CS 682: Project 5

~Augmented Reality Assistant for Astronauts~

By- Parth Akre Karthik Mukka Pravallika V.

TABLE OF CONTENT

1. INTRODUCTION

1.1. Background

This is a continuous project, previously the students have successfully implemented a part of NASA SUITS requirements which is capturing the biomedicals of the astronauts. For this they were able to capture the live Body temperature, Fluid level, Oxygen level, Heart rate/ Pulse rate of a person by using the following sensors: Fluid Sensor, Pulse sensor, temperature sensor etc.

1.2. Problem Statement

As explained above obtaining the biomedical data is a small part of the original project requirements. What NASA wants us (students) to do, is to develop a software which can guide the astronauts throughout his space trip. A software (LMCC) which should be able to communicate with the HMD (we are HOLOLENS 2 as the HMD for this project). This LMCC will act as the server for the HMD. The LMCC operator has the ability to control the HMD and be able to communicate with the astronauts through HMD. So, we are creating two User Interfaces one is for LMCC and other is for HMD.

Requirements :

The requirement for our project is to create an User Interface (HMD) which should be performing the following requirements:

a. Mapping and Navigation:Shall include a 2D map showing points of interest and live asset locations.
b. EV Telemetry: This should display the real time biometric information of the astronaut.
c. Caution and Warning System: Shall alert the wearer of off-nominal ranges in suit or biometric telemetry and when a particular vital drops or rises above its threshold value.
d. Scientific Data Reporting and Log:Shall provide alerts for various stages of sample scanning and analysis.
e. Pin Dropping Feature: Shall allow the astronaut to drop pins within the

augmented reality environment during the EVA.

2. Hardware/Software

Hardware components:

- ➔ Rover with native camera.
- ➔ Realsense camera D435i
- ➔ HoloLens 2
- ➔ Raspberry pi 4
- ➔ Sensors- Temperature

Software Components:

- ➔ JavaScript, HTML, CSS for LMCC
- ➔ C# for HMD
- ➔ Python for Rover and RealSense camera
- ➔ Framework: Unity, Windows Mixed Reality Toolkit and visual studio

3. Implementation

3.1 HMD/Hololens

Implemented a GUI for the Hololens2 on the Unity Framework by writing C# scripts that perform in accordance with our functionality and attach them to the game-objects that we imported by using Windows Mixed Reality Toolkit. Once the scripts are attached we can create a scene on Unity to display our GUI. This scene is now ready to be displayed on the Hololens once we build the Visual studio solution for it. Now we can connect it to our hololens by inputting the IP address of the Hololens on the network it is currently active on in the debugging tab of the solution and deploy it. Once this is done it is ready to be rendered by the HMD for the user to interact with via touch on the virtual panel with eye gaze and pinch feature. We have also included voice commands to increase the ease of access for differently abled users.

3.2 LMCC

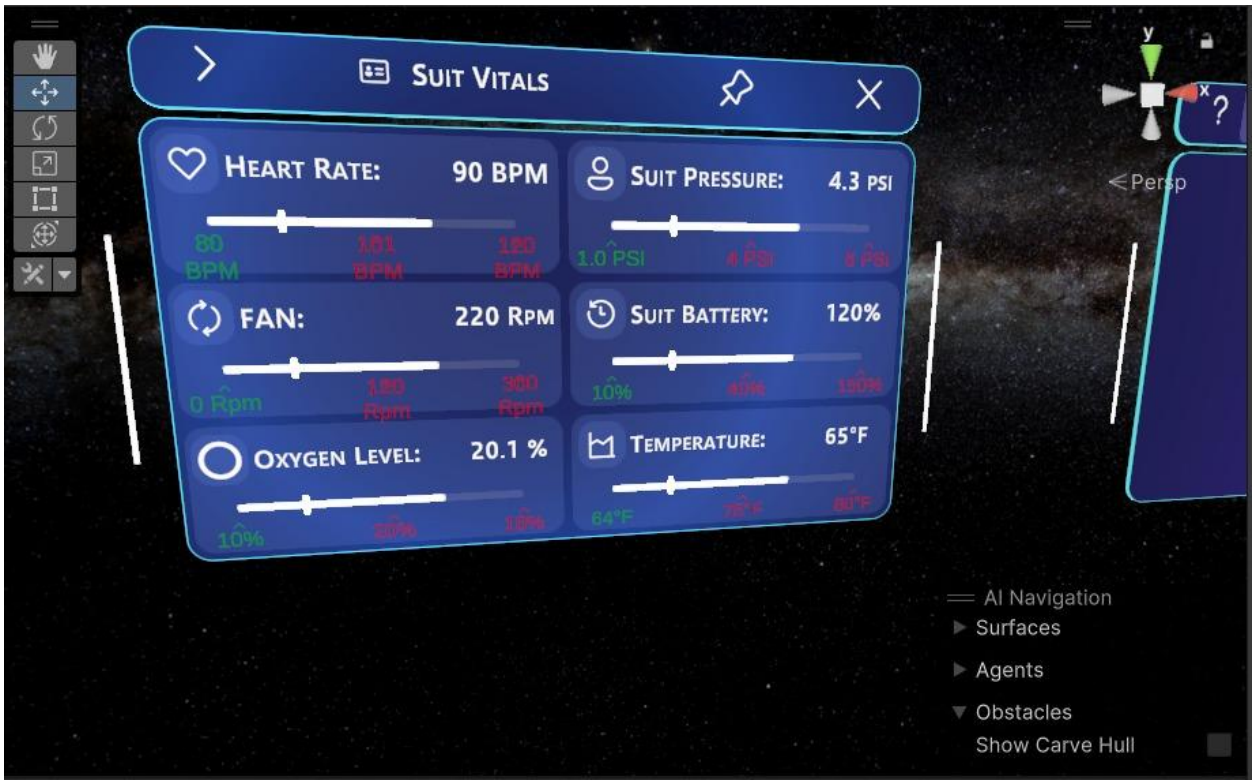
LMCC is a web application which will further communicate with the Hololens. NASA has provided the requirements of what needed to be integrated into this web application. LMCC's development started from this project. To the web page we have added a "Task Menu" where the LMCC operator can assign the tasks to a particular astronaut, and then the astronaut can visualize what all the tasks have been assigned to him. This is the feature of this task. And the web page also has maps integrated in it. These Maps should be able to detect the location of the astronauts and all the assets like rovers etc. We still have the EV telemetry feature integrated into the webpage. EV Telemetry shows the biomedicals of the astronauts. These are a few features we have worked on for LMCC.

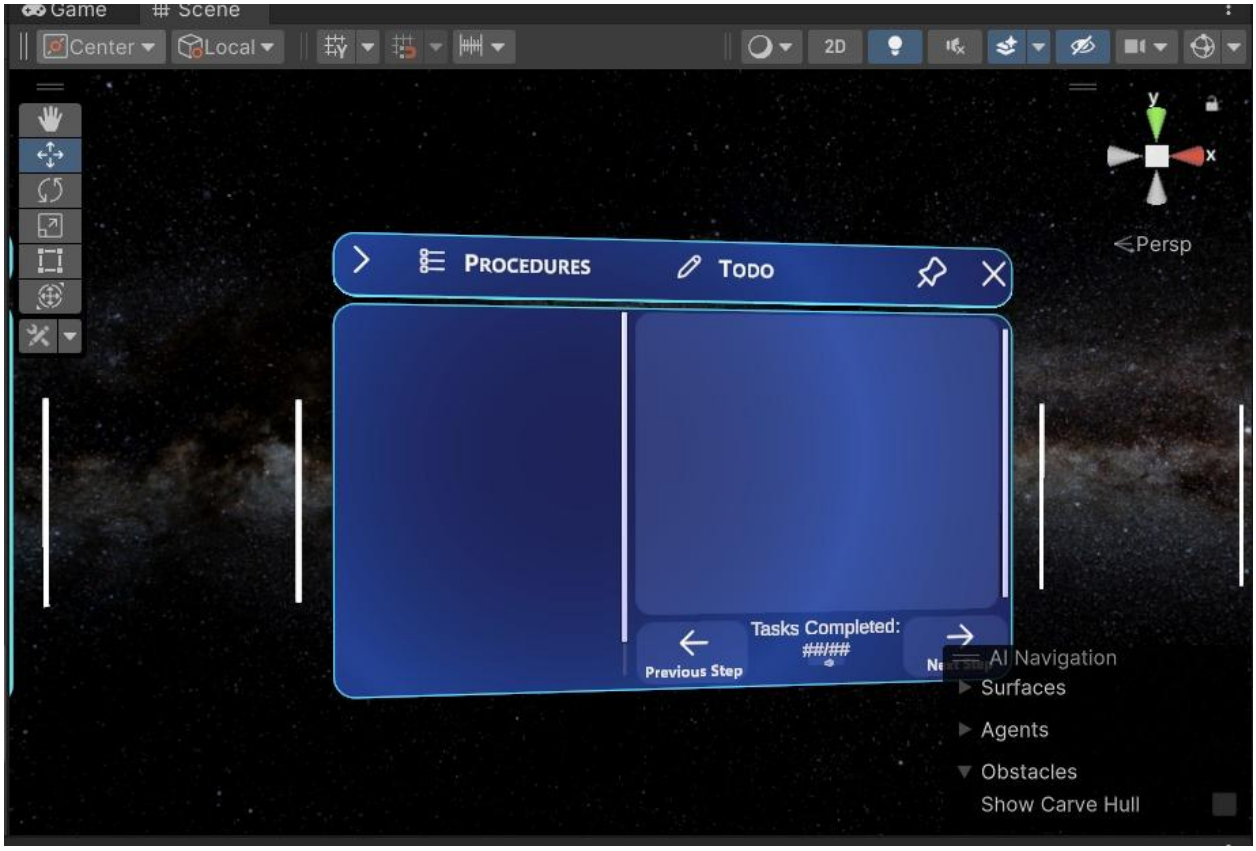
3.3. Realsense Camera

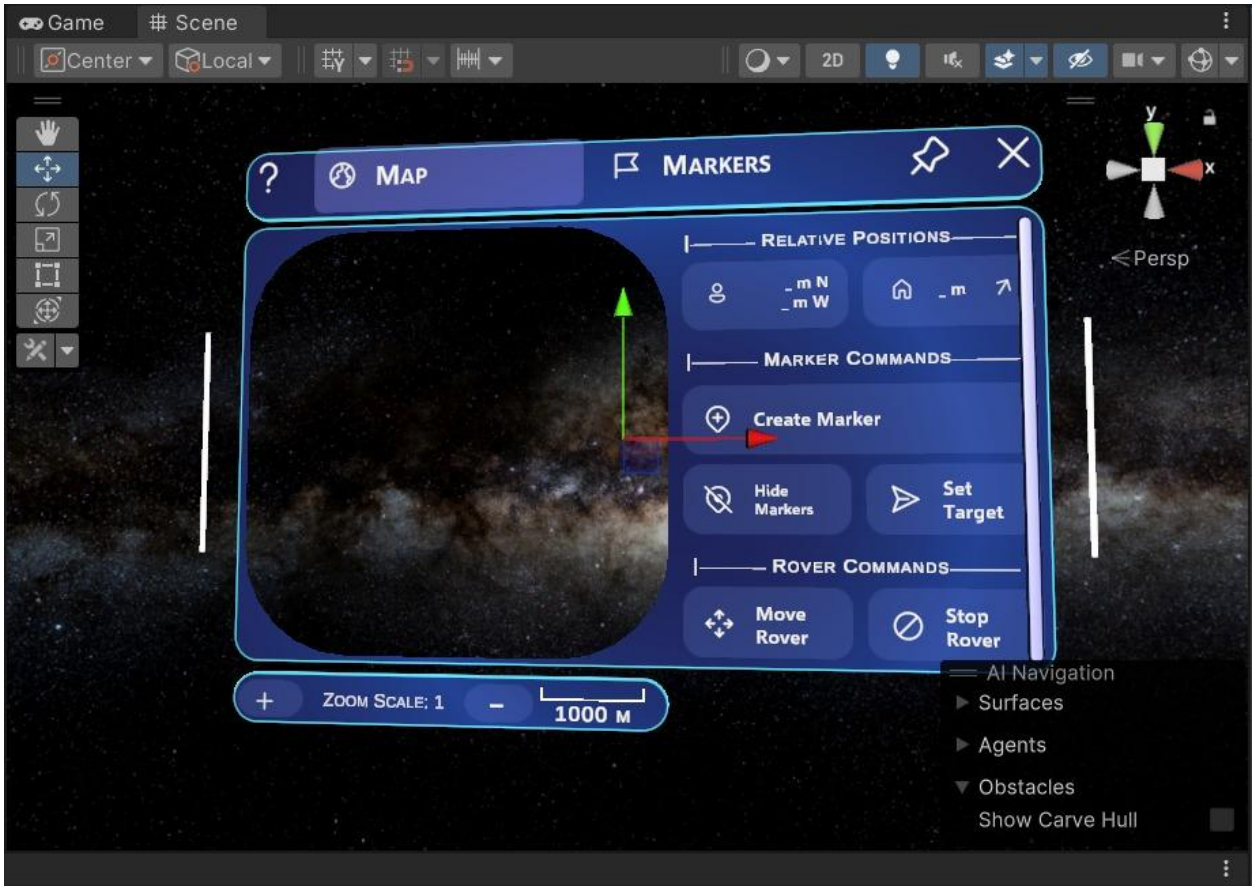
For this project we are using the Realsense camera D435i. What this camera does is, it will calculate the distance of the object from the camera. This camera is mounted onto the rover with its native camera. This is a feature of capturing the realsense camera feed and visualizing it on the LMCC webpage. We were able to calculate the distance of the object from the camera successfully. All that is left is to integrate the camera feed onto the webpage.

4. Result

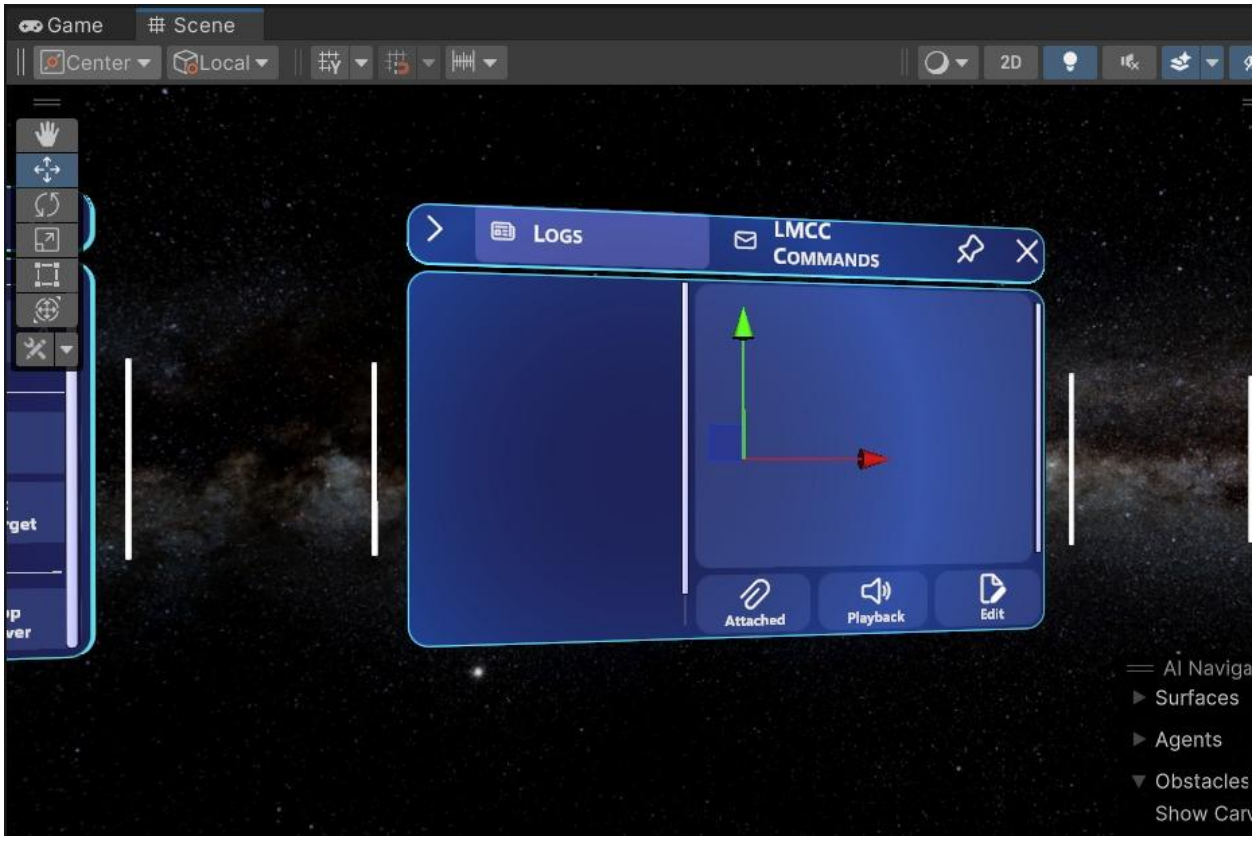
4.1. HMD




Game
Scene
Center
Local
2D
Persp
PROCEDURES
TODO
Tasks Completed:
#####
Previous Step
AI Navigation
Surfaces
Agents
Obstacles
Show Carve Hull

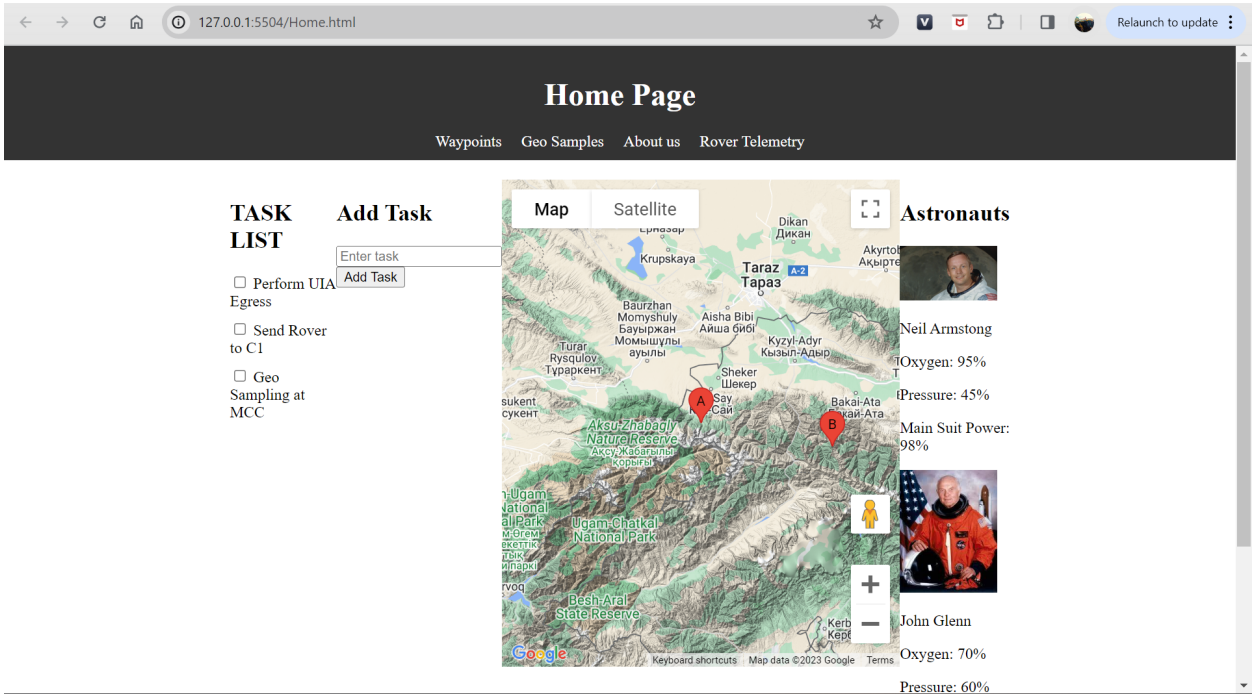

Game
Scene
Center
Local
2D
Map
Markers
Relative Positions
_m N
_m W
_m
Marker Commands
Create Marker
Hide Markers
Set Target
Rover Commands
Move Rover
Stop Rover
Zoom Scale: 1
1000 m
Persp
AI Navigation
Surfaces
Agents
Obstacles
Show Carve Hull


Game
Scene
Center
Local
2D
Enter Log
2023:09:02 15:07
Enter Subject...
Enter Content...
Icon
Persp
AI Navigation
Surfaces
Agents
Obstacles
Show Carve Hull


Game
Scene
Center
Local
2D
Logs
LMCC
Commands
Attached
Playback
Edit
AI Naviga
Surfaces
Agents
Obstacles
Show Carv

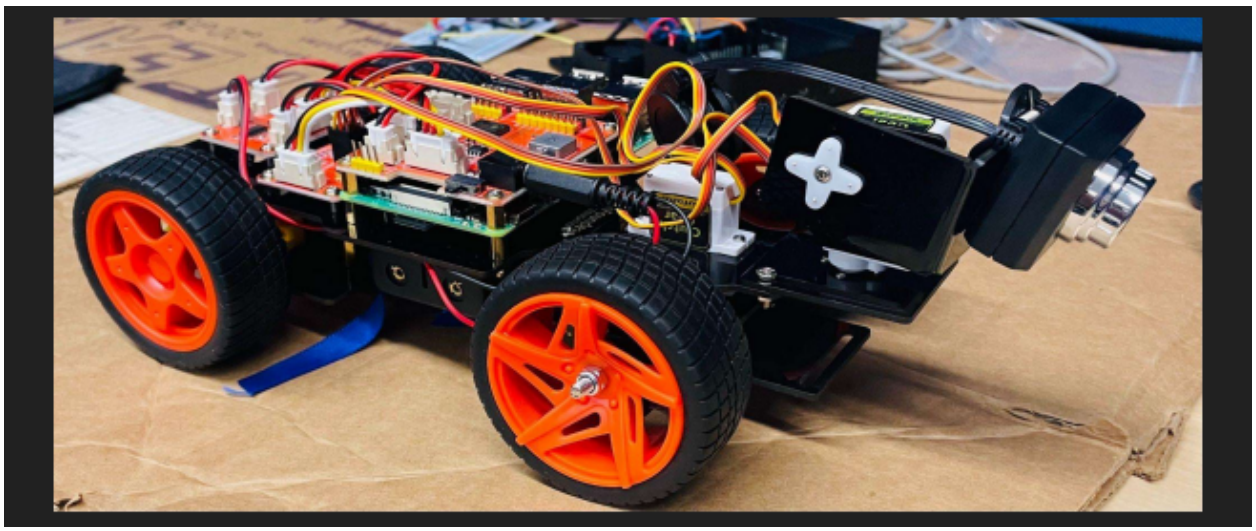

Game
Scene
Center
Local
2D
Persp
Heart Rate
Suit Pressure
Fan
Temp
O2 Level
Battery
AI Navigation
Surfaces
Agents
Obstacles
Show Carve Hull

4.2. LMCC



4.3. Rover

We need to remotely control the rover which has been successfully implemented. There is a video which has captured the movement of the rover. These are few pictures of the rover.



This rover has a built in native camera, which later is visualized by the LMCC operator.

4.4. RealSense camera feed

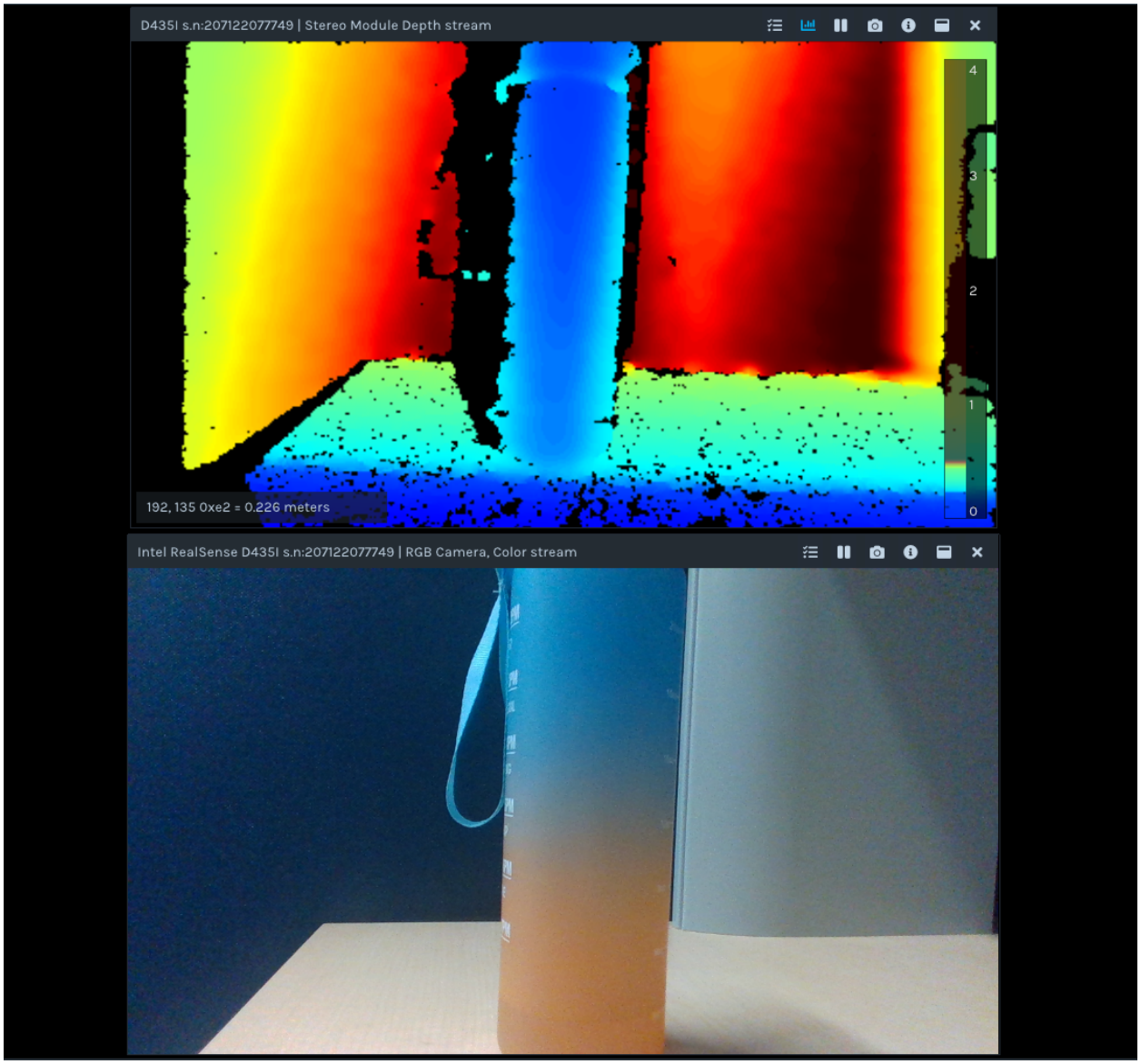


Fig 4.4.1

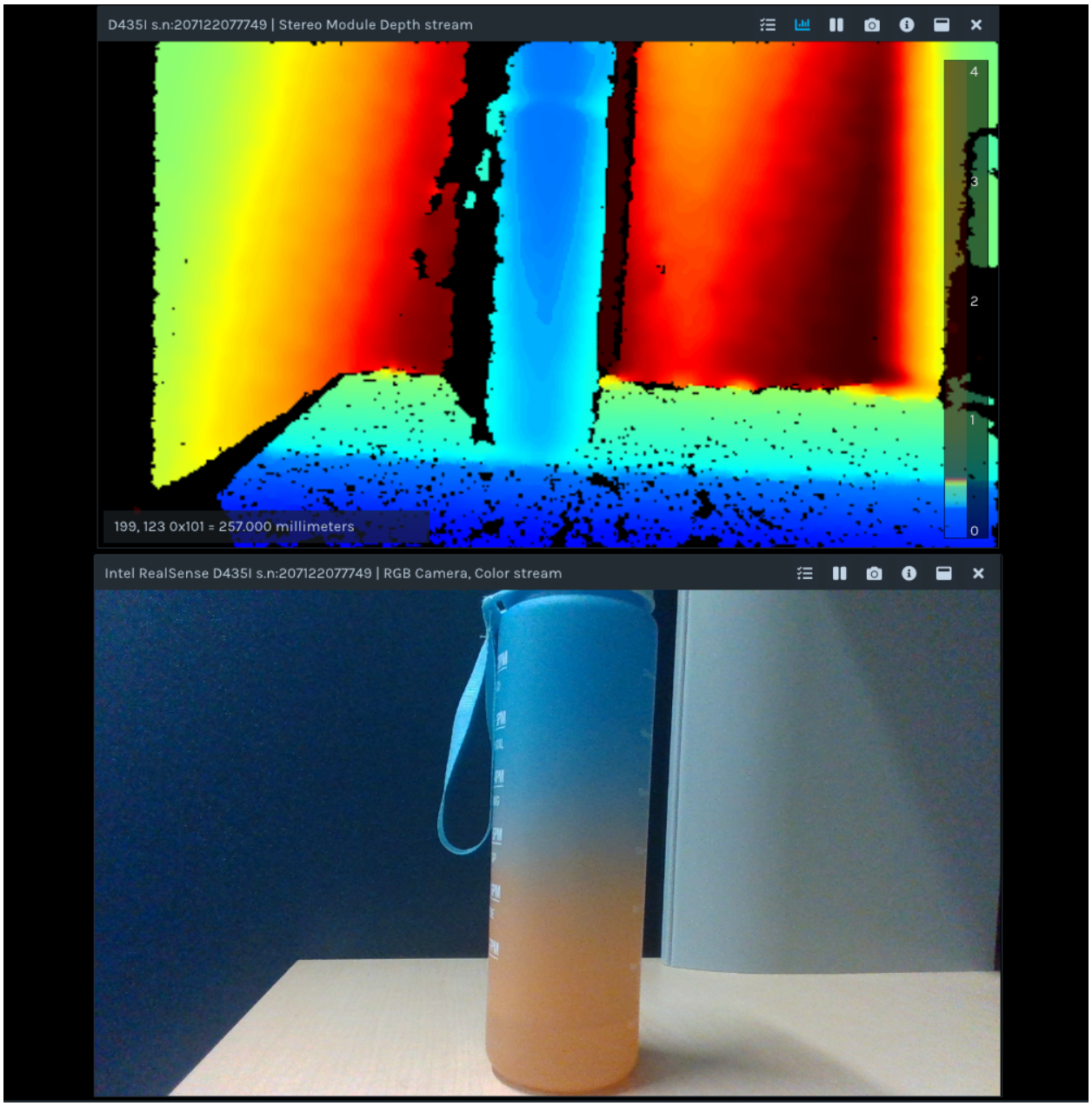


Fig 4.4.2

This is the screenshot of the realsense camera feed. As you can see the camera is detecting that the water bottle is the object and it is calculating the distance of the object from the camera. There is a difference in distance of the object from the camera in fig 4.4.1 the object is 0.226 meters far from the camera and coming to fig 4.4.2 it is 0.257 meters far from the camera. This means that the object is moved far away from the camera. The implementation of the RealSense camera.

5. Conclusion

In conclusion we have implemented the GUI for the HMD and UI for the LMCC. In addition for our project we have done some of the original requirements like realsense camera and rover. We have yet to obtain the communication between HMD and LMCC.