Storyboards versus SwiftUI

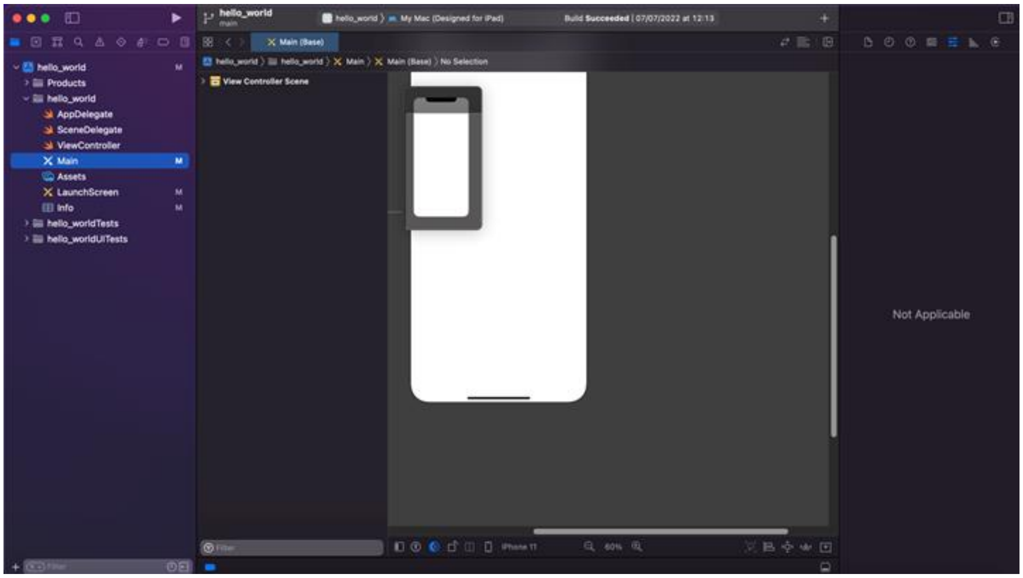
Introduction

An iOS developer today has more options for creating a user interface (UI) layout on iOS than ever before. Most developers are familiar with an interface builder called Storyboard and a toolkit released by Apple called SwiftUI.

Let's explore the differences between Storyboards and SwiftUI and the benefits of using Storyboard for creating UI.

Storyboard

Storyboard is the most popular way to create a UI for iOS applications after its release with iOS 5. By default, any new iOS project created has a **main.storyboard** file included in the generated files.



A storyboard in Xcode is an empty canvas screen that helps you create various UIs for an application. This file allows you to move different types of views such as a button and text on its surface.

With this interface builder, you can create anything from controllers used to manage interactions on the UI to text and image views. You can add controller screens like a **TableView** controller or navigation controller by simply dragging and dropping from the object library.

Advantages

- It's easy to learn, even for beginners.

- With Storyboard you can create a prototype of an app in a few hours. This is a useful method of visualizing the app without having to write too much code at an early stage of development.
- The Storyboard object library has a wide range of Views such as button and text.
- The developer community is large, so it’s easy to get help whenever you hit a roadblock.

Disadvantages

- Source control may be intimidating since the storyboard code is in an XML-type format. It may be difficult to resolve conflicts when mergin code with another developer.
- As the number of screens keeps growing, managing storyboards can become chaotic as there may be too many files.

SwiftUI

SwiftUI was released alongside iOS 13. It is a toolkit released by Apple to help developers create UI in a declarative way using Swift language.

Advantages

- It’s easy to build and targets multiple iOS platform apps (such as macOS, iOS and iPad apps).
- It offers a combination of drag and drop through Xcode and writing code in a Swift file.
- The code is basic and clean.
- SwiftUI views can be easily reused by creating smaller module files.

Disadvantages

- It supports only iOS 13+ and Xcode 11+, so you cannot build apps with lower OS versions using this.
- It is still relatively new, so there aren't as many component options in the SwiftUI as in the Storyboard development kit. To meet app requirements, you must create custom views, such as a search bar.
- There is no seamless way of switching from Storyboard to SwiftUI. This means developers have to continue using Storyboard for existing apps.

- The community is small, and, as a result, there is currently less ongoing support for developers.

Conclusion

After evaluating the advantages and disadvantages of Storyboard and SwiftUI, it is recommended that you choose Storyboard to create UIs rather than SwiftUI, which is still fairly new.