RIDWAN HOSSAIN
100747897

2022/05/24

- Carla simulator (testbed.py) seems to freeze after driving for a little bit. This seems to be directly related to the serial write function sending speed values to the Arduino board as when that statement is suppressed, the simulator runs fine and does not freeze.

- get_speed function is now returning correct speed values on the client machine, next step is to correctly communicate them to the Arduino board.

- The Arduino board is not correctly interpreting integer values. A resource online suggests a way to resolve this issue is to simply send the values to the board as a string, and then convert them back into integer values on the board's side: https://stackoverflow.com/questions/56316443/how-to-send-an-int-from-python-to-an-arduino-in-order-to-use-it-as-an-argument-f

- Board is having trouble receiving strings, some time will be spent troubleshooting and researching this issue,

- Found an alternative method to send integer values to the Arduino board by using the struct.pack() method provided in the pyserial library. This results in integer values sending, however, a lot of stray '0's, '-1's and '-12's are being read within the serial monitor, causing the speedometer to bounce and act unstable. I will use a temporary variable on the Arduino side program to hold the most recent non zero value and send that value to the speedometer. This should allow the speedometer to ignore the stray signals on the serial monitor.

- After ignoring all the stray signals, the board is now able to correctly receive the speed values sent from the python side. Next step is to correctly scale the range of values the board produces to the range of values that the speedometer on the gauge cluster can represent.

- The statement '**int mappedSpeed = map(incomingSpeedValue,0,115,0,155);**' in the arduinotest Arduino program was used to achieve the correct speed scaling factor, where the vehicle speed in the Carla simulator is correctly translated to km/h on the gauge cluster's speedometer.

- Next step to finish working on the gauge cluster is to correctly display the rpms on the tachometer, after which I will move onto the android infotainment system.

- **Note:** Running testbed.py without the serial monitor open causes the Carla simulator to freeze on launch for some reason, so make sure it's open when you're running testbed.py.

- Discovered that the speedometer and tachometer Arduino code needs to be in the same Arduino program, so I need to pack the appropriate values into one message to send to the board each time.

- After successfully packing multiple values together, looked into how to obtain the engine’s rpms in Carla, as there is currently no way to access that information without going into the engine level code of Carla (Unreal Engine).