Deliverable 2: Test Plan

The testing process

Testing will be carried out by running a runAllTests script from the linux command line that will walk through the testCases folder and use the specification files to execute each test case. This will be shown in an html file comparing the results to what is expected.

Requirements Traceability

The Tanaguru software should be able to properly calculate the contrast between two colors, as well as produce a more favorable color combination if needed.

DistanceCalculator.java: calculate(Color colorToChange, Color colorToKeep):
The calculate method takes two java Color objects as input and then, using each's RGB values, finds the Euclidean distance between them.

Tested items

We are testing various methods from the Tanaguru Contrast Finder software.

DistanceCalculator.java: calculate(Color colorToChange, Color colorToKeep):
Our tests include having the two far ends of color, white and black, as well as having the higher color in both the first and second input, having the same color as both inputs, and two tests of different colors in between the range of RGB.

And 4 other methods

Testing Schedule

The schedule for testing will follow along with the class deadlines.

Test recording procedures

When the testing framework is invoked, a html file will be created and opened in a browser to display the results of the tests.

Hardware and software requirements

The testing framework will be run from the Linux command line using Java 7. The rest will be included in the testing framework.

Report

After looking through both the Tanaguru Contrast Finder and STEM source code, we decided to go with Tanaguru due to STEM's extensive code dependencies. When making the test cases for the calculate method, we found that what the documentation says the method returns is incorrect. It says it uses the distance formula but the code itself uses cubic operations rather than squared. Because of this we decided to base our expected outcomes on the Euclidean formula, even though that means tests will fail.

Test Case Specifications Format:

Test ID //Name of the test file

Tested class //Name of the class being tested

Tested method //Name of the method being tested

Method Parameters //Parameters for the method being tested

Executable Compile //Command line for compiling test executable

Executable Run With Arguments //Command line for running test executable with arguments

Expected outcomes //Expected outcomes of the test

calculate_1 DistanceCalculator.java Calculate Color(255,255,255), Color(0,0,0) javac /testCasesExecutables/calculate.java java /testCasesExecutables/calculate 255 255 255 0 0 0 441.67	calculate_2 DistanceCalcualtor.java Calculate Color(0,0,0), Color(255,255,255) javac /testCasesExecutables/calculate.java java /testCasesExecutables/calculate 0 0 0 255 255 255 441.67
calculate_3 DistanceCalculator.java Calculate Color(0,0,0), Color(0,0,0) javac /testCasesExecutables/calculate.java	calculate_4 DistanceCalcualtor.java Calculate Color(255,200,34), Color(255,255,17) javac /testCasesExecutables/calculate.java

java /testCasesExecutables/calculate 0 0 0 0 0 0 0.0	java /testCasesExecutables/calculate 255 200 34 255 255 17 57.57
calculate_5 DistanceCalculator.java Calculate Color(200,7,16), Color(57,57,57) javac /testCasesExecutables/calculate.java java /testCasesExecutables/calculate 200 7 16 57 57 57 156.94	