Mark Ha '13
Adviser: Brian Kernighan
Fall 2012 Thesis Checkpoint Report
11/12/12

"Easy Android" - Developers' Tool for Android Beginners

The name, "Easy Android," is subject to change, but hopefully it aptly conveys my goals with this project. To summarize briefly, I am creating an application that simplifies process of developing an Android application for the user. Normally, in order to create an Android application, developers need to have an assortment of tools before they can even start, and after that, they need to familiarize themselves with the Android application infrastructure. My goal is to reduce that overhead significantly, including both the software and knowledge required to create an Android application.

One of the earliest decisions I had to make with this project was who my audience was. On the one hand, I could create a tool that increases the productivity of engineers who are already familiar with Android, and maybe have made an application or two already. At the other extreme, I could target people with no programming experience whatsoever who want to start making Android applications. Ultimately, I decided to target users with at least some general programming experience, and hope to create something that enables such a person with a clear design or specification to create their own Android application with relatively little effort.

In terms of the concrete work I have done so far, I outlined in my presentation that the first steps would be to develop the core functionality and complete a barebones end-to-end version of my tool. My rough deadline for the end-to-end version was the end of November. I started off focusing on the "coding" part of the application - that is, replacing actually coding by the user with simple command line input, with optional arguments for customization. As an example of how this works, my application currently allows for the use of Button, TextView

(normal text), and EditText (editable text form) objects. So if a user types "button -height 100 -width 50" and then asks for the corresponding code ("print") for example, my application outputs the contents of a Java file that the user can literally copy and paste into an Android project, build it, and put it on to their phone. So, though this will of course expand to more than three types of objects and will have to include more features, the very basic functionality of this is complete.

Next was the building, compilation, and packaging process, which is what most editors do for a developer. Combining Android tools with Apache Ant software, creating and building an Android project is possible from the command line, though perhaps not so convenient. Currently, my application simplifies that entire process for the user, so that two commands, "create" and "build", give the finished product. Therefore, this step is almost complete, but there are a couple unresolved problems to work on. One is that while I have the functionality to add files to the created project, I do not currently automatically integrate those files into the created project, because Android has a special file (AndroidManifest.xml) where project files need to be declared. The other main problem before achieving true end-to-end functionality is that currently I use the Android tools and Apache Ant software to do this step, which is less than ideal, since my original goal was to eliminate all the overhead associated with installation and setup. However, because the software is already relatively lightweight, I may prioritize other things going forward.

To summarize, I should be able to wrap up the end-to-end functionality by the end of November, though I may assume that the user downloads the Android and Apache Ant software. The next step is to develop the "coding" process, and offer more options for users using my application, so that they can do more than just create a few buttons. But in terms of my rough timeline that I set at the beginning of the semester, I am fairly on track.