**Friday, December 1st, 2023**

I continued setting up the software on the Jetson. However, the remote access I configured on Wednesday didn’t work, so I had to set up the Jetson with a monitor again. Currently, the way I am trying to perform remote access is:

z

Jetson connects to Hotspot, receives both LAN/Internet connection

Wi-Fi Hotspot from my computer

Fairfax (Internet access)

However, although the WiFi hotspot from my computer was working yesterday, it is not working today. I have tried rebooting my computer and the Jetson, forgetting the network on the Jetson, restarting the Wi-Fi radio on the Jetson and my computer, however, my computer is not exposing Internet access to the Jetson at all. This is a problem because if the Jetson does not have internet access I cannot download anything on it.

While connecting to the Fairfax Wi-Fi seems like the obvious solution, I found early in the year that the Fairfax/FCPSGuest Wi-Fi networks do not allow computers to see other computers on the network – thus I would not be able to remotely access the Jetson. (Getting the trusted certificate on Linux is tricky but doable, however).

I talked with Sonny and Akshat on how they are remotely connecting to their drone. They are using a desktop with a Wi-Fi adapter which allows it to connect to the drone. I think it may be useful to get a Wi-Fi router or access point so that I don’t have to rely on my computer as a router. As a temporary solution, I tried plugging the Jetson into Ethernet so that I would at least have internet access. However, the Ethernet port connects to the SysLab’s internal network which is not very helpful. I tried updating my computer to fix the issue but it did not work.

**Monday, December 4th, 2023**

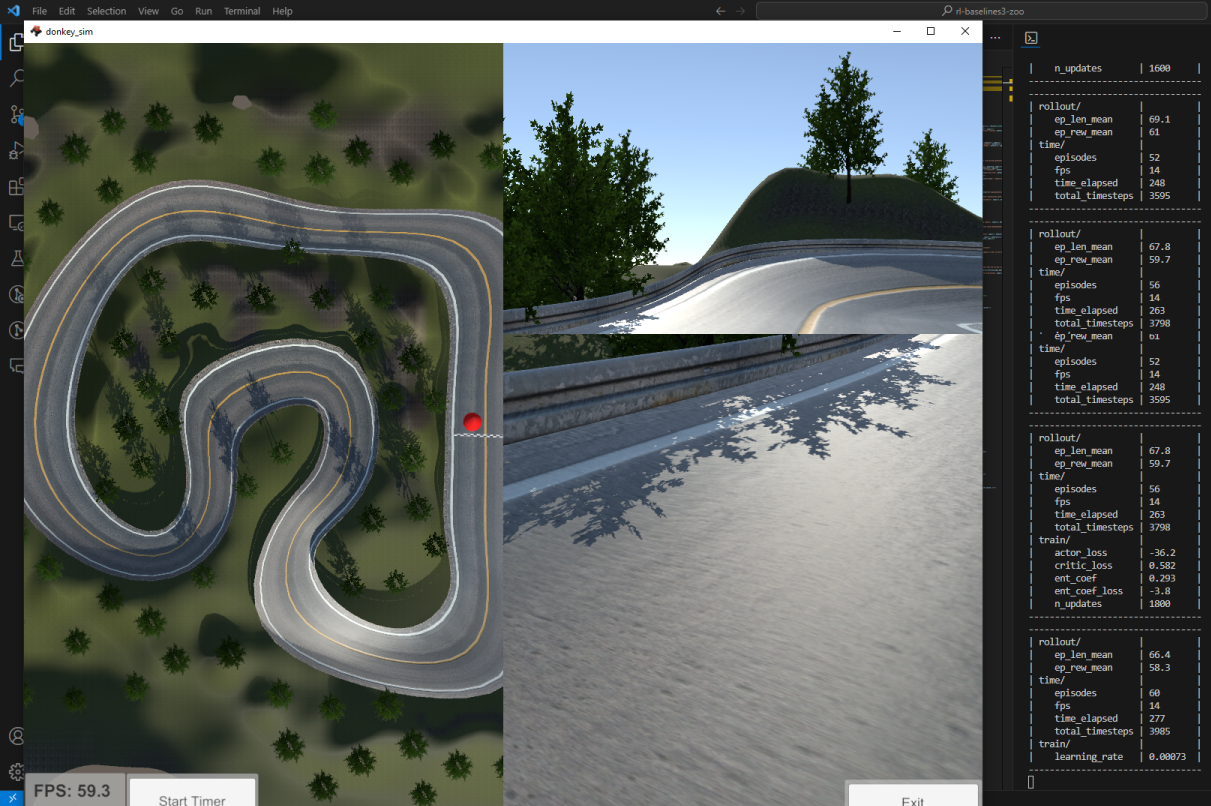
I prepared a purchase order to get the car running un-tethered. I included a Wi-Fi router for remote access, a new motor controller (so that I don’t waste any further time with the old ones we have), and an antenna.

I then worked on getting data streamed with the camera. The API for the camera is the same regardless of what platform you’re on, so I connected it to my laptop to save time. I verified I could use OpenCV to stream the camera feed. I was also able to view the data from the camera’s built in IMU (which includes an accelerometer, gyroscope, and magnetometer). This may be a useful supplement to the RPM values I get from the VESC motor driver for estimating the car’s motion.

**Wednesday, December 6th, 2023**

Shreyan laser-cut a new mounting plate with the correct dimensions and attached the camera, Jetson board, and USB hub to it.

Since I decided to not spend further time on the physical car until the new ESC arrives, I turned back to the final version of my reinforcement learning algorithm. Previously (around October), I used a CNN to extract features which were passed through PPO. After millions of timesteps of training it was successfully able to navigate the track. Obviously, the final result of my project is going to be training the car in real life, thus it needs to be more sample efficient. To summarize what I’ve said in previous journal reports, I plan on using an autoencoder to separate the process of learning an abstract representation of the road from the actual process of learning a good driving policy.

I followed the tutorial (<https://www.youtube.com/watch?v=ngK33h00iBE&list=PL42jkf1t1F7dFXE7f0VTeFLhW0ZEQ4XJV&index=1>) from one of the maintainers of the Stable-Baselines (SB3) library (which is one of the most popular open source reinforcement learning libraries for Python). The rl-baselines3-zoo repository (<https://github.com/DLR-RM/rl-baselines3-zoo>) has pre-made implementations and tuned hyperparameters for a few environments including my car simulation. I logged in remotely to my computer at home which has a more powerful CPU and an Nvidia GPU so that I could run the training more efficiently. It took a while to match up the correct OpenAI Gym, SB3, and PyTorch versions to the specific version of CUDA that my GPU has. However, I did get an example working using the TQC algorithm. (The visualization looks weird because I’m logged in remotely, so the screen updates very slowly):

I couldn’t find much about the TQC algorithm online other than the paper (it is very recent). From what I can tell, it is a variation of Twin Delayed DDPG (TD3). TD3 uses 2 Q-functions and performs policy updates at a lower frequency than Q-function updates. It also adds noise to actions in order to increase exploration early on in the reinforcement learning process. The improvement of TQC over TD3 is that it predicts a distribution for the value function instead of just one value. I’m not sure why this is helpful, but it does better on the standardized RL benchmarks, and it is cutting edge research.

I talked with Dr. Torbert about the status of the parts I ordered. He said that the first purchase order from last week would most likely arrive early next week. The new motor controller and router will either come at the end of next week or after Winter Break.