RIDWAN HOSSAIN
100747897

2022/06/07

- Will continue making test cases for the new features added to the updated Carla bench.
- **Note:** The tests that require the use of the Carla simulator have their client side code in the **parse_events** function.
- **Hand_Brake_Test (Test id = 22)**
    - The purpose of this test is to ensure that the added functionality that allows the user to apply the handbrake and have it locked until used again is functioning as expected
    - The expected behavior is that if the handbrake is pulled while stationary and the throttle is applied, the car will not move (or at least not move fast). To obtain a reasonable speed the car should be under when the hand brake is pulled, some manual testing is required to acquire a reasonable threshold.
    - After doing some manual testing, it was found that if throttle is applied when the hand brake is active and the car is stationary, the vehicle speed does not exceed 1 kph for any of the vehicles tested. As such, the threshold will be < 1 kph.
    - **Note:** To access throttle, brake, and steering controls for testing purposes, previous statements can be found in the **_parse_vehicle_wheel** function.
    - After making the test file, the test passed successfully.
- **Shifter_Test (Test id = 23)**
    - The purpose of this test is to test whether certain button inputs change the vehicle's gears to the correct ones when in manual mode.
    - This will be done by simulating button presses from the shifter.
    - The **parse_events** function was modified to check if a shifter test is being run, and if so, simulate a certain button press as passed to the function, and afterwards, return the current gear the vehicle is in.
    - After some extensive work and modifications to numerous functions, I was able to simulate button presses from the shifter. The next step is to see if I can cause that button press to trigger a gear change, just like normally using the controller.
    - After some extensive work, I was able to externally change the vehicle's gear from the test file. However, after being able to do so, a problem arised. After changing the gear externally once, I am not able to do it again within the same test case. As such, the test file contains multiple test cases, each time the Carls world is reinitialized, allowing the gear to be changed multiple times in one test.

- After making the necessary modifications, I was able to make the test verify that changing into each gear using simulated button presses when in manual mode resulted in the vehicle shifting to the correct gear. The test passed successfully.

- It was discovered that as the test technically generates 8 worlds for each of the 8 gears, not all of them are destroyed once the test is complete, and as such, when the simulator is ran again, there are many actors (vehicles) that remain spawned, causing the simulator's performance to deteriorate significantly. I will work to resolve this issue.

- With the help of the **subTest** feature newly introduced in version 3.4 in Python unittest, I was able to clean up the test significantly and make it much more elegant. Instead of having repetitive code for testing each gear, I was able to format them into subtests that run in a for loop. Furthermore, this allowed me to run the function that destroys the world for each run of the loop instead of having a separate function for each gear. The test now runs successfully and does not cause the simulator's performance to deteriorate.

- **ADB_Command_Test (Test id = 24)**
  - The purpose of this test is to test whether the button presses on the steering wheel send the appropriate commands to the Android head unit through the ADB interface. This test will be focusing on basic commands, another test will be created for launching applications from the steering wheel.

  - The reason I am simply testing whether the buttons are sending commands properly opposed to seeing whether the command performs the appropriate function on the Android side is because these simple commands do not return anything on the ADB interface, so it would be difficult to make tests for. As such, I will be focusing on the button-to-command relation for this test.

  - Found that this test has the same issues as the previous test where multiple external button presses cannot be simulated at once without destroying the wheel. As such, the same structure as **Shifter_Test** was adopted.

  - After making the necessary modifications, this test is now running successfully,

- **ADB_LaunchApp_Test (Test id = 25)**
  - The purpose of this test is to see whether the steering wheel buttons launch the appropriate application on the Android head unit.

  - Since launching an app returns something in the ADB interface, this can be tested to see whether the command functions properly on the Android side.

  - Used the same method as before to simulate external button presses to launch the apps, and used the ADB's **adb shell ps *packagename*** command to see if the apps have been launched on the android side, as this command returns nothing if the app is not running, or the pid if its launched.

  - Using the pids, I was able to have the test detect whether specific packages (the applications being launched) have an associated pid or not.

  - After some work, the test is running successfully.

- **Notes from brief meeting with supervisor (IMPORTANT):**
  - He said he'd check if he has any existing Jenkins framework related files from the previous capstone team

  - He said, for now, he wants to generate some data from the Carla simulator. Some examples he mentioned is I could drive around, and collect some data from that in the form of a video, logs, charts, stuff like that.

  - The kind of data I can collect for example are speeds, average speeds, braking, the amount of throttle and brakes I'm using, collision data, driving over broken lines and such, and have timestamps for each in a log of some sort.

  - I could also maybe generate performance data as well, data from the tests I created and their runtimes and stuff like that, data from the hardware peripherals as well?

  - He said he'd send them the data I generated and then if they need some sort of specific data, they can tell us like maybe they want some data on autonomous driving functions or stuff like that.

  - **He said for now, do a bit of both, like what I'm doing right now with making tests and all that, while also generating data from all the stuff I mentioned above.** I was thinking I could alternate, like one day normal stuff, next day generate data, and then keep switching. So for today I'll continue what I was doing, which was generating tests for the new features I added.

  - In the future we can talking about designing our own Jenkins-based CI/CD framework for the bench

  - He was also saying that we may have to move the physical bench somewhere else. It'll be in the same lab, but maybe it'll be in a cubicle or some other area in the lab. Just something to think about, maybe I have a suggestion for a place in the lab.

- Tomorrow I will begin the process of generating data from the Carla bench, and the day after, continue making tests for the following:
  - Test the CAN messages
  - Possibly test the CAN attacks? (may not be possible as that's an entirely different executable)