

TEC-V

MILESTONE 3

By: Michael Dowling & Zealand Brennan



CLIENT

- DR. Wood
 - **Professor** | Ocean Engineering and Marine Sciences
 - **Program Chair for Ocean Engineering**



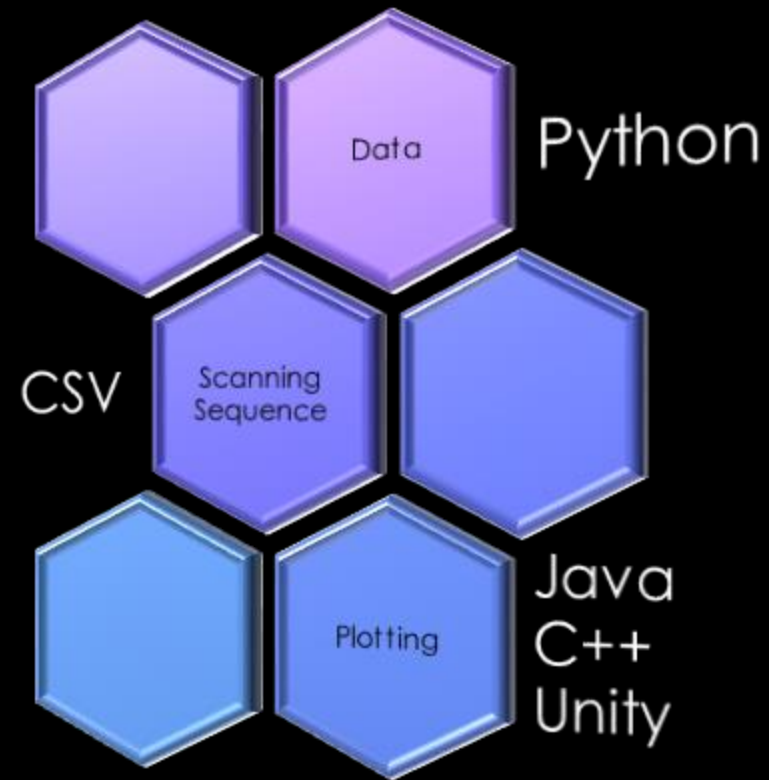


MILESTONE 2 OVERVIEW

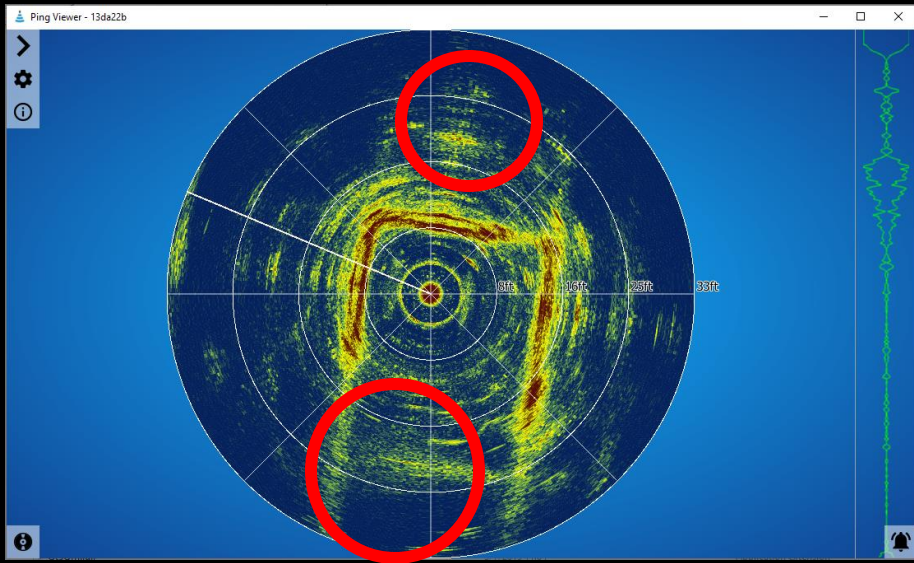
- ❖ False Data
- ❖ Depth Finder
- ❖ Compass and Telemetry
- ❖ Cloud Plot Application

TOOLS

- Data: Python
 - Sonar
 - Telemetry
- Plotting: Unity / C++



FALSE DATA



FALSE DATA

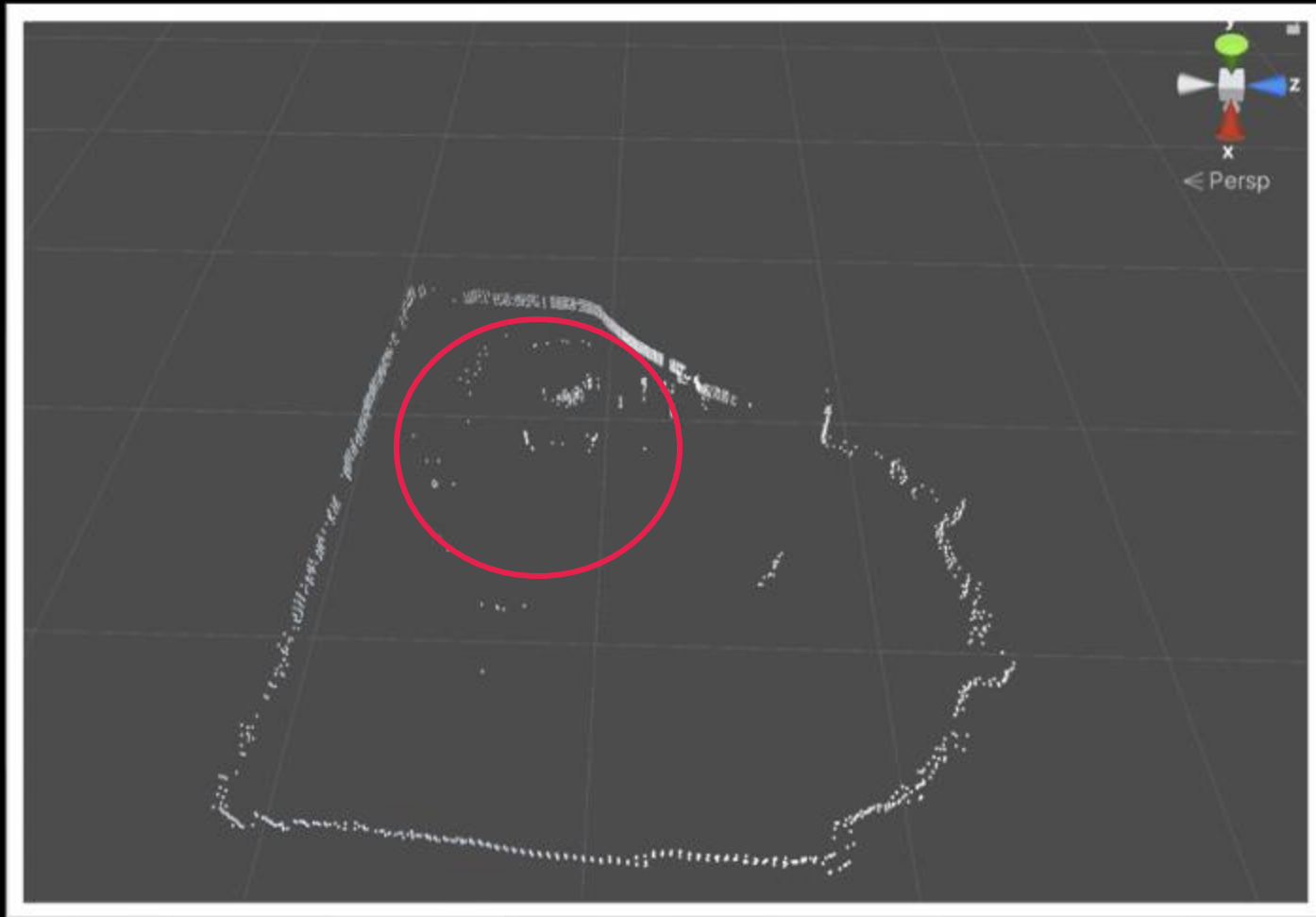
```
// Check if the current data and the data before and after it are within 1 meter range
if (i > 0 && i < dataLines.Count - 1)
{
    string priorLine = dataLines[i - 1];
    string nextLine = dataLines[i + 1];

    string[] priorData = priorLine.Split(',');
    string[] nextData = nextLine.Split(',');

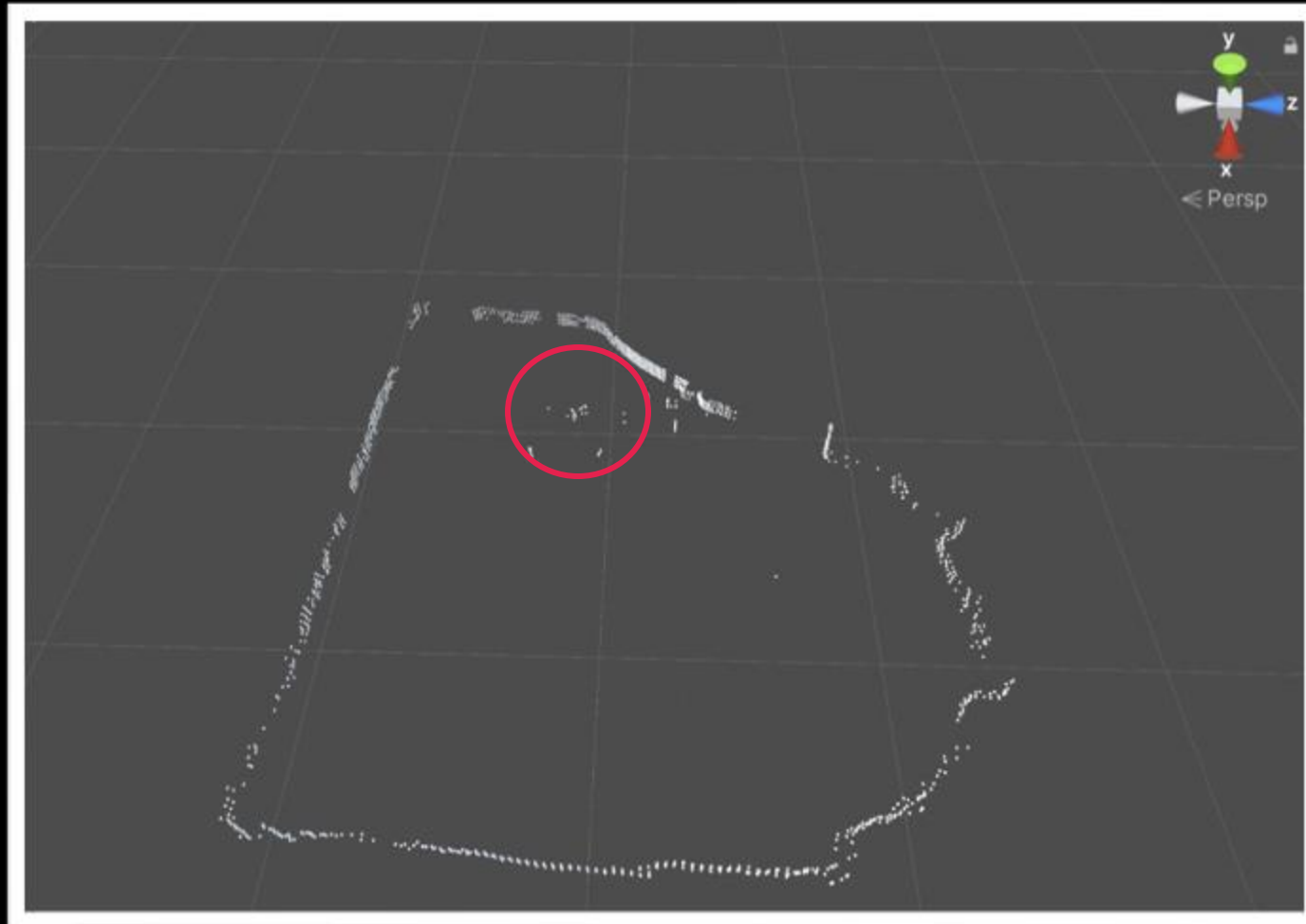
    float currentThirdValue = float.Parse(currentData[2]);
    float priorThirdValue = float.Parse(priorData[2]);
    float nextThirdValue = float.Parse(nextData[2]);

    if (Mathf.Abs(currentThirdValue - priorThirdValue) <= 1f && Mathf.Abs(currentThirdValue - nextThirdValue) <= 1f)
    {
        // Data is within 1 meter range, instantiate sphere
        InstantiateSphere(currentData, scale);
    }
}
```

FALSE DATA



FALSE DATA





TELEMETRY

GETTING TELEMETRY

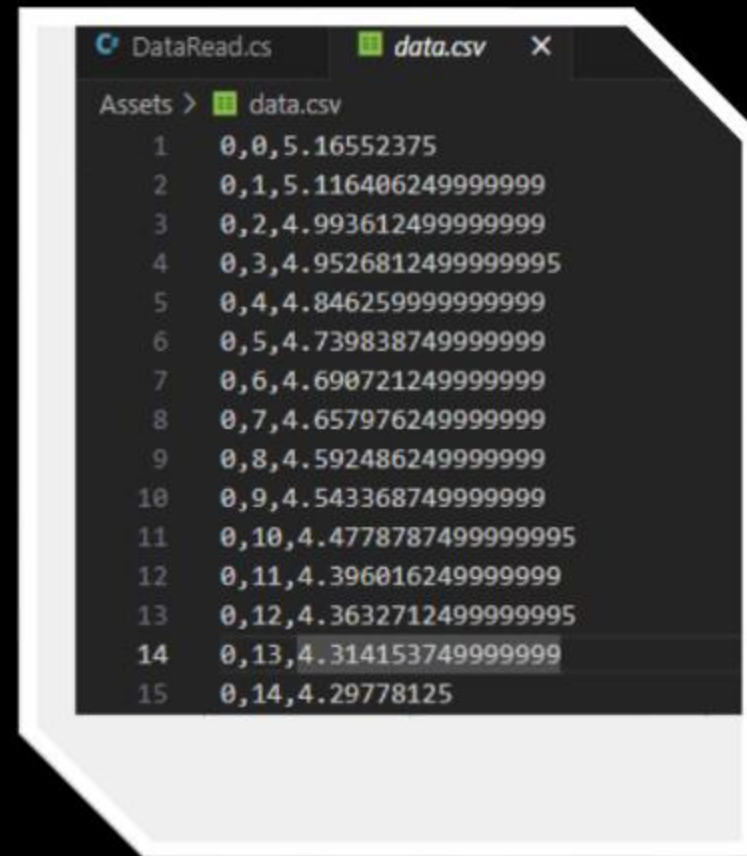
Main Components

- Depth
- Compass Heading
- Roll
- Pitch
- Yaw

ORIGINAL FORMAT

Data.csv

- Three categories
 - Depth (in progress)
 - Angle
 - Most likely distance to object



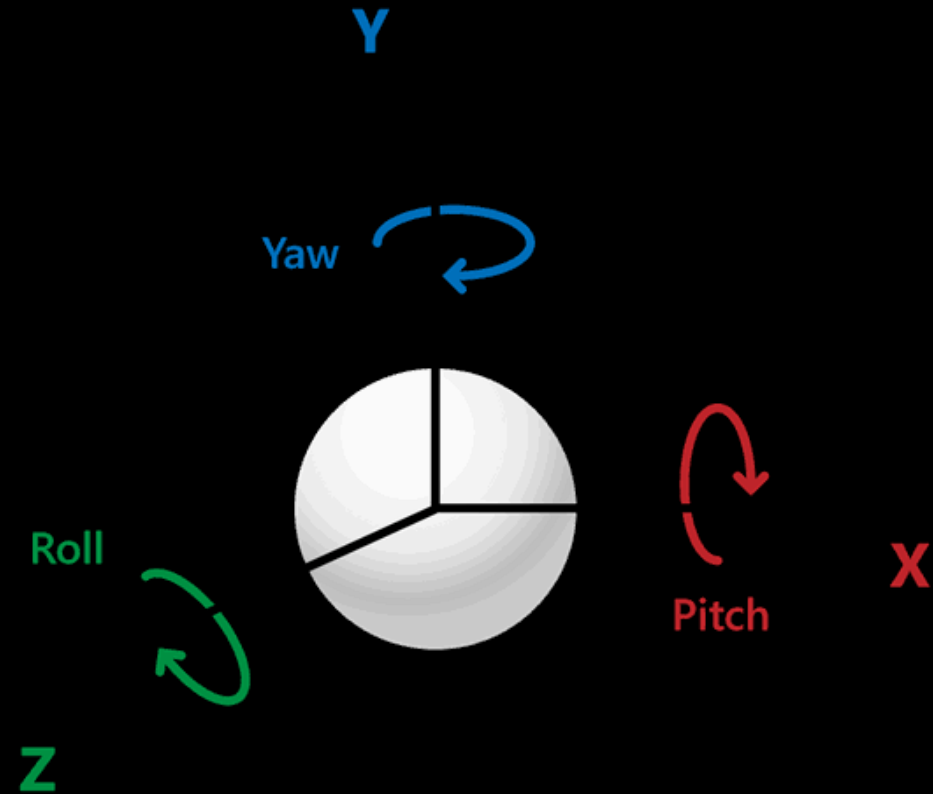
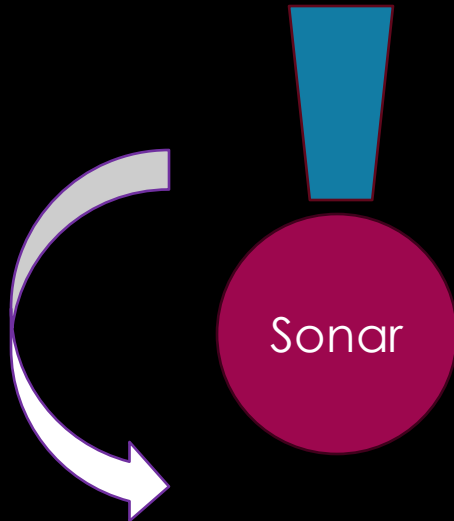
The screenshot shows a code editor window with two tabs: 'DataRead.cs' and 'data.csv'. The 'data.csv' tab is active, displaying a list of 15 rows of data. Each row consists of a line number followed by a comma-separated list of three values. The values are: 0,0,5.16552375; 0,1,5.116406249999999; 0,2,4.993612499999999; 0,3,4.9526812499999995; 0,4,4.846259999999999; 0,5,4.739838749999999; 0,6,4.690721249999999; 0,7,4.657976249999999; 0,8,4.592486249999999; 0,9,4.543368749999999; 0,10,4.4778787499999995; 0,11,4.396016249999999; 0,12,4.3632712499999995; 0,13,4.314153749999999; 0,14,4.29778125. The 14th row is highlighted.

1	0,0,5.16552375		
2	0,1,5.116406249999999		
3	0,2,4.993612499999999		
4	0,3,4.9526812499999995		
5	0,4,4.846259999999999		
6	0,5,4.739838749999999		
7	0,6,4.690721249999999		
8	0,7,4.657976249999999		
9	0,8,4.592486249999999		
10	0,9,4.543368749999999		
11	0,10,4.4778787499999995		
12	0,11,4.396016249999999		
13	0,12,4.3632712499999995		
14	0,13,4.314153749999999		
15	0,14,4.29778125		

WHY TELEMETRY DATA?

Rotation

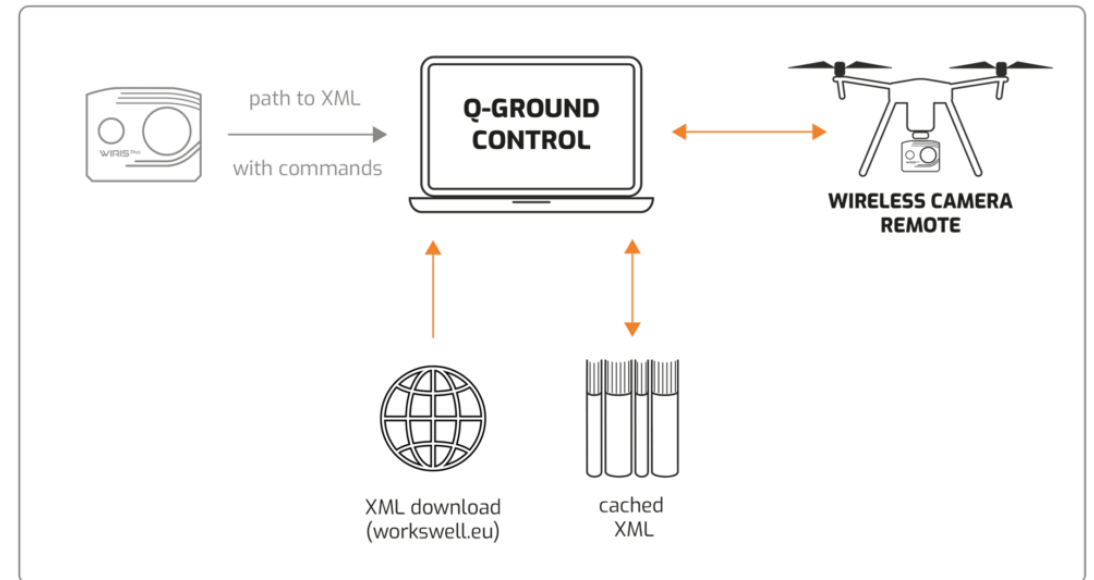
- During sonar scan:
 - If (rotate occurs)
 - Where am I looking?



GETTING TELEMETRY

Attempt Number 1

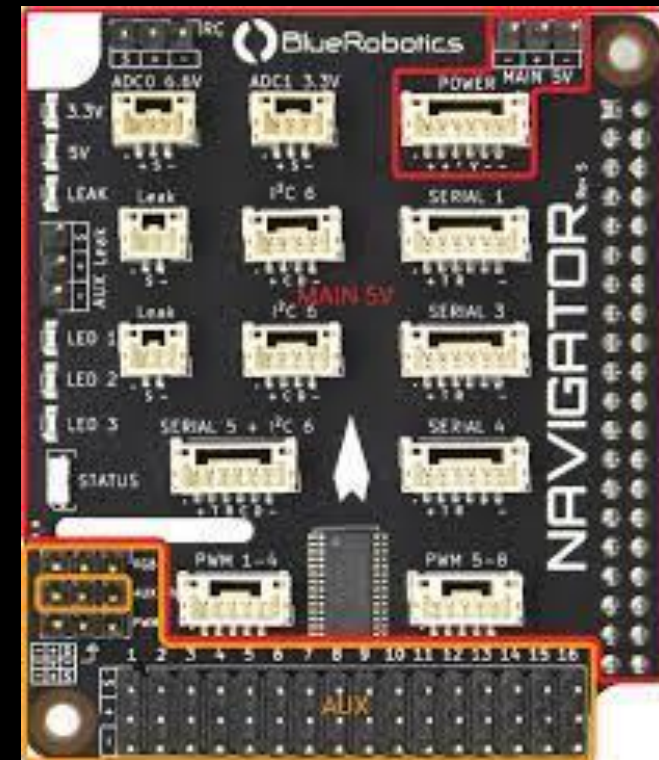
- Using the source “Mavlink”
 - Allows the user to take information directly from the flight control software on topside receiver.



GETTING TELEMETRY

Attempt Number 2

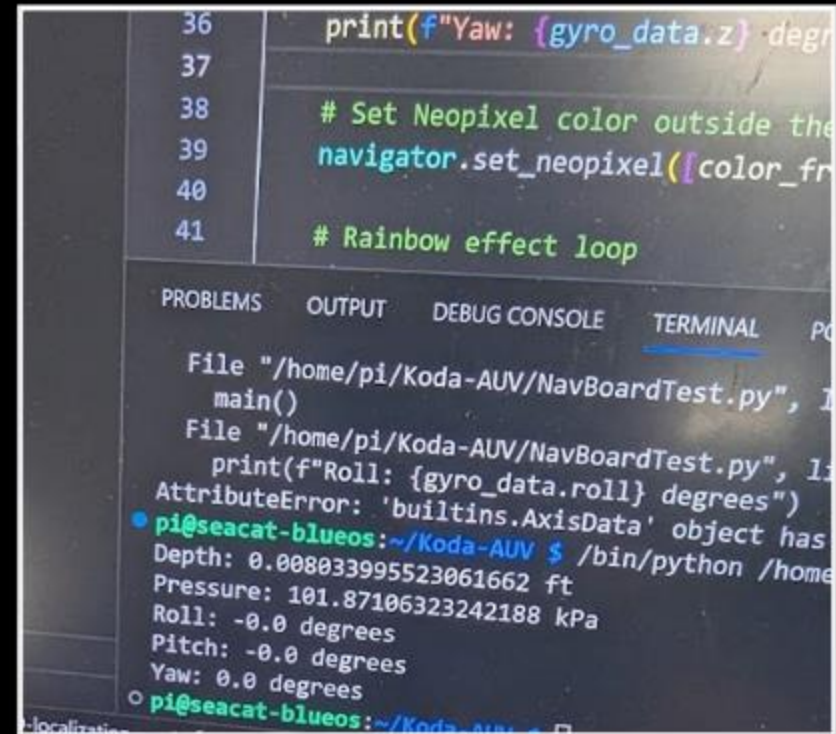
- Blue Robotics Nav Board
 - Get information directly
 - Real time
 - Can save data to same csv file as sonar.



GETTING TELEMETRY

Attempt Number 2

- Blue Robotics Nav Board
 - Pros:
 - Real time
 - Cons:
 - Creates huge delay in topside information



The image shows a screenshot of a code editor and a terminal window. The code editor displays Python code for a navigation board test. The terminal window shows the output of the code, including a rainbow effect loop and a print statement for Yaw.

```
36 print(f"Yaw: {gyro_data.z} degr
37
38 # Set Neopixel color outside the
39 navigator.set_neopixel([color_fr
40
41 # Rainbow effect loop
```

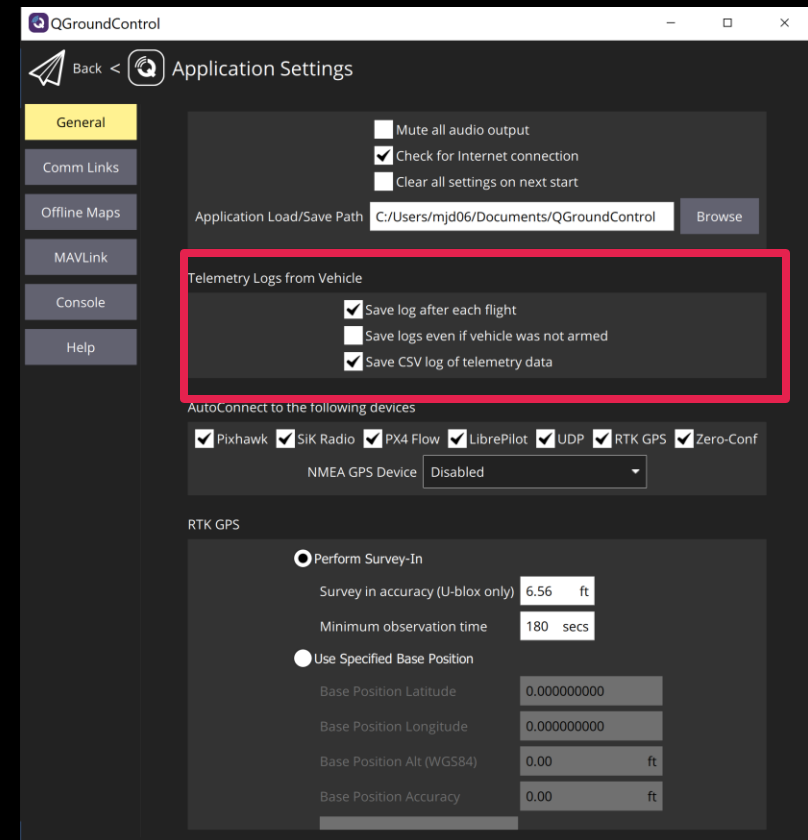
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PC

```
File "/home/pi/Koda-AUV/NavBoardTest.py", l
main()
File "/home/pi/Koda-AUV/NavBoardTest.py", l
print(f"Roll: {gyro_data.roll} degrees")
AttributeError: 'builtins.AxisData' object has
pi@seacat-blueos:~/Koda-AUV $ /bin/python /home
Depth: 0.008033995523061662 ft
Pressure: 101.87106323242188 kPa
Roll: -0.0 degrees
Pitch: -0.0 degrees
Yaw: 0.0 degrees
pi@seacat-blueos:~/Koda-AUV $
```

GETTING TELEMETRY

Attempt Number 3

- Q-Ground: (Flight control app)
 - Saves telemetry data during flight



GETTING TELEMETRY

DATA11.CSV

data.csv		data11.csv ×		merged_data.csv	
data11.csv					
1	Depth,Angle,Distance,Timestamp				
2	0,0,2.9375085937499996,2023-11-20 15:42:44.905183				
3	0,1,2.9375085937499996,2023-11-20 15:42:44.906065				
4	0,2,2.9334707812499996,2023-11-20 15:42:44.906411				
5	0,3,2.9334707812499996,2023-11-20 15:42:44.906672				
6	0,4,2.9334707812499996,2023-11-20 15:42:44.906959				
7	0,5,2.9334707812499996,2023-11-20 15:42:44.907263				
8	0,6,2.9294329687499996,2023-11-20 15:42:44.907493				
9	0,7,2.9334707812499996,2023-11-20 15:42:44.907700				
10	0,8,2.9334707812499996,2023-11-20 15:42:44.907898				
11	0,9,2.9334707812499996,2023-11-20 15:42:44.908126				
12	0,10,2.9294329687499996,2023-11-20 15:42:44.908343				
13	0,11,2.9294329687499996,2023-11-20 15:42:44.908528				
14	0,12,2.9294329687499996,2023-11-20 15:42:44.908772				
15	0,13,2.9375085937499996,2023-11-20 15:42:44.909017				
16	0,14,2.9455842187499996,2023-11-20 15:42:44.909225				
17	0,15,2.9496220312499997,2023-11-20 15:42:44.909485				
18	0,16,2.9496220312499997,2023-11-20 15:42:44.909758				
19	0,17,2.9617354687499997,2023-11-20 15:42:44.910002				
20	0,18,2.9698110937499997,2023-11-20 15:42:44.910237				
21	0,19,2.98192453125,2023-11-20 15:42:44.910526				
22	0,20,2.99403796875,2023-11-20 15:42:44.911648				
23	0,21,3.00211359375,2023-11-20 15:42:44.911843				
24	0,22,3.01826484375,2023-11-20 15:42:44.911945				
25	0,23,3.02634046875,2023-11-20 15:42:44.912038				

VEHICLE1.CSV

vehicle1.csv	
1	Timestamp,roll,pitch,heading,rollRate,pitchRate,yawRate,groundSpeed
2	2023-11-20 15:48:45.219,7.7,6.3,37,0.0,0.0,0.0,0.7,0.0,0.000,-0.2,-0.0
3	2023-11-20 15:48:46.216,-0.1,10.0,36,0.0,0.0,0.0,0.9,0.0,0.000,-0.0
4	2023-11-20 15:48:47.215,-3.2,13.7,38,0.0,0.0,0.0,1.4,0.0,0.000,0.0,
5	2023-11-20 15:48:48.216,2.3,12.7,48,0.0,0.0,0.0,2.6,0.0,0.000,0.0,-
6	2023-11-20 15:48:49.224,1.5,2.0,57,0.0,0.0,0.0,2.7,0.0,0.000,-0.0,-
7	2023-11-20 15:48:50.216,4.1,2.8,62,0.0,0.0,0.0,2.1,0.0,0.000,-1.2,-
8	2023-11-20 15:48:51.216,4.6,-10.4,58,0.0,0.0,0.0,1.3,0.0,0.000,-0.5
9	2023-11-20 15:48:52.221,-1.2,-1.1,53,0.0,0.0,0.0,0.9,0.0,0.000,-0.0
10	2023-11-20 15:48:53.217,-1.2,-0.4,51,0.0,0.0,0.0,0.7,0.0,0.000,-0.0
11	2023-11-20 15:48:54.217,-0.9,3.2,50,0.0,0.0,0.0,0.7,0.0,0.000,-0.0,
12	2023-11-20 15:48:55.221,-0.0,6.4,50,0.0,0.0,0.0,0.8,0.0,0.000,-0.0,
13	2023-11-20 15:48:56.216,1.7,11.1,49,0.0,0.0,0.0,1.6,0.0,0.000,0.1,-
14	2023-11-20 15:48:57.217,1.9,2.5,45,0.0,0.0,0.0,1.4,0.0,0.000,-0.0,-
15	2023-11-20 15:48:58.220,2.2,0.8,37,0.0,0.0,0.0,1.6,0.0,0.000,0.0,-
16	2023-11-20 15:48:59.218,5.0,4.9,29,0.0,0.0,0.0,2.2,0.0,0.000,-0.0,-
17	2023-11-20 15:49:00.217,0.7,-3.7,7,0.0,0.0,0.0,2.0,0.0,0.000,-0.1,-
18	2023-11-20 15:49:01.218,-0.3,2.5,358,0.0,0.0,0.0,2.0,0.0,0.000,0.0,
19	2023-11-20 15:49:02.218,0.0,-0.9,45,0.0,0.0,0.0,2.2,0.0,0.000,-0.1,
20	2023-11-20 15:49:03.217,-1.6,-0.7,45,0.0,0.0,0.0,2.6,0.0,0.000,0.0,
21	2023-11-20 15:49:04.217,0.2,4.8,23,0.0,0.0,0.0,2.5,0.0,0.000,0.1,-
22	2023-11-20 15:49:05.218,2.2,7.8,16,0.0,0.0,0.0,2.5,0.0,0.000,0.0,-
23	2023-11-20 15:49:06.216,2.1,4.3,12,0.0,0.0,0.0,2.6,0.0,0.000,0.0,-
24	2023-11-20 15:49:07.217,1.7,4.8,7,0.0,0.0,0.0,2.5,0.0,0.000,0.0,-
25	2023-11-20 15:49:08.219,2.5,5.0,8,0.0,0.0,0.0,2.5,0.0,0.000,0.0,-
26	2023-11-20 15:49:09.217,1.3,6.0,5,0.0,0.0,0.0,2.5,0.0,0.000,0.0,-
27	2023-11-20 15:49:10.218,1.6,5.3,0,0.0,0.0,0.0,2.4,0.0,0.000,0.0,-
28	2023-11-20 15:49:11.219,2.4,4.7,3,0.0,0.0,0.0,2.3,0.0,0.000,0.0,-
29	2023-11-20 15:49:12.218,2.3,6.2,9,0.0,0.0,0.0,2.2,0.0,0.000,0.0,-
30	2023-11-20 15:49:13.218,1.7,6.0,15,0.0,0.0,0.0,2.0,0.0,0.000,0.0,-
31	2023-11-20 15:49:14.217,1.2,6.6,15,0.0,0.0,0.0,1.9,0.0,0.000,0.0,-
32	2023-11-20 15:49:15.218,1.7,6.1,12,0.0,0.0,0.0,1.7,0.0,0.000,0.0,-
33	2023-11-20 15:49:16.218,2.5,5.5,14,0.0,0.0,0.0,1.5,0.0,0.000,0.0,-
34	2023-11-20 15:49:17.219,1.3,6.4,17,0.0,0.0,0.0,1.2,0.0,0.000,0.0,-
35	2023-11-20 15:49:18.219,0.8,6.9,17,0.0,0.0,0.0,1.0,0.0,0.000,0.0,-
36	2023-11-20 15:49:19.218,1.8,6.3,15,0.0,0.0,0.0,0.8,0.0,0.000,0.0,-
37	2023-11-20 15:49:20.225,1.4,6.8,12,0.0,0.0,0.0,0.6,0.0,0.000,0.0,-
38	2023-11-20 15:49:21.217,0.9,7.2,8,0.0,0.0,0.0,0.5,0.0,0.000,0.0,-
39	2023-11-20 15:49:22.218,1.4,6.8,5,0.0,0.0,0.0,0.4,0.0,0.000,0.0,-
40	2023-11-20 15:49:23.227,1.3,7.2,1,0.0,0.0,0.0,0.4,0.0,0.000,0.0,-

MERGE DATA

Time Stamp

Timestamp

6,2023-11-20 15:42:44.905183

6,2023-11-20 15:42:44.906065

2023-11-20 15:42:44.906411

Timestamp,roll,pitch,heading,rollRate,pitchRate,yawRate,

2023-11-20 15:48:45.219,7.7,6.3,37,0.0,0.0,0.0,0.7,0.0,0

2023-11-20 15:48:46.216,-0.1,10.0,36,0.0,0.0,0.0,0.9,0.0

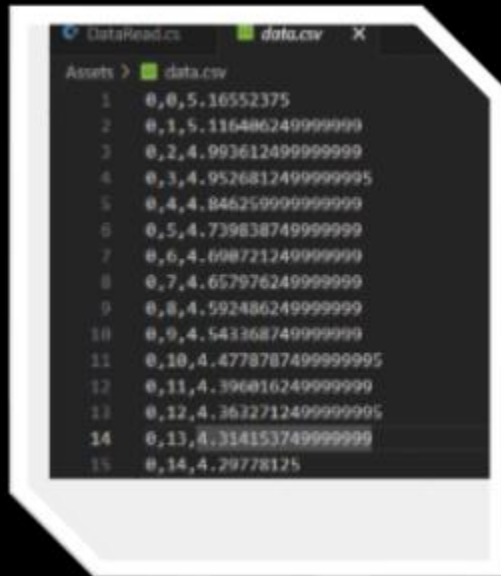
2023-11-20 15:48:47.215,-3.2,13.7,38,0.0,0.0,0.0,1.4,0.0

MERGING DATA

All_Data.csv

- Components:
 - Sonar
 - Telemetry data
- Based off timestamp
 - Time stamp not correct

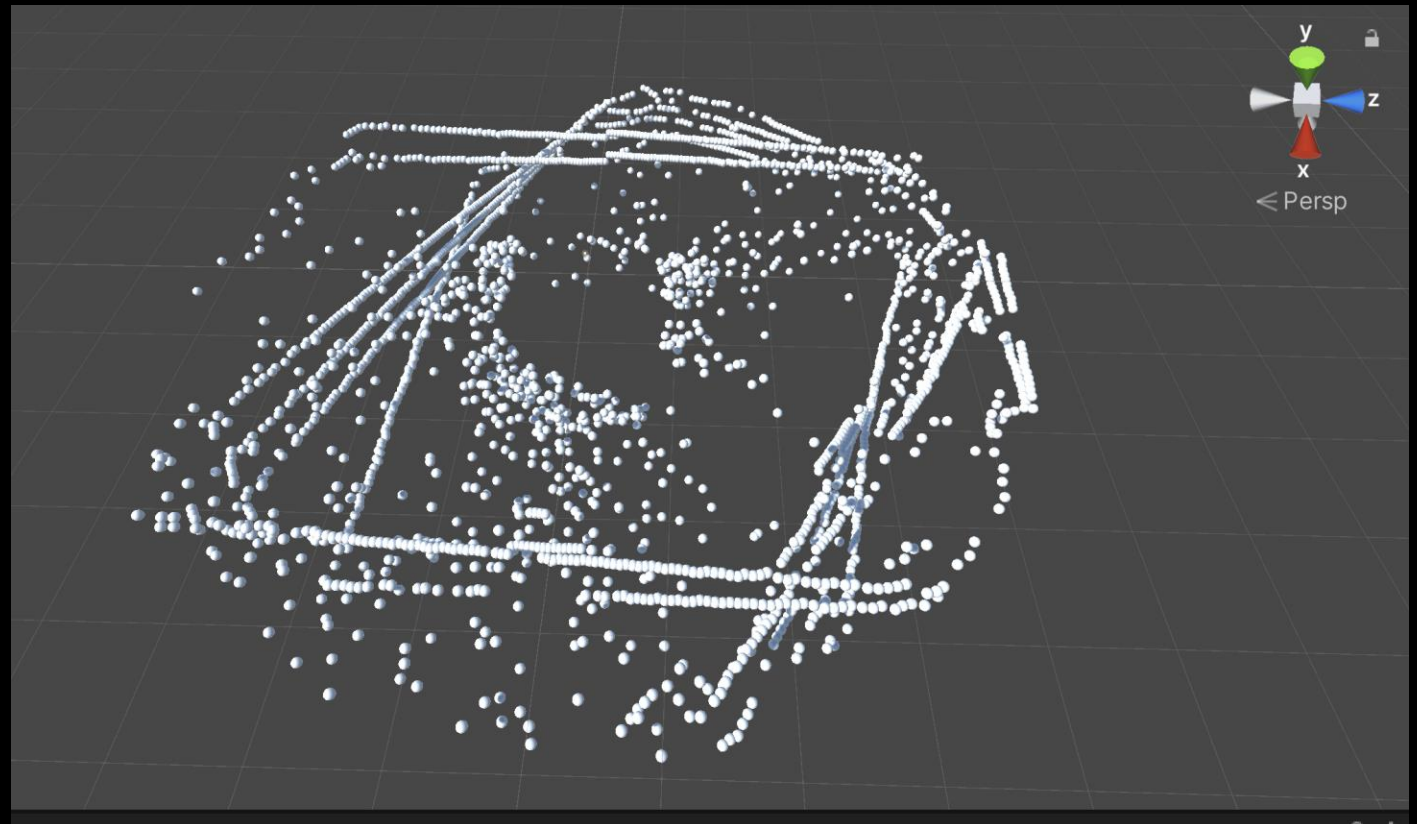
UNITY



```
while (!endoffile)
{
    string Data_String = stReader.ReadLine();
    if (Data_String == null)
    {
        endoffile = true;
        break;
    }
    var datavalues= Data_String.Split(',');
    Debug.Log(datavalues[0].ToString() + ',' + datavalues[1].ToString() + ',' + datavalues[2].ToString());
    Instantiate(sphere, new Vector3(Mathf.Cos(float.Parse(datavalues[1]))*
        (Mathf.PI/200))*scale*-1* float.Parse(datavalues[2]),
        float.Parse(datavalues[0])*scale/5, Mathf.Sin(float.Parse(datavalues[1]))
        * (Mathf.PI / 200)) * scale *float.Parse(datavalues[2])),
        new Quaternion(1, 1, 1,1));
}
```

11-22-23

- Rotation Test



TESTING

11-22-23

- Melbourne commons 10 a.m. to 1 p.m.
- Goal:
 - Test sonar data retrieval
 - Collect Data for Cloud Plotting
 - Roll rate for next sonar upgrade



MILESTONE 3:

Task	Michael	Zealand
False Data	Create an algorithm to remove false data points / fill in the shadows within the data to create a cleaner image.	
Depth Finder	Identify the protocols to find and retrieve this data, may need to be done through Arduino. The goal for this is to have accurate measurements of the current depth.	
Compass and Telemetry	Identify the protocols to find, retrieve, and save the information. This is so that once we start rotating the AUV we can track the current heading to assist with data transcription.	
Cloud Plot Application		Work on creating an environment that will transpose the data and allow for Autonomous testing in a virtual environment.

MILESTONE 4:

Task	Michael	Zealand
False Data Improvements	Create an algorithm to remove all false data points.	
Rotation Algorithm	Create a function to tun the two scans into the same orientation	
AUTONOMY	Use Gazebo to test partial pathing for current data sets	



Demo:

<https://www.youtube.com/watch?v=VTigK4eMFWs>

WEBPAGE LINK

TEC-V

https://bluecodehydra.github.io/FIT_Project-TEC_V/data.html