RIDWAN HOSSAIN
100747897

2022/06/16

- Will continue from yesterday and add the **obstacle detection data** and lane invasion data to the graphing script
- For some reason, the data being displayed on the y axis do not correspond with the values in the data at all, and there does not seem to be an issue with the code. Looking into why this might be.
- It was found that as with the collision data, multiple obstacle detection readings correspond to the same **Rec_times**, causing them to stack (source: https://community.plotly.com/t/plotly-bar-chart-values-not-matching-data/14583/2). I just need to go through the data and remove any duplicates, this time, **keeping the smallest readings recorded for each time** as the readings with closer (smaller) distances to the vehicle are more important.
- After adding the necessary changes, the resulting graph produced a correct and coherent graph. The next step is to graph the final data item, **lane invasion data**. This data will be graphed **categorically**, plotting a point for each type of line types broken at each point in time.
- The lane invasion data was successfully graphed. Now that all the graphable data was visualized, I exported all the interactive graphs as html files in a folder called **Visualized Data**, located within their respective dataset folder.
- Listed below are the data items that have been visualized into interactive graphs:

Data Item	Graphed	Not yet Graphed
Speed	✓	
Server & Client Frame Rates	✓	
Throttle Application	✓	
Brake Application	✓	
Steering Input	✓	
Height	✓	
Accelerometer Readings	✓	
Gyroscope Readings	✓	

Collision Data	✓	
Lane Invasion Alerts	✓	
Obstacle Detection Data	✓	

- **This concludes the graphing of all the data I collected from the first dataset. The graphing script can easily be used to graph any dataset produced from the Carla bench in one simple run of the code. The next steps now are as follows:**
    - Produce more data for other maps and use the script to visualize the data.
    - Produce an informal script for the demo on Tuesday.
    - Continue to make test cases for the additional functions I created (CAN messages and CAN attacks if possible).
- The next dataset to be produced consists of driving a compact car in a world populated with traffic and pedestrians.
- Below is a chart showing what datasets have been produced and which ones are yet to be produced:

Dataset			**Produced**	**Not yet Produced**
Vehicle Type	**Traffic**	**Map**		
Compact	No	Town10HD_Opt	✓	
Compact	Yes	Town10HD_Opt	✓	
Sedan	No	Town10HD_Opt	✓	
Sedan	Yes	Town10HD_Opt	✓	
Sports	No	Town10HD_Opt	✓	
Sports	Yes	Town10HD_Opt	✓	
SUV	No	Town10HD_Opt	✓	
SUV	Yes	Town10HD_Opt	✓	
EV	No	Town10HD_Opt	✓	
EV	Yes	Town10HD_Opt	✓	
Compact	No	Town06	✓	

Compact	Yes	Town06t	✓	
Sedan	No	Town06	✓	
Sedan	Yes	Town06	✓	
Sports	No	Town06	✓	
Sports	Yes	Town06	✓	
SUV	No	Town06	✓	
SUV	Yes	Town06	✓	
EV	No	Town06	✓	
EV	Yes	Town06	✓	

- The reason the two maps were chosen is because **Town10HD_Opt** contains compact roads and simulates a city area well, and **Town06** consists of long highways and even off-road terrain. These 2 maps provide a good variety of roads and environments to collect data from. In conjunction with these 2 maps, a combination of traffic presence and each vehicle type will be used to produce datasets. By the end, I plan to produce a total of **20 datasets**, which shouldn't take too much time since I streamlined the data logging and visualization pipeline so well.