Automated Testing Script with Tanaguru

Cass Outlaw, Scott Stolarski, Drew Bigelow, Justin Arends

CSCI 362 Software Engineering

What is Tanaguru?

- Tanaguru is an accessibility website that finds the contrast between the font and background. This aides those who suffer from color blindness.
- Tanaguru is written in java
- Tanaguru is an example of Humanitarian Free and Open Source Software (HFOSS)

Why Tanaguru?

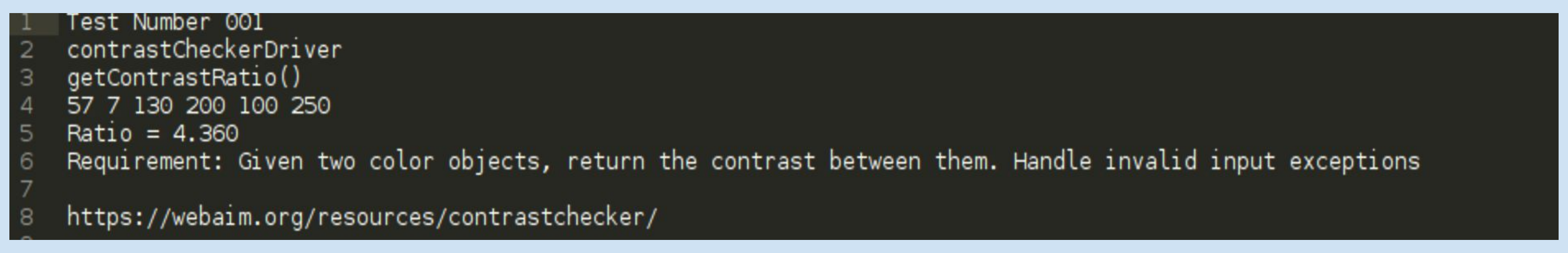
- The Tanaguru GitHub, **at first glance**, looked well documented with an easy to understand file structure and a helpful interactive website.
- Tanaguru included fully built test cases on their GitHub, which appeared to be well organized. This turned out not to be true.

Testing Process

- Methods within Tanaguru will be tested
  - getContrastRatio(Color color1, Color color2)
  - getLuminosity(Color color1)
  - getSaturation(Color color1)
  - rgb2Hex(Color color1)

Test Cases

- Test case Template to be saved as a text file
  - Test number
  - Class being tested
  - Method being tested
  - Test input to be given to the driver
  - Expected outcome
  - Requirement being tested
  - Source where output is verified

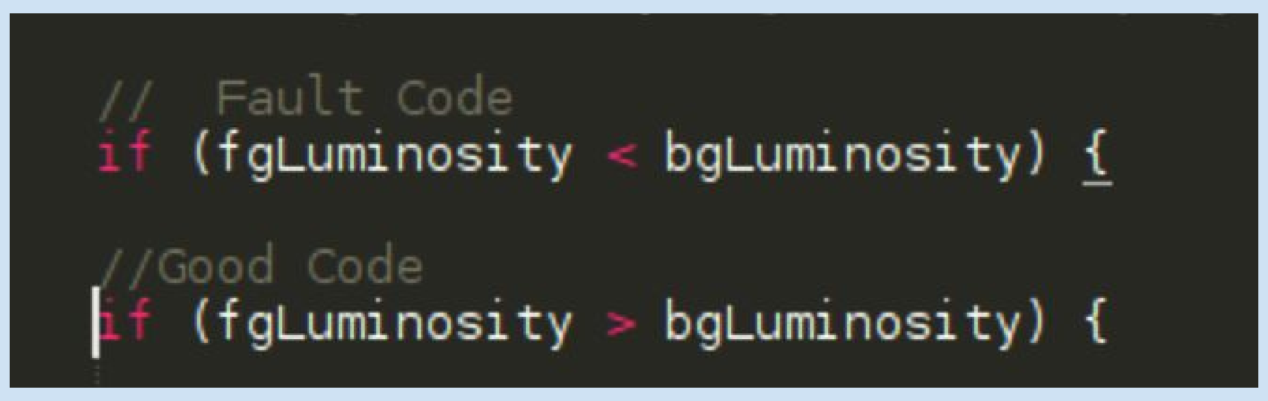


```
Test Number 001
contrastCheckerDriver
getContrastRatio()
57 7 130 200 100 250
Ratio = 4.360
Requirement: Given two color objects, return the contrast between them. Handle invalid input exceptions

https://webaim.org/resources/contrastchecker/
```

Fault Injection

- We purposefully added errors to the Tanaguru code to insure the robustness of our tests.
- Errors included simple changes, such as switching mathematical operators.



```
//  Fault Code
if (fgLuminosity < bgLuminosity) {

//Good Code
if (fgLuminosity > bgLuminosity) {
```

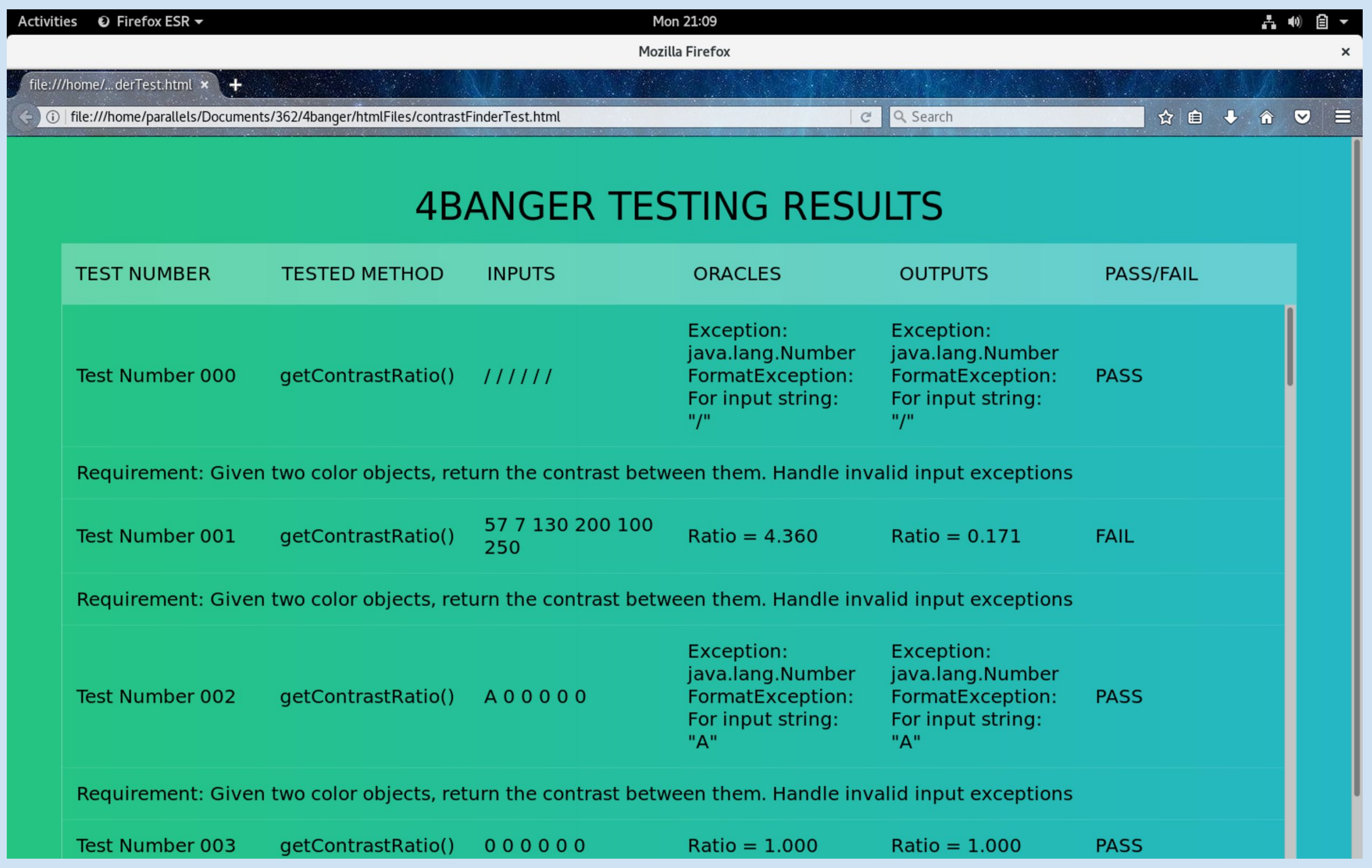
- The fault injection verified that our system is robust and can operate without the Color-Contrast-Finder functioning the way it is supposed to

The Script

- We use a bash script to run our test cases
  - At first, the script required the drivers and amount of tests to be hardcoded into it.
  - This would be problematic if people outside of our group would want to add test cases.
- To add a new test case, the user only needs to add the case itself and a driver.

Output

When the script is done running the tests, the output is formatted and displayed on a webpage



Activities
Firefox ESR
Mon 21:09
Mozilla Firefox
file:///home/...derTest.html
file:///home/parallels/Documents/362/4banger/htmlFiles/contrastFinderTest.html
Search

4BANGER TESTING RESULTS

TEST NUMBER	TESTED METHOD	INPUTS	ORACLES	OUTPUTS	PASS/FAIL
Test Number 000	getContrastRatio()	/ / / / / /	Exception: java.lang.Number FormatException: For input string: "/"	Exception: java.lang.Number FormatException: For input string: "/"	PASS
Requirement: Given two color objects, return the contrast between them. Handle invalid input exceptions					
Test Number 001	getContrastRatio()	57 7 130 200 100 250	Ratio = 4.360	Ratio = 0.171	FAIL
Requirement: Given two color objects, return the contrast between them. Handle invalid input exceptions					
Test Number 002	getContrastRatio()	A 0 0 0 0 0	Exception: java.lang.Number FormatException: For input string: "A"	Exception: java.lang.Number FormatException: For input string: "A"	PASS
Requirement: Given two color objects, return the contrast between them. Handle invalid input exceptions					
Test Number 003	getContrastRatio()	0 0 0 0 0 0	Ratio = 1.000	Ratio = 1.000	PASS

Lessons Learned

- Dealing open source projects is difficult
  - The documentation of the Tanaguru code was limited or nonexistent
  - The instructions to compile and run Tanguru were outdated
- We needed to make sure the script was not reliant on other elements of the testing framework
  - Our original script needed a hardcoded list of test cases and drivers
  - This was changed to allow the addition of test cases without modifying the script