Scott Dudley

**Sprint 3 Research:**

***GRPC Networking: (8 Hours)***

At the end of Sprint 2, our group decided to step away from Django for our networking. During our Sprint 2 presentation, Professor Patel suggested using GRPC for the networking between our desktop app and the robot cars. Most of my responsibilities for Sprint 3 were for making sure the main application worked, using GRPC for the networking. GRPC is an RPC created by Google that works with multiple programming languages. We decided on a Python server for our Linux Raspberry Pis, and a Visual C# client for our Windows desktop application. At first, I researched how to do both sides of the connection, server and client, to understand how GRPC worked, but then we decided on splitting the responsibilities, where I worked on the client side, and another teammate, Alex, worked on the server side. Neither of us had ever used C#, so Alex and I both created our client, PiCarConnection.cs, while I worked on the UI files we had ready from Sprint 2 to handle client operations and adjusted some coding to avoid the application crashing. I tested the connection between the server on one of the raspberry pis and the client on my desktop and made sure that everything connected correctly. We were able to demo the networking working in our Sprint 3 presentation. All of the sources that I used for this section, as well as the links to the files that I worked on from github are provided below:

GRPC Client/Server Research:

<https://grpc.io/docs/tutorials/basic/go.html>

<https://alexandreesl.com/2017/05/02/grpc-transporting-massive-data-with-googles-serialization/>

<https://ops.tips/blog/sending-files-via-grpc/>

<https://dzone.com/articles/grpc-sample-in-c-and-python>

Client C# file:

<https://github.com/sramocki/SeniorProjectRobot/blob/master/Source/RobotClient/RobotClient/PiCarConnection.cs>

UI C# files:

<https://github.com/sramocki/SeniorProjectRobot/blob/master/Source/RobotClient/RobotClient/MainWindow.xaml.cs>

<https://github.com/sramocki/SeniorProjectRobot/blob/master/Source/RobotClient/RobotClient/Registration.xaml.cs>

***Meeting at TARDEC Base: (2 hours)***

We met with the staff of TARDEC who we’ve been connecting with through email at their base. We were able to receive our hardware from our TARDEC contact, Dominic. While we were able to get a tour of the base, we also discussed our requirements and use cases with Dominic. We decided to change the scope of the project slightly to fit a few requirements that Dominic wanted to see and omitted some of our stretch-goal requirements.

***Building the PiCar: (9 hours)***

Since, we were able to get the hardware from TARDEC, we decided that Christian and I would each take home one robot to build the kit. I downloaded NOOBS onto a mini-USB drive that we got with the raspberry pis. This loads Raspian, the default Linux OS for raspberry pis, onto the pi. I was able to configure everything from the raspberry pi end, using my setup.bash script. While everything loaded, I built the robot kit, which included mechanical and electrical assembly. We found out that the kit needed a certain kind of battery, 18650 batteries. I went to several stores and only found one that sold them, but overpriced them, so we ordered four of them, two for each robot, along with battery chargers, so we wouldn’t have to get more. We got them from Amazon, but the shipment got lost, so we got the batteries the night before the Sprint 3 presentation. I wasn’t able to configure the servos before getting the batteries, since it needed to have enough power. When I received the batteries, I started the servo configuration but found out that one of the servos was overheating, so I immediately stopped the configuration and ordered replacement servos that I received after Sprint 3 ended. For Sprint 4, we will have the robots completely built, configured, and moving. The manual for building the robots is provided below along with the link of the setup.bash file I worked on and used from github:

PiCar-V Manual:

<https://www.sunfounder.com/learn/download/U21hcnRfVmlkZW9fQ2FyX1YyLjBfZm9yX1Jhc3BiZXJyeV9QaV9QaUNhci1WXy5wZGY=/dispi>

Setup Script:

<https://github.com/sramocki/SeniorProjectRobot/blob/master/Source/Setup/setup.bash>

***Total Work for Sprint 3: 19 hours***