ASTE 404 HW 7 Fall 2021 USC

Nikita Persikov

***Abstract*—This report summarizes the work performed for HW7 in ASTE-404 by Nikita Persikov. The ASTE 404 HW 7 assignment was completed. Unit tests were implemented. The tests passed. Documentation was generated using Doxygen. A repository was created on GitHub through GitKraken.**

***Index Terms*—Google Test, Doxygen, Github, LaTeX, GitKraken**

I. Introduction

A c++ class was created to define an object that behaves as a MATLAB vector. It had a dot operation and a magnitude operation (called mag). The *dot* operation is defined below:

$$\vec{a} \cdot \vec{b} = \sum_{i=0}^{2} a_i b_i = a_0 b_0 + a_1 b_1 + a_2 b_2 \tag{1}$$

The *mag* operation was defined as follows:

$$|\vec{a}| = \sqrt{\vec{a} \cdot \vec{a}} \tag{2}$$

The equation shown in 1 is coded up as

```
friend T dot(const _vec3<T>&a, const _vec3<T>&b)
{
    return a[0]*b[0] + a[1]*b[1] + a[2]*b[2];
}
```

This requried operator overlaoding. For example, addition was implemented like this:

```
friend _vec3<T> operator+(const _vec3<T>&a,
    const _vec3<T>&b)
{
    return _vec3<T>(a[0]+b[0],a[1]+b[1],
        a[2]+b[2]);
}
```

We also learned how to use Doxygen, `Google Test`, and LaTeXThese tools are summarized in Table I. I used GitKraken since I knew how to set up repositories that way. (I kept parts of the template table just to see how the Doxygen link worked).

Technology	Use
Doxygen	Developer guide documentation system
GTest	Unit Testing for C++
Github	Source Control Repository
GitKraken	Source Control User Interface
LaTeX	Technical paper writing language

II. Source Control

A Github repository was set up, as shown in Figure 1. Instead of using the command line, this was set done using GitKraken, which is a user friendly way of creating, managing, and doing git operations on repositories. The result is shown in Figure 2

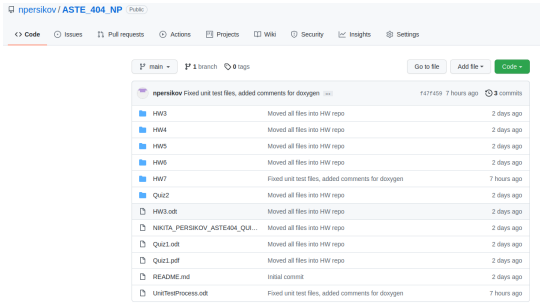

Fig. 1. Screenshot of the brand new Github repo

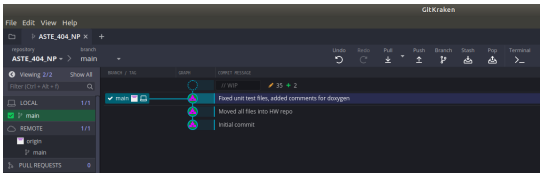

Fig. 2. Screenshot of the GitKraken view of the repo history/branches

The Google Test repo found in [1] provided setup instructions. A link to my HW repository can be found here.

III. Unit Testing

The code itself was written in Visual Studio Code. The unit testing was created using Google Test. I used the method from [2] and got this passing result for the simple dot() unit test.

Fig. 3. Screenshot of the first unit test attempt

For the advanced version with the class (PART C), I got this result with the same method of compiling the cmake text file:

MS Student, Department of Astronautical Engineering e-mail: persikov@usc.edu

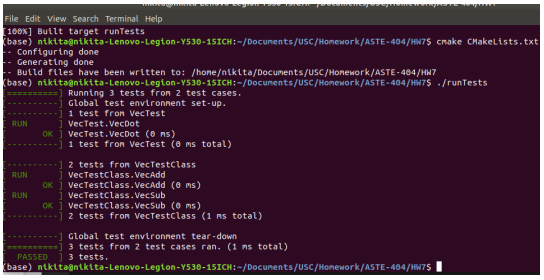

Fig. 4. Screenshot of the second unit test attempt

IV. Doxygen Documentation

The following are screenshots of my generated html documentation file. For some reason, the graphical hierarchy did not generate.

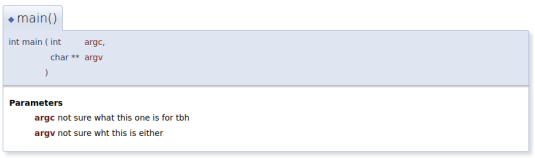

Fig. 5. Parameter descriptions written using @param tag in the c++ code processed in Doxygen

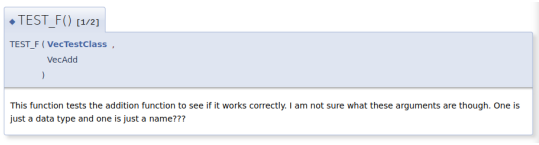

Fig. 6. This image shows a unit test with a description

Fig. 7. This shows both a function description and parameter descriptions

V. Conclusion

The code worked as intended and passed all unit tests. Version control has been successfully set up. Doxygen was used to create an html documentation file that correctly displays parameter and function desccriptions. Finally, a LaTeXreport was compiled to summarize the results.

References

[1] https://github.com/google/googletest.
[2] https://www.eriksmistad.no/getting-started-with-google-test-on-ubuntu/.