Loadable Kernel Module Final Report

Software Systems - Spring 2014

Lyra Silverwolf

1) Who was in the group?

I worked with Kai Austin, Rachel Biniaz, and Aidan McLaughlin as a loose confederation or a support group. We are presenting together and we read the same material/ did the same exercises, but we did them at different paces and we did not create one cohesive product.

2) What subject(s) are did you explore?

I explored Objective-C mostly in the context of iOS development, as well as iOS development and design principles among many other things. I also got a lot of experience using Xcode (the IDE for iOS and OS X).

3) What resources did you use? Books, web pages, class syllabuses, exercises?

In order to begin learning Objective-C syntax and structure I raced through codeschool.com's free Objective-C course in the first few days. After getting familiar with the language, I moved on to *Head First iPhone and iPad Development* in order to get started on learning how Objective-C is implemented in iOS. I made it through the first 5 chapters (of 8) from that book and learned most of the fundamentals of designing for iPhone and iPad. The final 3 chapters cover databasing and use of built in hardware like the camera and GPS.

4) What was your learning goal? What did you plan to get out of this module? How would you know if you achieved your goal?

In this LKM I set out to learn about what tools Objective-C has to offer and how they can be used in iOS. I am interested in the app space and want to move into iOS development. I thought that I would know that I had achieved my learning goals if I was able to create an iPhone application using the sample code from the *Head First* book and other resources I could find.

5) Did you achieve that goal?

Resoundingly yes. I learned a lot about Objective-C from the codeschool.com tutorial, and realized that it is a lot more different from C than I expected. All method calls are executed by messages which are super long lines of information encased in square brackets with tons of colons sprinkled in. Then, while sprinting through the first ⅔ of the *Head First* book, I learned most of the fundamentals of iOS programming and made 3 different apps. I made sure to do all the exercises and create all the code I encountered in the book to solidify my newly gained knowledge.

In the end I was able to make an app that used a plist full of data to display information about records in a hypothetical record store. This application had a table view as its main view, and a detail view containing more information about the record available if any of the entries of the table were clicked on. I can now add more data quite easily, and I will be doing a demo of the app in class for the demo period. During this demo I will show the class how easy Xcode is to use and now awesome iOS development is by adding another view connected to a button, on the fly. Preparing this demo (that directly deviated from the code provided in the book) proved to me that I could put together what I learned and prove to myself and the class that I learned a lot about programing for the iPhone.

6) What did you learn?

- The basics of Objective-C
- iOS design principles
- How to get around Xcode
- How to debug with Xcode
- Model-View-Controller design paradigm
- Using the Twitter API
- Classes: Interface and Implementation
- Message passing
- Hierarchical data
- Building Data Access Objects
- Implementing segues between views
- The control that apple has over apps

7) How did it relate to Software Systems?

Without the knowledge of the C language I gained in software systems, I would not have been able to dive headfirst into programing for the iPhone like I did. Objective-C is more different from C than I expected, but they are still fundamentally similar on a low level, and many concepts such as pointers and character arrays leak over from C. I think that this project was really a great way for me to be able to tie together a lot of what I learned in a high level way that I can imagine myself doing a lot in the future. I'm not sure how much more bare C I might write in my life, and I might not be programming for embedded systems when I leave Olin, but mobile prototyping is something I have a passion for. iOS development is the piece that will keep me thinking about software systems and the relevant topics far into the future, hopefully.

8) What would you have done differently if you could do it again?

I would probably have interspersed the codeschool.com Objective-C tutorial with the Head First book, because the speed at which I went through the Objective-C tutorial may not have been the most conducive for my long term retention of the language. However, the Head First book starts off really slowly and doesn't get to the meat of Objective-C until chapter 2.5, so I would have feathered the two learning resources together if I had known better. However, I still think I learned a lot from this project, and it is something that I intend to keep doing long after I turn in this report.