

2022/06/14

- **Will continue from yesterday and complete the following items:**
 - Add whether the autopilot is enabled in the vehicle telemetry logs for the data logging pipeline.
 - Pair a screen recording function with the data logging.
 - Generate some sample data for a map, and develop some scripts to process and visualize the data which makes sense to be graphed.
 - Produce more data for other maps and use the scripts to visualize some of 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).
- Added whether the autopilot is enabled in the data logging pipeline. The next step is to implement screen recording from a code level and pair that with the data logging functionality. The following source may be helpful for implementing screen recording from python: <https://www.geeksforgeeks.org/create-a-screen-recorder-using-python/>
- After trying the above source and various other methods, it was decided against using screen recording done at the code level as all the researched methods utilized the **opencv** library and methods of taking screenshots. The issue with this is that this resulted in footage that was either significantly too fast or too slow and it required retiming to be done manually, by eye, through trial and error. This would make the recordings not at all useful for recording the Carla simulation as it is crucial that the rate of the recording be the same as the data being logged. **As such, it was decided to simply use a screen recording software and use the simulation time visible in the Carla simulator to match the screen recording with the logged data.**
- A software called **simple screen recorder** was installed on the Linux machine and will be used to perform the screen recording when collecting data. The one downside of this is that the recordings must be saved into their respective dataset folders manually, opposed to having the convenience of the data logging pipeline I constructed. However, this is a small hindrance as the rest of the data logging is highly streamlined by the pipeline.
- The next step is to generate some sample data for a map and develop some scripts to process and visualize the data which makes sense to be graphed.
- The first map I will sample data from will be the default server map, **Town10HD_Opt**. The plan for each map is to collect data from 5 different types of vehicles: **compact**,

sedan, sports/coupe, SUV, EV. Each run will include some normal manual driving, some sporadic and aggressive manual driving, and some autopilot. There will also be 2 runs for each type of vehicle, one without traffic and pedestrians, and one with. The driving behavior will vary from map to map as the road types and terrains will change.

- Produced first dataset using a compact car and no traffic. Also created a spreadsheet to keep track of what data each dataset folder contains, called **info.csv**.
- As the video files are very large, mp4 files have been added to the gitignore file for the repository and will instead be uploaded to the google drive folder.
- The next step is to develop scripts to process and visualize the data which can be graphed from the first dataset. This will most likely require the use of the **matplotlib** or **plotly** python library.
- The following sources may be helpful:
 - <https://plotly.com/python/plot-data-from-csv/>
 - <https://plotly.com/python/tick-formatting/>
- Was able to build a plot of the speed and add interactive features displaying the speed vs the recording time and whether autopilot was active at the time or not. The next step is to make interactive graphs for the remaining data:
 - Server Frame Rate
 - Client Frame Rate
 - Accelerometer Data
 - Gyroscope Data
 - Height (Not useful for most maps)
 - Throttle
 - Brake
 - Steering
 - Collision Data
 - *Possibly lane invasion and Obstacle detection data?*

The lane invasion and obstacle detection data can either not be graphed or is better viewed in the form of logs, which they already are in. There *may* be some aspects about the data which may be visualized.

- Went through and modified the scripts to differentiate between Manual control and Autopilot control.

- Listed below are the data items that have been graphed and the ones which have yet to be graphed:

Data Item	Graphed	Not yet Graphed
Speed	✓	
Server & Client Frame Rates	✓	
Throttle	✓	
Brake	✓	
Steering	✓	
Height	✓	
Accelerometer Data		✓
Gyroscope Data		✓
Collision Data		✓
<i>*Lane Invasion</i>		✓
<i>*Obstacle Detection</i>		✓

- The reason Lane invasion and Obstacle detection are asterisked is because it may be better to simply leave them in their log form, but it may be possible to graph some aspect of their data.