RIDWAN HOSSAIN
100747897

2022/05/25

- Found a developer who provided formulas for how to obtain the vehicle's engine rpms using the Carla interface on a GitHub qa board: https://github.com/carla-simulator/carla/issues/2989

- Added this functionality to the get_speed function in testbed.py and printed the vehicle's engine rpm values to the terminal window for testing purposes.

- Found that currently the speedometer is not very responsive to the vehicle's speed and does not update frequently. To change that, the baud rate for the data transfer to the Arduino board was updated to **38400**, which was found to be an optimal value through trial and error (too high means even more stray values, too low means less responsive).

- Noticed the engine rpms immediately dropped to 0 if the user lifted off the throttle, even at speed, which would not be realistic relative to a real engine's behavior. To mitigate this issue, added a temporary throttle variable called '**tempThrottle**' that holds the most recent throttle value whilst the vehicle is at speed. This allows the rpm calculator formula to still use a nonzero throttle input in order to calculate the current engine rpms whilst the car is moving but off throttle.

- Was able to pack the speed and rpm values into a single message sent to the Arduino board. Packing the values resulted in the speedometer being less responsive, so I increased the baud rate to the next highest double the current rate, which is **115200**. This resulted in the speedometer being more responsive. Next step is to display the received signals on the tachometer as well.

- Was able to have the RPMs display successfully on the gauge cluster's tachometer. Next step is to get the scaling factor between the software values and the hardware signals to an optimal level. This will be obtained through trial and error.

- It seems that the formula used to obtain rough rpm values does not sufficiently replicate the behavior of a vehicle's engineer as it gains and loses rpms through the gears. Instead of increasing the rpms throughout the rev range and resetting after each shift up and vice versa, the needle simply increases with the vehicle's speed, acting similar to the behavior of a speedometer. For better results, I looked into other ways to obtain the vehicle's engine rpm in Carla. May need to look into the windows version of the previous capstone group's project to see what methods were used to obtain an rpm value.

- Realized the reason the rpm values were acting strange was because the Arduino board was not receiving them properly. This issue was fixed. However, this caused another issue: Once both values are packed and send to the Arduino side, the speed values become inaccurate and strange, opposed to when only sending speed values, where they are correct and consistent.

- Next step is to figure out how to properly pack the values such that they are read correctly on the Arduino end, as currently, the RPMs are read fine, but the speedometer values have become scattered.