RIDWAN HOSSAIN
100747897

2022/06/10

- Will continue on from yesterday and add the remaining vehicle telemetry data to the data logging pipeline.
- Added GNSS and vehicle z-axis displacement to the data logging pipelines.
- **Note:** When working with the throttle data, I noticed that the throttle was jammed between 0 and 1, it was either fully closed or fully open. Looked into the code base to figure out why.
- After some searching, it was found that there was a faulty conditional statement that mapped the throttle to 1 (100%) if the throttle was above 0. I resolved this conditional statement and the throttle is now much more responsive and accurate. Now I can more accurately capture the throttle position data.
- The throttle and brake pedal positions were converted to be represented as percentages inside the datasets.
- Realized that I could add the simulation time to the dataset to easily match a video to a set of data, as the simulator is always displaying the simulation time. This will provide coherency when relating the data and the video.
- Increased the resolution of the simulation time for higher accuracy and added the simulation time to the data logs.
- Reworked the data logging pipeline to create a folder named after the map the data was recorded in. This is because I want to organize the files in such a way that data logs are saved in folders pertaining to the map they were collected in, and within those logs, one of the data items being logged is the vehicle, as during a recording, you may switch your vehicle, but the map stays constant. It is only possible to change the map by changing the code.
- **After implementing more items to the data logging pipeline, the below table represents the current data items being logged in a recording:**

Data Item	Added	Not yet Added
Simulation Time	✓	
Recording Time	✓	
Server Frame Rate	✓	
Client Frame Rate	✓	

Vehicle Speed (kph)	✓	
Vehicle Heading (degrees and heading)	✓	
Accelerometer Data	✓	
Gyroscope Data	✓	
Location Coordinates	✓	
GNSS Telemetry	✓	
Vehicle Height (Z-axis Displacement)	✓	
Throttle Position (Percentage)	✓	
Brake Position (Percentage)	✓	
Current Gear	✓	
Steering (-1 to 1 represents left lock to right lock respectively)	✓	
Vehicle Label (make and model)	✓	
Collision Data		✓
Lane Invasion Data		✓
Obstacle Detection		✓

- Next step is to implement collision detection and obstacle detection to the data logging capabilities. This source may be useful: https://carla.readthedocs.io/en/latest/ref_sensors/
- It was decided to have collision data be exported as a separate dataset because collision events may occur outside the sampling rate of the data.
- After some extensive work, I was able to obtain the collision data in the form of the time it occurs and the intensity. This data will be exported as logs whenever a collision occurs and will contain the following data:
    - Simulation Time
    - Recording Time
    - Collision Event Message
    - Collision Intensity (Can be graphed in the form of a bar chart)
- I will create another pandas dataframe and export the collision data as a separate csv.

- Collisions will be appended in the **_on_collision** function.
- Successfully implemented the addition of the collision data to the data logging pipeline. I would like to reorganize the file structure of the pipeline once again to improve the coherency. I will group sets of vehicle telemetry and collision data into folders called datasets, which will be numbered. These dataset folders will be contained within the map folders.
- Revised the file structure of the data logging pipeline as mentioned above. The next step is to log lane invasion data, such as crossing solid lines or broken lines.Lane invasion data will also be exported as a separate csv as it may occur outside of the sampling rate.
- Successfully implemented the addition of the collision data to the data logging pipeline. The next step is to figure out how to implement an obstacle sensor and obtain the data it reads. After this, I will also add this to the data logging pipeline.