Chapter 4

Progress and Experiences

We had already made our 25 test cases but not all worked properly and we had significant troubles with the ones that resulted in an error. We debated on how we should handle these exceptions with try catch blocks and still pass in illegal arguments into the method being tested. We decided on isolating the method in its own try block and doing a catch that would handle any exception that would rise. This also caused us to have slightly alter certain test cases so that we would not be injecting errors.

Test cases 21-25 do actually fail. These test the method calculate in distanceCalculator, which calculates the Euclidean distance between two colors. In the comments above the method, this link is given for the formula they use:

http://en.wikipedia.org/wiki/Euclidean_distance#Three_dimensions

This link takes you to the three-dimensional Euclidean distance formula:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}.$$

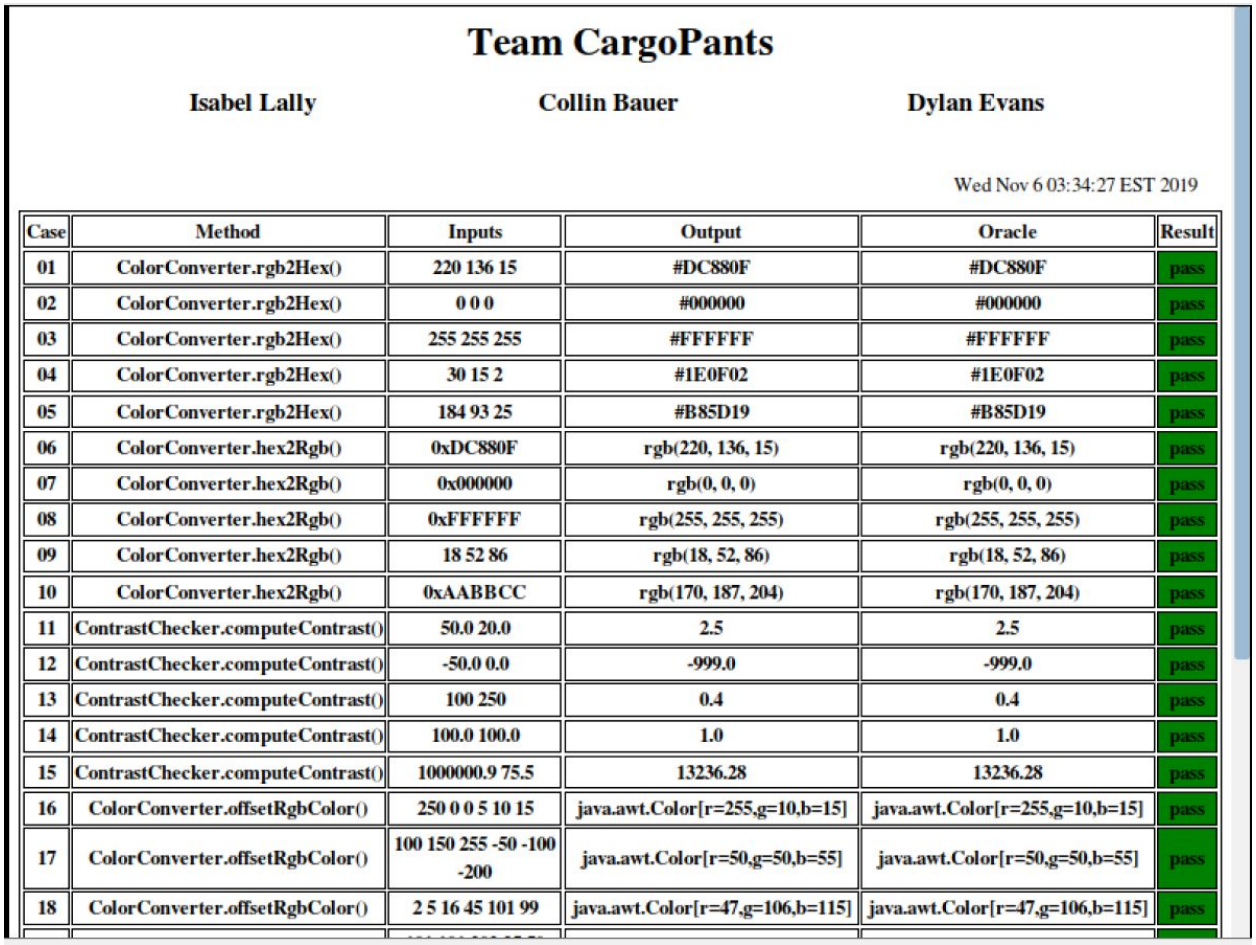
This is how the oracles of the five test cases were decided, however, the output values are different. Our calculations were double checked for any mistakes, just in case. The reason for the failures appears to be that instead of squaring the difference of the rgb values between colors, the developers cubed the values.

HTML output from runAllTests

As is, our scripts did run properly and gave the results from our tests, but they had to be altered so it would output to an HTML file. This required a lot of rewriting and the creation

of a template which our scripts would reference and build an output HTML from. This was done by using sed to read lines from the template and parsing it with regular expressions to replace an internal portion of it with the desired output.

Example output



Team CargoPants

Isabel Lally **Collin Bauer** **Dylan Evans**

Wed Nov 6 03:34:27 EST 2019

Case	Method	Inputs	Output	Oracle	Result
01	ColorConverter.rgb2Hex()	220 136 15	#DC880F	#DC880F	pass
02	ColorConverter.rgb2Hex()	0 0 0	#000000	#000000	pass
03	ColorConverter.rgb2Hex()	255 255 255	#FFFFFF	#FFFFFF	pass
04	ColorConverter.rgb2Hex()	30 15 2	#1E0F02	#1E0F02	pass
05	ColorConverter.rgb2Hex()	184 93 25	#B85D19	#B85D19	pass
06	ColorConverter.hex2Rgb()	0xDC880F	rgb(220, 136, 15)	rgb(220, 136, 15)	pass
07	ColorConverter.hex2Rgb()	0x000000	rgb(0, 0, 0)	rgb(0, 0, 0)	pass
08	ColorConverter.hex2Rgb()	0xFFFFFF	rgb(255, 255, 255)	rgb(255, 255, 255)	pass
09	ColorConverter.hex2Rgb()	18 52 86	rgb(18, 52, 86)	rgb(18, 52, 86)	pass
10	ColorConverter.hex2Rgb()	0xAABBCC	rgb(170, 187, 204)	rgb(170, 187, 204)	pass
11	ContrastChecker.computeContrast()	50.0 20.0	2.5	2.5	pass
12	ContrastChecker.computeContrast()	-50.0 0.0	-999.0	-999.0	pass
13	ContrastChecker.computeContrast()	100 250	0.4	0.4	pass
14	ContrastChecker.computeContrast()	100.0 100.0	1.0	1.0	pass
15	ContrastChecker.computeContrast()	1000000.9 75.5	13236.28	13236.28	pass
16	ColorConverter.offsetRgbColor()	250 0 0 5 10 15	java.awt.Color[r=255,g=10,b=15]	java.awt.Color[r=255,g=10,b=15]	pass
17	ColorConverter.offsetRgbColor()	100 150 255 -50 -100 -200	java.awt.Color[r=50,g=50,b=55]	java.awt.Color[r=50,g=50,b=55]	pass
18	ColorConverter.offsetRgbColor()	2 5 16 45 101 99	java.awt.Color[r=47,g=106,b=115]	java.awt.Color[r=47,g=106,b=115]	pass