Team Work

One of the problems we've been having is trying to figure out how to run all the tests that the Contrast-Finder has. We found out somethings were modified in the project that were not also modified in the build that the Tanaguru project has.

After spending a lot of time trying to figure out what needed to be changed in the build we decided that it may be a good idea to contact the leader of the Tanaguru Project, Frederic Halna. He was more than willing to help us. After a few emails with him, he invited us to the projects Slack page and let us ask whatever questions to the people who helped develop the project.

TEST PLAN

Process

We intend to write our testing script in Python, writing Java classes as necessary to instantiate and run the objects we are testing. We plan to write the script framework first, so that we have a clear plan as to the syntax and format we will need to use going forth. Once the framework is completed, we will finish and polish the test cases, and create the necessary Java classes and begin testing our tests to ensure everything works as planned. From that point, it will be inputting faults into the code, as well as iterating through the tests and ensuring that everything is ready for the final project.

Testing

We plan to unit test various methods throughout the project, aiming for about 5 tests per method chosen. We are trying to choose classes that do not have a lot of dependencies throughout the project, to avoid needing to compile many classes for a one-unit test.
As of the time of this writing, our test case template has not been finalized, as we want to write our framework and figure out what the best way to format the test cases will be.

Schedule

NOTE: This schedule is not complete, and is subject to change.
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

DELIVERABLE #2

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

TESTING TEMPLATE

Format:

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. Expected result (as return type?)

Test 1:

1. ID: 000
2. ContrastChecker
3. distanceColor
4. Fails if only passed one Color
5. new Color(100,100,100)
6. Error

Test 2:

1. ID: 001
2. ContrastChecker
3. distanceColor
4. Works for identical inputs
5. new Color(100,100,100),new Color(100,100,100)
6. 0.0

Test 3:

1. ID: 002
2. ContrastChecker
3. distanceColor
4. Fails if not passed anything
5. null
6. Error

DELIVERABLE #2

Test 4:

1. ID: 003
2. ContrastChecker
3. distanceColor
4. Works when passed different colors
5. new Color(100,100,100), new Color(200,50,123)
6. 166.21973408714141003380012224457

Test 5:

1. ID: 004
2. ContrastChecker
3. distanceColor
4. Works when passed black and white
5. new Color(0,0,0), new Color(255,255,255)
6. 441.672955930063709849498817084