



360|iDev

EVAN K. STONE

LAUNCHING YOUR APP THE SWIFTUI WAY



INTRO: WHO AM I?



WHO AM I?

QUICK BIO...

- ▶ Ancient history: Windows desktop and web development
- ▶ Developing for iOS full-time since 2011
- ▶ Started with Objective-C
- ▶ Swift since 2014



Enough of that.

Misnomer.

**LAUNCHING YOUR APP
THE SWIFTUI WAY**

APP LIFE CYCLE: THE SWIFTUI WAY

Time for a
little
history...



IN THE BEGINNING: THE UIKIT WAY

Xcode 10

The screenshot shows the Xcode interface with the following details:

- Title Bar:** AppLifeCycleXcode10 > Generic iOS Device AppLifeCycleXcode10: Ready | Today at 7:13 AM
- File Navigator:** Shows the project structure: AppLifeCycleXcode10 > AppLifeCycleXcode10 > AppDelegate.swift
- Code Editor:** Displays the `AppDelegate.swift` file content. The code defines the `AppDelegate` class which implements `UIResponder` and `UIApplicationDelegate`. It contains several methods for handling the application's life cycle.

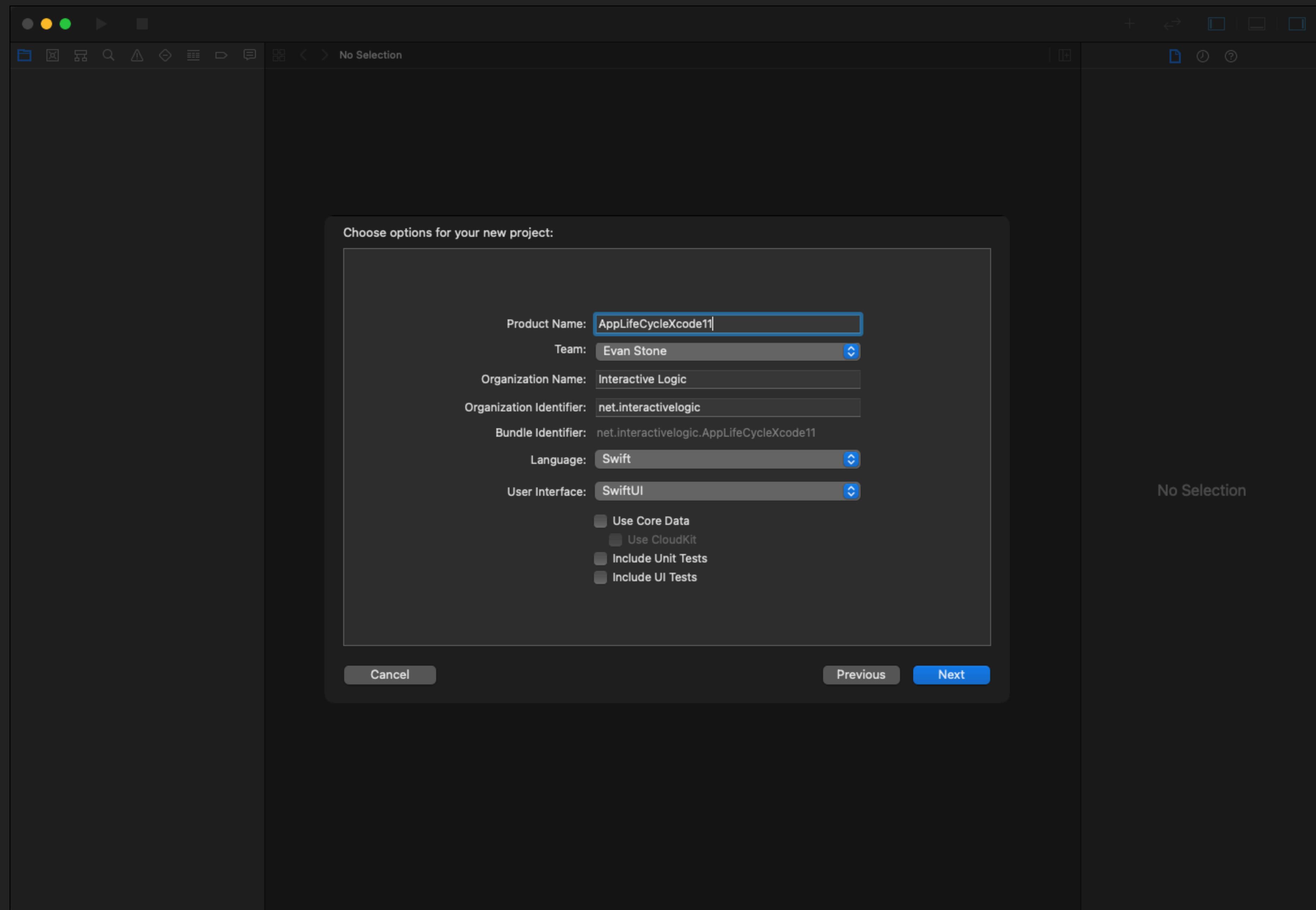
```
8
9 import UIKit
10
11 @UIApplicationMain
12 class AppDelegate: UIResponder, UIApplicationDelegate {
13
14     var window: UIWindow?
15
16
17     func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
18         // Override point for customization after application launch.
19         return true
20     }
21
22     func applicationWillResignActive(_ application: UIApplication) {
23         // Sent when the application is about to move from active to inactive state. This can occur for certain reasons.
24         // Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks.
25     }
26
27     func applicationDidEnterBackground(_ application: UIApplication) {
28         // Use this method to release shared resources, save user data, invalidate timers, and store enough information to restore your application if it is terminated later.
29         // If your application supports background execution, this method is called instead of applicationWillTerminate:.
30     }
31
32     func applicationWillEnterForeground(_ application: UIApplication) {
33         // Called as part of the transition from the background to the active state; here you can undo many of the changes made in applicationWillResignActive:.
34     }
35
36     func applicationDidBecomeActive(_ application: UIApplication) {
37         // Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, during its transition to the active state.
38     }
39
40     func applicationWillTerminate(_ application: UIApplication) {
41         // Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.
42     }
43
44 }
45
```

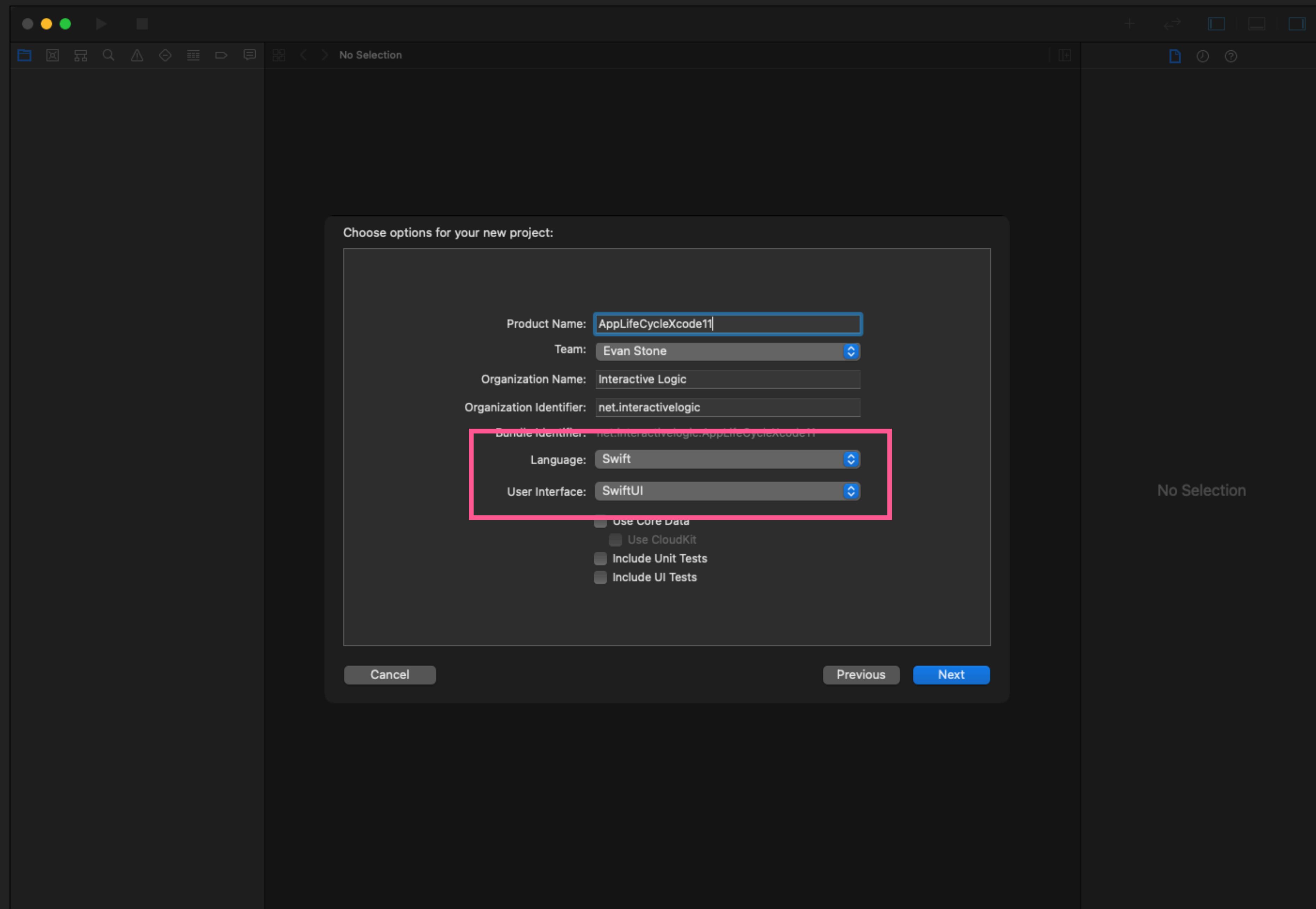
- Bottom Bar:** Includes a "Filter" button and a small icon.

Xcode 10.3

THE TRANSITION YEAR: UIKIT + SWIFTUI

Xcode 11





The screenshot shows the Xcode interface with the following details:

- Title Bar:** AppLifeCycleXcode11 > Generic iOS Device AppLifeCycleXcode11: Ready | Today at 2:39 PM
- File Navigator:** Shows the project structure: AppLifeCycleXcode11 > AppLifeCycleXcode11 > AppDelegate.swift
- Editor:** Displays the contents of `AppDelegate.swift`. The code is a template for an iOS application delegate.

```
1 //  
2 //  AppDelegate.swift  
3 //  AppLifeCycleXcode11  
4 //  
5 //  Created by Evan Stone on 8/17/21.  
6 //  Copyright © 2021 Interactive Logic. All rights reserved.  
7 //  
8  
9 import UIKit  
10  
11 @UIApplicationMain  
12 class AppDelegate: UIResponder, UIApplicationDelegate {  
13  
14     func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {  
15         // Override point for customization after application launch.  
16         return true  
17     }  
18  
19     // MARK: UISceneSession Lifecycle  
20  
21     func application(_ application: UIApplication, configurationForConnecting connectingSceneSession: UISceneSession, options: UIScene.ConnectionOptions) -> UISceneConfiguration {  
22         // Called when a new scene session is being created.  
23         // Use this method to select a configuration to create the new scene with.  
24         return UISceneConfiguration(name: "Default Configuration", sessionRole: connectingSceneSession.role)  
25     }  
26  
27     func application(_ application: UIApplication, didDiscardSceneSessions sceneSessions: Set<UISceneSession>) {  
28         // Called when the user discards a scene session.  
29         // If any sessions were discarded while the application was not running, this will be called shortly after application.startsRunning(_:).  
30         // Use this method to release any resources that were specific to the discarded scenes, as they will not return.  
31     }  
32 }  
33 }
```

Xcode 11.7

Where did everything go?

The screenshot shows the Xcode interface with the following details:

- Title Bar:** AppLifeCycleXcode11 > Generic iOS Device AppLifeCycleXcode11: Ready | Today at 2:39 PM
- File Navigator:** Shows the project structure: AppLifeCycleXcode11 > AppLifeCycleXcode11 > SceneDelegate.swift. The `SceneDelegate.swift` file is selected.
- Editor:** Displays the `SceneDelegate.swift` code. The code handles the life cycle of a scene, including connecting to a session, creating a window, and managing its visibility. It also handles disconnecting from a session, becoming active again, and resigning from an active state.

```
9 import UIKit
10 import SwiftUI
11
12 class SceneDelegate: UIResponder, UIWindowSceneDelegate {
13
14     var window: UIWindow?
15
16     func scene(_ scene: UIScene, willConnectTo session: UISceneSession, options connectionOptions: UIScene.ConnectionOptions) {
17         // Use this method to optionally configure and attach the UIWindow `window` to the provided UIWindowScene.
18         // If using a storyboard, the `window` property will automatically be initialized and attached to the scene.
19         // This delegate does not imply the connecting scene or session are new (see `application:configurationForConnectingSceneSession`).
20
21         // Create the SwiftUI view that provides the window contents.
22         let contentView = ContentView()
23
24         // Use a UIHostingController as window root view controller.
25         if let windowScene = scene as? UIWindowScene {
26             let window = UIWindow(windowScene: windowScene)
27             window.rootViewController = UIHostingController(rootView: contentView)
28             self.window = window
29             window.makeKeyAndVisible()
30         }
31     }
32
33     func sceneDidDisconnect(_ scene: UIScene) {
34         // Called as the scene is being released by the system.
35         // This occurs shortly after the scene enters the background, or when its session is discarded.
36         // Release any resources associated with this scene that can be re-created the next time the scene connects.
37         // The scene may re-connect later, as its session was not necessarily discarded (see `application:didDiscardSceneSessions`).
38     }
39
40     func sceneDidBecomeActive(_ scene: UIScene) {
41         // Called when the scene has moved from an inactive state to an active state.
42         // Use this method to restart any tasks that were paused (or not yet started) when the scene was inactive.
43     }
44
45     func sceneWillResignActive(_ scene: UIScene) {
46         // Called when the scene will move from an active state to an inactive state.
47         // This may occur due to temporary interruptions (ex. an incoming phone call).
48     }
}
```

Xcode 11.7

The screenshot shows the Xcode interface with the title bar "AppLifeCycleXcode11" and "Generic iOS Device". The project navigation sidebar lists "AppLifeCycleXcode11", "AppLifeCycleXcode11", "AppDelegate.swift", and "SceneDelegate.swift". The main editor pane displays the "SceneDelegate.swift" file content:

```
func sceneDidDisconnect(_ scene: UIScene) {
    // Called as the scene is being released by the system.
    // This occurs shortly after the scene enters the background, or when its session is discarded.
    // Release any resources associated with this scene that can be re-created the next time the scene connects.
    // The scene may re-connect later, as its session was not necessarily discarded (see 'application:didResignActive')
}

func sceneDidBecomeActive(_ scene: UIScene) {
    // Called when the scene has moved from an inactive state to an active state.
    // Use this method to restart any tasks that were paused (or not yet started) when the scene was inactive.
}

func sceneWillResignActive(_ scene: UIScene) {
    // Called when the scene will move from an active state to an inactive state.
    // This may occur due to temporary interruptions (ex. an incoming phone call).
}

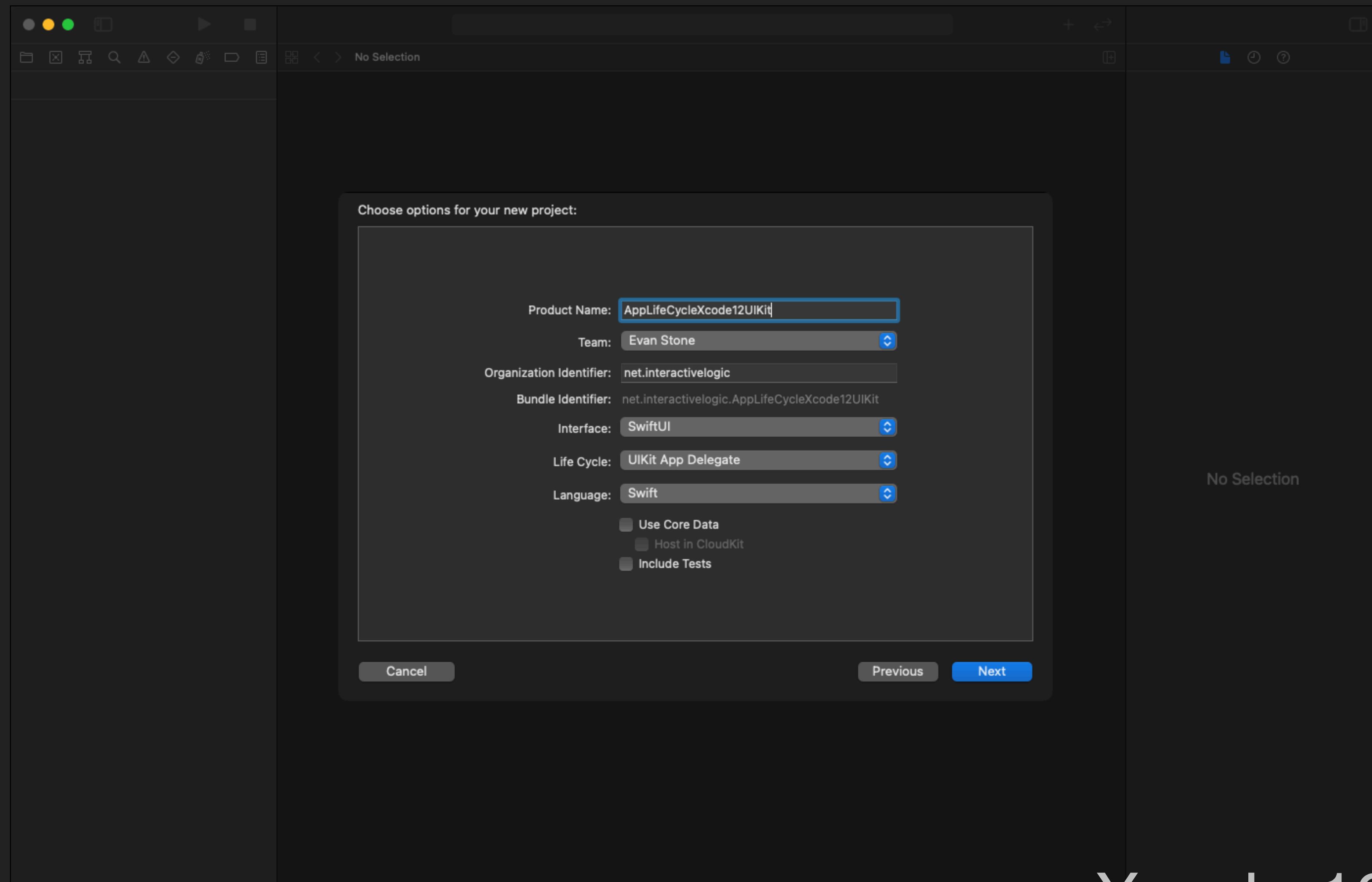
func sceneWillEnterForeground(_ scene: UIScene) {
    // Called as the scene transitions from the background to the foreground.
    // Use this method to undo the changes made on entering the background.
}

func sceneDidEnterBackground(_ scene: UIScene) {
    // Called as the scene transitions from the foreground to the background.
    // Use this method to save data, release shared resources, and store enough scene-specific state information
    // to restore the scene back to its current state.
}
```

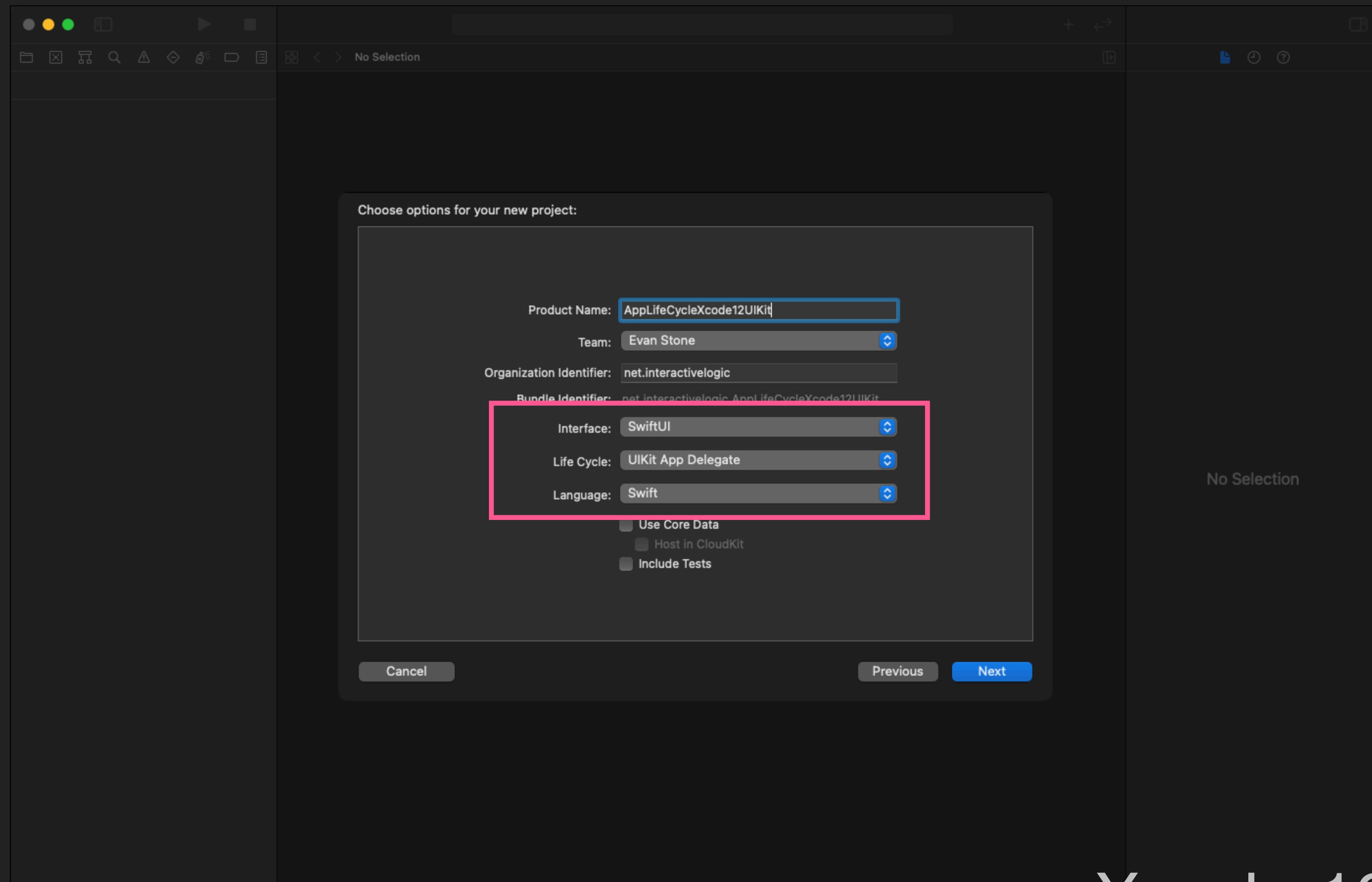
At the bottom left of the editor, there is a "Filter" button. On the right side of the window, there is a large watermark-like text "Xcode 11.7".

THE PRESENT: THE SWIFTUI WAY

Xcode 12



Xcode 12.5.1



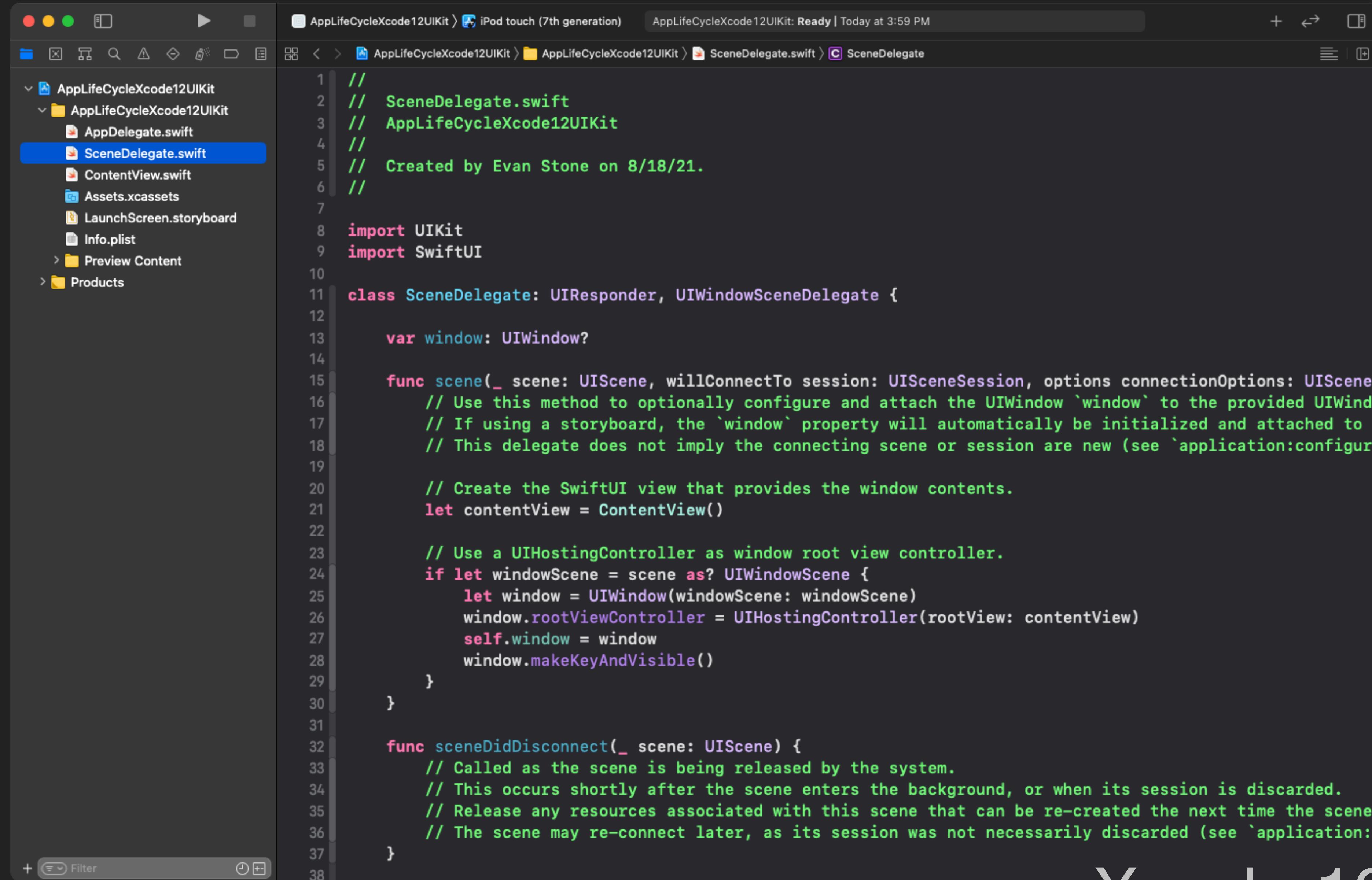
Xcode 12.5.1

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure for "AppLifeCycleXcode12UIKit". The "AppDelegate.swift" file is selected.
- Editor:** Displays the content of the "AppDelegate.swift" file. The code is as follows:

```
1 // AppDelegate.swift
2 // AppLifeCycleXcode12UIKit
3 // Created by Evan Stone on 8/18/21.
4 //
5 import UIKit
6
7 @main
8 class AppDelegate: UIResponder, UIApplicationDelegate {
9
10    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
11        // Override point for customization after application launch.
12        return true
13    }
14
15    // MARK: UISceneSession Lifecycle
16
17    func application(_ application: UIApplication, configurationForConnecting connectingSceneSession: UISceneSession, options: UIScene.ConnectionOptions) -> UISceneConfiguration {
18        // Called when a new scene session is being created.
19        // Use this method to select a configuration to create the new scene with.
20        return UISceneConfiguration(name: "Default Configuration", sessionRole: connectingSceneSession.role)
21    }
22
23    func application(_ application: UIApplication, didDiscardSceneSessions sceneSessions: Set<UISceneSession>) {
24        // Called when the user discards a scene session.
25        // If any sessions were discarded while the application was not running, this will be called shortly thereafter.
26        // Use this method to release any resources that were specific to the discarded scenes, as they will no longer be available.
27    }
28
29 }
30
31 }
32
33 }
```

Xcode 12.5.1



The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure under "AppLifeCycleXcode12UIKit". The "SceneDelegate.swift" file is selected and highlighted with a blue background.
- Editor:** Displays the content of the "SceneDelegate.swift" file. The code is written in Swift and defines a class that implements the `UIResponder` and `UIWindowSceneDelegate` protocols.
- Document Outline:** Located at the top center, it shows the current file path: "AppLifeCycleXcode12UIKit > SceneDelegate.swift".
- Search Bar:** At the top right, it displays the text "AppLifeCycleXcode12UIKit: Ready | Today at 3:59 PM".

```
// SceneDelegate.swift
// AppLifeCycleXcode12UIKit
// Created by Evan Stone on 8/18/21.

import UIKit
import SwiftUI

class SceneDelegate: UIResponder, UIWindowSceneDelegate {

    var window: UIWindow?

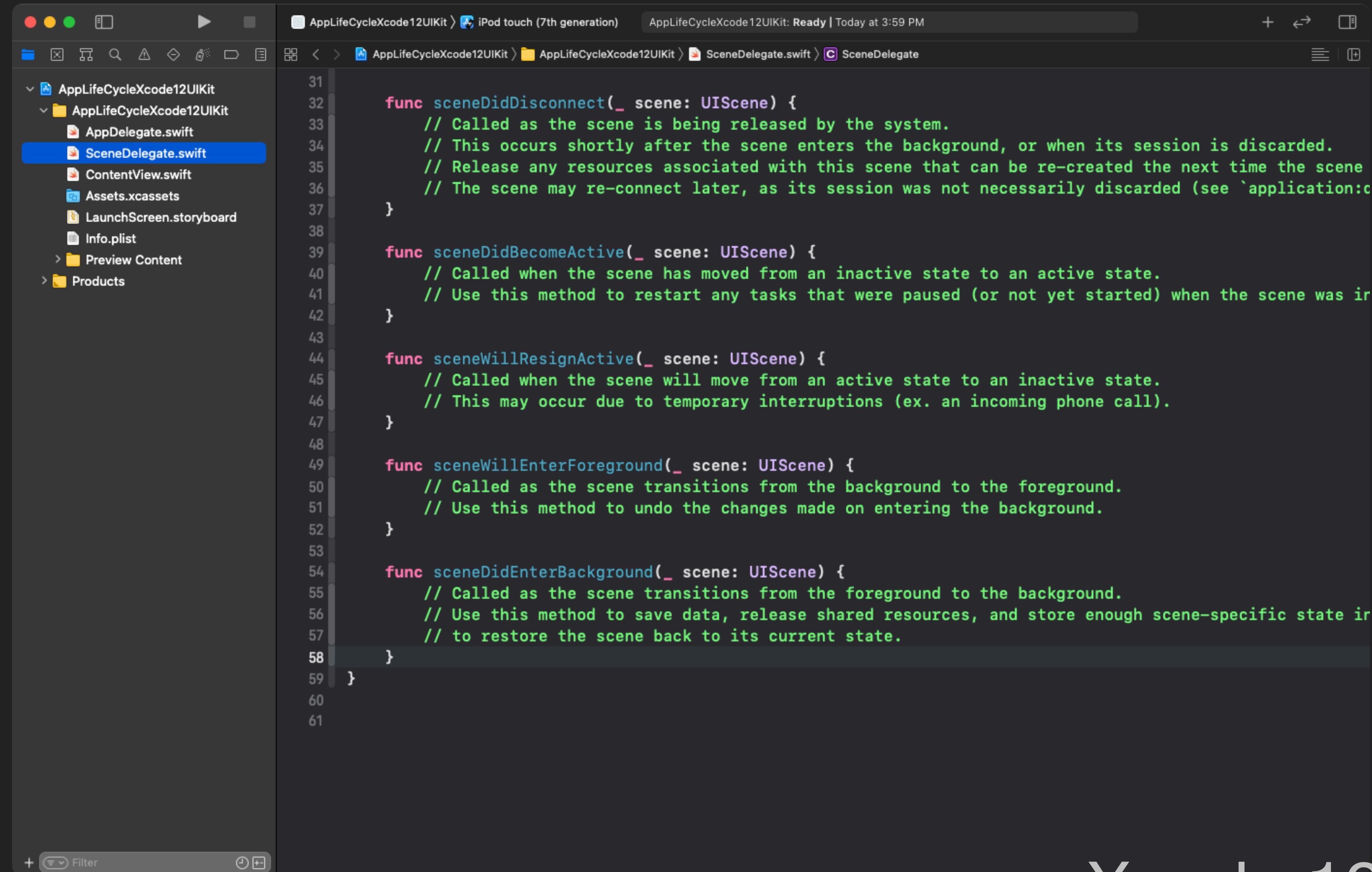
    func scene(_ scene: UIScene, willConnectTo session: UISceneSession, options connectionOptions: UIScene.ConnectionOptions) {
        // Use this method to optionally configure and attach the UIWindow `window` to the provided UIWindowScene.
        // If using a storyboard, the `window` property will automatically be initialized and attached to the scene.
        // This delegate does not imply the connecting scene or session are new (see `application:configurationForConnectingSceneSession`).

        // Create the SwiftUI view that provides the window contents.
        let contentView = ContentView()

        // Use a UIHostingController as window root view controller.
        if let windowScene = scene as? UIWindowScene {
            let window = UIWindow(windowScene: windowScene)
            window.rootViewController = UIHostingController(rootView: contentView)
            self.window = window
            window.makeKeyAndVisible()
        }
    }

    func sceneDidDisconnect(_ scene: UIScene) {
        // Called as the scene is being released by the system.
        // This occurs shortly after the scene enters the background, or when its session is discarded.
        // Release any resources associated with this scene that can be re-created the next time the scene connects.
        // The scene may re-connect later, as its session was not necessarily discarded (see `application:configurationForConnectingSceneSession`).
    }
}
```

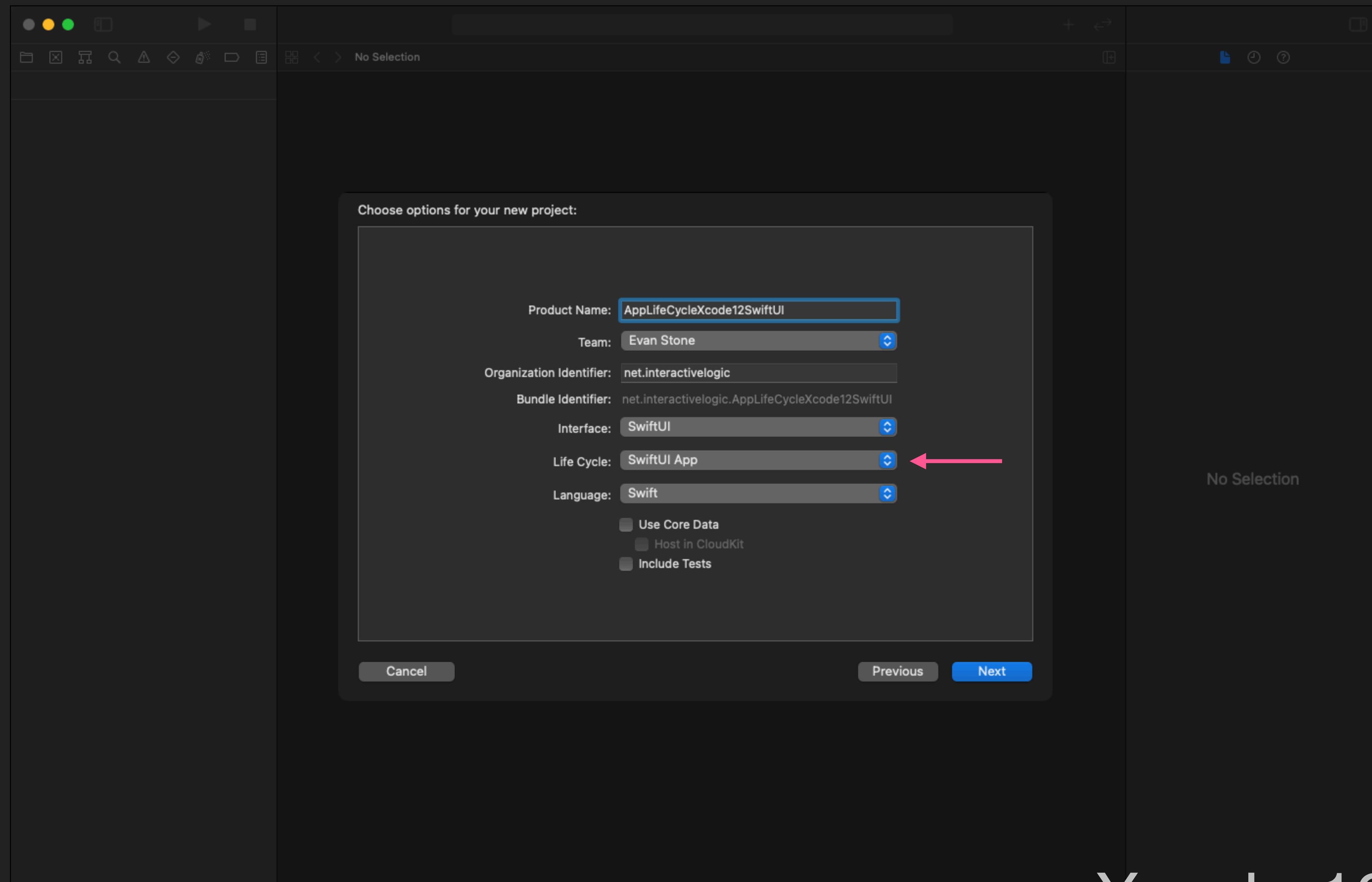
Xcode 12.5.1



A screenshot of the Xcode 12.5.1 interface. The main window shows the code editor with SceneDelegate.swift open. The code defines several optional methods for managing the app's life cycle:

```
31 func sceneDidDisconnect(_ scene: UIScene) {
32     // Called as the scene is being released by the system.
33     // This occurs shortly after the scene enters the background, or when its session is discarded.
34     // Release any resources associated with this scene that can be re-created the next time the scene
35     // The scene may re-connect later, as its session was not necessarily discarded (see `application:didRescueSession`).
36 }
37
38 func sceneDidBecomeActive(_ scene: UIScene) {
39     // Called when the scene has moved from an inactive state to an active state.
40     // Use this method to restart any tasks that were paused (or not yet started) when the scene was inactive.
41 }
42
43
44 func sceneWillResignActive(_ scene: UIScene) {
45     // Called when the scene will move from an active state to an inactive state.
46     // This may occur due to temporary interruptions (ex. an incoming phone call).
47 }
48
49 func sceneWillEnterForeground(_ scene: UIScene) {
50     // Called as the scene transitions from the background to the foreground.
51     // Use this method to undo the changes made on entering the background.
52 }
53
54 func sceneDidEnterBackground(_ scene: UIScene) {
55     // Called as the scene transitions from the foreground to the background.
56     // Use this method to save data, release shared resources, and store enough scene-specific state information
57     // to restore the scene back to its current state.
58 }
59 }
60
61
```

Xcode 12.5.1



Xcode 12.5.1

The screenshot shows the Xcode interface with the following details:

- Title Bar:** AppLifeCycleXcode12SwiftUI > iPod touch (7th generation) | AppLifeCycleXcode12SwiftUI: Ready | Today at 2:35 PM
- File Navigator:** Shows the project structure:
 - AppLifeCycleXcode12SwiftUI
 - AppLifeCycleXcode12SwiftUI
 - AppLifeCycleXcode12SwiftUIApp.swift (selected)
 - ContentView.swift
 - Assets.xcassets
 - Info.plist
 - Preview Content
 - Products
- Editor Area:** Displays the contents of AppLifeCycleXcode12SwiftUIApp.swift:

```
1 //  
2 //  AppLifeCycleXcode12SwiftUIApp.swift  
3 //  AppLifeCycleXcode12SwiftUI  
4 //  
5 //  Created by Evan Stone on 8/18/21.  
6 //  
7  
8 import SwiftUI  
9  
10 @main  
11 struct AppLifeCycleXcode12SwiftUIApp: App {  
12     var body: some Scene {  
13         WindowGroup {  
14             ContentView()  
15         }  
16     }  
17 }  
18
```
- Bottom Bar:** Filter, Refresh, and New File icons.

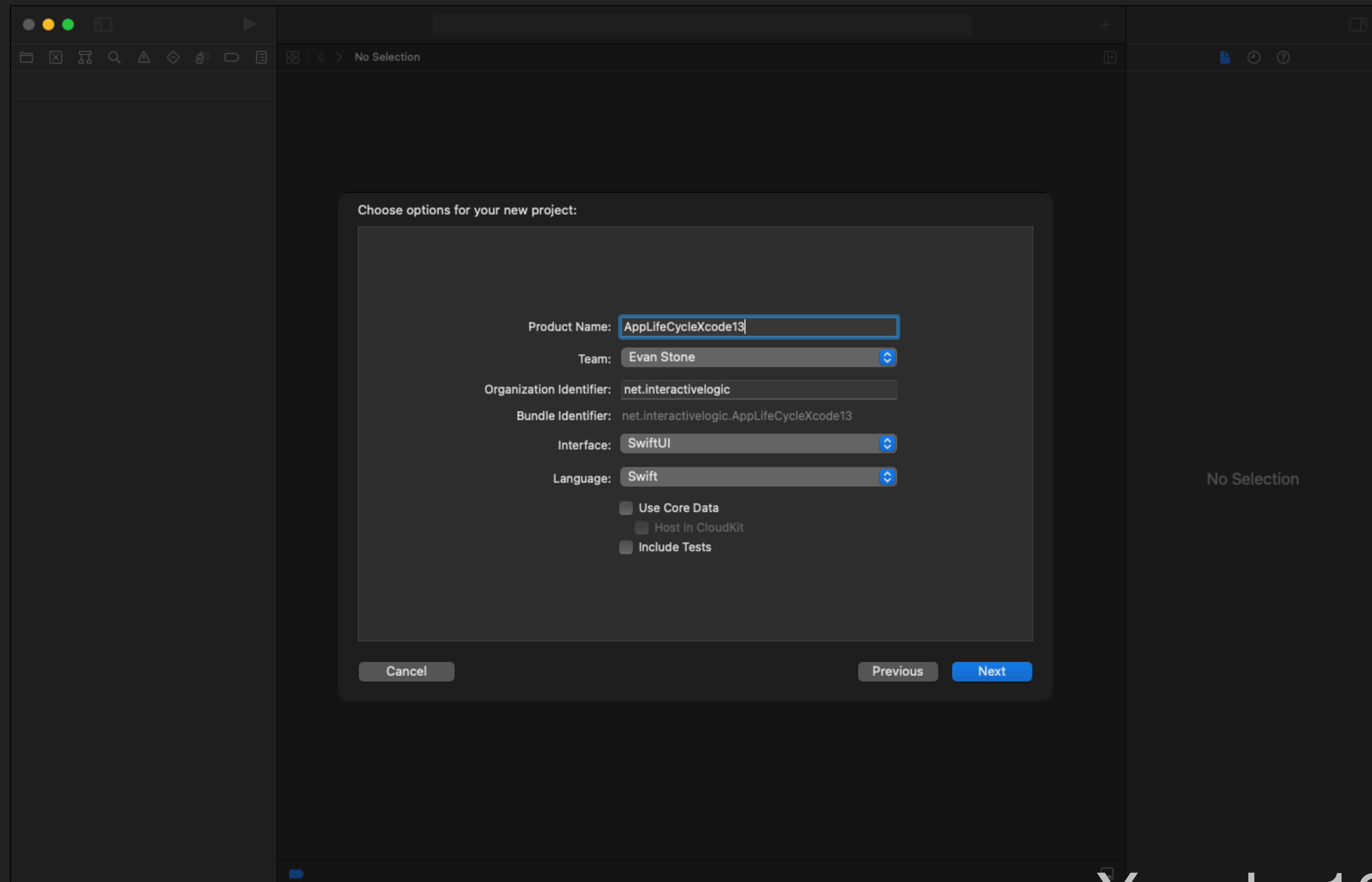
Xcode 12.5.1



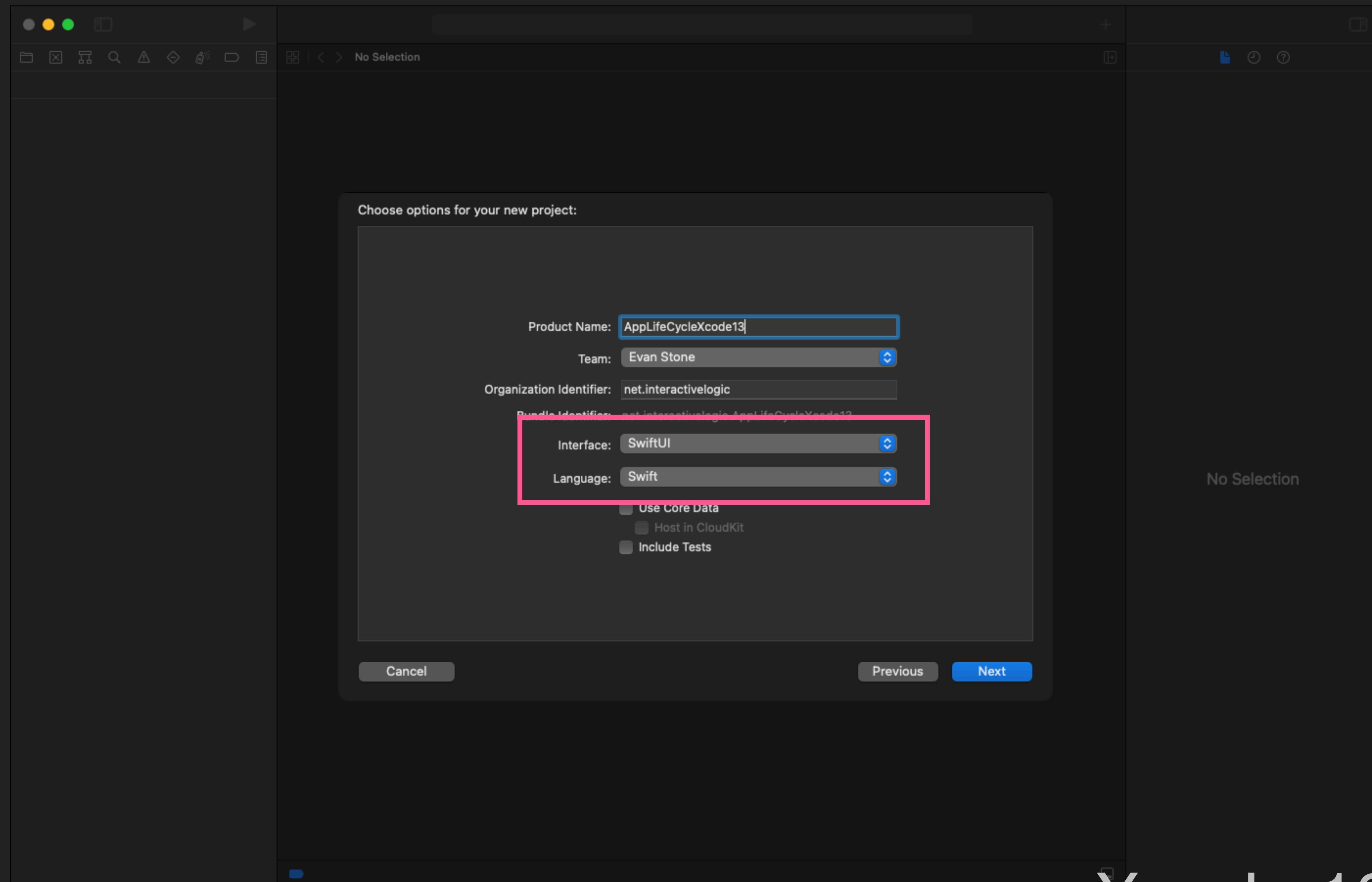
Where did everything go?
(again)

**THE FUTURE:
THE SWIFTUI WAY**

Xcode 13



Xcode 13b5



Xcode 13b5

The screenshot shows the Xcode interface with the following details:

- Title Bar:** AppLifeCycleXcode13
- Document Outline:** Shows the project structure: AppLifeCycleXcode13 > AppLifeCycleXcode13 > AppLifeCycleXcode13App.swift
- Editor:** Displays the contents of AppLifeCycleXcode13App.swift. The code is as follows:

```
//  
//  AppLifeCycleXcode13App.swift  
//  AppLifeCycleXcode13  
//  
//  Created by Evan Stone on 8/18/21.  
//  
import SwiftUI  
  
@main  
struct AppLifeCycleXcode13App: App {  
    var body: some Scene {  
        WindowGroup {  
            ContentView()  
        }  
    }  
}
```

The code defines a Swift UI application structure named AppLifeCycleXcode13App. It imports SwiftUI and uses the @main annotation to mark it as the main entry point. The body is defined as a WindowGroup containing a ContentView.

Xcode 13b5

Demo.

APP LIFE CYCLE HISTORY CHART



WWDC	Xcode	iOS	UI Framework	Life Cycle
2018	10	12	UIKit	App Delegate
2019	11	13	UIKit/SwiftUI	App & Scene Delegates
2020	12	14	SwiftUI*	App/Scene
2021	13	15	SwiftUI*	App/Scene

WHICH SHOULD I USE?

ACKNOWLEDGEMENTS

ACKNOWLEDGEMENTS

- ▶ *The Ultimate Guide to the SwiftUI 2 Application Life Cycle* by Peter Friese
<https://peterfriese.dev/ultimate-guide-to-swiftui2-application-lifecycle/>
- ▶ *Managing app in SwiftUI* and *Managing scenes in SwiftUI*, by Swift with Majid
<https://swiftwithmajid.com/2020/08/19/managing-app-in-swiftui/>
<https://swiftwithmajid.com/2020/08/26/managing-scenes-in-swiftui/>
- ▶ *How SwiftUI can now be used to build entire iOS apps* by John Sundell
<https://wwdcbysundell.com/2020/building-entire-apps-with-swiftui/>
- ▶ *How to add an AppDelegate to a SwiftUI app* by Paul Hudson
<https://www.hackingwithswift.com/quick-start/swiftui/how-to-add-an-appdelegate-to-a-swiftui-app>

WHERE TO FIND ME . . .

WHERE TO FIND ME...

- ▶ Twitter: [@evankstone](#) and [@iOSDevBreak](#)
- ▶ Blog/Consulting: [interactivelogic.net](#)
- ▶ iOS Dev Break Podcast: [iosdevbreak.com](#)
- ▶ Transformation: [transformationapps.com](#)
- ▶ Bleacher Report: [bleacherreport.com](#)

Interactive Logic



transformation

B-R

