The Testing Process

Phase one: Test the system's ability to find colors based on their names

Phase two: Test the system's ability to compare contrast between two colors

Phase three: Test whether or not the system returns the correct hue when given a color

Phase four: Tests the system's ability to properly convert Hexadecimal to RGB

Phase five: Tests to see if the system accepts non-capital Hexadecimal inputs

Testing schedule

Oct 1- Oct 8: Set up the design for the automated testing framework, including documentation to how it will work and the like

Oct 9 – Oct 16: Build the testing framework with previous design as a guide. Really just working to get it functioning

Oct 17 – 24: Test the framework to just make sure it’s working as intended

Oct 25 – 28: Update the previous documentation to reflect the current implementation of the system

Oct 29: Present our test framework and explain its functionality.

Nov 4: Team meeting to make sure 10 test cases are completed and running properly

Nov 11: Team meeting to make sure all 25 test cases are complete and running properly

Nov 12: Present the running of our test cases from the test framework and explain any questions regarding the tests function.

Nov 13 – 14: Conceptualize the design of the faults for the system and document this process

Nov 15 – 17: Build the faults and inject them into the code.

Nov 18: Test and make sure the faults are working as they should. Update the necessary documentation.

Nov 19 Present the results of our fault design

Nov 21/ Nov 26 Present the overall process that got us to this point and demonstrate the fruits of our labor

Test Recording Procedures

All tests will have the following categories filled out for each of their tests:

Test Case ID > The ID of the specific test case

Requirement Tested> Specifying which of the project requirements we will be addressing

Component tested> The greater component being tested

Method tested> The specific method being tested

Test Input Data> The test data that will be used for the test

Expected Result> What you expect the test to produce

Actual Result> What the test actually produces

Hardware and Software Requirements

The software required for this process include a Disc image File for Ubuntu 18.04, Oracle VM VirtualBox or a comparable virtual machine, Docker, Git, Java, and the source code from the Contrast Finder Repository.

The hardware required is at least 5277 MB of memory to allocate towards the Virtual Machine.

Constraints

Constraints that may come up in the foreseeable future could include computer malfunctions for members, members getting ill, ext. These all come down to losing a member for an unspecified amount of time, which will simply lead to us redistributing the workload between two members until the third is able to rejoin the group.

Test Cases

Test Case 1

Requirement: The method takes a string that is a color name and checks to see if the color is a valid entry.
Component Being Tested: ColorNameLookup

Method: getColorNameFromStr(String colorStr)
Test Inputs: Yellow
Expected Outcome: Yellow

Test Case 2

Requirement: The method takes a string that is a color and checks to see if the color is a valid entry.
Component Being Tested: ColorNameLookup
Method: getColorNameFromStr(String colorStr)
Test Inputs: YELLOW
Expected Outcome: Yellow

Test Case 3

Requirement: The method takes a string that is a color and checks to see if the color is a valid entry.
Component Being Tested: ColorNameLookup
Method: getColorNameFromStr(String colorStr)
Test Inputs: yellow
Expected Outcome: Yellow

Test Case 4

Requirement: The method takes a string that is the Hexadecimal value of a color and returns the RGB of the same color
Component Being Tested: ColorConverter
Method: hex2Rgb(String colorStr)
Test Inputs: 00AA00
Expected Outcome: rgb(0,170,0)

Test Case 5

Requirement: The method takes a string that is the Hexadecimal value of a color and returns the RGB of the same color
Component Being Tested: ColorConverter
Method: hex2Rgb(String colorStr)
Test Inputs: 0000aa
Expected Outcome: rgb(0,0,170)