Module Reflection

https://github.com/bgraham89/uoeo-eportfolio/tree/module/oop

OOP

During this objected-oriented programming module, I've explored topics such as the principles of OOP, software requirements analysis, OOD, the UML language, software quality theory, software metrics and design patterns. This theory has been very helpful for me, being a self-taught programmer. It's filled in some gaps, and brought me up to the standards associated with a bachelor's degree, and then pushed me beyond that by requiring active research, and critical reasoning skills.

I already have a job as a game developer, and I've been implicitly using OOP daily, but I've noticed that since this module, I've been much more active at my workplace in trying to understand more technical aspects of what I do. For example, I I now have a firmer understanding of the concepts available to me in Typescript such as the differences between interfaces and types and implementations of abstract base classes, and I have a better understanding of the overall framework implementation, which involves factories and arbiters. I attribute this development to the research skills and terminology I've gained during this module. Furthermore, parts of the module inspired me to build a script to convert our typescript framework into a class diagrams, using stuff I picked up from the UML and software quality theory parts of the course too. I found educational value in the visualisations that UML provide. I'm hoping to have my side project fully implemented by September, (which would be my one-year anniversary there).

Python

During the course I've explored Python to a level where I'm confident with OOP constructs, and the pythonic deviations of it, and I now have stuff to showcase related to that. I wish I had had the time to practise building coroutines and using asynchronous construct of python, as these are concepts that I think would be beneficial in more technically demanding python roles. I tried to focus on the main course topics and the research around those to ensure I fulfilled the module objectives, because OOP is widely renowned. It was difficult not to side track too much with Python, as I've used python quite a lot, so I'm comfortable it. So putting too much of my attention on it, could have been distracting to working with classic OOP principles. That being said, to be critically reflective requires not just accepting a solution to the problem that's been identified, but it requires trying to build solutions too. I believe I found a decent balance of talking about the principles expected and my own, but only feedback will tell. I've at least got tangible evidence of my understanding of OOP on my eportfolio.

Regarding what to do next with python, I'm not entirely sure. It would make sense for me to try and learn tensorflow or pytorch for the application potential, or a backend framework like flash or Django. But my current work domain is in game development where languages like C# and C++ dominate. One domain I've noticed that connects these languages is robotics. I think exploring Robot Operating System (ROS) is a reasonable direction for broadening my experience, overlapping with

C++ for an introduction, while keeping my prior experiences in python and machine learning relevant. I'm pretty sure the alternative of web development is not a field I want to go into, at least not while considering my interests. For money and an entry point, it might not be a bad idea though.

Other Practise

During the course I've had the chance to practise the design and implementation of autonomous vehicles. Other microprojects included designing a poker engine, exploring model-based reasoning, and designing my eportfolio. I also participated in multiple discussions with my peers.

Within my eportfolio I reflected on different parts of the course in more detail, but to summarise:

- The autonomous car project helped me develop research, software design and software implementation skills. Generally, I believe I've gotten better at sifting through academic papers for relevant research, measured by attempt to design an autonomous car system. I would like to conduct an actual software literature review and hone these skills. I can use car project to showcase some experience with OOD, and OOP too. Embedding the project into the domains of game development, or even hobby-level robotics is another reasonable to spend my time. My priority though is to finish the course, while I'm working full time. So, this reflection will be useful to read back on once I'm done, for inspiration if I need it.
- The poker engine design convinced me of the value of certain UML diagrams, by facilitating an improved design for an old home project.
- Identifying the concept of model-based reasoning helped me make explicit some of the interests that I've had for a long time. In the past I've been on a theoretical physics degree, I switched degree to one based on mathematics and psychology. Later on I became an English language teacher. All this while being broadly interested in many fields. The concept of modelling is apparent in all of these fields, and it labels the aspects of these domains that I've been interested in. So, this simple concept ended up being significant to me, and I'll use it as a thematic direction in the future to coordinate my efforts. The challenge of making a breadth of interest into a depth for the sake of a personally meaningful profession is one that'd like to conquer. I guess the next thing to do would be to perform an SLR of this concept, or the domain it's situated within.
- Designing my eportolfio has been rewarding in that I've been able to explore the presentation of information, reflect on, coordinate, and showcase my efforts and understanding of OOP, computer science and science generally.
- The discussion with my peers was awesome, I genuinely wish I could be apart of paper-centred discussions more frequently. It's important to be able to articulate ideas, and construct critical arguments for the sake of science. Professionally, this brings respect too, as it's what's expected of seniors. I'd rather work for somebody who has a mastery of critical reasoning, so that's the standards I aspire to gain. I'm expecting the course to facilitate more of that development.

I think one the biggest areas I need to improve on is academic integrity. The challenge for me in the next module, if I pass this one, is to restrict the content of my portfolio to material on explicit models, for 100% integrity.