Summative Assessment: e-Portfolio

Reflective Essay

Introduction

Overall, the module work itself was quite extensive compared to the previous module, which I was not expecting. I had to learn several new concepts of the Python language relatively quickly, a stressful but ultimately enjoyable task. I gained quite a bit of knowledge relating to the Python language, UML diagramming, and different types of UML diagrams and their place in the software development lifecycle. I also believe my writing has improved as a result of the essays and discussion pieces in the formative exercises in each unit. In this essay I will talk about the things I learned throughout the module, the things I found challenging, and how I could possibly improve in the future. I will also go over the assignments I did in units 7 and 11 and talk about my experience and takeaways from them. I’ll sum up the essay with a brief reflection on where I feel I can improve going forward.

## Takeaways

Working my way through the module I learned about UML diagramming, and although I knew they existed, had never used them. They gave me an insight into what it weould be like to professionally design a piece of a software and allowed me to apply it to a project, albeit inefficiently, which I will go into detail later on. I had previously used Python extensively for scripting as it is rather fast compared to PowerShell, and as I’m sure anyone who has worked in the tech industry knows, time is an extremely valuable resource. I had not used Python in any intermediate capacity and found that it had quite a few, in my opinion, nuisances. These included having to pass “self” into each class method, the odd way of importing other files/modules, the strict whitespace requirement which I was aware of became much more of an issue as classes and files grew in size, and threading is a little odd for specifically Python 2.7.15 which was the Python version in the Codio environment we were required to use. I find the way threading is implemented in .NET for example much easier to understand and utilize effectively. Python’s odd naming (\_\_init\_\_) and other class methods such as \_\_repr\_\_ or \_\_str\_\_ felt almost hacky to use. I’m sure that future modules will also be using Python, but outside of coursework I believe I will avoid using Python for anything more complicated than a script, for something that required OOP principles I would use .NET. I look forward to developing my skill in Python, and perhaps my opinion will change over time as I become more comfortable with the language.

## Assignment Evaluations

Assignment 1 was, having previously had no knowledge of UML diagrams, a challenge. The sequence diagram that I was unable to complete in time for submission was, in essence, a simple thing to understand but proved to be a little difficult to wrap my head around. I skipped it for the first assignment, but I managed to not only wrap my head around it but do what I think was a competent version of a sequence diagram for the second assignment (system implementation). I would have liked to see the system design essay either have some of the required references removed or the word count increased, as I felt 500 words and 7 required references a bit too distracting from the point of the essay, the design of a system that includes three functions of an autonomous vehicle. Regarding my project, I personally felt that this was a decent attempt at a system design, albeit a bit basic. It turns out that designing a system without spending a significant amount of time thinking about how the system(s) interoperate is demanding. Were I to do this assignment over, I would spend much more time on the diagram design and learning best practices, but I believe the module overall was a good introduction to UML diagramming and the software development lifecycle.

Assignment 2 was very challenging for me, I found. As I started working out my design in my Python program, I found that the original system design failed to be applicable as I scaled and modified the scope of the program. Ultimately, an unfortunate combination of a serious lack of time, scope creep, and what I can most likely attribute to inadequate system design regrettably led to me cutting corners, specifically with the functionality of it. I chose three subsystems to support, Automatic Emergency Braking (AEB), Adaptive Cruise Control (ACC), and Lane-keeping Assist (LKA), and LKA ended up being cut from scope which will definitely not help my grade there. I believe that if I had more time to go back and redo the system design, I could have more efficiently used the time I had to code the program which probably would have avoided or at the very least minimized the issue of removing or leaving out functionality.

## Summary

In summary, this course overall was a good study of object-oriented programming concepts and principles, and while I had some issues with assignment requirements and my personally available time to devote to the course, I had a couple of great takeaways including UML diagramming and becoming more familiar with Python. I feel relatively more competent as a programmer, but I feel that UML diagramming will almost certainly be something I will use for personal projects and for potential larger-scale automation at my workplace. Unfortunately, I was not able to devote the time required to partake in any forum discussion and it wasn’t previously listed as a requirement or something that would eventually be compulsory and count toward an assessed assignment. That is partially my fault for not reading the final assignment requirements closely enough. Other than the group projects I believe I could personally improve scope and management more effectively, and I suspect using and being more comfortable with UML modeling will assist with that.