

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.

2 Lessons Learned

Emma: Previously, I didn't have experience developing in Android because I took 2110 after the Android Project was scrapped from the syllabus. I had to get used to a lot of new aspects of coding/Java that I never dealt with before, such as XML files, intents, activities, fragments, and etc. Through this project, I better understand the underlying aspects that go into creating something as simple as a button on a screen, and all the different steps it takes to go from having said button pop up on the screen, have a click registered, and transfer data. The trials and errors I went through while working on my app taught me a lot about streamlining development as well as writing generally good, understandable code. It was definitely interesting working on this project starting with nothing. In 1110, making the game wasn't too difficult because all of the style/formatting was given to us. We just had to implement some code that we already wrote in lab or for homework.

I also had to do a lot of work getting to understand GitHub and how to push/pull from remote repositories. Previously, I only ever uploaded com-

pleted projects to GitHub and never went back to expand on them. We definitely had a lot of issues with pushing/pulling branches and Richard had to do a lot of fixing/merging.

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.

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>