RIDWAN HOSSAIN
100747897

2022/05/30

- **Note:** Upon start up, a new error appeared throwing an index out of range exception for the jsinputs array, indicating that there was an error with initializing or parsing the inputs from one of the joysticks, either the steering wheel, pedals, or gear shifter.

- After inspecting the code and some manual testing, it was discovered that the gear shifter and steering wheel had swapped indexes in the array of joystick inputs. This had never happened before, but now seemed to be permanent. In order to remedy this issue, I went through 'testbed.py' and swapped all references of event.joy == 1 and event.joy == 0. I made sure to put the comment "**index out of range**" next to the statements where the swapping occurred. It was also included next to statements that may need to be swapped in case this ever happens again, which may be unlikely since this is an extremely rare occurrence.

- I will remap some buttons as a result of the joystick index swapping, and afterwards, begin to address the next steps as discussed with the supervisor this morning:
    - Download MATLAB on the Linux machine and see if roadrunner is included. If not, inform the supervisor.
    - Try to get the computer communicating with the Android infotainment system, such as opening a music app

- **Buttons remapped:**
    - Handbrake has been mapped to the left circular button on the right hand side of the steering wheel, code has been added to hold state information such that the handbrake does not have to be held down and can simply be pressed once to be activated and another time to be deactivated.

    - The top circular button on the right hand side of the steering wheel now toggles the simulator's camera angle

- Began installing MATLAB and Simulink on the Linux machine, and in the meantime, researched how to communicate to an Android device (the android infotainment system) through Python. The following source was found: https://itnext.io/how-you-can-control-your-android-device-with-python-45c3ab15e260

- **It seems that the current license for MATLAB does not include the Roadrunner software. It was not present in the list of softwares to download when installing.**

- Installed the ADB developer tool on the Linux machine

- Inside the **platform-tools** folder, used the **./adb devices** command to begin running the daemon server and obtain a list of connected android devices.

- After cross referencing the Android head unit's serial number, and the serial number visible after using the **./adb devices** command, it was found that the serial number of the Android head unit is **SGWOOFVOO7NJRO7H**. This was done using this source: https://www.xda-developers.com/install-adb-windows-macos-linux/

- Extra documentation for the ADB can be found here: https://developer.android.com/studio/command-line/adb

- Installed the **pure-python-adb library** on the Linux machine, which will be used to to interface with the ADB and our device.

- Resolved error where module was not found even through module was installed (needed to use '**sudo pip3** install pure-python-adb')

- Was able to utilize the pure-python-adb library to connect to and communicate with the Android head unit. Made a python script to change the volume on the head unit in 2 second intervals for testing purposes.

- **A list of commands to control the connected Android device can be found here:** https://forum.xda-developers.com/t/q-adb-input-keyevent-for-long-press-on-power-button.2063741/#post-64890206

- Researched how to launch an application using the ADB interface. Found this source: https://stackoverflow.com/questions/69644744/how-can-i-launch-an-android-app-on-a-device-through-python

- Looked into how to obtain the package name of an Android application.Found that the Linux command **adb shell pm list packages** lists all installed packages on the connected Android device.

- For testing purposes, made a script to launch the chrome application through the adb interface. Was able to launch the chrome app successfully. Next step is to install spotify on the Android head unit, obtain it's package name, and launch that app through the ADB interface.