

# 2022/06/08

- Researched into how to record and retrieve simulation data from the Carla simulator. Found useful information on their docs:  
[https://carla.readthedocs.io/en/latest/tutorial\\_ego.py](https://carla.readthedocs.io/en/latest/tutorial_ego.py)
- From reading the documentation, it seems that the Carla website provides a script that can be run alongside the client program in order to record sensor data.
- Found that a modification to the main code made yesterday has negatively altered steering input responsiveness. I will look into this issue and resolve it immediately.
- The steering issues was tied to an error in the range of values being used for a conditional statement, and has been resolved. Steering functions have been restored.
- Was able to run the Carla provided simulation data recorder script (**tutorial\_ego.py**) alongside our updated Carla simulator (**testbed.py**), however, it was noted that running both the simulator and recorder at the same time deteriorates the simulator's performance dramatically. I'm looking into why that might be.
- It was discovered that recording the simulator results in a photo of every frame being captured and saved to a directory of the rgb camera in the simulator, which may be the reason the performance deteriorates so much when the recorder is also running. I removed the portion of the recording script which captures every frame.
- This remedied the issue regarding the performance drop. It was discovered the recorder can also spawn in a vehicle of its own and perform self driving functions and record that data. However, the recording and data is not saving upon ending the recording. I will look into why the recording is not saving.
- After extensive research and online searching, the reason as to why the recording was not saving was not found. After some further investigation into the Carla documentation ([https://carla.readthedocs.io/en/latest/advanced\\_recorder/](https://carla.readthedocs.io/en/latest/advanced_recorder/)) it was found that having the '/' character in the **start\_recorder** function results in the save path being absolute, which explained why the results of the recording was not saving. After changing the file path to an absolute one, a sample recording was finally able to be saved.
- To ensure the validity of Carla recordings, I wanted to be able to see the recording being replayed and the data that it had captured. After some more investigation into the Carla documentation and some reading over, I was able to construct a script that spawns in an actor in the Carla **server** (**CarlaUE4.sh**) and reenact the actions of the vehicle in the recording. I was also able to output some of the data captured from the recording to the terminal window. The next step is to try and extract as much useful data from the recording as possible, and to do so, I'll have to keep reading through the documentation and see what I can come up with.

- Gave it some thought and realized that if the data is to be viewed by external parties that may not be using any Carla software, this form of data may not be useful to them. As such, I have decided to change directions slightly and manually output the data captured while driving into a csv file, and I will also visualize that data in different formats, such as a video, charts, etc. **However, this does not mean the work done today has been a waste. As I have now figured out how to record and reenact the actions of a vehicle, this can be used to view the same path of a vehicle from different sensor types, such as an rgb camera, log camera, depth sensor, lidar, etc.**
- The next step is to export the recording function to **testbed.py** in order to be able to record manual driving which is controlled by the user.
- I was able to implement the recording functionality into testbed.py and also export a sample vehicle run to a rec file. I will have to compare the quality of the replay to that of a log file but it seemed a log file recording provided better replay.
- It was found that the log format provides slightly smoother replays, however, both have issues. **It might be best to simply record the screen as I am driving manually using a screen recording software and accompany that video with the exported data. As such, I am going to remove the recording function from testbed.py as it is producing an excessive amount of unnecessary data whenever a recording is made, causing commits to the repository to take an extremely long time.**
- **The next step is to export vehicle telemetry data when recording in the form of a csv. Some data we can export is:**
  - Server fps (can be graphed)
  - Client fps (can be graphed)
  - Speed (can be graphed)
  - Compass heading
  - Accelerometer data
  - Gyroscope data
  - Location data (can be graphed)
  - GNSS data
  - Height data (can be graphed) (not useful on flat maps)
  - Throttle data (can be graphed)
  - Steering data (can be graphed)
  - Brake data (can be graphed)

- Collision data (Can be graphed)
- Line breakage data
- **All the above data combined, listed out in the form of a log**
- It is best to start with the last item first, as all the other data can be derived from the last item
- An accompanying the above data, a video of the driving from which the data is pulled from
- This data can also be retrieved from different maps and take note of the differences, particularly with maps that have changing elevation and varying road types.