**Diary**

**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

**Weekly To-Do – Week 1**

* ~~Meet with Ryan McGovern regarding robotic arm~~
* ~~Install all software on laptop~~
* ~~Read CherryPy documentation~~

*Friday 17th January 4pm*

Contacted Ryan McGovern regarding the robot arm and arranged to meet him on Wednesday at 2pm in the IOT space in the CSB so he can show me how the robot arm works and how to program it etc.

I have also installed Python 3.8 on my Macbook and installed CherryPy using pip. I am now reading the Cherrypy basics documentation to help with my understanding of how the robot arm is communicated to using HTTP.

At the minute, I’m very excited to get started with this project because it’s something that I’ve never really done before and I know that I can learn a lot from it. However, I’m also quite worried because it’s such an unfamiliar territory.

*Saturday 18th January 6pm*

Finished the ‘Basics’ documentation on the CherryPy website so I know how CherryPy works (<https://docs.cherrypy.org/en/latest/basics.html>). I also have started to look into Object-Oriented programming in python as I only know the core concepts of python currently by reading articles online and watching videos on YouTube. I’ve never really learnt an HTTP-based language like CherryPy before, so it is quite difficult to understand at times, however when I try to continue looking at it / doing research on it, I eventually find I understand it more and that makes me feel relieved.

*10pm*

By 10pm, I have finished learning about OOP in python (inheritance, polymorphism, method overloading / overriding) by programming along with a YouTube tutorial and now feel more comfortable coding projects in python. Over the next few days I plan to start looking into Arduino programming and look at some of the tutorials / advanced aspects of CherryPy. In the following days I’m planning to attempt to start learning Arduino Programming. I’m slightly worried about this as I have never used an Arduino before. I’m also currently worried about the mathematical aspect of the upcoming project as I didn’t do A Level maths. However, if I do research on maths topics that I’ll need for this, I should be fine.

**Weekly To-Do – Week 2**

* ~~Setup Anaconda on laptop~~
* ~~Try to understand existing code for robotic arm~~
* ~~Debug robotic arm code~~
* ~~Understand how to move robotic arm in different ways~~
* Research GoPro remote control

*Monday 20th January 6pm*

Today I started researching Arduino programming, I’m not finding it too bad as I’ve programmed a lot in C# before at A Level, so it’s relatively similar. I also started looking into buying an Arduino to try and help with my knowledge of it but I’m currently unsure what one to buy, so I will question Ryan McGovern on this on Wednesday when I’m meeting him to show me the robotic arm. I have also started watching a long YouTube video on Arduino programming as I find watching videos easier to learn than consuming large amounts of text (<https://www.youtube.com/watch?v=QO_Jlz1qpDw>). I’m planning to start looking into photogrammetry software and more advanced programming with CherryPy in the coming days as well as looking into trigonometry as I believe it is used in the programming of the robotic arm. I’m slightly worried that I’m moving too fast currently and need to slow down and do more research into the topics before moving on.

*Tuesday 21st January 10am*

This morning I started by doing some more research on Arduino programming by reading some example code that other people had made projects with as well as reading some Arduino documentation. After this, I started to look at the Tutorials section on the CherryPy website, some of which I found quite difficult and advanced because there was a lot of examples that I was unfamiliar with / hadn’t heard about before, but after spending a long time testing it out myself, I finally understood it. I’m planning to continue working my way through the tutorials for CherryPy to ensure I have a solid understanding of the language. At the minute, I’m slightly worried about the scope of the project and what it should actually do. However, I can ask John about this at some point which will help calm my mind a bit. Having learnt the Arduino programming and about CherryPy, I feel more excited about this project now that I’m getting stuck into it.

*Wednesday 22nd January 6pm*

Earlier on today I met with Ryan McGovern who developed the robotic arm and he showed me how it worked and how to control the arm using the software. Before I met with him, I was quite worried / nervous because I didn’t know anything really about robotics and it felt quite intimidating. However, after he showed me how it worked and I started to look into the code of it, I started to feel much more comfortable.

Today I installed the Anaconda environment needed for the robotic arm project which was a challenge as I was trying to install it on Mac OS where it was made on Windows 10 leading to many errors. Due to this, it took me several hours to install it as it required a lot of research and troubleshooting to get it to work. When I was doing this, it was quite challenging mentally as I felt like giving up at times, however whenever I completed it, I felt accomplished and relieved. This has helped to show me that if I keep pushing on and not giving up, I will complete it eventually even if it takes a few days.

*Thursday 23rd January 1pm*

In the morning I started by debugging some of the code for the roboticArm which I had downloaded on GitHub. One of the problems was that pylint was giving a false error for a numpy method due to an outdated version of pylint. I tried to fix this error by updating pylint however this didn’t fix it which was quite annoying. However, by continuing to work on this error for 2 hours, I was finally able to fix it by reading pylint documentation and also StackOverflow threads regarding others having this same error. When trying to fix this error, I was very confused because it had never happened to me before and I had to read through threads on GitHub which I was very unfamiliar with. However, when I discovered that it was nothing I could fix for the meantime, it made me feel more at ease.

Later on in that day, I went into the IoT lab in the Computer Science Building again to make sure my build of the project was working with the robotic arm. Whenever I got everything set up and running, when the robotic arm was calibrating one of the joints (joint 4) wasn’t calibrating correctly. Due to me being relatively unfamiliar with the mechanics behind the robotic arm, I was unsure why it wasn’t working which made me quite stressed, so I emailed Ryan McGovern who had shown me the robotic arm and he thinks that one of the limit switches aren’t working properly and that we are going to meet on Wednesday afternoon so he can show me how to fix this problem if it arises again, which helped my stress.

Reading through the code, especially for the kinematics sections of it have made me quite worried since I don’t understand most of it because it contains a lot of maths, especially trigonometry which I haven’t studied since GCSE. To help my understanding of this, I plan to watch videos and read online lessons on trigonometry to help myself understand this better.

*Saturday 25th January*

*1pm*

This afternoon, I started looking into ways to remotely control a GoPro to connect to the robotic arm in order to take pictures of the objects in the future for photogrammetry purposes. After spending a few hours looking at different ways to achieve this, I found a way that I can connect the GoPro to an Arduino using soldering, however it can fail and damage the GoPro so I’m slightly nervous about this method. Another method is that I can connect the GoPro to a WiFi network and use an API from GitHub to allow the GoPro to be remotely controlled using an HTTP interface such as CherryPy. However, this method requires a lot more equipment such as an extra router to allow the GoPro to be connected to a network. I’m currently unsure of which method I should use to do this but I’m going to continue doing research on this area to find the best method possible. I’m currently unsure with a few areas of the code and how to modify it in order to allow for remote control of the GoPro such as taking pictures. Due to this, I’m worried about the workload because of how unfamiliar it all is but at the same time I’m quite excited to get stuck into it. I’ve delayed myself from reading more of the book about photogrammetry for now or else I feel that it will make me more stressed as it is very detailed.

*10pm*

Tonight, I spent about an hour reading through some of the code again and trying to understand it. I have found that by leaving something for a day or even a few hours and coming back to it helps me immensely because it helps to clear my head and not get stressed. Now coming back to it, I understand it slightly more, I still don’t understand some parts of it, but I can question Ryan McGovern on these parts next week. In the coming days I plan to continue looking at the GoPro remote controlling and also how I’m going to attach the GoPro to the end effector of the robotic arm.

**Weekly Summary**

I’m relatively happy with the work that I’ve completed this week. I’m starting to feel more confident with the robotic arm code and I feel like after a few more days of playing around with it, I’ll be able to move on to the next step in the project. I’m quite worried about how much maths I’m going to have to use in this project and how I’ll actually conduct the photogrammetry process as well as how I’m going to actually move the robotic arm around the real-world object and what method I’m going to use to do that.

**Weekly To-Do – Week 3**

* ~~Research GoPro remote control~~
* ~~Research different GoPro APIs~~
* ~~3D print holder for GoPro on robotic arm~~
* ~~Attach GoPro holder to end effector of robotic arm~~
* ~~Figure out how to connect camera to robotic arm~~
* ~~Create How-To for robotic arm~~

*Monday 27th January 1pm*

Today I’m starting by continuing my research on how to remotely control the GoPro and I think I have a solution of getting another Raspberry Pi which can be permanently connected to the GoPro’s network and I’ll be able to control it from there. From reading through the code of the robotic arm over the last couple days, I’m feeling more confident with it than I was a couple days ago, however I’m slightly confused regarding the scope of the project such as whether it should be able to create a digital representation of any object in a room and how I pick which object it performs this action on. I’ve also been slightly worried about how the kinematics demos work and the different types of kinematics, however I can ask about this on Wednesday in the IoT lab with Ryan McGovern. I’m going to try and acquire a GoPro and a Raspberry Pi so that I can test out this concept of remotely controlling the GoPro to see if it would work or not. I also need to start looking for a way in which I can attach the GoPro to the arm which I think I’ll need to 3D print a piece for this. I’m not too worried about this part because I used a lot of CAD software and 3D printing at A Level in Technology & Design.

*8pm*

This evening, I’ve started working on my how-to guide by downloading a Wikipedia template and editing it to show how the robotic arm is being used. I’m finding it fairly easy because it’s only using HTML and CSS. I’m planning to make my how-to guide on how to use the robotic arm since there currently isn’t any proper documentation on it so it would be very useful for others trying to learn how to use the arm.

*Tuesday 28th January 1pm*

Today, I continued my research on remotely controlling a GoPro and I think I’m going to use two raspberry pi’s now instead of a raspberry pi and an Arduino as I’ve found the **goprocam** library which allows me to remotely control the gopro using HTTP commands when a the device is connected to the GoPro’s WiFi. However, I’m currently unsure of how I’m going to automate this process. Now that I understand the main python code for the arm, I’ve started looking at how the commands are actually sent to the raspberry pi from the web interface using CherryPy. Now that I’m making progress in this area, I’m feeling a little less stressed since I feel closer to the end of this big step since it’ll be my first big deliverable whenever I get it implemented into the system. After having a meeting with John today, I feel much more confident / happier with the scope of the project and I know what exactly I should do in the project.

*8pm*

This evening, I’ve continued to work on my how-to guide. I’ve got most of the basics of how to use the arm documented and now I just need to document more advanced parts of it. I’m waiting to ask Ryan McGovern tomorrow about a number of things such as how the commands are actually sent and hoping he’ll be able to explain the code to me that I’m unfamiliar with. When reading through the code, it was worrying me because I felt that I should know how it works but when I stopped and thought about it, I felt better.

*Wednesday 29th January 2pm*

Today, I met with Ryan McGovern in the IoT lab in the CSB where I had the opportunity to ask him a number of questions about the code of the arm which helped greatly. It’s helped to put my mind at rest since I now understand how the commands are actually sent to the Raspberry Pi and how to use CherryPy which will help me greatly with controlling the GoPro on the arm.

I also designed a part using Fusion that will be able to screw onto the end effector of the robotic arm and allow the GoPro to be held securely. I got this 3D printed using one of the 3D printers in QLabs and attached to the GoPro. After doing this, it made me feel more excited about the project again since I was actually pushing out a deliverable and was a big step in the right direction. Now that I have this mount built, it will be very useful for others since they can attach a GoPro to the mount too and use it instead of my own.

After the work I got completed today, I’m also feeling more confident with my project as I know what direction I’m going to be going now. However, I’m still worried about a couple things such as how the robotic arm is going to know where to move to which will take some mathematical formulas which is one of my weak points. I’m also worried about how exactly the GoPro will be controlled at this point in time but through research I should be able to find a way.

*Thursday 30th January 1pm*

Today, I continued my research on how to remotely control the GoPro and I think I’ve found a way. I’m going to use a USB cable to connect the two Raspberry Pi’s so that I should be able to execute a python script on the second raspberry Pi which will be connected to the GoPro’s WiFi. I’m now planning to use a different library which will allow me to import a library called **goprocam** and run commands to perform actions on the GoPro. This method will also allow me to download all the files from the GoPro which was another feature I was unsure of how to achieve before today. I tested this library out using the GoPro and connecting my laptop to the GoPro’s WiFi and it works perfectly.

*Saturday 1st February 9pm*

Today, I finished off the how-to webpage and correctly formatted it to make it easy to read and understand. In my how-to page, I included details on the software needed for the robotic arm, how to setup the environment, how to connect to the robotic arm, the web interface commands, how to run demos, how to use kinematics in order to move the robotic arm and how to write demos. I decided to make the how-to guide on this as there wasn’t currently any other documentation on how to control / use the robotic arm and I had to learn by asking questions and people showing me how to use it. However, with this how-to guide, any person with at least some technical knowledge should be able to easily control the arm by following the step-by-step guides in this guide. I decided to make the how-to guide look plain but simple as I think it will be the easiest to read and it’s split up into different sections to allow users to quickly skip to any part of the webpage when they need a recap on how to do something.

I also finalized how I’m going to achieve the remote control of the GoPro. I’m planning to use ssh commands via USB from the host device (raspberry pi connected to the robotic arm) which will ssh into the ‘slave’ device which will be a Raspberry Pi Zero, connected to the GoPro’s WiFi. This will allow me to remotely run the python script on the Pi Zero to take a large number of pictures needed for photogrammetry. Now that I’ve figured this out, I feel much more relieved as remotely controlling the GoPro was a very big part of the project and will allow me to move onto the next big part once I’ve worked this out.

**Weekly Summary**

I feel like I’ve made a good amount of progress this week now that I have the end effector part for the robotic arm made so that I can attach a GoPro to it. It feels good to actually have made something now that can be useful to other students in the IoT lab if they’re wanting to use the robotic arm for photography using the GoPro along with the tool-changing mechanism that Ryan McGovern is working on. Also, with the how-to guide on how to use the robotic arm completed, this will help other students a lot since there was no documentation at all on how to use the arm and the only way I was able to learn how to use it was to spend hours playing about with the code and asking questions. I feel like I’ve improved in my knowledge of python as well as how to use the robotic arm now that I’ve made the guide for it. In the coming week, I need to focus on how I’m going to remotely control the GoPro and any other cameras, as well as how I’m going to remotely control other portable cameras such as PiCameras, producing a much more affordable solution for smaller companies etc.

**Weekly To-Do – Week 4**

* ~~Acquire raspberry pi zero for remote control~~
* ~~Remotely control GoPro (possibly connect to raspberry pi or Arduino)~~
* ~~Implement remote control GoPro code into main system~~
* Read book about photogrammetry (Photogrammetric Computer Vision by Wolfgang Forstner)
* ~~Download / Install photogrammetry software~~
* Attempt photogrammetry with GoPro
* Review results of photogrammetry with GoPro
* Create / Find Turntable
* Create / Find Lighting System

*Monday 3rd February 2pm*

Today, I worked on the remote control of the GoPro for most of the day with the ssh technique that I had explained above. After a few hours of trying and doing large amounts of research on websites such as StackOverflow, (<https://raspberrypi.stackexchange.com/questions/66431/headless-pi-zero-ssh-access-over-usb?noredirect=1&lq=1>)I finally found a way to ssh into the raspberry pi zero where I could run a python script from the boot directory which was able to remotely control the GoPro. During the time I spent doing trial and error, I felt a lot of stress as it felt like a simple task before I started it, but when I actually tried it, it proved to be a lot more difficult. However, over the past 3 or 4 weeks I have learnt through this process to continue and persevere even when something isn’t working correctly, or I can’t find a solution for a problem. Now that I have this working, I feel a sense of relief as that’s an instrumental part of this project working, although it isn’t implemented yet. I feel like having this part of the project completed, it will be beneficial to others in the future having a remote-control GoPro set up in the IoT lab which can be controlled relatively easily, and I can make it easier to use in the future.

One thing that I’m worried about is that if the GoPro will be good enough / take good enough photos for the photogrammetry process. So, my next step is going to be to setting up / researching how to use photogrammetry software and attempting to do photogrammetry of a small object to see how it turns out and if it is low quality, I can try to use other solutions like a DSLR, however I’m concerned / worried it would both be too heavy for the robotic arm and would be more difficult to remotely control. If I can’t use a DSLR and have to stick with the GoPro, I may have to try and focus on post-processing of the digital recreation to make sure it does look better. Since this isn’t really as easy to do compared to making the robotic arm move better, this may be more beneficial to others than simply having the robotic arm moving since it has unlimited possibilities.

*Tuesday 4th February 3pm*

Today, I completed the implementation of the GoPro into the main system so that now it can be called from demos by creating an instance of the arm class and calling the connectGoPro() method with parameters of the filename on the Pi Zero and the number of photos for the GoPro to take. This can also now be called from the web interface. I currently still have to update the Pi currently connected to the *Arduino* in the IoT lab in the CSB to update the code that’s on it and ensure it has the correct packages installed to allow it to run correctly.

Today, I also added a few extra bits of detail to the how-to webpage so that it’s ready to be uploaded. After the meeting today, I was able to edit the webpage and diary to make sure that they’re up to standard and so that everything I’m writing about displays enough detail.

The meeting today was very helpful as it helped boost my moral since I feel more confident whenever John feels like I’m doing well in the project and making good progress because I sometimes find it difficult to tell whether I’m on track or not with the project I’m working on.

*Sunday 9th February*

Today, I started researching photogrammetry so that I can start trying to make a digital representation of an object using a GoPro handheld so I can tell if it’s good enough to perform photogrammetry. I downloaded AgiSoft PhotoScan Pro and PhotoShop. I need PhotoScan so that I can create the digital representations and PhotoShop so that I can create masks for the photos before I find a way to automatically do that. I started researching automatically applying masks last night too and I believe that I’ve found a way to do it.

I also purchased a lightbox so that I can ensure I have optimal lighting for the objects that I’m photographing have adequate light levels. I also found that I need to use a low ISO and low zoom level. I’m currently worried about how the photogrammetry will work and what is involved in it. I’m not sure where to go currently as well whether I should work on trying to improve the GoPro capturing as it currently takes a long time to take photos and download photos over WiFi. I do, however need to try and complete photogrammetry with the GoPro to see if it’s good enough. I installed photoshop and Metashape so that I can apply masks with photoshop and perform photogrammetry using Metashape. I’m going to try and do it with sample photos first by using these files (<https://www.agisoft.com/downloads/sample-data/>). I also am using this guide on how to perform the photogrammetry ([https://www.youtube.com/watch?v=9\_F-b2hxP\_o](https://www.youtube.com/watch?v=9_F-b2hxP_o&t=175s)) with Agisoft Metashape.

**Weekly Summary**

I didn’t get much work done this week because I’ve been sick so didn’t have much time to do work. However, with the work that I did get done this week, I feel relatively happy since I have the remote-control GoPro methods implemented into the main repository of the robotic arm code so that in demos you can take photos using the GoPro. After Tuesday, I’ll be able to know what exactly to focus on next, whether to continue with the photogrammetry route or to focus more on remote control of different portable cameras. I feel like I need to improve on the value of work I’m doing to try and set my goals to make more of a lasting impact and produce a deliverable at the end of the week so that students in the future or any other people that go onto my GitHub can improve on it and use it.

**Weekly To-Do – Week 5**

* Attempt other photogrammetry capture methods (other than GoPro)
* A screenshot of a cell phone

  Description automatically generated~~Implement GoPro remote control into web interface~~
  + ~~Take Photos~~
  + ~~Take Videos~~
  + ~~Turn on / off~~
  + ~~Download all photos / videos~~
  + ~~Change Resolution~~
* Implement Livestreaming of GoPro into web interface
* Add validation to GoPro interface

A screenshot of a cell phone

Description automatically generated

*Tuesday 11th February*

After the meeting today, I now feel more confident with the next steps for my project. After discussing my project with John, I’m now planning to focus on the remote control of different remote cameras for now since this has more lasting impact for other students and people in the future by creating web interfaces for different cameras like GoPros and PiCameras so they’re easier to use for anyone by just starting a web server. This could also create lasting impact for QUB by rigging cameras in the Makerspace etc which can be remotely controlled and viewed as an affordable solution to IP Cameras.

Today, I started researching how to create a web interface for remotely controlling the GoPro. After the meeting today, I’m planning to focus on the web interface this week to try and at least get the backend complete. I’ve never made a web interface before for a raspberry pi so I’m not too sure currently how I’m going to do it currently. I think that I can use CherryPy to do it like the web interface made to control the robotic arm, so I started trying to understand the files in the http\_api folder of the robotic arm. Although I looked briefly at CherryPy at the start of this project, I haven’t used it so far for any proper use.

*Wednesday 12th February*

After spending a good amount of time researching CherryPy yesterday by reading the documentation of CherryPy (<https://docs.cherrypy.org/en/latest/> ), and reading the robotic arm web interface code, I think I’m ready to start programming the web interface for my GoPro. I started by using CherryPy where it would just execute the python script, however this changed the URL, and so it reloaded the webpage which I don’t want to happen. So, I started researching again and found that I could use Ajax (<https://www.w3schools.com/xml/ajax_intro.asp>), which I had never used before in order to fix this problem. So, I started researching how to use ajax to run python scripts which took me quite a while. However, I am happy with my progress today as I got the CherryPy part implemented meaning it will be easy to implement all other features of the GoPro since I only have the take photos feature implemented so far too. Once I finish this feature, I feel that it will be very useful to others since I haven’t seen anything like this created before (specifically for a GoPro). I’m also currently slightly worried if the GoPro is going to be good enough or not to achieve photogrammetry due to its fisheye lens, however, after consulting one of my film-student friends, they said that they can show me how to disable the fisheye view. After the last couple days, after learning a lot of new content and doing a lot of stressful troubleshooting I’m quite tired. Tomorrow I hope that I’ll be able to complete the backend of the system.

*Thursday 13th February*

This morning, I started early so that I could try and get the backend finished earlier. I started to try and implement the Ajax part of the project which was quite challenging because I had never done any ajax before and was just using CherryPy documentation on Ajax and trying to replicate it for my situation which proved to be much more difficult than I initially thought. Whenever there was an error, it made the situation quite stressful because I had no idea how to fix the error without spending a lot of time on websites such as StackOverflow trying to find a solution and I found there wasn’t many solutions for CherryPy with Ajax on StackOverflow, or the internet in general which was very annoying since I was only going off about 2 different articles to try and find solutions. Finally, I got the take photos feature working with Ajax, however the way in which I had implemented it, I wasn’t going to be able to add any more features which felt very annoying after spending hours doing the initial implementation to only have to change the whole thing. After doing more research to find a more robust solution, I finally found a way to implement multiple POST / GET methods into Ajax which made me feel quite relieved. After trying to implement this, I kept getting the same error, ‘404 Not Found’. Due to me not knowing how to properly use Ajax with CherryPy, I was trying to call a different webpage instead of the CherryPy method. After doing more research once again, I found a different solution to CherryPy, called Flask (<https://opensource.com/article/18/4/flask>), which was mostly the same except it had a lot more solutions and documentation. So, I chose to switch to Flask and finally it worked and was able to both take photos and start recordings. After finally having this working, I felt very relieved because it felt like all the stress that had built up that day was finally away. Since it took longer than expected, I’m going to try and complete the backend over the weekend or early next week.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

*Saturday 14th February*

Today, I implemented many more features into the web interface such as changing settings of the GoPro, starting and stopping recording, turning the GoPro on and off etc. The frontend of the web interface still currently looks bad, but I can fix that easily early next week since it shouldn’t take too long at all. I can now start to write the documentation for this feature on how to use it and upload it to GitHub, which I need to learn how to use. However, I’m going to a workshop on Wednesday on how to use GitHub in the CSB which should help a lot.

I feel like this project will be very useful for others (using a raspberry pi as a web server, connected to a pi zero, connected to a GoPro) since I couldn’t find anything like this online before, so I’m very happy I can finally push a proper deliverable. I feel very relieved and accomplished because it’s the first time I’ve pushed a deliverable to a project that I actually enjoy working on.

A screenshot of a social media post

Description automatically generatedA screenshot of a cell phone

Description automatically generated

**Weekly Summary**

I’m extremely happy with my work that I’ve got completed this week having basically finished the web interface for the GoPro now which can be easily expanded for other cameras etc. Since it’s all open-source, other students in the future can improve on my web interface whether it be in terms of design or functionality. I’m very happy since I’ve improved my knowledge a lot this week now that I know how to make fully working web interfaces using Flask and Ajax, which are two things I had no clue how to do at the start of this week. I feel like I need to improve on the design aspect of my deliverables to make them more visually appealing to customers, especially since my target demographic are photographers and people with not a lot of technical knowledge that just want to remotely control their GoPro. Next week, I’m going to start to look into implementing PiCameras into the web interface as well as they’re a much more affordable solution compared to GoPros. I also need to add validation to the web interface so that the user cant’t access it if the GoPro / PiCam is not connected.

**Weekly To-Do – Week 6**

* ~~Implement PiCamera into web interface~~
* ~~Implement livestreaming into PiCamera interface~~
* ~~Add validation into both interfaces~~
* ~~Do frontend of web interface~~
* Implement livestreaming into GoPro interface

*Monday 17th February*

Today, I spent a long time implementing more features for the web interface as well as creating a design for how the frontend of the webpage should look by the time I start that part of the project tomorrow morning. I implemented features such as downloading all the photos to the device who is currently on the web interface, rather than downloading to the web interface as well as powering on/off the GoPro and changing the resolution, along with touching up a few bugs with the taking photos and videos. After tomorrow morning when I complete the frontend of the webpage, that should be the GoPro web interface nearly complete, however, I want to experiment with the livestreaming functionality of the GoPro’s API, which I need to look more into first of all to see what the latency and resolution is like of the livestream.

*Tuesday 18th February*

This morning, I spent a good amount of time working on the frontend of the system to make it appealing to the typical user (such as people working in photography) to ensure it looks clean, but at the same time has all the functionality needed. The website still currently has no validation in the inputs, however this shouldn’t be too much of a problem since I’m planning on creating a how-to guide for the gopro web interface anyway.

After I finish this part of the web interface, I’m going to do some research into the remote control of other portable cameras such as DSLRs and even very small cameras such as the Raspberry Pi Camera which could be used in large amounts of places since they’re so small and cheap. I’m slightly worried about this though since I don’t want to fall behind on the photogrammetry part of the project and focus TOO much on the remote control of cameras part, however it does mean that I would produce more deliverables which would be very useful to others in the future whether it be future QLab students or anyone in the world that could download it and easily set it up using my how-to guides. After the meeting today, I know to just continue doing what I’m doing which helps me feel confident with the progress I’ve been making since over the past couple weeks I’ve been trying to work harder so I can catch up for not doing much work three weeks previous. Also, John told me about the InventNI competition which I’m very excited about since he feels like the project I’ve been working on is good enough to put into the competition. This helps raise my confidence levels which will help me work harder since over the last week or so I had been worrying if I was doing enough work and creating enough deliverables which can be useful to students in the future.

**A screenshot of a social media post

Description automatically generated**

*Thursday 20th February*

Today, I started to do some validation for the inputs to ensure that they’re not blank and if they are blank to display an error message. This makes sure that any users who don’t enter an input for number of photos or seconds of video don’t crash the system if they enter an invalid input. This ensures that inexperienced users won’t crash the system. I also added validation to only allow the user to go onto the webpage if the GoPro or PiCam is connected by creating an error.html page which they get redirected to.

A screenshot of a cell phone

Description automatically generated

I also started to learn how to use the PiCamera, connected to the Pi Zero. I found that the PiCamera is much easier to use and learn than the API for the GoPro since it doesn’t use a third-party API like the GoPro. I’m going to let the user take photos and videos with the PiCamera as well as see the live feed of it which should be relatively easy to do. I learned how to use the PiCameras API using this documentation (<https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/4>).

*Sunday 23rd February*

Today, I started programming the code for the PiCamera and implemented it into the system where I ran into a number of problems such as it not connecting properly and mostly with the design of the web interface since the PiCamera’s live feed wouldn’t reposition properly, so I had to redesign the webpage which was quite annoying and time consuming since I’m not that great at CSS which is something I’m going to work on in the future.

After a few hours, I finally got the PiCamera’s web page complete, now I just need to wait for an extender cable to come to allow me to connect the PiCamera to the Pi Zero instead of just the Pi 3B since they have different ports.

**Weekly Summary**

I’m relatively happy with the work I got done this week. I finished the design of the website and added more functionality to the GoPro as well as added the PiCamera part to the website with validation for both parts. Overall, this week has made the web interface much more user friendly so that it can be available to more users in the future. I’m slightly frustrated that I didn’t get the GoPro live feed working, even after hours of research but I feel that the work I did get done instead will be much more valuable in the long term. I learnt a lot of new skills this week such as a refresher on how to use CSS since I hadn’t used it in a long time and how to do validation using Ajax and Flask. I’m currently feeling quite unsure of what I’m going to be moving onto next, but after the meeting on Tuesday I should feel much more confident with the upcoming part of my project now that I’ve got past this large hurdle.

**Weekly To-Do – Week 7**

* ~~Install XCode~~
* ~~Create basic application with Swift~~
* ~~Get app running on iPhone X~~
* GoPro livestreaming / live feed
* Create camera application with Swift

*Monday 24th February*

Today, I started to look at the livestreaming of the GoPro which is very stressful to do since there’s a lot of examples online with very little documentation, but they aren’t exactly what I need (implementing it into a web interface with flask). I spent a few hours researching this and trying out a number of different examples, but I didn’t get any properly working so far, so I’m going to leave it for now and come back to it at some point.

After chatting to John about the possibilities of my project today, it’s made me much more excited about the future of my project since the area I’m hoping to go into with turning iPhones into remote cameras since they’re much more affordable and have better technology than most good IP cameras.

I also did some more validation today by creating an error page to test if the Pi 3B is connected to the Pi Zero and if the Pi Zero is connected to the GoPro. This ensure that the system won’t break if users try to take photos etc. when the GoPro isn’t connected and makes the webpage much more user friendly, making it available for more users.

**A close up of a logo

Description automatically generated**

*Tuesday 25th February*

Today, I finished polishing off the web interface, fixing a few parts of CSS and changing some layouts on the webpage, before committing the latest version to my GitHub.

After my meeting with John today, I now know where to go next with my project, to focus on the IOS development. I spent a few hours researching whether it would be better to code the iOS apps in Objective-C or Swift, and I came to the conclusion that Swift would be better since it’s more modern and runs better than Objective-C, making it much easier to learn and code. Since it has easier syntax, it will be easier for other students in the future, who will be reading my code on GitHub to understand it and carry on from where I left off.

Once I knew I was going to use Swift, I downloaded and installed XCode, set up my developer Apple account and started learning Swift using online tutorials on tutorialspoint.com. Before this however, I started looking at the base code for creating a single-view iOS app and it started to worry me a bit because it seemed quite complicated and since in the last month I’ve learnt and worked with many different languages such as Python, Java, JavaScript, Ajax, HTML etc, I was worried that I’ll get mixed up between different languages if they have minor differences in their syntax.

*Wednesday 26th February / Thursday 27th February*

Over these two days, I continued learning Swift from variables, to different collections (arrays, dictionaries), to functions and new concepts which I found very confusing at first such as closures and having functions inside of other functions parameters which quite frustrated me since it was so confusing at first to understand.

Another thing that frustrated me during this period was that the syntax of Swift in general is a weird combination between Java and Python with a few other parts added in which seem unnecessary and pointless to be necessary syntax such as having to use an underscore instead of an iteration identifier such as ‘x’ if you don’t use ‘x’ in the for loops body. Another thing that has annoyed me with Swift is that whenever you initialize variables if you type ‘var score: Int= 0’, it will throw a compilation error since you didn’t put even whitespace on either side of the equals sign. All these and other things about Swift annoyed me and made me feel like I wanted to give up at times since it felt pointless to try and learn another whole new language since I didn’t know C++ or Objective-C either. However, I pushed through and slowly learnt all the basic and OOP concepts of the language. Thankfully, in comparison to a number of other languages and APIs that I’ve recently used, there has been a good amount of documentation for Swift which helped a lot.

In the coming days, I won’t be able to do as much since I have project work to do for other modules which annoys me since it feels like I haven’t done enough for this project / module this week since all I’ve done is learn a new language and not actually made any deliverables.

Over the weekend, I’m going to try and create a tester IOS app to put my new skills of the language to the test so that I can hopefully have something completed by the end of the week to help me boost my spirits a bit. I’m still slightly worried about the scope of the project and what the product is specifically going to be used for, however through market research I’ll be able to work out where my product would work best.

I used this Swift documentation to help me learn the language(<https://swift.org/documentation>). Although it’s not very user friendly and not good for beginners, I found it easy enough to understand by playing around with each of the concepts that it explains in very little detail.

*Friday 28th February*

Today, I spent a few hours trying to go through the base IOS app template and understand what each file of it does, since there are about 4 or 5 automatically generated files when you select you want to create a Single-View app in XCode. This took me about 2 hours of watching different videos on YouTube and reading many different articles, until I finally understood how to actually use all the functions of XCode and understand the automatically generated code in the files.

Some of the articles / YouTube videos I used:

* <https://www.youtube.com/watch?v=09TeUXjzpKs&t=9821s>
* <https://www.youtube.com/watch?v=aiXvvL1wNUc>
* <https://www.youtube.com/watch?v=Ulp1Kimblg0>
* <https://www.codecademy.com/learn/learn-swift>

Once I had this done, I started to try and add some small functionality into the app such as buttons and switches which only changed the text on a label, but I feel like I’m slowly making progress into learning how to create IOS apps with Swift. Since I have programmed GUI applications with C# before, this helped somewhat in terms of programming GUI applications and how to access the variables of each of the elements on the app.

I’m still quite worried I didn’t make enough progress this week since I had a lot of work to do for other projects and other deadlines to meet, so I need to try and work on managing my time better so that I can allocate even time to each of my projects for each of my modules.

A screenshot of a cell phone

Description automatically generatedNext week I’ll be able focus on this project much more and I hope to start to figure out how to use ARKit and access things such as the Camera which I’ll need for this. I’m also slightly worried about the current scope of this part of the project since I’m not 100% sure on what the app should exactly do and what limit I should go to in terms of the features / functionality of it. I’m also slightly confused about where and how I’ll use ARKit in the app, however after Tuesday when I have my meeting, I should be able to surpass these problems.

*Sunday 1st March*

This evening, I spent a good few hours using the playground part of XCode to try different functions and closures and try to get a grip on using error handling in Swiftto help me feel more comfortable using it so that tomorrow I can start trying to make proper projects with it until I feel comfortable enough using it to work on my actual project. I still feel quite confused about some functions and especially closures and generalizations, however if I continue to work on these using the Playground feature in XCode, I’ll finally feel confident enough using them to create a fully functional app in the coming days.

A screenshot of a cell phone

Description automatically generated

When using the playground, I had a number of problems such as it took a very long time to execute, by doing some research, I found that to fix it, I had to change the Playground settings platform from iOS to macOS here.

Also, when attempting to get the default app template to run on an iPhone X, it wouldn’t run because the iOS versions weren’t the same. I noticed that the deployment target was IOS 13.4 and the iPhone was running 13.3 even though it showed as up to date. So, to fix this, I went into the settings of the deployment device on XCode and changed the Deployment Target from 13.4 to 13.3 and that fixed the issue.

**Weekly Review**

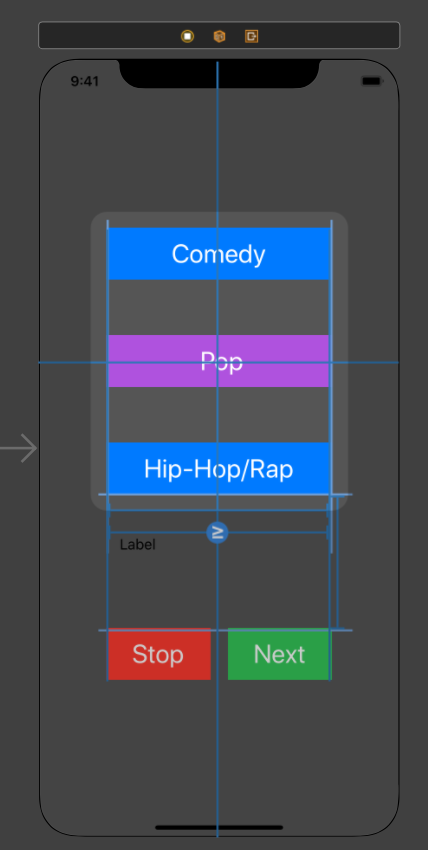
I’m pretty happy with the work that I’ve got done this week since I have a better idea of my projects direction and I’m more excited about this part of the project than the photogrammetry part of the project since I find app development and using AR, AI and machine learning / data analysis more interesting. I feel I didn’t do a lot this week since I just spent a lot of time trying to learn Swift and how to use X-Code since it was quite a difficult language to learn because it feels like a strange mixture between two languages I already know; Java and Python, so I found it very challenging at times. After the meeting on Tuesday, I’m hoping to have a crystal clear idea on what the project will entail and what technologies / APIs I should use to achieve the end goal.

**Weekly To-Do – Week 8**

* ~~Research ARKIT~~
* Creating Working ARKit App for 2D
* ~~Create Working ARKit App for 3D~~
* ~~Bounding Boxes around Objects~~

*Monday 2nd March*

This morning, I spent a lot of time creating a working music-player application using Swift on X-Code and emulating it with X-Code. After doing this project, I feel much more confident with Swift and feel like I am going to be able to complete this upcoming part of my project of creating an iOS app to complete data analysis using the footage from an iPhone Camera. I’m still currently unsure of the complete scope of the project since I’m not sure on how APIs like ARKit are going to be used in it. To help me work on the project, I’m planning to create a requirements document so I know exactly what I have to create to reach my end goal, however this can be done using the Planning document. I feel quite confident working with XCode and Swift now to create iOS apps but there’s still some things I feel very unsure of such as how to use ARKit and other APIs as well as how to use the camera on the iPhone. I’m also still not good at the design aspect of the apps using constraints etc on the storyboard view of the app:

A screenshot of a cell phone

Description automatically generated

To create this music-player app, I just followed this tutorial on YouTube: (<https://www.youtube.com/watch?v=aiXvvL1wNUc>)

The app used the iTunes API to pick a song genre and play songs from the genre and then the stop button to stop the music and the next button to go the next song.

*Tuesday 3rd March*

Today, I continued working with Swift to try and get a hang of it so that I could hopefully feel confident enough soon to develop the ARKit app. I started looking into accessing the camera using Swift and how I would take photos / record videos with the iPhone.

I spend a couple hours watching and reading different tutorials and attempting different methods and APIs on how to best do it. After experimenting, I eventually found one that worked well and was the most configurable. When I have been trying to experiment and learn Swift over the past few days, it has been quite stressful because whenever I don’t know why somethings wrong, when using a new IDE and when I don’t know why something’s not running correctly, it’s challenging to go through the code and debug it like I can with Java since I have much more experience with Java. However, I’m slowly starting to feel much more confident with using it and how to debug issues with it.

*Wednesday 4th March*

Today, I spent most of my day working on my blog post and fixing up some other documents such as my how-to guides. However, I did also make progress on my project; I started to look into integrating ARKit into an app by reading the ARKit documentation as well as watching tutorials on how to use ARKit. I downloaded the ARKit Scanner app on the iPhone X which allowed me to create a scan file of any object using a very user-friendly app. I then watched YouTube tutorials on how to create a camera app on how to use this scan to detect it any show a description of the item scanned. However, after a good amount of time debugging it, it wasn’t picking up the object. I think that the problem is because the object wasn’t clear enough against the background, so I’m going to try again tomorrow or try with 2D objects and do a 2D scan such as a playing card.

*Thursday 5th March*

Today, I continued experimenting with ARKit by adding multiple objects and scans. I added these objects and tried to tets it with the iPhone app and the result was that it became very slow. I also fixed the AR SpriteKit labelling so that it was closer to the bounding box.

I’m unsure on how to improve the speed and performance of the scanning of objects since it currents takes around 2-5 seconds to scan an object currently and is very dependent on the lighting conditions. I also started to look into how to draw a bounding box around the objects whenever they’re detected which I found relatively easy.

**Weekly Review**

Overall, I feel happy with the work that I accomplished this week, with starting to understand how to use ARKit, however I still need to keep using it until I’m confident with it because currently I’m mostly following YouTube tutorials and going through the code afterwards, trying to understand what each part does / means. I didn’t get a lot done this week compared to other weeks because I had to spend a couple days trying to improve the documents such as my planning to make sure I have a clear overview of the scope of the project and to ensure I can make the project have as much lasting impact as possible.

**Weekly To-Do – Week 9**

* ~~Add downloading to server from PiCam / GoPro~~
* ~~Add downloading from server to client on web interface of images / videos~~
* ~~Learn web framework for iOS Swift’~~
* ~~Create Web Server on iOS~~
* ~~Add camera view to web server app~~
* ~~Get hardware device information from iPhone using Swift~~
* ~~Display hardware device information as JSON on web interface~~

*Monday 9th March*

I started off today by setting up the Raspberry Pi GoPro / PiCam web server again and starting to look into using SCP to successfully download the images / videos from the cameras, to the web server and eventually to the client.

I spent a couple hours trying to troubleshoot how to use SCP with SSHPass since I kept getting errors with very little debugging / error information provided. I found this quite frustrating at the time because I was exactly following an online guide on how to use it and it wasn’t working. Finally, after I got it working, I implemented this into the web server code, which once again, wasn’t working as expected for some reason since when I clicked the ‘Download’ button, it would only sometimes trigger the http GET event randomly.

*Tuesday 10th March*

Today, I was working on using Flask to send files to the client on the web server. I started this morning off by fixing the download button so now that it works correctly by sending GET requests and currently downloads the images to a folder on the web server. I fixed this by switching to a better internet connection other than the Makerspace WiFi. This is what was making it process all the requests so slowly as well. However, I still find it to be quite inconsistent at times, sometimes the download button will work, sometimes it won’t and sometimes the download button will work and download the files to the webserver, but not send them to the client, so in the future I will need to do some more testing on this.

This has frustrated me a bit since I don’t know why it wasn’t working.

*Wednesday* 11th March

Today, I started researching the web framework for Swift that I could use, like Flask, to send HTTP requests. I wanted to find a very lightweight library so it wouldn’t take up many resources on the iPhone at all to run it. After some googling, I found a web server called Swifter which seemed well documented compared to the rest of them and easy to use. It took me an hour or two to install this because I was very unfamiliar with how to install packages with either Swift Package Manager or Carthage, and when I did install it with Carthage, it didn’t detect when I tried to import it and caused a runtime error when I ran it, even without the import statement or any code from Swifter. This was very frustrating because it seemed like such a simple task, but it didn’t even work!

After getting this working, I learnt how to use it by reading through its source code and its very minimal documentation so that I could understand how to use it before creating a simple web server and playing about with it to test all of its functionality to ensure it would be suitable for the project, and it seems like it will be.

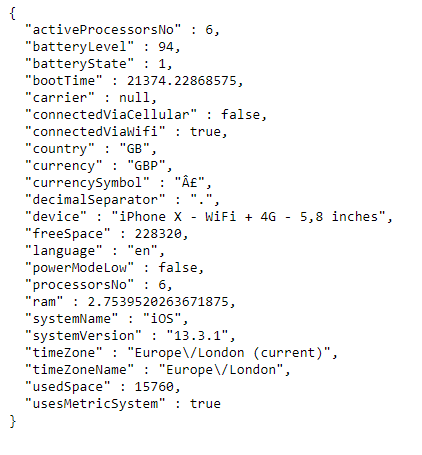
Today, I also added a basic camera view to the Swift app above the *Stop Server* button so that I may use in the future to take pictures from to send them to the main web server.

*Thursday 12th March*

Today I started looking into getting hardware device information from the iPhone and saving this information in JSON. After searching around google for about an hour, I came across a library called **Luminous** which had all of the functions I would need to get diagnostic information regarding the camera. This was easy enough to implement and I made variables for all of the variables I thought would be useful for debugging purposes. I then started to look into transforming these into a JSON file and display it on the iPhone’s web server. This was easy enough to do by reading up on some different tutorials on how to use the JSONSerialization library. After this, it was extremely easy to display it on the iPhone’s web server using Flask since I had a good amount of experience with it before.

**Weekly Review**

I’m very happy with the work I got done this week by having the GOPro and PiCam web interfaces polished off as well as getting the device hardware information from the iPhone and sending it to the web server.



**Weekly To-Do**

* Take pictures with iPhone and display on web server
* ~~Start central web server for iPhone to asynchronously update json data / image~~
* ~~Polish GoPro / PiCam Webservers~~

*Wednesday 18th March*

Today I started by looking into taking pictures using the iPhone, which I had to re-implement the camera part of my current web server with. I tried many different methods with this, when implementing Luminous (JSON Data bit), I saw that there was a function to take a snapshot of the current view, so I tried for a few hours to use this instead of taking a photo using the camera, however this didn’t work and was just outputting a white screen. This was very frustrating and unmotivating after working on that part of the project for the last few hours and not achieving anything.

So, I started looking into alternative methods by just taking an image using the camera, I had previously implemented parts of a camera app to show the live feed, but couldn’t find the guide again to complete it so I had to restart that part of the app. This only took about an hour, and after this I got an image downloading to the document’s directory and the camera roll (for testing purposes).

*Thursday 19th March*

Today I started to research how to investigate how to create the async web server for the iPhone, GoPro and PiCam so I can asynchronously update the json data and the images for each device. I’m felt very unsure about how to do this, which worried me, however when I started researching, I found a library called flask\_socketio which was made by a popular developer on GitHub, so I used it since it was well documented. Thankfully there was a good amount of documentation for this library and I spent today testing out the library and trying to figure out how to properly use it.

**Weekly Summary**

Overall, I’m unhappy with the work I got done this week. I felt very unmotivated and out of pattern since I was at home all day, where I found I could only work well when at the QUB campus, so I didn’t get nearly as much work as I wanted to get done, completed this week. However, next week I’m going to try to catch up on the work I didn’t get done this week, for next week.

**Weekly To-Do**

* ~~Take pictures with iPhone and display on web server~~
* ~~Asynchronously update webpage for iPhone images~~
* ~~Fix downloading of GoPro and PiCam web servers~~
* ~~Asynchronously update webpage for GoPro / PiCam images~~
* Comment Code

*Monday 23rd March*

Today, I started to implement the taking of pictures with the iPhone and displaying them on the web server. I had several issues with this. First, I started to try doing it with converting the image to base64 however this didn’t work because the image was much too high of a resolution (1080p) for base64. So, I had to research other methods and tested other methods throughout the day.

After spending most of the day trying to research the best solution to use for displaying the image of a web server, I finalized on having to download the image to a folder using the requests python API, displaying it on the web server, and delete the file just before another image downloads to the folder.

This has frustrated me quite a lot because I wanted to find a faster solution for finishing this project off for now, however I felt like my time would be better spent finding a better solution until I finish the other parts of the project.

One thing I’m unsure of right now is how to have multiple connections with flask\_socketio, since I currently have the iPhone data connection running. From reading the documentation I can’t find out clearly how to do this, however I’m hoping it’ll be relatively self-explanatory.

*Tuesday 24th March*

Today, I started to implement the asynchronous image updating from the iPhone onto the web server. I had a good idea of how I was going to do this because it seemed relatively self-explanatory since I had done it once before for the data of the iPhone, except this time with an image instead of text. After playing about with the flask\_socketio code for a while in a tester class, I found out how to have multiple connections, by using the *force\_new\_connection=True* parameter when initialising the connection / socket in the Javascript file. After basically creating a carbon-copy of the code I already had for the iPhone’s data, except changing a few variable names, it was working well, and the image was updating approximately every 2.5 seconds.

I could potentially improve the speed of this slightly by using a faster internet connection or connecting the raspberry pi to ethernet instead of WiFi so it would transfer the image faster. I would need to investigate this for the future anyway for cyber-security reasons. Now that I have this working, implementing the other two solutions for the GoPro and PiCam should be easy enough.

I have still been finding it somewhat difficult to adapt to a new working environment, working from home instead of at QUB due to the COVID-19 pandemic since I find it much harder to focus at home with more distractions around me.

*Thursday 26th March*

Today I started off by looking at the code for the GoPro and PiCam web interfaces to change the infrastructure slightly by adding a new route to display the image that has just been downloaded from the Pi Zero to the Pi 3B on the web server so that the requests API can access this and download it to the central web server, to eventually display it on the central web server. This took longer than I initially expected because for both the GoPro and PiCam web servers, I had to add new methods and fix the file structure slightly since I felt it was quite messy so that a different user could more easily go onto the GitHub repository and implement this system themselves. I then had to find a way to easily display this image on the web server where I could maintain its resolution / quality, while still uploading / downloading quickly.

*Friday 27th March*

Today I finished off the central web server by once again, implementing the GoPro and PiCam images from the newly created app routes for each of the web servers by using the python requests API in combination with flask\_socketio as before with the iPhone image and data requests. These also update about every 2.5 seconds. I finished off by touching up the layout of the webpage by turning it into a 3-column webpage to separate the iPhone, GoPro and PiCam data. This came with no real challenges at the end since I was confident with how to use flask\_socketio by today.

**Weekly Summary**

Overall, I’m happy with how it turned out since going into it I had no idea how to start this central web server or what technologies to use. However, in the end, I found that it wasn’t too difficult at all once I learned how to use the flask\_socketio API.

**Weekly To-Do**

* ~~Comment Code~~
* ~~Ensure GitHub is Up to Date~~
* ~~Start Testing Process~~
* ~~Fix CSS of PiCam Web Interface~~

*Monday 30th March*

Today, I spent most of the day polishing everything up from commenting all my code, ensuring my GitHub repository was up to date and fixing the CSS of my PiCam web interface since beforehand it was a bit messy and didn’t look great. I’m hoping that in this coming week that I get some of my work ethic back whenever I start to do the testing process, but I don’t feel that excited to do the testing which makes it harder to start doing.

*Wednesday 1st April*

Today I started to do the testing for this project. I spent most of today doing research for the testing by gathering links and sources which could help with my testing such as the different manufacturing sectors and how much IoT technology can help the value of the manufacturing industry in the next 10-20 years. This took longer than expected because it was hard to find relative information for Northern Ireland and a clear overview of the sectors in NI / the UK.

*Thursday 2nd April -> Saturday 4th April*

Over these past few days I’ve been iterating on the testing of my project. I didn’t feel the need to create a separate entry for each day because I don’t feel that there’s enough to talk about each day since all I’ve been doing is adding to the testing phase. I started by looking into the different environments that the product could be used in and then looked at different ‘what-ifs’ and different scenarios such as how to power the cameras, and what if the cameras stopped working / how to tell they’re not working as well as less technological feasibility such as ethical feasibility.

Thanks to the daily meetings with John which I have been finding incredibly helpful, I’m able to know what I need to work on with my testing section since I find it a bit confusing sometimes to know what to talk about.

*Monday 5th April*