

Tanaguru Contrast Finder Automated Testing Framework



Reid Foster, Joe Mezera and Stefan Velloff
College of Charleston

Construction and Testing

After developing the testing framework, our team chose five methods to use for testing. For these methods we created five test cases each. In the way the framework has been constructed, new test cases and respective drivers are simple to add as long as the laid out template is followed. This means this system could easily be expanded on for testing of the entirety of the java code in the system. To further test the system, faults were also injected into the code to insure that the system would catch errors and other changes in results.

Figure 1:
Output of
running the
Framework.

Tanaguru TestAutomation Results								
Test ID	Tested class	Tested Method and Parameters	Driver	Arguments	Expected outcomes	Last Result from Running Test	Success or Fail	
calculate_1	DistanceCalculator.java	calculate(Color(255,255,255), Color(0,0,0))	testcasesexecutables.TestCalculate	255 255 255 0 0	441.67	367.77	Fail	
calculate_2	DistanceCalculator.java	calculate(Color(0,0,0), Color(255,255,255))	testcasesexecutables.TestCalculate	0 0 0 255 255 255	441.67	367.77	Fail	
calculate_3	DistanceCalculator.java	calculate(Color(0,0,0), Color(0,0,0))	testcasesexecutables.TestCalculate	0 0 0 0 0 0	0.0	0.0	Pass	
calculate_4	DistanceCalculator.java	calculate(Color(255,200,34),Color(255,255,17))	testcasesexecutables.TestCalculate	255 200 34 255 255 17	57.57	54.45	Fail	
calculate_5	DistanceCalculator.java	calculate(Color(200,7,16),Color(57,57,57))	testcasesexecutables.TestCalculate	200 7 16 57 57 57	156.94	139.77	Fail	
getContrastRatio_1	ContrastChecker.java	getContrastRatio(Color(0,0,0),Color(0,0,0))	testcasesexecutables.TestGetContrastRatio	0 0 0 0 0	1.0	1.0	Pass	
getContrastRatio_2	ContrastChecker.java	getContrastRatio(Color(0,0,0),Color(255,255,255))	testcasesexecutables.TestGetContrastRatio	0 0 0 255 255 255	21.0	21.0	Pass	
getContrastRatio_3	ContrastChecker.java	getContrastRatio(Color(255,255,255),Color(0,0,0))	testcasesexecutables.TestGetContrastRatio	255 255 255 0 0 0	21.0	21.0	Pass	
getContrastRatio_4	ContrastChecker.java	getContrastRatio(Color(57,57,57),Color(213,247,156))	testcasesexecutables.TestGetContrastRatio	57 57 57 213 247 156	9.686825265872823	9.686825265872823	Pass	
getContrastRatio_5	ContrastChecker.java	getContrastRatio(Color(187,5,190),Color(68,106,159))	testcasesexecutables.TestGetContrastRatio	187 5 190 68 106 159	1.0184255955300985	1.0184255955300985	Pass	
getHue_1	ColorConverter.java	getHue(Color(128,128,0))	testcasesexecutables.TestGetHue	128 128 0	60.0	60.0	Pass	
getHue_2	ColorConverter.java	getHue(Color(0,0,0))	testcasesexecutables.TestGetHue	0 0 0	0.0	0.0	Pass	
getHue_3	ColorConverter.java	getHue(Color(45,50,255))	testcasesexecutables.TestGetHue	45 50 255	238.57144	238.57144	Pass	
getHue_4	ColorConverter.java	getHue(Color(200,255,0))	testcasesexecutables.TestGetHue	200 255 0	72.94118	72.94118	Pass	
getHue_5	ColorConverter.java	getHue(Color(2,0,241))	testcasesexecutables.TestGetHue	2 0 241	240.49792	240.49792	Pass	
offsetRgbColor_1	ColorConverter.java	offsetRgbColor(Color(128,128,0), 1, 2, 3)	testcasesexecutables.TestOffsetRGBColor	128 128 0 1 2 3	129 130 3	129 130 3	Pass	
offsetRgbColor_2	ColorConverter.java	offsetRgbColor(Color(0,255,255), 255, -10, -10)	testcasesexecutables.TestOffsetRGBColor	0 255 255 -10 -10	255 245 245	255 245 245	Pass	
offsetRgbColor_3	ColorConverter.java	offsetRgbColor(Color(120, 120, 120), 60, -60, 60)	testcasesexecutables.TestOffsetRGBColor	120 120 120 60 60 60	180 60 180	180 60 180	Pass	
offsetRgbColor_4	ColorConverter.java	offsetRgbColor(Color(0, 0, 0), 255, 255, 255)	testcasesexecutables.TestOffsetRGBColor	0 0 0 255 255 255	255 255 255	255 255 255	Pass	
offsetRgbColor_5	ColorConverter.java	offsetRgbColor(Color(255,245,245), -255, -245, -245)	testcasesexecutables.TestOffsetRGBColor	255 245 245 -255 -245 -245	0 0 0	0 0 0	Pass	
rgb2Hex_1	ColorConverter.java	rgb2Hex(Color(0,1,2))	testcasesexecutables.TestRGB2Hex	0 1 2	000102	000102	Pass	
rgb2Hex_2	ColorConverter.java	rgb2Hex(Color(128,128,12))	testcasesexecutables.TestRGB2Hex	128 128 12	80800C	80800C	Pass	
rgb2Hex_3	ColorConverter.java	rgb2Hex(Color(0,255,0))	testcasesexecutables.TestRGB2Hex	0 255 0	00FF00	00FF00	Pass	
rgb2Hex_4	ColorConverter.java	rgb2Hex(Color(0,255,255))	testcasesexecutables.TestRGB2Hex	0 255 255	00FFFF	00FFFF	Pass	
rgb2Hex_5	ColorConverter.java	rgb2Hex(Color(255,255,255))	testcasesexecutables.TestRGB2Hex	255 255 255	FFFFFF	FFFFFF	Pass	

Figure 2:
Output of
running the
Framework
with injected
Faults.

Tanaguru TestAutomation Results								
Test ID	Tested class	Tested Method and Parameters	Driver	Arguments	Expected outcomes	Last Result from Running Test	Success or Fail	
calculate_1	DistanceCalculator.java	calculate(Color(255,255,255), Color(0,0,0))	testcasesexecutables.TestCalculate	255 255 255 0 0	441.67	441.67	Pass	
calculate_2	DistanceCalculator.java	calculate(Color(0,0,0), Color(255,255,255))	testcasesexecutables.TestCalculate	0 0 0 255 255 255	441.67	441.67	Pass	
calculate_3	DistanceCalculator.java	calculate(Color(0,0,0), Color(0,0,0))	testcasesexecutables.TestCalculate	0 0 0 0 0 0	0.0	0.0	Pass	
calculate_4	DistanceCalculator.java	calculate(Color(255,200,34),Color(255,255,17))	testcasesexecutables.TestCalculate	255 200 34 255 255 17	57.57	57.57	Pass	
calculate_5	DistanceCalculator.java	calculate(Color(200,7,16),Color(57,57,57))	testcasesexecutables.TestCalculate	200 7 16 57 57 57	156.94	156.94	Pass	
getContrastRatio_1	ContrastChecker.java	getContrastRatio(Color(0,0,0),Color(0,0,0))	testcasesexecutables.TestGetContrastRatio	0 0 0 0 0 0	1.0	1.0	Pass	
getContrastRatio_2	ContrastChecker.java	getContrastRatio(Color(0,0,0),Color(255,255,255))	testcasesexecutables.TestGetContrastRatio	0 0 0 255 255 255	21.0	0.047619047619047616	Fail	
getContrastRatio_3	ContrastChecker.java	getContrastRatio(Color(255,255,255),Color(0,0,0))	testcasesexecutables.TestGetContrastRatio	255 255 255 0 0 0	21.0	0.047619047619047616	Fail	
getContrastRatio_4	ContrastChecker.java	getContrastRatio(Color(57,57,57),Color(213,247,156))	testcasesexecutables.TestGetContrastRatio	57 57 57 213 247 156	9.686825265872823	0.1032329966271871	Fail	
getContrastRatio_5	ContrastChecker.java	getContrastRatio(Color(187,5,190),Color(68,106,159))	testcasesexecutables.TestGetContrastRatio	187 5 190 68 106 159	1.0184255955300985	0.9819077648801406	Fail	
getHue_1	ColorConverter.java	getHue(Color(128,128,0))	testcasesexecutables.TestGetHue	128 128 0	60.0	300.0	Fail	
getHue_2	ColorConverter.java	getHue(Color(0,0,0))	testcasesexecutables.TestGetHue	0 0 0	0.0	0.0	Pass	
getHue_3	ColorConverter.java	getHue(Color(45,50,255))	testcasesexecutables.TestGetHue	45 50 255	238.57144	121.428566	Fail	
getHue_4	ColorConverter.java	getHue(Color(200,255,0))	testcasesexecutables.TestGetHue	200 255 0	72.94118	287.05884	Fail	
getHue_5	ColorConverter.java	getHue(Color(2,0,241))	testcasesexecutables.TestGetHue	2 0 241	240.49792	119.50207	Fail	
offsetRgbColor_1	ColorConverter.java	offsetRgbColor(Color(128,128,0), 1, 2, 3)	testcasesexecutables.TestOffsetRGBColor	128 128 0 1 2 3	129 130 3	3 130 3	Fail	
offsetRgbColor_2	ColorConverter.java	offsetRgbColor(Color(0,255,255), 255, -10, -10)	testcasesexecutables.TestOffsetRGBColor	0 255 255 255 -10 -10	255 245 245	245 245 245	Fail	
offsetRgbColor_3	ColorConverter.java	offsetRgbColor(Color(120, 120, 120), 60, -60, 60)	testcasesexecutables.TestOffsetRGBColor	120 120 120 60 60 60	180 60 180	180 60 180	Pass	
offsetRgbColor_4	ColorConverter.java	offsetRgbColor(Color(0, 0, 0), 255, 255, 255)	testcasesexecutables.TestOffsetRGBColor	0 0 0 255 255 255	255 255 255	255 255 255	Pass	
offsetRgbColor_5	ColorConverter.java	offsetRgbColor(Color(255,245,245), -255, -245, -245)	testcasesexecutables.TestOffsetRGBColor	255 245 245 -255 -245 -245	0 0 0	0 0 0	Pass	
rgb2Hex_1	ColorConverter.java	rgb2Hex(Color(0,1,2))	testcasesexecutables.TestRGB2Hex	0 1 2	000102	020100	Fail	
rgb2Hex_2	ColorConverter.java	rgb2Hex(Color(128,128,12))	testcasesexecutables.TestRGB2Hex	128 128 12	80800C	0C8080	Fail	
rgb2Hex_3	ColorConverter.java	rgb2Hex(Color(0,255,0))	testcasesexecutables.TestRGB2Hex	0 255 0	00FF00	00FF00	Pass	
rgb2Hex_4	ColorConverter.java	rgb2Hex(Color(0,255,255))	testcasesexecutables.TestRGB2Hex	0 255 255	00FFFF	FFFF00	Fail	
rgb2Hex_5	ColorConverter.java	rgb2Hex(Color(255,255,255))	testcasesexecutables.TestRGB2Hex	255 255 255	FFFFFF	FFFFFF	Pass	

Figure 3: Test Case Template

1. Test ID
2. Tested class
3. Tested Method and Parameters
4. Test Case Driver
5. Arguments
6. Expected outcomes
7. Last Result from Running Test (These last two lines only appear if the tests have been run before)
8. Success or Fail

Figure 4: Example Test Case

```
calculate_1
DistanceCalculator.java
calculate(Color(255,255,255), Color(0,0,0))
testcasesexecutables.TestCalculate
255 255 255 0 0 0
441.67
367.77
Fail
```

Final Thoughts

The design and development of this project was a new experience for all of us. TCF was a great project choice for our group despite its poor documentation. In fact, several typos and logical errors we found in the code make it a perfect candidate for testing. The development of the script and drivers was a good implementation of encapsulation by making each aspect ignorant of each other. This allows us to easily add to and modify the framework to expand its capabilities. With this project we have gained experience with open-source software, the java package structure, and interaction between many different coding languages.

The Tanaguru Contrast Finder software is an open-source application that, as the name implies, finds the contrast between two colors. Its main purpose is to find colors with a good contrast ratio to provide an adequate amount of readability. This is particularly useful for those who are visually impaired, but a poor contrast is bad for anything that requires readability.

Introduction

Here we have developed a testing framework for the java backbone of this software. This framework is easy to expand and to execute, as it is run by a script and drivers, allowing a simple command to execute all of the tests. Adding tests and drivers is also simple as long as one follows our template and the framework structure.

Requirements

The requirements for using this system are java, python, and a linux virtual machine.

Resources & Acknowledgements

Our team would like to thank Dr. Bowring for his support during this project.

Our project repository:

<https://github.com/csci-362-02-2020/Team-6>

Tanaguru Source Code:

<https://github.com/Tanaguru/Contrast-Finder>