

Culminating Reflection

“Food falling through the gutters are grate.”

The Process

I learned an abundance of new SwiftUI skills through this project, most of which are related to the user interface. For the first half of my time spent developing my iOS app, I was researching and implementing various features that SwiftUI had to offer. These included, but were not limited to: LazyVStacks, animations, onTapGesture/LongPressGesture, offsetting, customizable colours, rotationEffect, using GeometryReader for backgrounds, environmentObject. I gained a greater understanding of the SwiftUI language and how it worked as well. I had only created smaller, personal projects in the language before the Milestones app, and had always felt confused about the restrictions that it posed on my code. When I previously worked with Swift (note distinction from SwiftUI), I had gotten used to using for...in loops and if statements freely. After making the switch, however, I was forced to make dramatic syntax changes and re-learn many concepts that I had mastered from prior experience. Through completing the development process for this app, I resolved many of the unanswered questions I had regarding SwiftUI. For example, I learned that using traditional for loops outside of Views would overload the operating system, which is why an error shows when it's attempted.

In addition to just code, I also learned how to make use of other SwiftUI features. One major improvement I made to the Milestones app was adding background images, as this greatly enhanced the user's experience and engagement. I was initially hesitant to commit to this change, but it turned out that the process for adding images required far less effort than I expected. It was as simple as downloading the desired image, adding it to the assets files of the Xcode project, and implementing the Image() element in the code. I repeated this set of steps for multiple images, and also gained a thorough understanding of how to make adjustments to them within the app. Another topic I looked into was the connecting of different Views. The traditional approach is to simply use NavigationView and NavigationBar to direct the user, but I felt that this was insufficient for my means. NavigationView was slightly unintuitive, and the UI components didn't match the overall “look” of my app. So, I found a solution from an online source that involved creating a Directions() structure. This framework would accept user input through a menu, and guide users to chosen locations by using the viewNum variable. This alternative method was the route I chose in the end, and has yet to disappoint.

I ended up satisfying most of the goals for classes/structures that I set out with. I uploaded MilestonesView, SteppingStonesView, MilestonesApp, MenuContent, AddSteppingStoneView, and AddMilestoneView to my github submission. Out of these, AddSteppingStones view was left intentionally void of code, as everything that I had tried to write for it was crashing my entire program. This problem, although discouraging, could most likely have been resolved if given a few more days of time. The only file from my UML diagram that I didn't create was

MilestoneHistory. Nearing the project deadline, I realized that I had perhaps paid too much attention to the user interface and not enough to functionality. At this point, I had several options: continue to focus on UI, start basic functionality for all missing structures, or complete the functionality for one high-quality structure. I turned to my AGILE method for guidance, and ultimately decided to go with the third option to ensure the quality of my project. This meant that I had to abandon the MilestoneHistory structure, a decision that I don't regret.

From this experience (or ordeal, depends how you look at it), I gained substantial knowledge of proper project and time management. My initial culminating idea was to create a chess training application in Java, which I eventually abandoned due to the complexity of functional chess programs. However, this wasn't before I spent five whole days doing research, which left less time to work on my alternative plan. I made up for this period by quickly completing a second proposal and dedicating a whole weekend to coding in SwiftUI. This change of plans, although undesirable, gave me the opportunity to exercise flexibility in the face of challenges. Additionally, there are many, many steps to building a complete iOS app, something that I learned the hard way. I had to research optimal methods of compartmentalizing my Milestones program in both Xcode and Github, and greatly underestimated the time it would take to finish the most important user interfaces. I was forced to make up for this mistake by micro-managing the time I invested into other project components. This issue of time management grew even more prominent as the deadline grew closer, and it got to a point in the final days where I was setting completion goals for half-hour chunks of time. In the end, though, coding this goal-setting app taught me how to create more feasible goals under time pressure myself.

There was only one major problem that I encountered besides the normal coding obstacles, and that was the issue of connecting my project to a Github repository. Xcode has settings that supposedly make the source control process highly efficient, but I spent a significant amount of time figuring out how to create a remote repository on Github.com through Xcode. Every attempt I made in the beginning resulted in an error stating that the software "Failed to load repository owners". After some trial and error, I realized that the issue had to do with my saved credentials, but not before I accidentally replaced my entire culminating project with a file of the same name (I resolved this highly problematic situation by using my iCloud account to retrieve the recently lost files). In the end, I re-entered my Github info and was finally able to make the initial commit to a local repository. A lesser notable issue had to do with the SwiftUI documentation. The .doccarchive file that I was exporting wasn't accessible through any other means except Xcode and a web server, which meant that I couldn't just upload it to Github and share the link in my final submission. I solved this problem by taking screenshots of the documentation tab in my own version of Xcode and uploading them to my supplementary document.

The Conclusion

Overall, I found this culminating project to be greatly enjoyable, despite not being able to finish my final product. The many hurdles I had to overcome along the way only aided my learning in the end, so I have a positive impression of the journey as a whole. Hopefully, I'll be able to continue where I left off on this project in the future, and apply the many skills I've gained from coding it beyond just high school.

By the way, thank you for your continual guidance over the past three weeks and throughout the semester, it really encouraged me to pursue my ideas to the best of my ability. It's been an exciting and educational few months, and your class was a big reason I chose to pursue computer engineering at Waterloo.

Wishing you a safe and stress-free summer!

- Anthony Chen