Stephen Marhefka & Matthew Borbonus

Luis Olivera

CS 1980

25 October 2019

Midterm Progress Update & Presentation

Our project is doing rather well despite our initial expectations. Due to the nature of this project, we have encountered multiple problems that have barred us from progressing as quickly as we’d have liked, but so far, we have overcome them. This project is not exactly what we originally anticipated back when we wrote the proposal, but nonetheless we have been proceeding forward with optimism. For our progress report, we shall go over our current implementation of our software, some changes that we made, and our current problems and concerns with our project.

While we are far from finished, we have managed to implement our first set of features rather smoothly. Our biggest concern, programmatic control of the GPS, was initially a little rough and we were unsure of how to approach this. Due to user privacy concerns, control of the GPS was harder than first imagined, and our choice of solution requires the phone to be rooted and the application to be a system app (Android prevents developers from accessing these features in a user application). This is due to the edit settings permissions that our solution requires to execute the code. While this is a minor inconvenience, it is not a big concern since we are more interested in obtaining a proof of concept than a production ready app, an issue we have discussed with our point of contact. The other features we intend to implement, such as the Google Map interface as well as displaying user location and speed, were much simpler than we originally thought. Android Studio made it very straightforward to add these features because, it can automatically generate a page that has the Google Map interface preloaded (though you do have to get your own API key from Google), and the Android API provides simple methods for obtaining and displaying the user’s location, as well as acquiring the user’s speed. One other API feature that we implemented is a listener that detects location changes by the user. This will prevent us from having to write code to either detect this ourselves and can allow us to more easily obtain updates to the users current location and velocity.

We have made a few changes since the start of our project that are different than originally outlined. One main difference is how we will measure the power consumption of the device. Our original plan to measure power was to either to find a system call or hook into the kernel to monitor consumption, but after meeting with Dr. Mosse, he will be providing us with a wall-to-device power meter that will give us exact measurements of how much power the device is pulling. With this method we will have the ability to see exactly how much power we are saving by turning the unit off, and can see how big of a hit, if any, turning the device back on will have, and will allow us to tune our algorithm accordingly. Naturally, since this measures all power consumption, we will have to establish the base parameters; measuring over a period of time to see the default power draw with the GPS on, with the GPS off, and how much it draws turning the unit off and on many times in succession to monitor the power hit of repeated turn ons and offs to account for variability that exists out of our control. After we have obtained this data, we will then use our app for its intended use, and perform extensive testing along chosen predefined routes. This testing and comparison phase will allow us to conclude whether we are saving an appreciable amount of power or not.

One of our ongoing problems with the project currently, is learning the APIs necessary to perform the actions that we need. While we have both done minor amounts of mobile development, we haven’t done a considerable amount in the past to make us proficient in this area. Additionally, every step we make in our project depends entirely on the previous step functioning as intended. For example, our entire project at the start of the semester hinged on seizing control of the devices GPS programmatically. If we had not been able to develop a solution to this problem, nothing else we could have developed would have mattered. Thankfully, we were able to devise a solution to the problem that was somewhat understandable, but with the added cost that our app only works on a rooted phone (to enable this power state toggling. Currently, we are investigating how we can tie our algorithm into existing GPS navigation systems (such as Google Maps), and if there is an API that allows us to integrate routes into our system. If we cannot construct a simple GPS navigation system in our application, we will have to settle with statically constructed routes to test the battery consumption and algorithm efficiency.

Ultimately, our project seems to be going in the right direction, with some minor changes having to have been made. The possibility of finishing the entire project seems well within our grasp, but any complications of navigation and establishment of routes have a small possibility of preventing full completion of the project. We have theoretical countermeasures to help combat this, such as using static routes, but hopefully it will not come down to using them at all. Otherwise, we seem to be progressing at a reasonable rate and Dr. Mosse seems to be somewhat pleased with what we have accomplished so far. In the end, we hope that the data we produce during the testing phase of our project will provide results that show our idea is actually capable of saving battery power. Perhaps other companies, maybe even Google itself, will be interested in this idea if it turns out favorably!