The Process

First deciding what project to work on

We started with a really outgoing optimistic idea where we were going to connect an oculus rift and the leap motion to a quad copter.

This was changed to something more reasonable, where we found out about a much smaller drone the crazyflie, however the crazyflie turned out to be very touchy and unstable, as we tried to understand the crazyflie we found that a much more realistic project would be to get the leap motion and the crazyflie to communicate and eventually use the leap motion to control the actions of the crazyflie.

Python was the recommended developer language for the crazyflie however after coming across many issues when trying to install and setup the tools needed on windows there were too many issues and we all ran into many different problems. We had many difficulties when trying to get it to work with python.

Problems with Python – communications and using different versions across platforms and installing dependencies, versions 2 and 3 seemed to work on the virtual machine.

(Ryan our dear friend collapsed and was taken to hospital this upset us greatly and put our work back by a few weeks as we took time to come to terms with his illness).

When trying on the Mac we had more luck getting code to work.

However we looked for options where we would all be able to edit and test the code and the overall project, we then looked into JavaScript and managed to get this working

We managed to get basic version running in JavaScript this allowed us start the crazyfile using the leap motion to get the crazyflie to start flying. We did this with a sharp movement over the leap motion and raised our hand, directing the crazyflie to lift off and hover. However this was unstable and unreliable as it would stick at starting telemetry, upon investigation we found that this was due to the loss of packets (loss of packets should have been dealt with in the one of the JavaScript libraries however it doesn’t look like it was) as the crazyflie and crazyradio communicated. This was also affected by how close the crazyflie was and the number of electronical devices/ interference in the area.

We also looked into virtual machines.

Problems

* Difficulty enabling AMD virtualization with the MSI 970 Gaming motherboard for it is labelled as SVM Mode
* To run x68bit virtual machines there was a cash of Windows Hyper-V causing virtual box to only allow x86 OS’s
* We then had issues with Python 3.4 running the example code
* We then installed the provided VM by bitcraze
* We then had to install all the other python dependencies that caused some issues
* We got the VM to pick up the leap motion but the data communication of the hand positions was extremely slow and was not very good to use to control the crazyflie

… (so terrible we can’t even speak of it)

We are now using Cylon to get data from the leap motion and then send commands to the drone (AR.DRONE built by Parrot)

The firmware for the drone is proprietary but an SDK is provided to create clients to control the drone.

Drone Controls

* To take off, a clockwise gesture with your index finger (x2, twice)
* To land, an anticlockwise gesture with your index finer (Once)
* To fly up, hand raises
* To fly down, hand lowers
* To fly right, hand tilts right
* To fly left, hand tilts left

We did find there was an issue when we removed our hand from over the leap motion or when we made a fist and it couldn’t record the fingers. When this happened the droid would be send the previous command and continue in whatever direction it had been left in, we fixed this by editing the code to stop the droid and make it hover when it can’t detect a hand or fingers for commands.

This prevents the Droid from flying off when we remove our hand from over the leap motion.

The Droid also has issues picking up the left hand and seems to only count the right hand, if you try to use your left hand to control the droid it does detect the hand and its gestures however they are inverted and it won’t move the droid in the same way when you use your right hand.

Looking at the spec we realised we may not get excellent and as we are aiming for the highest possible grade we started looking into other technologies we might be able to add to the droid and leapmotion we have so far.

Current Technology

* JavaScript with node
* Leap motion
* AR.DRONE (built by Parrot)

Looking into

Google Goggles API we could use

Google Cardboard

Open CV facial detection – possible to program an action where a key tap gesture over the leap motion could trigger the drone to take a picture with the on board camera then using OPEN CV to detect any faces.