Reflective reviews

N. Kean

NMIT: [13522542]

Table of Contents

[Reflective reviews 3](#_Toc119403094)

[Project Milestone 1 3](#_Toc119403095)

[Project Milestone 2 4](#_Toc119403096)

[Project Milestone 3 5](#_Toc119403097)

[References 6](#_Toc119403098)

Reflective reviews

This review consists of journal entries related to work done around each of the 3 milestones for course SDV-602.

# Project Milestone 1

Having never touch python before, weeks 1-3 of the course were an exercise in mapping the way I’m used to coding (Perl) to the Python way. On July 24th I converted my Wordle solver written in Perl to Python; A surprisingly easy task given the similarities of the two languages. Lists, dictionaries, and sets all map straight to Perl and thinking of tuples as simply frozen lists completes the set.

In week 4 the homework was to knock up a simple GUI adventure game. This was the first real opportunity to put what I’d learned into practice. By using a data structure of a dictionary of dictionaries, the top level keyed off location nodes and the inner level keyed off actions, it meant that successfully navigating this structure would pretty much guarantee dictionaries were understood. Jumping straight into a GUI for this piece of homework turned out to be super useful as getting to grips with PySimpleGUI and its various idiosyncrasies would eat up a lot of the development time of my project.

Completing the LinkedIn Learning course “Python for non-programmers” was instructive but tedious. Getting used to VS Code, rather than developing in vim, was a hurdle but eventually worth it.

I thought that structuring the course around the development of a single project (per student) was a left-field sort of idea, having been used to the idea of “Teach it then test it”. Milestone One included a design and documentation section which I thought perhaps belonged more in the “Professional and Technical Writing” course. It did try to force a program design on me from the beginning though.

# Project Milestone 2

The two headaches with this milestone were getting to grips with PySimpleGUI (proper layout of elements in a window) and splitting the application code into MVC sections. I’ve always been a big fan of monolithic files of code with single-use functions being in the same file as the calling code and only library functions needing their own files. I ended up relying on the “Fn+F12” (jump to definition) of VS Code to find things, not to mention the ever-useful: find . -type f -print0 | xargs -0 grep “something” ☺

MVC gave me an ”ah ha” moment when the event processor was converted from a bunch of “if-else” statements to looping through a list of controls. Only then did I see any benefit. I still spent a lot of time trying to figure out where I put functions though.

To polish my Python skills and understand PySimpleGUI better, I started writing my own version of Conway's Game of Life (Kean, 2022). When I had it working, it felt slow, so I learned how to use the Python profiler to see where the bottleneck was, only to find out that the PySimpleGUI overhead was slowing the program down. Even after optimising it to lower the number of PSG calls made, it wasn’t as fast as I’d liked. One day I may rewrite it using the pygame library instead of PySimpleGui. Perhaps I will get around to doing my own hexagonal cell version one day after all.

For this milestone, the matplotlib examples from week 6 were heavily leaned on for creating real, rather than placeholder images. While I enjoyed the ease of using the pandas library to load excel files, I was a bit disappointed with the speed at which it ran. Since I understood the internal structure of an XLSX file well, it was hard to see why loading a file took so long. Still, much better to learn how to use a standard Python library than waste time writing my own XLSX reader!

Rather than limp through the LinkedIn Python courses, I found it much more useful to use YouTube as my video tutor. The mCoding (Murphy, 2022) channel was especially useful, although the YouTube algorithm would eventually start recommending me videos that were exactly what I was needing. Machine Learning FTW!

# Project Milestone 3

I’ve never been a fan of JSON, perhaps because I’ve never been a fan of JavaScript in general - give me a nice XML interface any day. Doing database manipulation via a URL felt strange, coming from an Oracle background where SQL\*Plus was my interface of choice even after “better” interfaces (Toad, SQL Developer, etc.) became available. Having hacked away at DBI::Oracle in Perl for 20+ years, changing to a non-transactional JSON interface felt like a step backwards.

Decorators look like they could simplify so much code. No need to build my own closures to do caching anymore; just use the @functools.cache decorator!

When we learned about list comprehensions, so many weeks ago now, I remember thinking along the lines of “when am I ever going to use that?”. How wrong I was. Having missed the whole “Pan/Zoom” adjustment section of Assignment Two, I needed to implement that into my final version. Oh, I can filter my data list with one line of code? Great. I can apply a function to every element in a list with 1 line of code? Nice.

Threading in Python is great! It is much easier than thread management in Perl. Plonking my PyLyfe application into my project (as an easter egg) was incredibly easy - simply import it and add a control to fire it off in its own thread. I used PyLyfe to learn how to use multiprocessing/multithreading since at that time my project wasn’t in a place where that could be implemented easily. Suddenly I could have a “Stop” button that actually works; I switched configuration parameters to on-the-fly options that could be toggled in the middle of a generation. I’d not even considered this until I started playing with threads.

If I had more time on this project, I’d look at using AWS or some other service for storing the data though since I ran into a lot of performance problems with the JSON service on newsimland.

# References

Kean, N. (2022) *The Game of Life. Written in Python.* GitHub.

https://github.com/NJK-NMIT/PyLyfe

Murphy, J. (2022) *mCoding.* YouTube.

https://www.youtube.com/c/mCodingWithJamesMurphy