

Alek Palmersmith  
June 29, 2014  
Music Technology Innovation  
Berklee College of Music in Valencia

## **APP-LICATION OF WHAT I'VE LEARNED IN MTI**

### **Introduction**

My culminating experience is the intersection between iOS app development and music. I have focused on learning to program and design applications, experiencing first-hand what it takes to create an app that looks and feels professional. I will break down my learning process into the main areas of focus, with a particular emphasis on the rapid prototyping of specific features, creating an app-a-day. Although my focus has been more on the learning process, rather than the final product, I do have a number of functional prototypes, with each representing a separate initial exploration of a concept to be further explored. I also explored potential for collaboration with app development through my work with Gerami Groover.

### **Description of the Culminating Experience Project**

App-a-day is what I started calling my rapid prototyping method for app development. During this stage, I take one simple feature, concept or idea, to focus on exclusively for an entire day. This clears the mind of unnecessary roadblocks, and allows for pure creative output. The concept of daily creative output in a particular area of focus is not uncommon in other fields. For example, aspiring songwriters benefit greatly from writing a song a day, music producers gain similar experience by listening to and analyzing a new album every day, a mixing engineer mixing a new track each day, etc. With software programming specifically, I initially felt challenged by the new vocabulary and highly complex nature of designing something from scratch with a written programming language. My way of dealing with this was to think of simple and fun apps to create, which could later have a variety of outcomes: (1) completed and made into final products, (2) included as a specific feature in another app, or (3) discarded entirely. With any of these three outcomes, I gained valuable experience.

A brief summary of the types of prototypes I created in this process include: recording audio with device's microphone, playing back local audio files, viewing multiple videos on the same screen at once, working with libpd as main audio engine, generating and displaying random numbers on screen, using device's motion detection, creating a synchronized playback of multiple audio stems, recognizing user gestures to control sound parameters, using built-in physics engine and animations to give user interface elements more life, using open source frameworks to do more complex tasks such as displaying an audio waveform and customizing its appearance, creating musical quizzes that test the user's ears, building a 12-tone series generator, and many more similar experiments.

Prototyping is wonderful for generating ideas, yet at some point I needed to focus on producing a complete app. This led me to study app building from a wider perspective, to take into account the various stages of this process, and the many roles of those involved in an app development team. I discovered that professional app development follows a process similar to what is familiar to me, music production. Similar to pre-production work, there is the equivalent of business requirements and functional spec writing to determine what the app will do, and design sketching for a rough idea of what the app will look like. Then, similar to production stage, there is development and user interface design. And finally, just as there is a post-production stage of music production, app building has a stage of beta testing and bug fixing.

The majority of my time has been spent in the development stage, learning the language and mindset of a programmer. Just as many experienced producers acknowledge the importance of the pre-production stage, I have come to a similar conclusion with app development. Neglecting this initial planning stage can be quite counterproductive. I was also naive at the beginning of my project to think that I could get away with focusing my energy almost entirely on programming, with design almost as an afterthought. I could not have been more wrong in my approach! Design is something that really goes hand in hand with programming, which is one of the ways Apple clearly stands out amid its competition.

Learning to program apps means literally learning a new language, Objective-C in my case. Technically this is a language between the programmer and the computer. However, just as importantly, being fluent in programming languages also allows for interesting collaborations

with other people who “speak” the same language. One example that comes to mind is “Music Hack Days” that go on all around the world, in which people are grouped together, with the common goal of a useful and inspiring creation at the end of 24 hours. On a smaller scale, I have noticed that it is not uncommon to see Meetup’s scheduled in metropolitan areas, where app developers get together to work on creative projects outside their day job. I would love to be a participant in these groups, as I see it as an opportunity for creative collaboration without the external pressures of normal work environment, similar to a jam session in which musicians often participate.

My first experience with a collaboration of this kind has been my work with Gerami Groover, and her project, “Revelación Latina.” She is passionate about Latin musical styles, and has gathered lots of interesting material on the subject. Our collaboration has been focused on making this content accessible on iOS app format, for the greater public to experience. Our plan was to create an educational and entertaining iOS app about the musical styles of Flamenco and Tango. I took the role of programmer and co-designer. My goal was to present the content in a way that takes full advantage of iOS device. Due to struggles with communication and time limitations, the app is only in early prototyping stage. I think this project can serve as the basic skeleton of a future app that Gerami could create with a team of designers and developers, as she now has the content ready.

During my app-a-day prototyping stage, there were some apps that I continued to go back to, adding features, improving user interface, and fixing errors. Some worth mentioning here (and shown in Culminating Experience video) are “Split Mix”, “Comprestures”, and “Echstasy”. “Split Mix” is an app that synchronously plays back audio stems of a full mix, each independent of each other. This allows the user to cut or solo any element of the mix, at any time. Additionally, I set up a compressor, delay and reverb, that can be toggled on and off. The second app, “Comprestures”, was based on something that I had contemplated for a long time, though I experienced renewed inspiration upon reading “Two Dimensional Gestural Control of Audio Processing” by Tom Wilson and Steve Fenton from AES E-library. In this paper, engineers were surveyed on the most effective two-dimensional gestures to manipulate audio parameters, such as compression, with multiple examples. I decided to take their excellent work, and bring it to life

on iOS, to be able to test the gestures while listening to their affect on the sound. “Echstasy” is highly similar to “Comprestures”, but focuses on delay parameters instead of dynamic range compression.

I draw inspiration from a number of released music apps, though there are three that I particularly enjoy: “The Orchestra” by Touch Press, “N Jorge Drexler” by Wake App, and “The Bermuda Tapes” by WhyHunger. What I love about “The Orchestra” is the use of multiple camera shots of the same orchestral performance. This gives the user the freedom to watch any or all of these videos as they wish. I think the same concept would be highly effective for a popular music recording session as well, with a camera on each musician. I also really enjoy the commentary given by conductor and musicians about the pieces being performed and the historical context, as well as the orchestral instrument learning section. “N” is an app with a simple layout, containing three original songs by the Uruguayan songwriter, Jorge Drexler. The app gives the user control over one specific element of each song. With one song, user picks the lyrics to be sung on subsequent phrase, another gives user choice between various singers per phrase, while the other song gives the user control over the musical arrangement. I find this musical arrangement interactivity the most effective, as you can bring in and out a variety of instruments and sections as the song is playing in real time, everything from guitar and vocals to a full orchestra. I find the “The Bermuda Tapes” innovative in its highly original illustrative design and interesting storyline and gameplay that puts the music into context. This app really takes the user on a journey in such a seamless way that the technology becomes transparent.

### **Innovative Aspects of the WORK**

Innovation is often combining existing elements together into a new context. Throughout the app development process, my goal has been to build on the work done by others, while contributing something original. For example, I took Apple’s built-in gesture recognizers (pinch, pan, tap, swipe, etc.), with open source audio library, The Amazing Audio Engine, and the work done by AES researchers, to prototype two dimensional gestures in an app form. In a similar way, I took the idea of temporarily soloing sections of a musical ensemble from “The Orchestra” and “N” apps, and used musical example of Radiohead’s released stems of the song, “Nude” combined with basic audio effects control. I find this methodology to be effective in moving

music and technology forward in a way that respects and adds value to the original creators, while also adding my own voice.

### **New Skills Acquired**

Before the Spring semester, I had no prior experience in software design or any programming language whatsoever. Acquiring this new skill has given a new dimension to my work, and, as I mentioned earlier, could lead to interesting future collaborative projects. My work with design has also been a major area of growth for me. Visual art is not one of my strengths, so creating designs was certainly a challenge. In the process, I gained a deep respect for the mobile app designers who manage to create such beautiful, simple, and user-friendly interfaces.

I would like to briefly go over some of the tools I used in app building process. For initial design sketching and prototyping I used the iPad app, Paper. These tools made it easy to draw freely on screen, trying out ideas, while conveniently constrained by the actual size the elements will eventually be contained within. For creation of app icon designs, launch images, and other design elements, I used Sketch and Pixelmator. For easy creation of the multiple size resolutions of these elements (retina, non-retina, iPhone 4, iPhone 5, iPad Mini, iPad, etc.) I used the app, Prepo. I used Promotee to create beautiful automatic product shots of my app screenshots. Github and CocoaPods were both used as efficient ways of downloading and using open source frameworks in my app, like The Amazing Audio Engine, mentioned earlier. Apple's built-in Terminal app, though daunting at first glance, proved very useful in the process as well. A majority of my time was spent actively writing code within Xcode, OSX's Interface Development Environment.

Although more information can be found in footnotes, I would like to mention some of the important resources I used along the way for assistance in learning programming and design. Chris Ching, of CodeWithChris.com, has wonderful tutorials on iOS app development, as he breaks complex programming terms into simple analogies that anyone can relate to. Stanford's free course offering (through iTunes U) on app development was useful in better understanding software design best practices. The class, App Choir, gave me guidance in the app development process, especially in how it built on my previous knowledge of Max/MSP to relate similarities

and point of differences. Lynda.com courses on Objective-C, iOS development, and Sprite Kit also proved to be useful assets in the process. Another important resource is Apple's own documentation and example projects. I also used Meng To's suggestions on iOS 7 design with his website design+code. Towards the end of my culminating experience journey, I discovered CocoaPods, which manages numerous open source frameworks that can be added to your app with little effort. These include interesting extensions for video, audio, animations, and more. Even when not using these frameworks in my apps, I found it useful to skim through their code to see how they were achieving their end result, similar to how a mixing engineer might look at an open mix session done by someone else to see how they used the tools to achieve their sound.

Another skill I have acquired is the ability to dissect the individual components of an effective app. Beyond just noticing whether the user interface or functionality is effective or not, I now have a better idea of the types of approaches they probably used to achieve their result. For example, the use of animation, physics engine, auto layout, color scheme, the type of data model set up, when and from where the information and files come, and the order in which elements get loaded onto screen or in the background. This sense for piecing apart the work into its individual components is good for analysis and reference. A musician uses this skill to hear the specific harmonic, melodic, or rhythmic elements of the individual instruments of an arrangement, and a mixing engineer does a similar separation at times to note specifics that might go unnoticed by the general public (the compression on the snare drum, a high pass filtering of the vocal track, or subtle reverb tail on the acoustic guitar).

### **Challenges, both anticipated and unexpected**

A number of my challenges ended up being great learning experiences. At times, I would lose hours trying to fix a programming error, only to find out it was a simple mistake that could be easily avoided in the future. File management was a challenge, especially when working with external frameworks. Using CocoaPods for keeping these frameworks organized helped immensely. I found iOS user interface design to be particularly difficult, as I constantly struggled with simplicity versus control. Often I would find myself putting too many elements on the small iPhone screen, and then having to find ways of taking unnecessary elements away, or animating something in and out only when needed. Another challenge was organizing and communicating

with Gerami on the collaborative project. As I have found out, it is nearly impossible to start programming something without having the content that will fill the app. All decisions on design, gesture controls, amount of screens, among other similar decisions, can only be made once it is clear exactly what material we are working with. As we were on a tight deadline, I wasn't able to produce the app by the deadline as promised. With my time split between final paper, culminating experience video, my presentation preparation, and the app, it was difficult to deliver any of these with the high quality I would have preferred.

### **Future Ramifications and/or plans for the WORK**

As mentioned previously, my attention has been focused on the learning process, more than the actual products I have created during this time. I have taken the time to study with a number of excellent learning resources, and in my prototyping stage, I have learned from many initial mistakes. I think a clear and concise summary of my app-a-day journey would be a helpful resource for others wanting to learn to develop iOS apps. I have considered posting this as an article on Medium.com. A reworked paper detailing my experience working with the gestures for manipulating audio parameters would be beneficial to the AES community, so I have considered writing a short AES e-library article to be able to detail my discoveries.

I also would like to create an app based on some of my previous prototypes that could act as a template for musicians to present (and slightly customize) their material in a new way on iOS device. As presented by Stephen Webber's talk on the Album of 2025 and discussed in our Music Technology Innovation Seminar class, we have yet to see a standard new medium for music distribution. I would like to experiment with this idea in the form of iOS apps. One simple aspect of this can be seen in my "Split Mix" prototype explained earlier. I think releasing stems of an audio mix, and allowing for some alternate changes, as I explored with that app, could be an interesting component of the new album. Also explored in other apps, I think the addition of visuals, such as blurred album artwork as background texture, UIDynamics physics engine and UIAnimation, give life to the interface, using gestures to control the audio material in a new way. Multi cam video shots of the music videos, behind the scenes commentary from songwriters and musicians, and more, all give more to the user's aural and visual experience. These are just simple additions to the already established inclusion of lyrics, photos, credits, and more.

As I mentioned earlier, a major advantage of learning to program is the potential for collaboration. Not only in events such as Music Hack Days, and MeetUp events with other programmers, but also through sharing my apps in open source on sites such as Github and CocoaPods for others to use, and even to improve upon. I think everyone wins in this situation; it is a positive step forward for music-related apps, and hopefully for the music industry as a whole. I will continue to build experimental prototypes, and complete apps that I want to share, and even build an app or two for OSX Yosemite (to mix it up a little from just developing mobile applications).

## **Conclusion**

As with any skill, learning the craft of developing mobile applications is a lifelong endeavor. I have worked hard this semester to gain a solid foundation in the art of programming, while also gaining a lot of other skills in the process. I have learned the need for a beautiful and effective design, the importance of prototyping, the potential for collaboration with others, open source software sharing, and renewed respect for all the designers and programmers who bring the best apps to life. As I continue to write apps, I become more fluent in the languages of computers, whether it be Objective-C, or the newer language, Swift. Just as I would write down a melody I am hearing onto staff paper, the same is now true with app ideas and a programming language. Writing apps is just another way of taking my creative ideas and writing them down, to be later fully conceptualized into the final product. I look forward to creating apps with others and for others that will inspire forward motion for the music industry.

## **References**

Tyson, Michael. "The Amazing Audio Engine. Core Audio, Cordially." <http://theamazingaudioengine.com> (accessed April 20, 2014).

Wilson, Tom, and Steve Fenton. "Two Dimensional Gestural Control of Audio Processing." <http://www.aes.org/e-lib/browse.cfm?elib=17142> (accessed January 10, 2014).

Ching, Chris. "Code With Chris - How To Make an iPhone App." <http://codewithchris.com/> (accessed February 10, 2014).

Brinkmann, Peter. Making Musical Apps Real-time audio synthesis on Android and iOS. Sebastopol: O'Reilly Media, 2012.



"The Orchestra." iPad app by Touch Press. <http://www.touchpress.com/titles/orchestra/> (accessed September 1, 2013).

"N by Jorge Drexler." <http://wakeapp.com/n/en/> (accessed August 15, 2013).

WhyHunger. "John Lennon: The Bermuda Tapes." <http://lennonbermudatapes.com/theapp.php> (accessed March 15, 2014).

"Nude Remix." RSS. <http://radioheadremix.com/nude/> (accessed November 1, 2013).

"Pick up your tools.." Paper. <http://www.fiftythree.com/paper> (accessed May 25, 2014).

"Bohemian Coding Sketch 3." Bohemian Coding. <http://bohemiancoding.com/sketch/> (accessed April 25, 2014).

"Pixelmator." <http://www.pixelmator.com> (accessed January 8, 2014).

"Prepo." Mac App Store. <https://itunes.apple.com/us/app/prepo/id476533227?mt=12> (accessed June 1, 2014).

"Screenshot Maker Pro." App Store. <https://itunes.apple.com/us/app/screenshot-maker-pro/id673545271?mt=8> (accessed June 15, 2014).

"Promotee | NetwalkApps." <http://netwalkapps.com/app/promotee> (accessed June 16, 2014).

"Build software better, together." GitHub. <https://github.com> (accessed March 10, 2014).

"CocoaPods.org - The Dependency Manager for Objective C." <http://cocoapods.org/> (accessed June 8, 2014).

"Kuler." Adobe Kuler. <https://kuler.adobe.com> (accessed April 25, 2014).

"Developing iOS 7 Apps for iPhone and iPad." iTunes U. <https://itunes.apple.com/en/course/developing-ios-7-apps-for/id733644550> (accessed February 15, 2014).

"Developer Video Courses and Tutorials from lynda.com." lynda.com. <http://www.lynda.com/Developer-training-tutorials/50-0.html> (accessed January 5, 2014).

"Design Code: Learn iOS design and Xcode." <http://designcode.io> (accessed June 5, 2014).

"Apple Developer." iOS Dev Center -. <https://developer.apple.com/devcenter/ios/index.action> (accessed January 1, 2014).