**Weekly Learning Log**

**The Challenge**

The objective of my project is to use cameras such as GoPros which will be attached to the end effector of an already built robotic arm to perform photogrammetry in order to create digital representations of real-life objects

**Week 2**

I spent all of this week learning CherryPy, OOP and Arduino programming. I found it all relatively easy, however when learning CherryPy I found that the base documentation isn’t great, so I had to look at other articles as well.

* ~~Learn CherryPy basics~~
  + I need to learn how to use CherryPy so I can understand the current codebase for the robotic arm as well as so I know how to make web servers which I may need in the future for my project if I’m making a web interface for the camera attached to the end effector of the robotic arm. I found the basics of CherryPy challenging at points, but I just had to keep trying and experimenting with it until I fully understood it.
  + Learnt using <https://docs.cherrypy.org/en/latest/>
* ~~Learn OOP in python~~
  + I need to learn OOP in python so I know how to create object-oriented programs with Python such as when creating an API for remotely controlling a camera on the robotic arm or one to automatically perform photogrammetry. I found this very easy to learn because I had learnt OOP in C# and Java before, so learning OOP in python was quite easy.
  + Learnt using <https://realpython.com/python3-object-oriented-programming/>
* ~~Learn Arduino programming~~
  + I need to learn how to program using an Arduino to understand the Arduino code that controls the motors of the robotic arm currently as well as because I may use an Arduino at some point in the future, such as to remotely control a camera or to fix an issue with the current Arduino. I found learning how to code with Arduinos relatively easy since you can do most things (control the pins) with a few different commands.
  + Learnt using <https://www.youtube.com/watch?v=QO_Jlz1qpDw>
* ~~Learn how to use conda~~
  + I had to learn how to use anaconda through the command line so that I could easily set up the environment using the .yml file so I had all the packages needed to successfully run the robotic arm codebase. I had a lot of challenges with this because the .yml file was for packages on Windows, however when I was trying to run it on MacOS, there were a lot of errors, so to fix this I had to keep experimenting until I fixed the issue
  + Learnt using this cheat sheet: <https://docs.anaconda.com/anaconda/user-guide/cheatsheet/>

**Week 3**

Overall, I’m happy with what I learnt this week. This week, I learnt how to use SSH, Fusion and how to use the GoPro APIs. I had a challenge when trying to learn Fusion since I had experience in other CAD software such as SolidWorks, so learning new software was challenging at points. Also, since the GoPro APIs weren’t very well documented, it made it difficult to learn them.

* ~~Learn how to use ssh~~
  + I had to learn how to use SSH, and more specifically SSHPass to allow me to SSH into the Raspberry Pi Zero from the Pi 3B programmatically. Learning both of these were relatively easy. I had to use SSHPass to automatically SSH in with a password.
  + I learnt how to use it here: <https://askubuntu.com/questions/282319/how-to-use-sshpass>
* ~~Learn how to use Raspbian command line~~
  + I had to learn how to use the Raspbian command line, which I found relatively easy. The commands I used the most were **ls** to show all the files, **cd** to move directory, **pwd** to show the current path directory and **sudo nano** to edit files as an administrator when sshing into the Pi Zero.
  + I learnt how to use it here: <https://www.elektormagazine.com/news/bash-command-cheat-sheet>
* ~~Learn how to use Fusion (CAD Software)~~
  + I learnt how to use Fusion so that I could 3D print a piece for the end effector of the robotic arm so I could screw on the portable, waterproof enclosure for the GoPro. I found this challenging at times trying to learn how to use new CAD software, however after playing with it I got the hang of it.
  + Learn how to use it here: <https://www.autodesk.co.uk/products/fusion-360/learn-support>
* ~~Learn how to use GoPro API~~
  + I had to learn how to use the GoPro API **goprocam** so that I can control the GoPro using Python commands. I found this challenging at points to learn because there is very limited documentation and guides on how to use this. So, I’ve created a how-to guide on how to use it now which helped further spark my knowledge of the API. You can find the how-to guide on my GitHub.
  + GitHub: <https://github.com/RyanStronge/CameraWeb>
  + Goprocam documentation: <https://pypi.org/project/goprocam/>

**Week 4**

I didn’t really learn much this week because I was implementing what I had already learnt this week into actual code by creating the remote-control GoPro system so that I could call the methods from another class to remotely control the GoPro. I did however start to learn how to use Agisoft MetaShape / PhotoScan to perform photogrammetry which I found relatively easy to understand.

* Learn Agisoft Metashape
  + I learnt how to use Agisoft Metashape by reading articles and watching YouTube videos. I did attempt to perform photogrammetry this week but was unsuccessful and didn’t find a solution.
  + I learnt how to use the software here: <https://www.youtube.com/watch?v=Y9K-a91w9UA>

**Week 5**

This week, I spent most of my time learning how to use Flask and CherryPy to create a web interface for the GoPro. I had a number of issues with CherryPy as I couldn’t get it working, so halfway through I switched the web server architecture to Flask which I found much better and simpler. I also had to recap how to use HTML and CSS in some areas such as divs and I also had to learn how to use Ajax.

* ~~Learn how to use Flask~~
  + I had to learn how to use Flask since I couldn’t get the web interface working when using CherryPy since there’s very little, difficult to learn, documentation for CherryPy I found. I couldn’t get CherryPy working with Ajax because whenever I clicked a button, which triggered Ajax, it wouldn’t run the CherryPy route method. In the end, I found Flask much easier to use to allow Python scripts to be run from HTML on a web server.
  + I learned how to use it with the Flask documentation: <https://flask.palletsprojects.com/en/1.1.x/quickstart/>
* ~~Learn how to use Ajax~~
  + I had to learn how to use Ajax so I could use POST and GET methods to trigger Flask methods to run the python scripts in order to remotely control the GoPro. I found learning Ajax quite challenging at first because the syntax is quite difficult and not self-explanatory. However, after trial and error, I eventually got comfortable with it.
  + I learned how to use it here: <https://www.w3schools.com/jquery/jquery_ajax_intro.asp>
* ~~Recap HTML / CSS~~
  + I felt the need to recap HTML and CSS in some areas such as <div> tags in HTML and layouts in CSS since I hadn’t used them in a long time. I found this very easy to recap and it only took an hour.
  + I recapped this using this article: <https://www.w3schools.com/tags/tag_div.ASP>

**Week 6**

This week, I spent some time learning how to use the PiCamera commands, however this took quite a short amount of time. Other than this, I didn’t learn much this week as I spent most of my time adding validation to the web interface, implementing the PiCamera part into my web interface and doing the frontend of the web interface.

* ~~Learn how to use PiCamera via Python~~
  + I had to learn how to use PiCamera via Python so that I could implement a PiCamera section into my web interface so the user can remotely control both a GoPro and a PiCamera. I also learnt how to use livestreaming with the PiCamera when learning this so I can show the live feed of the PiCameras on the web interface.

**Week 7**

I spent most of my time learning this week. I was learning how to use Swift and XCode in order to develop the iOS app for the next part of my project. I found XCode quite easy to use, however found Swift quite difficult to learn at the beginning.

* ~~Learn Apple iPhone programming (Swift)~~
  + I found this somewhat challenging at the beginning because the syntax was hard to use learn I found. However, after playing about with it for a few hours in the Playground feature of XCode, I got confident enough with it. I’m still slightly unsure on a few things such as the meaning of ‘?’ and the meaning of ‘!’ beside variables on it, but I will continue working on it.
  + I learned how to use Swift using the Swift documentation: <https://swift.org/documentation/>
* ~~Learn how to use XCode~~
  + I found XCode very easy to use. The only issue I ran into was when running scripts in the Playground mode, I took a very long time to execute since I have a relatively old Macbook (2015 Pro). To fix this, in the playground settings, I changed the platform from iOS to macOS.
  + I learned how to use XCode here: <https://codewithchris.com/xcode-tutorial/>

**Week 8**

I spent most of my time this week learning new parts of Swift and playing about with it, creating small apps and testing different features with the iPhone until I feel comfortable starting to work on more complex applications such as ARKit. I did, however, start to experiment with ARKit in the later part of the week which I found quite challenging.

* ~~Learn how to use ARKit~~
  + I researched how to use ARKit using many different sources such as Apple’s documentation as well as multiple well explained YouTube videos. I found it quite challenging to understand all of the methods used to detect an object at first, however after running the code and experimenting with it, I felt more comfortable with it.
  + I learned how to use it here: <https://developer.apple.com/documentation/arkit>
  + and here: <https://www.youtube.com/watch?v=FEqBW3cKF2k>

**Week 9**

This week, there wasn’t much to learn because I was mostly using Flask which I have used a lot in the last couple of months. I did however have to learn some smaller libraries such as the web server framework, Swifter, the hardware device framework, Luminous and the JSONSerialization library.

* ~~Learn how to use Swifter (Swift web server)~~
  + I had to learn how to use swifter as an alternative to Flask so that I could run and make routes for websites on the iPhone. It wasn’t very well documented, so I learnt how to use it by reading the source code and testing it.
  + I learnt how to use it here: <https://github.com/httpswift/swifter>
* ~~Learn how to use Luminous~~
  + I had to learn how to use Luminous to get the hardware device information from the iPhone. This was easy to learn since you just had to call methods for different types of information.
  + I learnt how to use it here: <https://github.com/andrealufino/Luminous>
* ~~Learn how to use JSONSerialization library~~
  + I had to learn how to use this so I could output the hardware information in a JSON file style to make it easier to read and understand.
  + I learnt how to use it here: <https://stackoverflow.com/questions/29625133/convert-dictionary-to-json-in-swift>

**Week 10**

This week there wasn’t anything to learn because I was just working on what I had already learnt.

**Week 11**