinations** function, you have to loop through the board and check for the combinations. When you find a combination you should simply call the **SelectCombination** function to highlight it (below you will find a detailed explanation on the **SelectCombination** function).

- **function SelectCombination(iIndex1:int, iIndex2:int)**

This function highlights a combination on the board by putting a horizontal yellow rectangle surrounding consecutive squares specified by 2 indices (**iIndex1**, **iIndex2**)

Eg: SelectCombination(3,5); will lead to the following result:



All the changes that you need to do are **inside** the **GetStatistics** and **GetCombinations** functions, nothing outside of them. You don't even need to call them, because I already do that.

Notes on the side:

In the code you can find a variable that will help you with looping boundaries.

iCellsNumber, which represents the number of squares we have in the board (which is 12 in our case).

Also, 2 functions are given to populate the board (**PopulateBoardStatic** and **PopulateBoardRandom**). You should start with using the static one until you get the correct answer then test the random one.

If your code is valid, the texts will be showing the correct values and the correct combinations will be highlighted.

Comments

In this and future assignments, you are required to include:

- A file header comment at the beginning of the fla. The format is shown in the "Comments.flc" file given to you in the beginning of the semester and should be present at the very top of all your code.
- Function header for each function you create. The format is shown in the "Comments.flc" file given to you in the beginning of the semester and should be present at the top of every function.
- Inline commenting for your code.

What to submit

You must submit the fla file (**ColorsAndCombinations.flc**) in a single .zip file named correctly (go to the class page on moodle and you will find the assignment submit link). **Do not submit any other files than the ones listed.**

If you've forgotten how to submit files, the details about how to submit are posted in the syllabus. Failure to follow the instructions will result in a poor score on the assignment (and possibly a zero).

Special note:

The due date/time posted is the positively latest you are allowed to submit your code. Since the assignments can easily be completed well before the deadline, you should strive to turn it in as early as possible. If you wait until the deadline, and you encounter unforeseen circumstances (like being sick, or your car breaking down, or something else), you may not have any way to submit the assignment on time. Moral: **Don't wait until the last day to do your homework.**