Stephen Marhefka & Matthew Borbonus

Luis Oliveira

CS 1980

22 November 2019

Smartphone GPS Duty-Cycle for Saving Battery in Navigation Project Final Paper

**Introduction**

In today's modern world, many people rely on mobile navigation systems to help them get where they need to go. While these systems are incredibly helpful, they do come at the expense of a significant impact on device battery levels. Thus, creating a system that could reliably help users get to where they’re going, along with the added benefit of a measurable decrease in power consumption, was something that we aimed to conceptualize. Our project intended to implement an algorithm in the form of an Android application to reduce power consumption on mobile devices by systematically turning on and off the GPS unit of the device as needed while being used for navigation.

The idea behind our project, was given that the user is currently travelling on a section of road with no expected turns or merges in the short term, we can turn off the GPS unit during this section of the route to reduce the energy load of the device, while ensuring the GPS unit is back on just before the user finished traversing this section of their route. At this point, once the GPS has acquired a signal, we would check to see whether the user has actually arrived at the point or not. Via testing of power consumption of the device and comparing the total wattage consumed with the GPS both on and off, we have determined that (at least in our testing devices) a significant amount of battery life can be saved by utilizing a system like this. This project was overseen by Dr. Daniel Mosse whom we met with frequently throughout the course of the term for direction and advisement.

**Development Process**

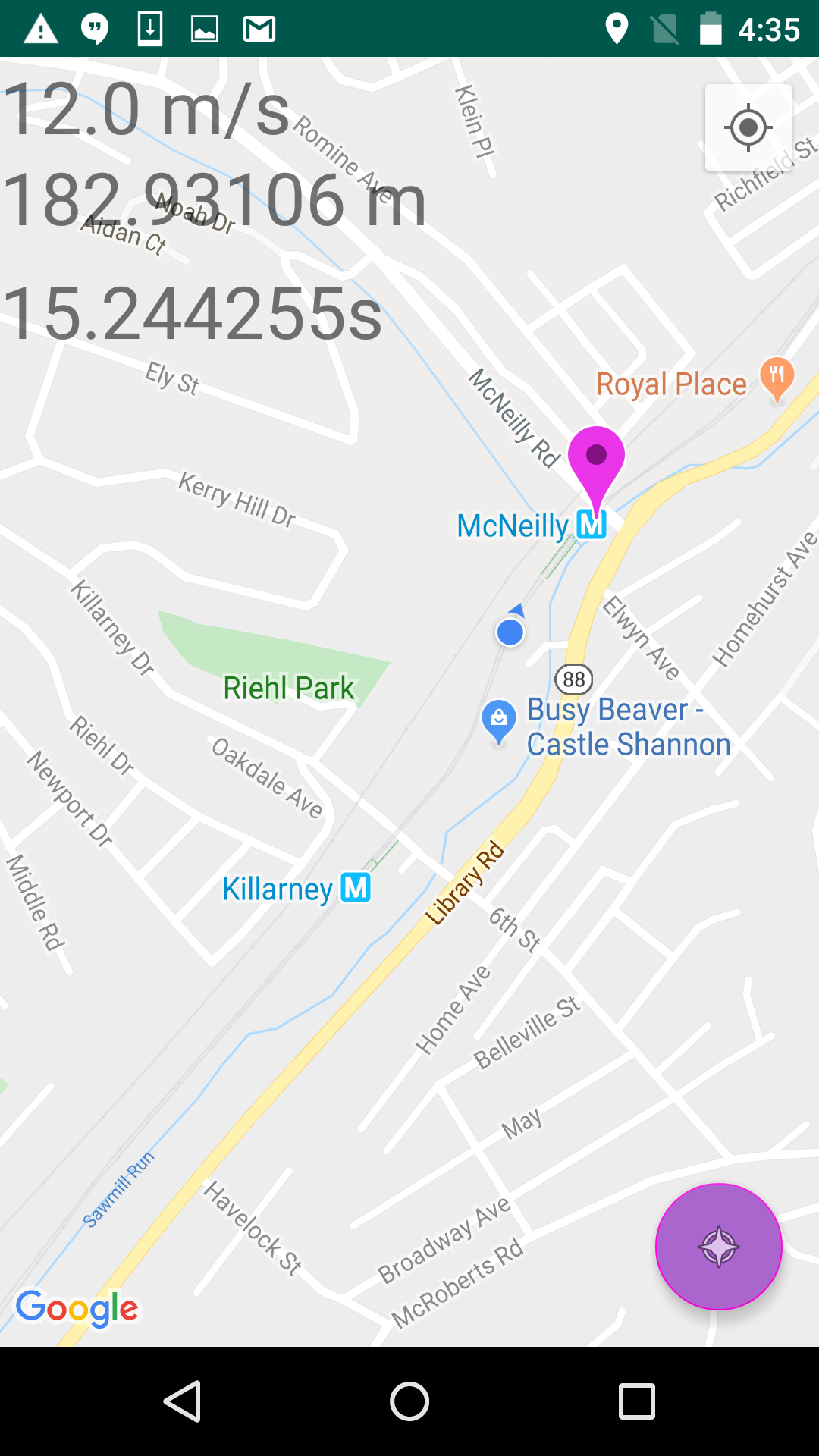
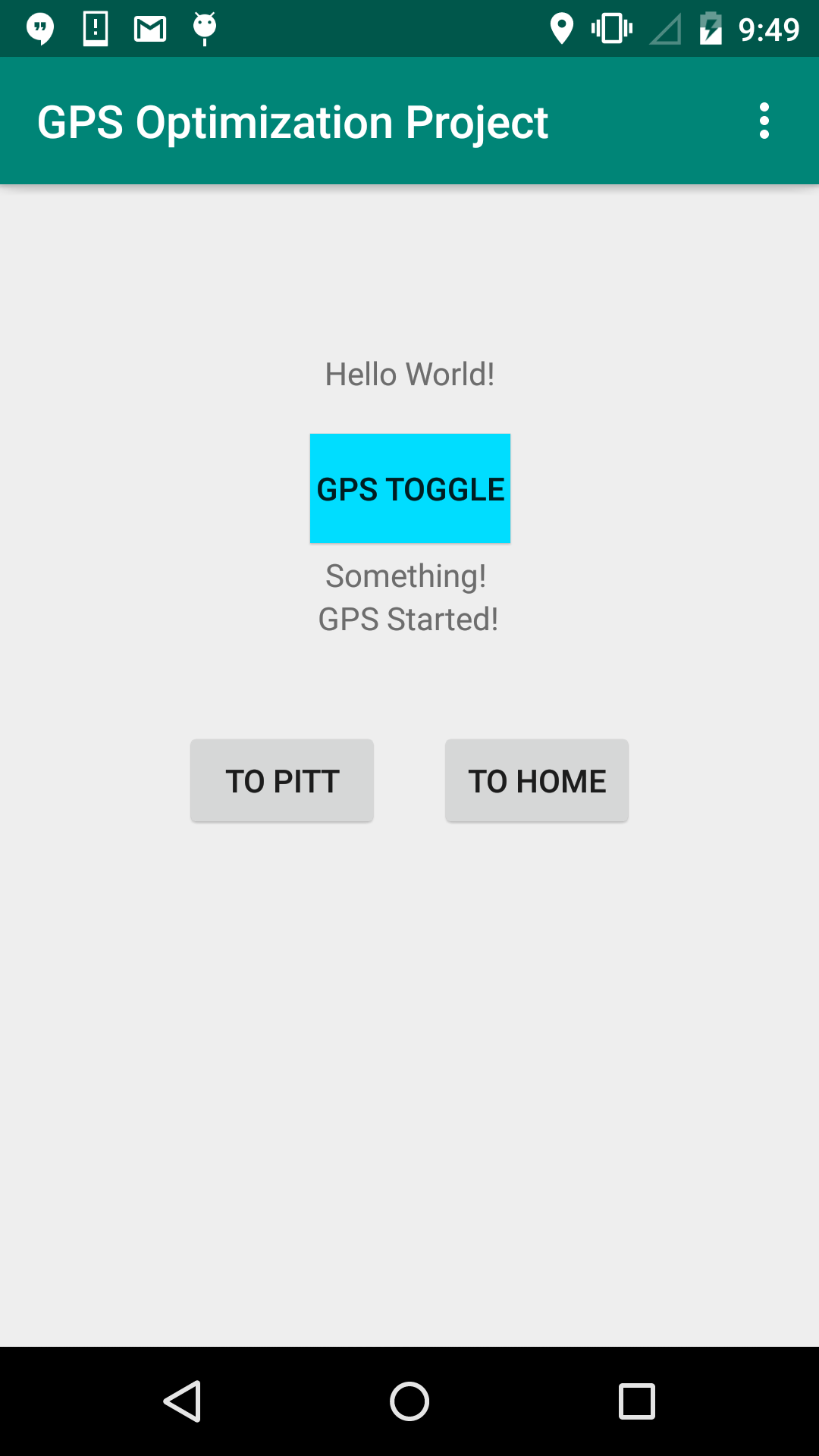
As stated above, our project was implemented in the form of an Android application. This choice was made primarily because the testing devices we had access to (Google Nexus 5) were Android devices, we both had experience programming in the Java programming language, we didn’t have access to Macs or the necessary software for iOS development, and Android development can be done for free, whereas development targeted at Apple devices is not. We also relied heavily on the Google Maps API for the presentation of our application in order to provide a basic interface to the user while they are navigating. Therefore, given the above reasons, development on the Android platform was the clear choice.

Our progress during the first half of the semester was rather slow but we picked up the pace of our development significantly during the latter half. We would meet in Hillman Library often two to three times a week to discuss the next phases of the project and to work on the actual code base for our application. We both conducted various types of tests on our own time such as tests of the application itself (which we built a logging system into), timing tests to determine the amount of time necessary for the GPS of our devices to acquire a satellite signal, and power testing to allow us to compare the power consumption of the device with the GPS on versus the GPS off.

**Application Requirements**

At its core, our application had the goal of being able to determine the amount of time required to travel from the user's current location to a destination (i.e. the next turn in the user’s route) and subsequently turn the GPS off for the duration of that time in order to reduce the power consumption of the device. Upon expiration of the timer, the GPS would be turned back on and would check to see if the user is within the vicinity of their destination. If the user is not within the acceptable radius of the next point, the GPS will stay on until the user has arrived. The time calculation we used takes into account the instantaneous velocity of the user provided to us via the Android API, along with the distance calculated between the user’s current location and their destination, where this distance calculation is also performed by the Android AP as well. In simpler terms, this would be , *v* being the instantaneous velocity and *d* being the distance from the user’s current position to the edge of the radius of the next intersection. Due to a lack of time, our application currently utilizes only statically entered routes, rather than a more robust system where a user could enter their desired destination and the route used could be generated by some third party such as Google or OpenStreetMap.

These static routes are hard coded into the application by explicitly stating the latitude and longitude of various points along the user’s route. The main page of our application allows the user to select ‘To Pitt’ (travelling from their home to the university) or ‘To Home’ (travelling from the university to their home). Depending on a string that we set prior to installing our application on a phone, we can set who the specific install will be for, therefore we can support multiple to and from routes for different people. Upon selecting either of these options, a list of Location objects will be built using the provided coordinates and this list will be iterated through as the user progresses through their route. The user will then be taken to a screen with a map where we overlay the users speed, distance to the next point, time required to reach the point in seconds, and a drop pin to show the location of their next intersection.



**Mentor Relationship**

To update Dr. Mosse on our progress and to get direction for the next steps to take, we would have biweekly meetings with him in his office in Sennot Square. We would review and demonstrate the things we had accomplished over the past two weeks and outline any issues we had. He would provide possible solutions to any issues we were facing, along with suggestions and guidance as to what our next two weeks should focus on and what features we should work on implementing next. Dr. Mosse was reachable by email and responded to any concerns or issues we had in a timely manner. We also worked briefly with some graduate students doing research with Dr. Mosse to understand how to use the USB power meter we were using as part of our data collection. They were very polite and were also reachable by email and responded in a timely manner as well.

**Technical Specifications**

To build an application on the Android operating system, we utilized Android Studio, which is a modified version of the Jetbrains IntelliJ IDEA integrated development environment, which is made by Google in order to make managing and building Android applications easier. Android applications are built using the Java programming language, and since we both have experience using it, we didn’t have to learn any other languages in order to get started. We did, however, have to familiarize ourselves with the Android API and the Google Maps API in order to implement the functionality we needed, but otherwise, no other frameworks or languages were needed.

In order to work properly, our application required that our testing devices were rooted in order to allow us to modify the settings needed to enable and disable the GPS as required. We utilized a rooting application called KingRoot in order to root the phones, and an application called Link2SD which allowed us to make our application into a system application to gain access to these settings. All of the phones we were using needed to be running Android Lollipop in order to work properly because KingRoot would only be able to successfully root the phones if it was running a version of Android Lollipop (5.x).

When creating our logging system, we debated the possibility of creating a method to wirelessly extract the log files from our testing devices, but due to our unfamiliarity with the Android API and lack of time, we were unable to create this type of system. Instead, we chose to store the logs in a structured text file on the device and manually retrieve them once we had physical access to the phones. However, the directory this file is stored in can only be accessed by users with root access. Thus, every time we want to extract the log file from the phones, we have to use a command-line tool called the Android Debugging Bridge (ADB) to first move the log file to an accessible directory on the phone. Once the file has been moved to this separate directory, it can then be transferred to our computers. This is due to the fact that the commands used to extract files from the phones do not have root access. The extra steps for log file retrieval are a bit of an inconvenience, but it is more than manageable since we are not doing it constantly.

**Testing**

The actual testing of the application has been an interesting experience thus far. Instead of just writing the algorithm and running just the application tests, we had to first gather some base data to get readings to compare against. First, we ran some timing tests to determine the average time it takes for the GPS to acquire a fixed signal, as well as calculating the standard deviation to determine how much leeway the GPS needs to reacquire a signal under normal circumstances. A separate application was constructed to record this timing data. We found that on average, the GPS signal took about 7 seconds to acquire, with a standard deviation of about 18.5 seconds**.** The fastest it ever acquired a signal was in about 0.06 seconds, while the slowest was about 144.88 seconds.Secondly, we also used a USB power meter to determine the average power usage during different states of the phone such as screen on and off as well as GPS on and off. With these power readings, we could determine how much power we save on average when running the app. Unfortunately, the power meter we are using does not store an internal log and thus to take constant readings over a period rather than just taking the ending reading of milliwatts per hour, it has to be connected via Bluetooth to a computer. While this is technically doable if someone is driving or taking public transportation, this is not exactly an easy (nor safe) task to do when biking or walking. Our only alternative at the moment is recording the battery level in the GPS logs, which is a different reading from the power trials and is unfortunately not as accurate (i.e. only provides battery level to the nearest percentage) but will help provide rough estimates. We performed a total of 40, 10-minute tests with the phone in 4 different configurations: screen on GPS on, screen off GPS on, screen on GPS off, and screen off GPS off. We were then able to compare these totals and the difference seen between having the GPS on versus having it off is significant. Comparing the 20 tests with the GPS on against the 20 tests with the GPS off, we saw on average a 32% decrease in power consumption.

Unfortunately, testing the GPS itself for timing and location accuracy is not exactly easy. The GPS signal by itself with no assistance from cellular data or Wi-Fi can be unreliable in certain environments. It would rarely work inside of buildings unless close to windows that allowed an optimal signal, but not all windows would allow an optimal signal. Sennott Square, for example, was in general not a GPS friendly location, which allowed for minimal demonstration when meeting with our mentor. Hillman Library, the primary meeting grounds of the developers was a similar scenario. In general, the GPS signal worked primarily outside, so often we had to walk outside to test a sizable portion of our code. The other factor we had to work with was the inaccuracy of the location data of a GPS signal. While the speed reading tends to be accurate, the location is inaccurate, sometimes to several meters. Even when the phone is completely still, the listener we used for detecting location changes would continuously update the location data to be slightly different every second. We had to account for these fluctuations in our code for if conditions based off of distance, such as larger radii for our intersections than what should be needed, and never using exact distance comparisons, instead using a margin of around 5 meters or more to determine if there was a significant change in location or if the difference is just a fluctuation.

Our final tests involve running a version of the application that will only be logging during the duration of the user’s route and not modulating the GPS power state, alongside a version that will be performing GPS power modulation. We will also be looking to compare the battery levels between the two phones and also review the accuracy and precision of our algorithm. The accuracy in this being how close to the destination do we turn the GPS on, and if we overshoot, by how much. The precision here being the number of intersections we miss, if any. This will be indicative of how useful or not our algorithm is. If we consistently overshoot or severely undershoot with our algorithm despite adjustments, then our application would not be worth it and the power readings not worth much. After running our tests, preliminary analysis seems to showcase that there is roughly a 30% difference in GPS constantly on versus GPS modulation, which corresponds with the results of the power tests.

**Further Refinements**

As with all projects, not everything goes exactly as expected and plans change, and our project was no exception to this. There were things that we were intending on doing that we didn’t get around to, whereas other areas we had planned on doing a certain way, ended up being accomplished some other way. Given more time or if this project were to be improved by future capstone courses, there are some areas where further refinements and work could be done to improve the application.

Additional testing, particularly having a more rigid and thorough testing framework when it comes to testing the functionality of the actual application would be helpful to rule out anomalous routes and provide more data to draw conclusions from, as to the effectiveness of our system. We performed two rounds of power testing (coming to a total of 40 tests), which clearly demonstrates that power consumption is lower when the GPS is disabled versus when it’s on. Although there are a few anomalies and unexplainable results in our power tests, they clearly show an overall difference between GPS power states, however due to potential differences between the devices these tests were performed on, along with inconsistent ambient light levels affecting how bright the phones displays were, this lead to more inconsistent testing data.

Again, due to a lack of time to explore other options, we were unable to use a dynamic routing system and instead had to utilize a system of static routes as described earlier. While this system was sufficient for our purposes, changes to routes and having different destinations was not easily achievable and became quite cumbersome to change since each point on the route had to be manually entered.

**Conclusion**

Ultimately, this was a fun project to work on with all parties involved. For our first dive into mobile development, we learned a lot about the Android API and a little bit about the Google Maps API. If we had more time, we could have explored the Google API further or switched over to a different API that doesn’t charge for using certain features. The potential showed by this project thus far is intriguing, perhaps intriguing enough that some company like Google might be interested in the idea.