ASTE-404 Homework 7 Report

Luis Diaz

***Abstract*—This report demonstrates the integration of core software engineering practices, including code development, unit testing, documentation, version control, and report writing with LaTeX. These processes collectively ensure robust, maintainable, and well-documented code [1].**

***Index Terms*—Google Test, Doxygen, GitHub, LaTeX**

I. Introduction

This assignment aimed to implement software engineering principles across multiple stages. We focused on five key areas: code development, unit testing, documentation, version control, and report creation using LaTeX. This report summarizes these practices, describing the tools and techniques applied.

References to online resources include various useful websites such as Slovak Cooking [2] and the personal blog of the project author [3]. Additionally, the **Particle in Cell** journal article [4] provides essential background on numerical simulations used in propulsion modeling.

II. Part 1: Code Development

Below is the C++ code defining a templated 3D vector class ‘vec3‘ with operations such as vector addition, subtraction, dot product, and magnitude calculation.

Listing 1. C++ Vector Code
```
#ifndef _VEC_H
#define _VEC_H
#include <ostream>
#include <math.h>

template<typename T>
class _vec3 {
public:
    _vec3<T>() : d{0, 0, 0} {}
    _vec3<T>(T a, T b, T c) : d{a, b, c} {}

    T& operator[] (int i) { return d[i]; }
    T operator[] (int i) const { return d[i];
        }

    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]);
    }

    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]);
    }

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

    friend double mag(const _vec3<T>& a) {
        return sqrt(dot(a, a));
    }

    friend std::ostream& operator<<(std::
        ostream& out, const _vec3<T>& a) {
        out << a[0] << " " << a[1] << " " << a
            [2];
        return out;
    }

protected:
    T d[3];
};

using double3 = _vec3<double>;

#endif
```

III. Part 2: Unit Testing

We used Google Test (GTest) to validate the correctness of the vector operations. Below is the test code to verify the vector dot product and addition functionality.

Listing 2. Unit Test Code with GTest
```
#include "gtest/gtest.h"
#include "vec.h"

TEST(VecTest, VecDot) {
    double3 a{0, 1, 2};
    double3 b{0, 0, 1};
    EXPECT_EQ(dot(a, b), 2);
}

int main(int argc, char **argv) {
    ::testing::InitGoogleTest(&argc, argv);
    return RUN_ALL_TESTS();
}
```

IV. Part 3: Documentation with Doxygen

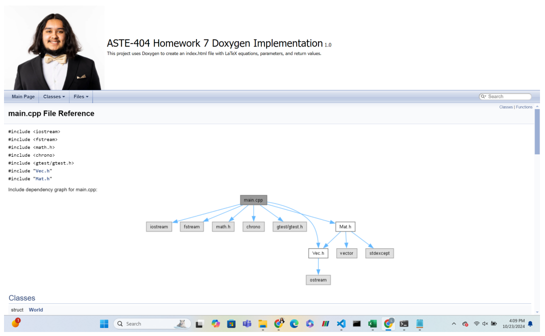
The code was documented using Doxygen to generate HTML documentation. Below is an example of the Doxygen comment for the main function.

Listing 3. Doxygen Documentation Example
```
/**
 * @brief Entry point for the program.
 *
 * This function initializes two vectors and
     performs operations
 * such as subtraction and dot product.
 *
 * @return int Program exit status.
```

Student, Department of Astronautical Engineering, USC.

```
 */
int main(int argc, char** argv);
```

Figure 1 shows a screenshot of the generated Doxygen documentation.



ASTE-404 Homework 7 Doxygen Implementation 1.0

This project uses Doxygen to create an index.html file with LaTeX equations, parameters, and return values.

main.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <math.h>
#include <chrono>
#include <gtest/gtest.h>
#include "Vec.h"
#include "Mat.h"
```

Include dependency graph for main.cpp:

Classes

struct World

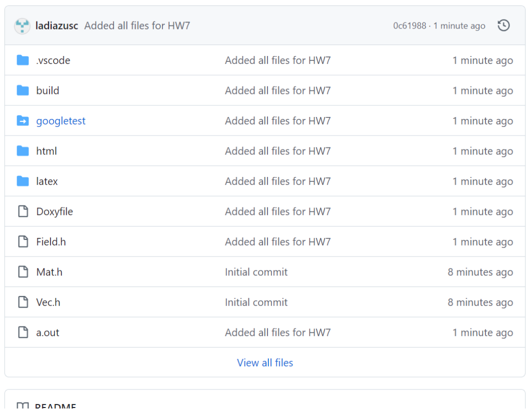
Fig. 1. Doxygen Documentation Output

V. Part 4: Version Control with Git

The project was version-controlled using Git and pushed to a GitHub repository. Below are the commands used during the workflow.

Listing 4. Git Commands
```
$ git init
$ git add main.cpp vec.h
$ git commit -m "Initial commit"
$ git remote add origin https://github.com/
   user/repo.git
$ git push -u origin master
```

Figure 2 shows a screenshot of the repository.



ladiazusc Added all files for HW7 0c61988 · 1 minute ago

.vscode	Added all files for HW7	1 minute ago
build	Added all files for HW7	1 minute ago
googletest	Added all files for HW7	1 minute ago
html	Added all files for HW7	1 minute ago
latex	Added all files for HW7	1 minute ago
Doxyfile	Added all files for HW7	1 minute ago
Field.h	Added all files for HW7	1 minute ago
Mat.h	Initial commit	8 minutes ago
Vec.h	Initial commit	8 minutes ago
a.out	Added all files for HW7	1 minute ago

Fig. 2. GitHub Repository Screenshot

VI. Part 5: LaTeX Report Creation

This document was created using LaTeX, ensuring professional formatting and precise mathematical typesetting. Figures, code snippets, and hyperlinks were embedded seamlessly.

VII. Conclusion

This assignment provided practical experience with key software engineering practices. The combination of development, testing, documentation, version control, and reporting promotes the creation of reliable and maintainable software solutions. Several online resources such as *Particle in Cell* and related sites [1]–[3] were valuable in contextualizing the concepts discussed.

References

[1] C. Brieda, “Particle in cell website,” https://www.particleincell.com, accessed: 2024-10-23.
[2] B. Brieda, “Slovak cooking website,” https://www.slovakcooking.com, accessed: 2024-10-12.
[3] A. Brieda, “Personal website,” https://www.iamlubos.com, accessed: 2024-08-21.
[4] L. Brieda, “Particle in cell,” *International Electric Propulsion*, vol. 10, no. 100, 2026.