Background Research

The idea is to create an application that simulates writing traditional calligraphy for various languages. However, before I could even start thinking about an application, I needed to understand the basic programming software that went into developing it.

It was interesting to learn that Cocoa is the name of Apple's "object-oriented application programming interface (API)", and it is one of the many parts that make up the tree of macOS programming (Cocoa). Programming languages and development tools are the branches. I originally only wanted to learn what languages were used in creating applications, but was surprised to learn that there were 2 languages that work with Cocoa. Objective-C was first (Muqri), then about 30 years later, Apple came out with Swift. Swift is essentially easier to understand and less complicated than Objective-C, and you can use development tools such as Xcode playgrounds to learn how to use Swift as well as prototype and create learning environments (Premaratne).

After sifting through a few more articles, I came upon the macOS Human Interface Guidelines. This was very eye-opening to understand UI from Apple's point of view. The guide brought up the mental model - the concept of an object or experience that people carry in their heads. A mental model is made up of metaphors, which are subtle links to a user's "mental model of a task or experience". They included the example of how file folders are a metaphor for how we want to store away things in our life, and also how the computer can stretch the capabilities

of things in reality (MacOS). The article brought up that motion metaphors should stay with people's expectations because it can cause confusion. However, fresh, new trends in interface design catch can peoples' attention - it gives a refreshingly interesting application while still staying true to the definitions of good design.

For my topic choice, an application that allows you to practice hand-lettering and learn calligraphy for various languages, I wanted to be able to understand the basics behind the back-end of an application before understanding the code for mimicking hand-lettering onto a digital screen. But since I am not a professional coder, I researched ways to achieve this through HTML5, javascript and jQuery. The code did not seem as complicated as I assumed, and I recognized the use of functions, loops, callouts, and events (Zipso.net). It utilizes event handlers such as mousedown, mousemove, and mouseup.

Works Cited

"Cocoa Event Handling Guide." Introduction. Apple Inc., n.d. Web. 09 Feb. 2017.

"MacOS Human Interface Guidelines: Design Principles." MacOS Human Interface Guidelines: Design Principles. Apple Inc., 2016. Web. 09 Feb. 2017.

Muqri, Mohammad Rafiq, Dr., and James R. Lewis, Mr. OBJECTIVE-C VERSUS JAVA FOR SMART PHONE APPLICATIONS. Diss. DeVry U, Pomona, 2012. N.p.: American Society for Engineering Education, 2012. Print. https://www.asee.org/public/conferences/8/papers/3338/download

Premaratne, Pasan. "Should You Learn Swift or Objective-C? The Short Answer Is Both." Treehouse Blog. Treehouse Island, Inc., 13 Sept. 2016. Web. 09 Feb. 2017.

"A Simple Touchscreen Sketchpad Using Javascript and HTML5." Zipso.net. Zipso.net, 23 Oct. 2016. Web. 09 Feb. 2017.