# Tanaguru Contrast Finder Testing Framework

By: Omer Omer, Joshua Bingham, Eduardo Abreu

# Table of Contents

# Choosing and building Project

The first project that we were originally looking at was the Martus project. Martus is a software that allows users to document incidents of abuse by creating bulletins, and storing them on redundant servers located around the world. For this project, we had a lot of problems trying to get all the tests to run and the files were all in random spots that would have been too much trouble to go back and organize everything.

After we decided that we didn't want to work on Martus we went with the Tanaguru Contract-Finder project. Tanaguru Contract- Finder is a web app that finds a valid background / foreground color contrast for accessibility. Some of the things that we needed to run the project was Git, Tomcat, Maven.

When first trying to build and run the project we had some difficulties. It seemed like the instructions they had up on how build and run it was out dated. So, we spend some time trying to fix this. After a few hours of failed attempts, we decided to try on contact the creator of the project, Frederic Halna. We let him know that we were students at the College of Charleston and that we started to work on his project. We told him that we were having trouble trying to build and run it. After

telling him all this he was excited to hear that we were working on his project and was more than willing got help us out. He invited us to the projects slack page and answered and questions we had. We were really thankful for his help.

# Test plan

We wrote our testing script in python, then writing Java classes as necessary in order to instantiate and run the objects we are testing. We wrote the script framework first, so that we have a clear plan as to the syntax and format we need to use going forward. Once we had the framework complete we began to finish up and polish the test cases, and create the necessary Java classes and begin testing our test cases to make sure everything worked as planned. After that all we had to do was inject faults into the code as well as iterating through all the test cases and making sure that everything was working properly.

Our projected Schedule:

October 15th: Have methods we will be testing decided on

October 22nd: Have 10 total Test Cases decided on and created

October 31st: Have test framework completed and working

November 7th: Have all 25 Test Cases decided on and created

November 14th: Have implementation of framework completed

November 21st: Inject faults into code to induce failures

Our schedule we had planned flowed the dates to when the deliverables were due.

# Testing Framework

Our script is written in python. The script run through each test case in our test cases folder. An example of how our test case look like is:

1. ID= #XXX
2. Class being tested (relative file path as string)
3. Method being tested (as string)
4. Requirement Being run (as string)
5. Inputs (as object(s))
6. Comandline input (as string)
7. Expected result (as return type?)

Each test case has a unique Id and indicates which class is being tested. It also puts which method is being tested and that allows us to use the right driver when

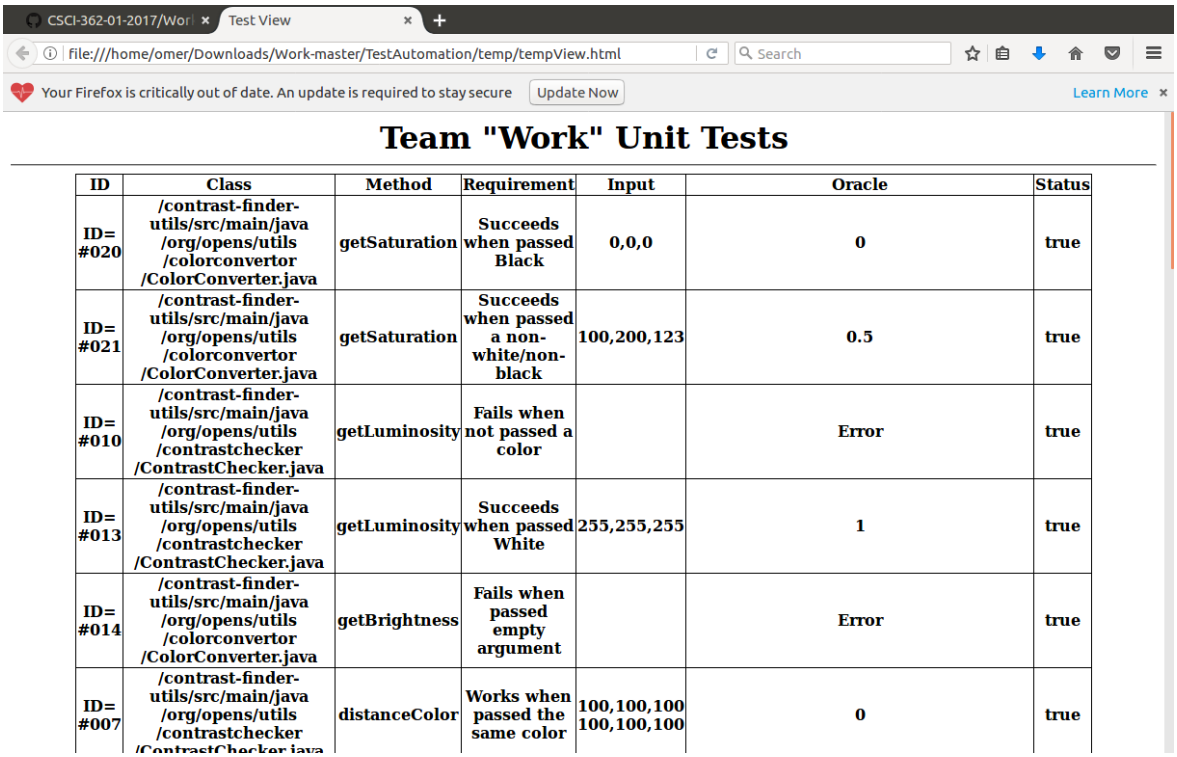
running our tests. We also included the Oracle for the test case. Then lastly we put our input and expected output. After we run our script it outputs a table of each test we run and then whether the test passed or failed.  Here is an example of a actaual test case we used:

1. ID = #001
2. ContrastChecker
3. distanceColor
4. Works for identical inputs
5. new Color(100,100,100),new Color(100,100,100)
6. Comandline input (as string)
7. 0.0

You can see which method is being tested and what our inputs and outputs are.

## Refining and more tests

We now have 25 test cases that work just fine. An example output of our script being run looks like:




## Team "Work" Unit Tests

| ID | Class | Method | Requirement | Input | Oracle | Status |
|---|---|---|---|---|---|---|
| ID= #020 | /contrast-finder-utils/src/main/java /org/opens/utils /colorconvertor /ColorConverter.java | getSaturation | Succeeds when passed Black | 0,0,0 | 0 | true |
| ID= #021 | /contrast-finder-utils/src/main/java /org/opens/utils /colorconvertor /ColorConverter.java | getSaturation | Succeeds when passed a non-white/non-black | 100,200,123 | 0.5 | true |
| ID= #010 | /contrast-finder-utils/src/main/java /org/opens/utils /contrastchecker /ContrastChecker.java | getLuminosity | Fails when not passed a color | | Error | true |
| ID= #013 | /contrast-finder-utils/src/main/java /org/opens/utils /contrastchecker /ContrastChecker.java | getLuminosity | Succeeds when passed White | 255,255,255 | 1 | true |
| ID= #014 | /contrast-finder-utils/src/main/java /org/opens/utils /colorconvertor /ColorConverter.java | getBrightness | Fails when passed empty argument | | Error | true |
| ID= #007 | /contrast-finder-utils/src/main/java /org/opens/utils /contrastchecker /ContrastChecker.java | distanceColor | Works when passed the same color | 100,100,100 100,100,100 | 0 | true |

To show if a test passed or failed we use true and false, true meaning the test passed and false meaning the test failed.

## Fault Injections

We injected faults to the code so we could make some of our test cases that passed now fail instead. The way we injected faults was quite simple. We made faults to the method distance calculator. This method was basically a math formula. We changed an addition to subtraction and by doing that we created faults. This did not create faults to test cases that use this method but have bad inputs. For these the

expected output is error and we still get error because it never does the calculation, it just reads that the input is invalid.

# Reflections

This project was a great experience for us. It was our first time working on open source software. Also, it was really interesting make a script that would read and run all of our test cases. Some of the things we need to work on was being more organized on when we meet. Since all 3 of us are full time students and work full time, finding time to meet was very difficult. This project was an amazing experience for all of us.