The Early Birds pecking away at open source software with a testing framework
Team name: The Early Birds

Jan-Michael Carrington, Vincent Nguyen, Phillip Wilson
CSCI 362, Fall 2016

COLLEGE of CHARLESTON

Introduction

In the software engineering class, we were assigned the task of finding an open source software we were interested in and create a testing framework to test the methods in the software.

The purpose of this semester long project was to give us experience with agile and iterative programming. It also taught us the importance of good documentation for later developpers to use or improve a software.

Matplotlib is a python 2D plotting library which produces quality figures in a variety of hardcopy formats. Since we all have not used Python since the first introductory coding class, this was a nice fresh reminder of how Python works.

Open Source Projects

At first, we started our work on the software Amari, however it quickly became evident with our current experience, we could not test it properly. We kept hitting walls of dependencies as we were trying to compile the source code from Github. We moved to pixi.ks, but discovered it was written in a language we were not familiar with, javascript. As a result, we flew to matplotlib.

The python 2D plotting library called matplotlib piqued our interest. Thanks to the amazing documentation provided by the software developpers of matplotlib, picking out methods to individually test was easy picking.

pixi.js logo

matplotlib logo

The Framework

The testing suite the team built focuses around the script runAllTests.py. It is ran from the command line. It goes through each test case file in the test case folder and runs the test on the specified method. An example test case file is shown below. The framework gets the result of the test and compares it to the expected output. The framework then returns a “Passed” or “Failed” value. The data is collected and exported to an automatically opened html.

```
jan-michael@janmichael-VirtualBox:~/temp2/The-Early-Birds-master/TestAutomation$
 ./scripts/runAllTests.py
Running tests please wait...

Tests complete.
jan-michael@janmichael-VirtualBox:~/temp2/The-Early-Birds-master/TestAutomation$
```

runAllTests.py started from the terminal

```
04
Converting a color to its RGBA value
Colors
colors.to_rgba(color)
'(1.0, 0.0, 0.0, 1.0)'
red
testCaseExecutable4.py
```

testCase.txt content

Automatically opened html with test reports

Fault Injections

Injecting faults into the source code was a lot of fun. First spotting where the library was built, and then fiddling around in its source code to insert a random fault into a line.

The code was found at:

/usr/local/lib/python2.7/dist-packages/matplotlib-2.0.0b4_2382.g042ebc7-py2.7-linux-x86_64.egg/matplotlib

And the injections were inserted in dates.py on line 186 and colors.py on line 246. Getting to break code so it fails a test cases is a lot more entertaining than it sounds.

Automatically opened html with test reports after fault injections

Reflections

This was the team’s first time working on open source software and after completing it, we can confidently say we have learned the importance of good documentation. There is no way we would have gotten this far without the proper documentation provided to us.

If we were to do this once more, we would manage our time a lot better. We heavily underestimated the time it would take us to get each deliverable done even though Dr. Bowring clearly told us this would be a time consuming semester long project.

Unfortunately we did not have the skillset to tackle on projects like SugarLab and Amari, but even with a simple open source software like matplotlib, this was a very rewarding experience.

Getting to work on real projects rather than the little toying scripts we are assigned in the other coding classes was extremely refreshing and challenging.