



SE2832 Lab 8: GUI Testing

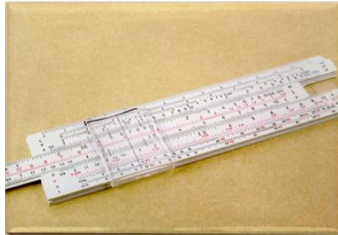
Due: May 7, 2014 23:59

1 Lab Objectives

- Perform user interface testing using a UI testing tool (specifically selenium)
- Perform testing on a web application using a testing tool.

2 Introduction

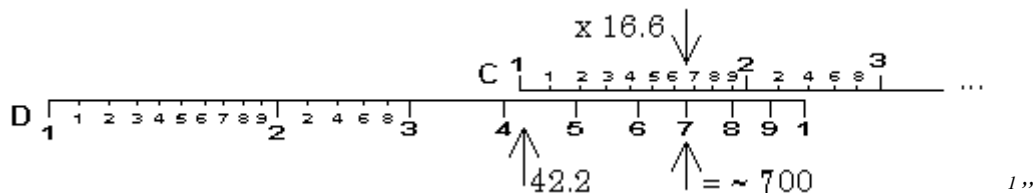
The calculator. Oh, where would we be without the calculator? Just picture the fun you would have. Instead of a \$100 TI89 calculator, you would walk into your class with this:



And, instead of entering programs into your calculator, you would memorize algorithms such as the following:

“To multiply two numbers on a typical slide rule, the user set the left index (start of the scale) on the C scale to line up with one factor on the D scale. (All labels refer to Pickett scales. Scale labels were not completely uniform between brands.) The user then found the second factor on the C scale and looked on the D scale for the product. By doing this, the user effectively added the logs (lengths) of the two numbers and looked up the antilog.

Multiplications with more than a single digit were carried out by making use of the smaller graduations to represent additional digits of decreasing significance. The precision available to the user was directly proportional to the size of the device (or the smallest lines the user could resolve.) The slide rule did not indicate the decimal point. That was done by the user - typically by estimation, "common sense" or by computing the characteristic. For example:



Wouldn't it be fun to figure out how to test this type of a system? It would be great fun. But, since the extent of your knowledge of this domain is limited to the previous paragraph and the scenes from Apollo 13 that you have watched (Note: Homework assignment: Watch Apollo 13 if you have never done so), that isn't going to be lab. (Big sigh of relief. There will be no calculations in lab.)

¹ <http://www.hpmuseum.org/srinst.htm>

Instead, we're going to test a web application which is less capable than a traditional slide rule. We are going to test a very basic calculator. This calculator has all the bells and whistles of a free calculator that you might get out of a gum ball machine, and the UI is as pretty as you would expect for a free web application. But, it's functional, and thus, you can test it (and you will find bugs...)

3 Step 1: Install Selenium

The first step of this lab is to install Selenium. Selenium works best with Firefox, though it can be used with other browsers. To install selenium, go to <http://docs.seleniumhq.org/download/> and download version 2.5.0 of the Selenium IDE. With Firefox, it will download the plug in and ask you if you wish to install it in Firefox. You will then need to restart Firefox.

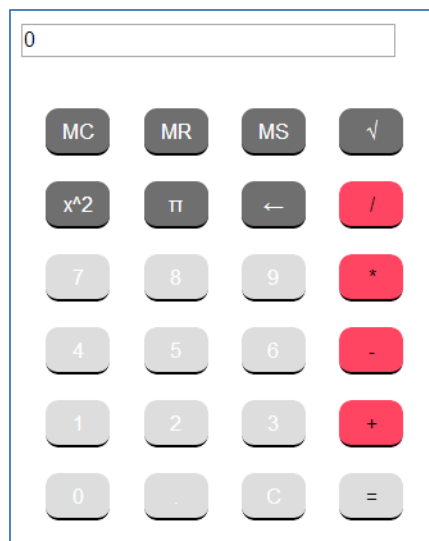


Figure 1: The calculator

4 Step 2: Developing Automated tests to use for regression

If you go to the course website, there is a version of a program that will do calculations. It has an absolutely stunning visual appearance, as is shown above. Using this as a baseline, you should develop a set of regression tests in selenium to to automatically test the operation of the GUI. Be as thorough as you think you need to be to properly test the program. Once you have finished the tests, save the test suite to an html file.

5 Step 3: Running regression tests

If you look at the website, there is a version of this complex web application which is broken, written using JavaScript. (Note: You are all expected to be experts in Java Script at this time due to your concurrent enrollment in your Web Applications course.) Download the program to your local machine and run the test script against it. When there are failures, fix the code in the JavaScript / html files. It is all very well documented and clean. In the end, you will have a fully functioning calculator in which all regression failures have been removed.



As you remove bugs from the HTML and JavaScript, put comments in the corrected code explaining the defect and why you are fixing it.

6 Deliverables / Submission

Now that you have completed your lab assignment, submit the following lab report detailing your experiences. The lab report should be submitted electronically through the course upload page.

1. Introduction
 - a. What are you trying to accomplish with this lab? This section shall be written IN YOUR OWN WORDS. DO NOT copy directly from the assignment.
2. Strategy
 - a. How did you go about determining your test cases?
 - b. How did you organize your test cases?
 - c. Why did you choose the test cases and how did you approach organizing them in a logical fashion.
3. Bugs
 - a. List, in a tabular format, the bugs which you found during testing. Where were they found in the source code and what was wrong?
4. Critique
 - a. What problems do you see with using this approach on a large scale web application?
 - b. How is this technique easier or harder than the other techniques you have used in class?
5. Things gone right / Things gone wrong
 - a. This section shall discuss the things which went correctly with this experiment as well as the things which posed problems during this lab.
6. Conclusions
 - a. What have you learned with this experience?

In addition to this, you should submit a zip file of your test cases and corrected source code through the website.

If you have any questions, consult your instructor.