



The Boyoz Tanaguru Contrast Finder Automated Testing Framework

Nathan Bell, Paul Joseph, Logan Siflar

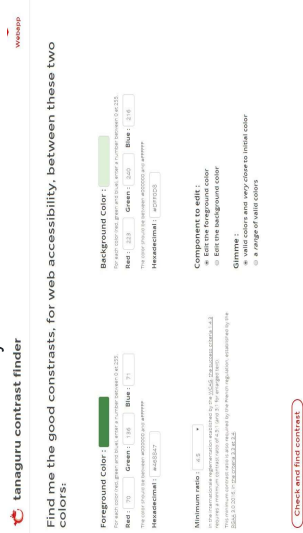
Project Goals/Objectives

The goal of this project was to create an automated testing framework for an open source application, then inject faults to ensure proper results are tested and printed. The testing was to be invoked by typing a single command (as seen in "Script Execution") in the command line once in the correct directory. After the testing was completed a results html page was to be opened in the users browser providing details on the tests that were run and the results of each test. Successful tests will print a green "pass" while failed tests (due to faults or otherwise) will print a red "fail".

Tanaguru Contrast-Finder

What is Tanaguru?

- Tanaguru is an open source website assessment tool that is dedicated to enhancing web quality and accessibility. They also focus on reliability and automation.



What is this project supposed to be doing and why is it important?

- Finds best contrast b/w two colors so website is readable

Test Cases

Test Case Outline:

Id: 1
Requirement: Converts a color object to a HSL value
Class: ColorConverter.java
Method: rgb2Hsl
Input: 000000
Expected Output: hsl(0, 0%, 0%)

At the end of our work we ended up with 25 test cases that follow the format above, but this is scalable. More test cases could be added as long as they follow the same test case format. If it were desired that more methods were to be tested new drivers that follow our naming convention of methodNameDriver.java would also need to be created along with those test cases.

Script Execution

- 0) Be sure to be on a linux or mac machine and have a JDK version 7+ installed.
- 1) Clone the github repository to your machine (git clone <link>)
- 2) Once you have the files on your system, navigate to the folder labeled TestAutomation (cd <directoryName>)
- 3) Run the command ./scripts/runAllTests.sh
- 4) After the script finishes executing a the results of the tests will be displayed in your browser

Results

ID	Test Case	Class	Method	Input	Expected Output	Actual Output	Pass/Fail
1	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
2	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
3	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
4	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
5	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
6	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
7	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
8	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
9	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
10	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
11	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
12	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
13	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
14	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
15	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
16	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
17	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
18	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
19	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
20	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
21	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
22	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
23	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
24	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass
25	Converts a color object to a HSL value	ColorConverter.java	rgb2Hsl	000000	hsl(0, 0%, 0%)	hsl(0, 0%, 0%)	Pass

Conclusions

All in all, this project gave us valuable knowledge on how to create and automatically run tests for software. Using Tanaguru's open source code, we developed individual test cases and drivers for each method we wanted to test. We then wrote a script that automatically read and ran each test case and outputted the result to a html file. Most of this was new to all of us, however through team collaboration we were able to figure out the ins and outs of testing and walk away with a new tool in our coding toolbox.

Additional Information

CSGI 362 02 Fall 2020

Dr. Jim Bowring

[csci-362-02-2020/The-Boyoz \(github.com\)](https://github.com/csci-362-02-2020/The-Boyoz)