

iOS Mini-App: UVa Bucket List

Richard Dizon & Emma Kopf
CS 4720: Team Mutsu
University of Virginia

October 7, 2016

1 Instructions and Information

App was developed in *Swift 3* using the *xcode* IDE off of the iPhone 6 emulator. Just open the app and it should run normally!

2 Lessons Learned

Richard: I found it really important to try to start the project off with a strong foundation so outside of lecture, I watched a few of the tutorials on iOS 10 development on Lynda.com and ended up implementing some of the functionality from the video examples, namely the sections on view data flow.

I was really interested in how the Swift language works, closely resembling a weird mix between Java and Scala (ironically enough, two of the main languages supported by IntelliJ IDEA which is the sister IDE of Android Studio). It was cool seeing how it had aspects of functional programming thrown into the syntax and the incorporated *lets* and exclamation points and question marks to make it more well-typed. It took getting used to, but it really forces developers to think more carefully about how data gets passed around.

It's interesting to me to see how well built and structured iOS architecture is; all the icons, fonts and labels are standardized, lists look the same as any stock iOS app and views just transition so cleanly natively. I definitely

felt a lot more comfortable developing in this environment than on Android even though I had never attempted iOS development, nor do I own an iPhone (though admittedly, after this experience, I'm starting to consider getting the new MacBook so I can use xcode on my own instead of having to go to Rice all the time) and feel pretty inclined to move forward with iOS for the final project.

Emma: I hadn't done any Swift development before this class, so I had to watch and read a lot of tutorials. Similar to Richard, I watched the Lynda tutorials which were really helpful; however, looking up things for help while coding was a bit difficult with the update to iOS 10. A lot of syntax from source code was older and had to be fixed to work with the current version of Swift. Fortunately, Swift recognized what was happening and gave suggestions about what the syntax has been changed to now.

Swift was definitely different from many of the languages that I've worked with before; the only time I've had experience with MVC models were with Ruby on Rails which I've only used briefly. It was really useful to see the storyboard and how data was passed from one view controller to another. For example, creating a new scene in the storyboard helped me visualize exactly what's happening with each and every call on a segue. The storyboard also helps visualize the app as a whole. It's also great for testing to make sure that you're getting what you expect to see.

I definitely learned to appreciate the MVC models that once frustrated me while working with Ruby on Rails. Swift had a bit of a heavy learning curve for me, but definitely made me think about every individual action that was happening and what it would take to make it all come together.

3 Sources Used

1. Lynda Tutorial - iOS 10 App Development Essentials 1: Create Your First App
2. <http://stackoverflow.com/questions/29734954/how-do-you-share-data-between-view-controllers-and-other-objects-in-swift>

3. <http://stackoverflow.com/questions/30773529/open-new-view-controller-by-clicking-cell-in-table-view-swift-ios>
4. <http://stackoverflow.com/questions/30773529/open-new-view-controller-by-clicking-cell-in-table-view-swift-ios>
5. <http://stackoverflow.com/questions/25867439/get-the-indexpath-from-inside-a-uitableviewcell-subclass-in-swift>
6. <http://stackoverflow.com/questions/25678373/swift-split-a-string-into-an-array>