Paul Atkinson

Sean Penney

Aaron Egger

CS 461, Senior Design

12/06/14

Progress Report

Group 24

APE Industries

The first step in our planning involved creating our technology review and problem statement. In our technology review, we decided that we would: use Android Studio over Eclipse or Netbeans; use Git as the backend over SVN or manually uploading and downloading tar balls; and support Bluetooth classic with the SPP profile as our bluetooth profile of choice. Later, we decided to expand our bluetooth support to Bluetooth Low Energy as well, as it has power saving options. Writing the problem statement gave us a better understanding on what is required from the client and the different components of the project. At this point in time we were not 100% sure what bluetooth profiles we would be able to support, so we started doing more research on it. We also started to figure out what test devices we would want to use.

Next we discussed bluetooth details with the ECE team. We decided that the app should support bluetooth low energy in addition to bluetooth classic. Bluetooth low energy makes sense for this project since we will be sending small amounts of data. The GATT profile will be used as the basis for bluetooth low energy devices. A list of fiveteen requirements was created for Don, which included both CS and ECE requirements. One of these requirements specified the bluetooth hardware we would test the app with. The Android devices will be a Moto X and a Nexus 7. The devices that we will connect to will be a bluetooth keyboard, bluetooth gamepad, and an Arduino. The keyboard and gamepad will use bluetooth classic. The Arduino will have a bluetooth low energy shield, that will allow for various bluetooth profiles. We will be using the RedBearLab bluetooth shield, which uses the Nordic nRF8001 Bluetooth Low Energy IC. This is compatible with my Arduino Mega 2560 and Nexus 7. There is open source Arduino code that we can use to implement bluetooth communication using the RedBearLab shield. According to the documentation, minimal work will have to be done to get the hardware setup. There are two pins that need to be set in the code, and jumpers on the shield need to be set appropriately. The bluetooth low energy code has example projects that implement different bluetooth profiles, such as the HID profile for keyboards or the heart rate profile. One project that would be useful as a starting point is the “ble\_uart\_project\_template.” This project simulates a UART connection over bluetooth. Nordic has a corresponding Android app that works with the ble\_uart\_project\_template. This Nordic app can be used to verify all the hardware is working, then we can work on creating our own Android code that connects to the Arduino.

More requirements were specified, related to the user interface and functionality of the bluetooth devices. We decided that the interface should have a section that displays data coming back from the device. The app and devices will have automated responses, so that timers can be set on the phone and someone could then go out of range. Research into how plugins may work in our app was done by Aaron. There is some information in Android forums about how to setup plugins, however, we will have to look into that more. Tasker was brought up as an example of how we may want to structure the plugins.

During the week of thanksgiving break, we finished our client requirements document and had it signed by our sponsor. We started work on both our preliminary poster and the elevator speech that we would have to give the next week. Sean ordered his Arduino bluetooth shield, Aaron ordered a bluetooth keyboard and Paul ordered a bluetooth gamepad. Sean started to look into bluetooth profiles for Arduino.

We created 24 requirements for the CS part of the project. Requirements were divided into the Android GUI, website, bluetooth, and misc. We named ourselves APE Industries. This name is derived from the integration of our last names: Atkinson, Penney, and Egger. It is also true that *we don’t monkey around.* We finalized our requirements this week. For the framework we will provide skeleton code for the creators of the modules to fill in. For the user interface we will be using a snap to grid drag and drop interface.

This we week wrote and presented our elevator speech. The preliminary poster was due this week which we completed and turned in. Sean’s bluetooth shield arrived, as well as Aaron’s bluetooth keyboard. Paul’s gamepad has yet to arrive, but should still arrive before programming on the device begins.

Our plan is to start on the actual coding of the project over winter break. Currently, all we have set up is a Github repo. Sean will set up the project for us to access and we will all figure out how to connect to our respective devices. The bulk of the project is going to be done over winter term though.