



Catalyst **Workshop**

About Me

- Benedikt Terhechte
- Work at XING (xing.com)

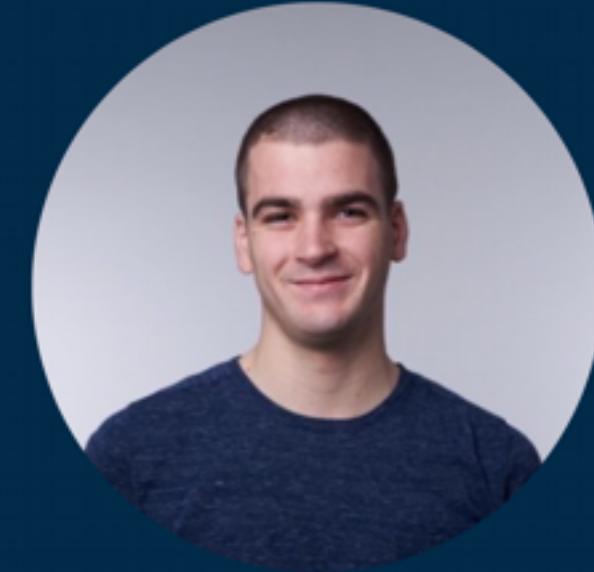


@terhechte



CONTRA VARIANCE

A  PODCAST



@BasThomas

Swift Weekly Brief

www.contravariance.rocks

Stickers!

Simple Swift Guides: www.appventure.me

The screenshot shows the homepage of APPVENTURE.ME. The header features the site's logo and the tagline "Simple. Swift. Guides.". Below the header, there is a search bar and a sidebar with navigation links for "SWIFT TOPICS" (All, Swift Tricks, Language, Hands On, Blog) and "GUIDES" (Pattern Matching, Enums, Map, Reduce & more, Associated Types, Tuples, Optionals, Reflection). A sidebar also includes a link to "CONTRA VARIANCE A PODCAST". The main content area displays a grid of six guide cards:

- Pattern Matching**: A brief introduction into Swift's tuples including a basic introduction and several useful tips for advanced usage.
- Tuples**: A brief introduction into Swift's tuples including a basic introduction and several useful tips for advanced usage.
- Enums**: When and how to use enums in Swift? This is a detailed practical overview of all the possibilities enums can offer you.
- Optionals**: Best practices for handling Swift's "Optional" and extending them to make them much more approachable.
- Reflection**: In this guide we'll examine the Swift reflection API, see how fast it is, and will try to show use cases where it can be applied successfully.
- Associated Types**: Understand how to model your way around some of the issues that arise when introducing associated types into your protocols.

At the bottom right of the page, the number "101" is displayed.

@terhechte

No Stickers!

Agenda

1. Scope of Workshop
2. Brief Catalyst Overview
3. iOS to macOS mappings
4. macOS UIKit additions
5. Notable Mentions
6. Notes
7. Workshop Task

Scope of the Workshop

Topics we will handle

- Catalyst
- NSMainMenu
- NSToolbar
- NSTouchBar
- Common issues
- Responder chain

Topics we will briefly handle

- Design considerations
- Dark Mode
- Settings.bundle
- Multiple Windows
- Non-Appstore builds

Topics we will not handle

- Releasing & App Store
- App Lifecycle
- Accessibility
- Including AppKit bundles

Brief Catalyst Overview

Catalyst

- Run iPad apps on macOS
- macOS 13.0 only
- Supports most of iOS
- (No ARKit, for example)

BETA Software 😎

```
Please contact someone with a photo of the
information printed below, along with a description of your
system configuration and what you were doing at the time that
the kernel panic occurred. We apologize for the inconvenience.

panic(cpu 0 caller 0x5600f3): "Unable to find driver for this platform: \"ACPI\".\n@/Users/nawcon/Builds/xnu-1504.9.26/lokit/
rnet/IOPPlatformExpert.cpp:1393
Debugger called: <panic>
Backtrace (CPU 0), Frame : Return Address (4 potential args on stack)
0x29223da8 : 0x21b518 (0x5db8ac 0x29223ddc 0x223986 0x8) _panic + 0x1e9
0x29223df8 : 0x5600f3 (0x5e54ac 0x31834f8 0x29223e28 0x31f9040) __ZN15IOPanicPlatform5startEP9IOService + 0x35
0x29223e18 : 0x540ac8 (0x323c700 0x31e8d80 0x1 0x5039aa) __ZN9IOService14startCandidateEPS_ + 0x7c
0x29223e78 : 0x53ecfd (0x31e8d80 0x323c700 0x31f9800 0x507872) __ZN9IOService15probeCandidatesEP120S0OrderedSet + 0x689
0x29223f28 : 0x53f28e (0x31e8d80 0x31efb00 0x8 0xffffffff) __ZN9IOService14doServiceMatchEm + 0x16c
0x29223f78 : 0x540f3d (0x31e8d80 0x8 0x29223fac 0x1) __ZN15_IoConfigThread4mainEPvi + 0x111
0x29223fc8 : 0x2a06bc (0x3183290 0x0 0x2f3c6530 0x69727473) __call_continuation + 0x1c

BSD process name corresponding to current thread: Unknown

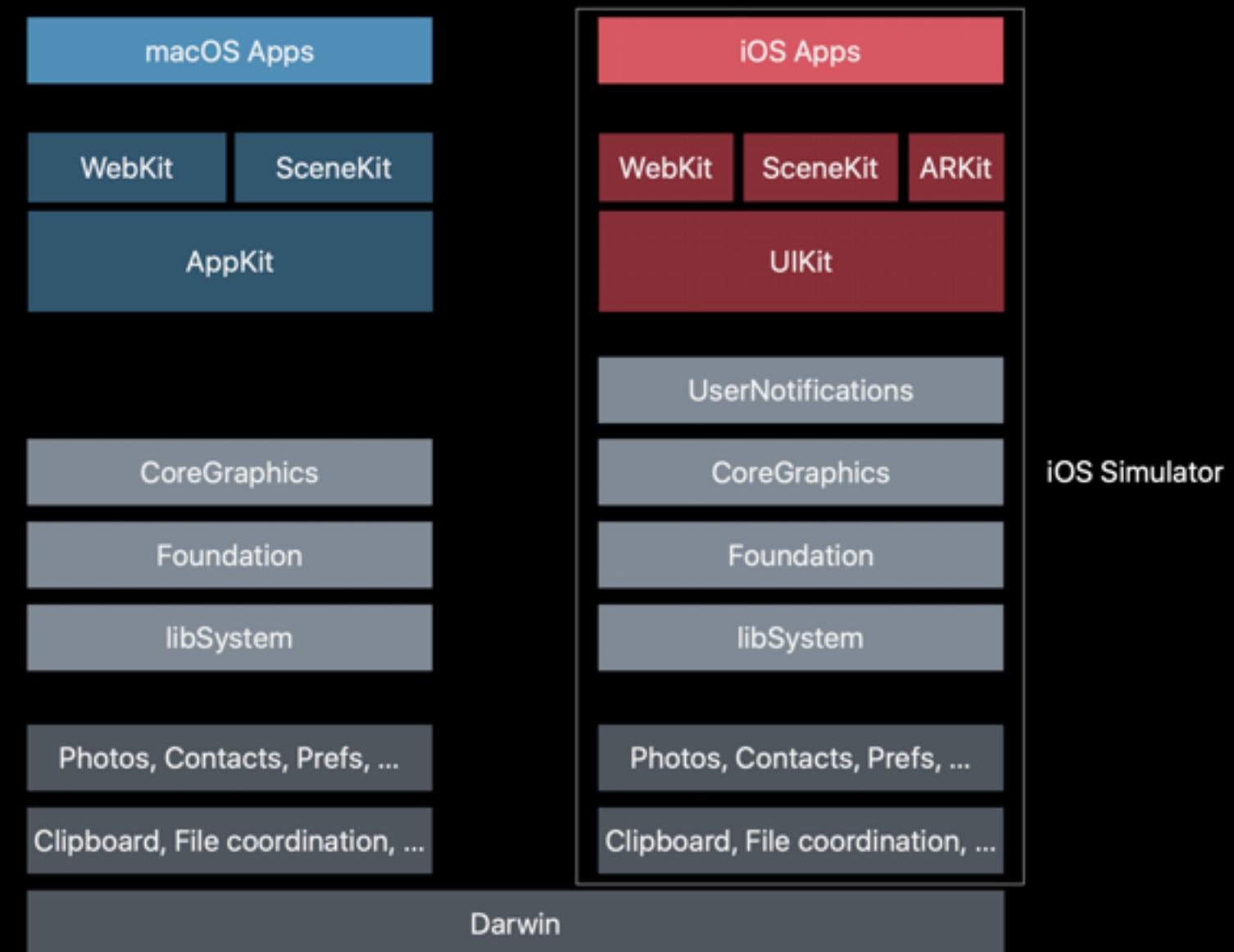
Mac OS version:
Not yet set

Kernel version:
Darwin Kernel Version 10.6.0: Sun Jan  9 16:31:48 EST 2011; legacy kernel v6 :xnu-1504.9.26/BUILD/obj/RELEASE_I386

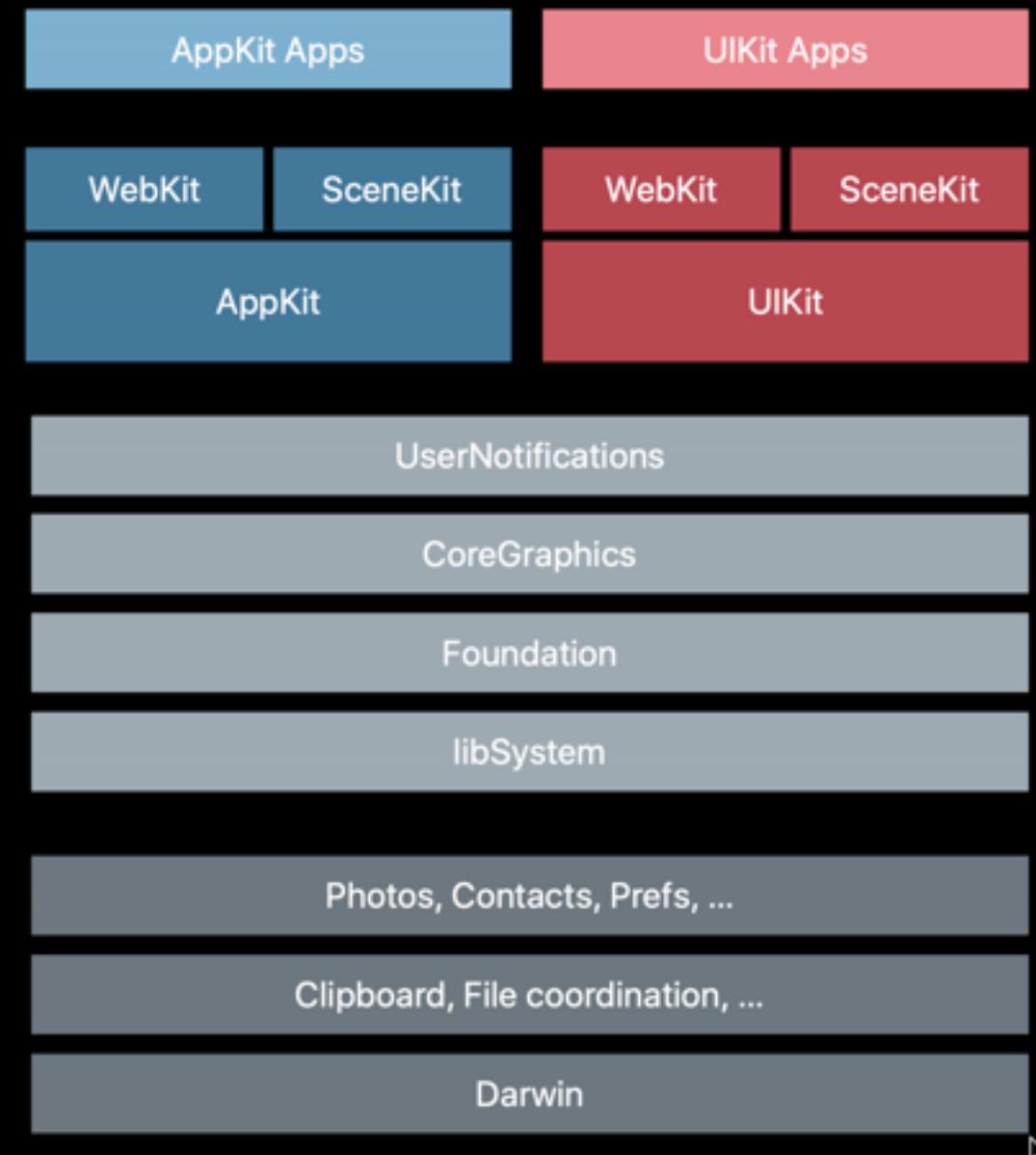
System uptime in nanoseconds: 26682610
```



Tech Stack Pre Catalyst



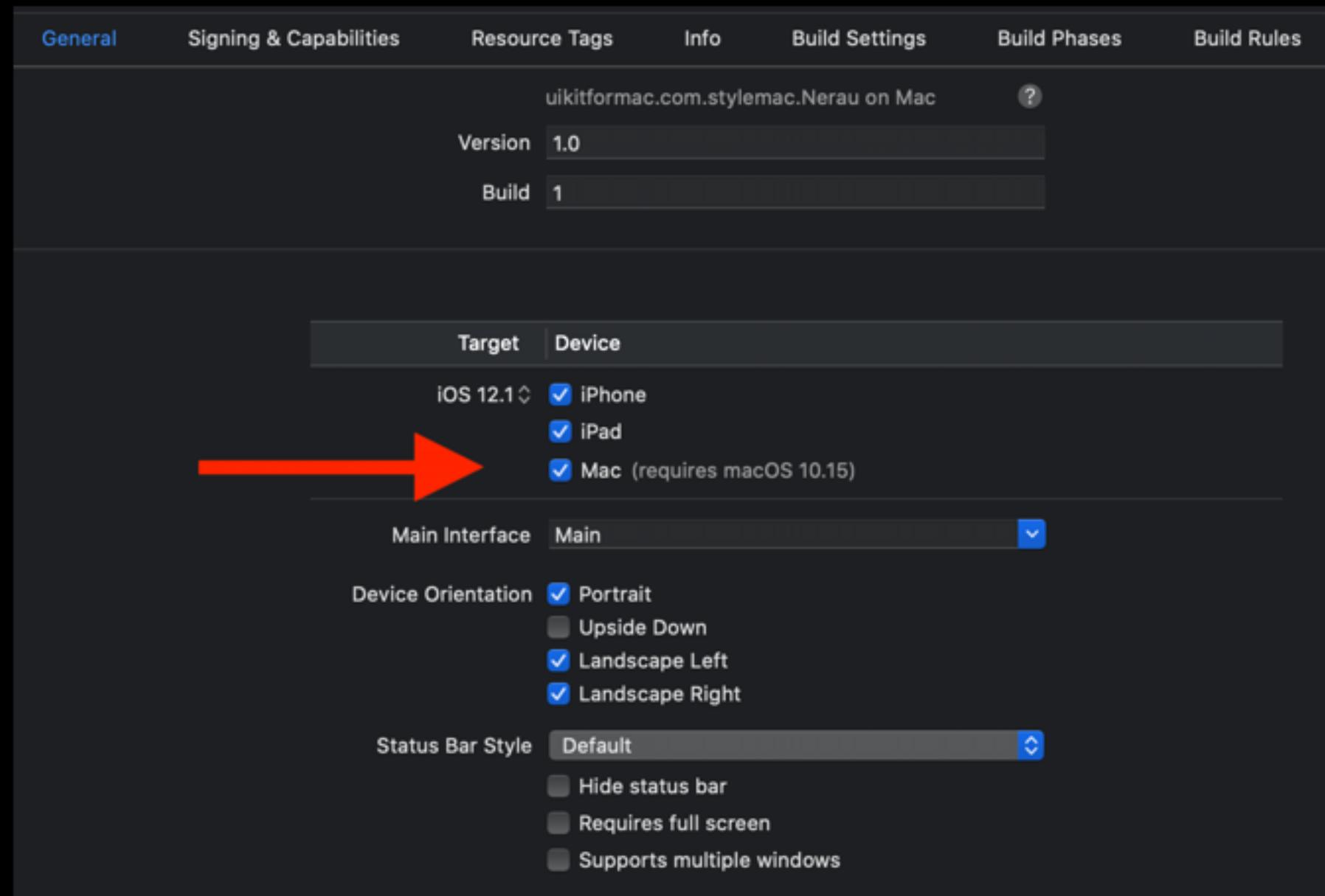
Tech Stack Post Catalyst



Exceptions



Activating it



What happens next?

Maps many iOS frameworks to macOS equivalents

Say UITableView to NSTableView

[More on this later]

Adds `AppName.entitlements`

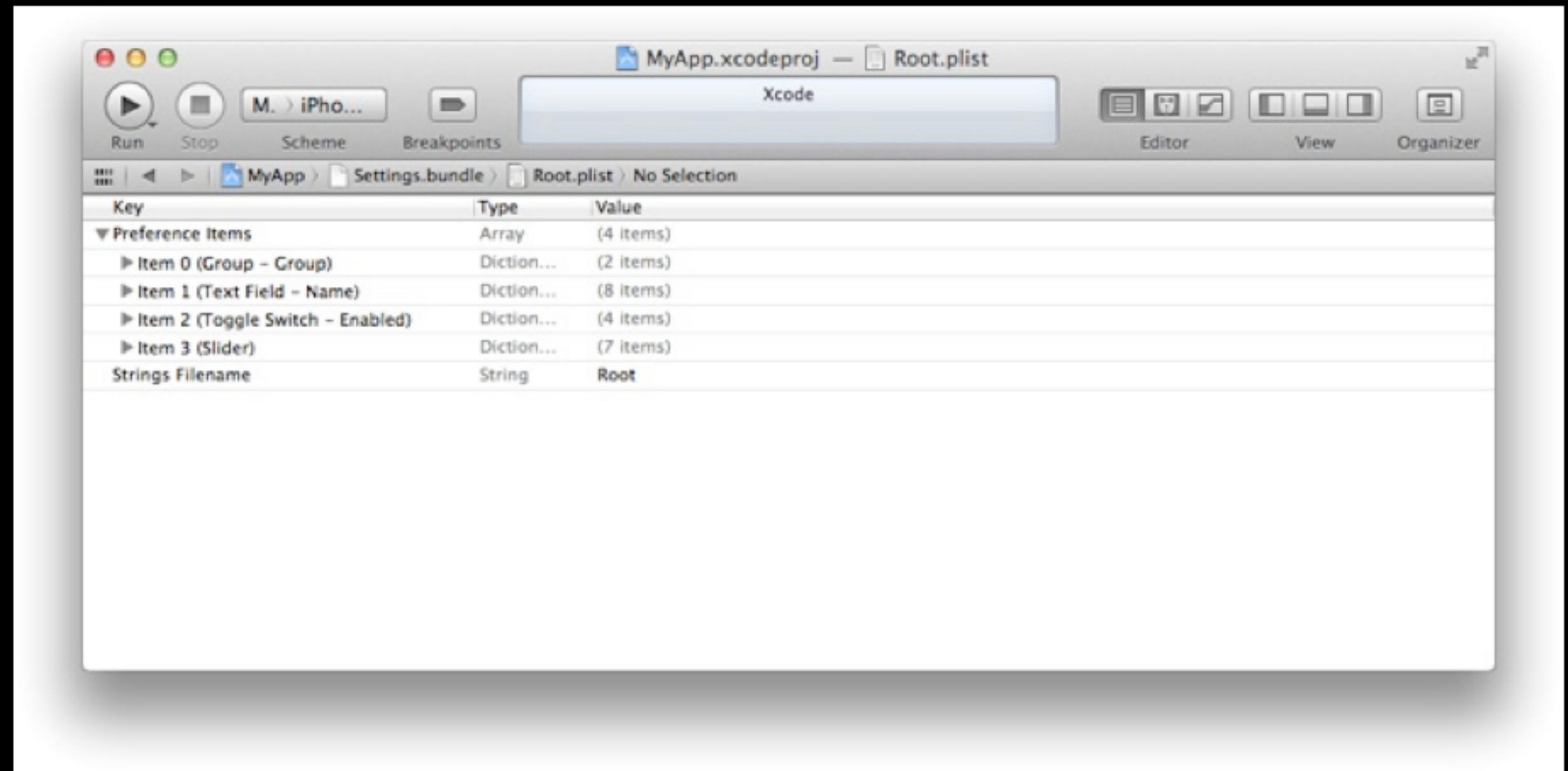
macOS Sandbox stuff

[Can be disabled!]

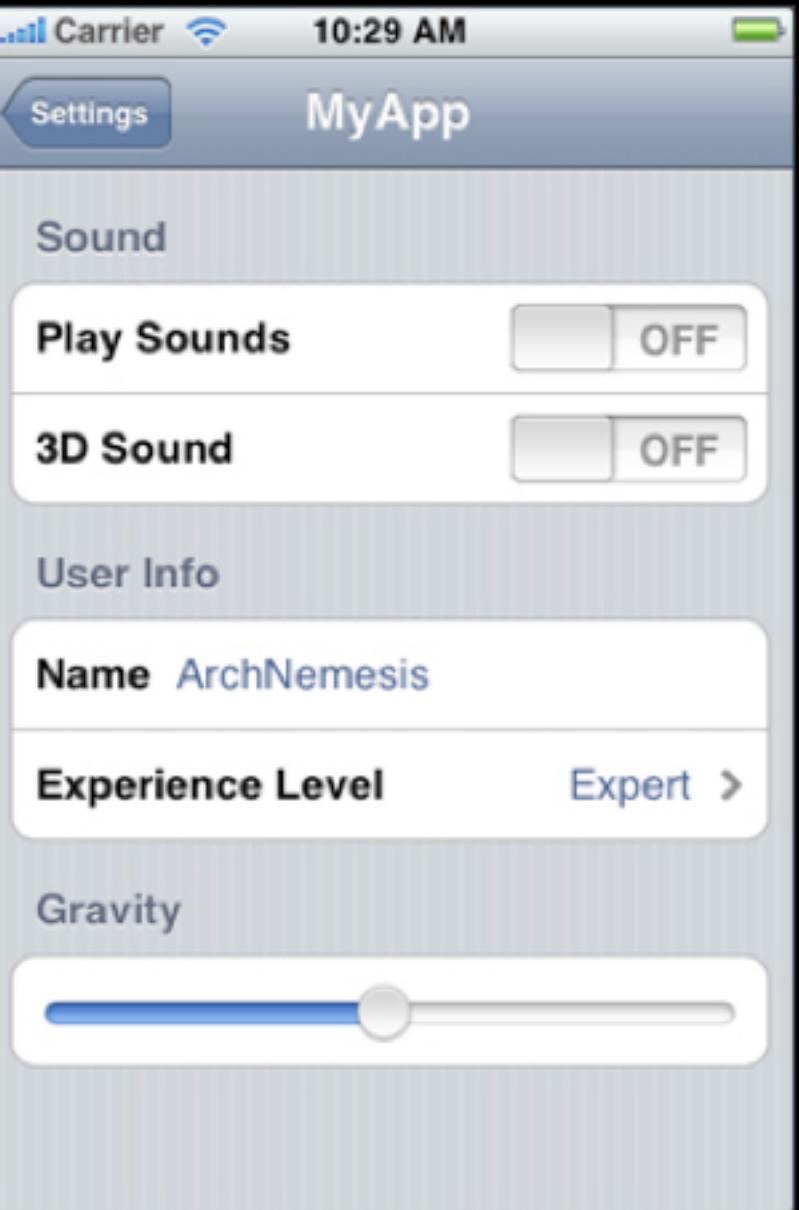
Preferences

Settings.bundle

[Remember?]

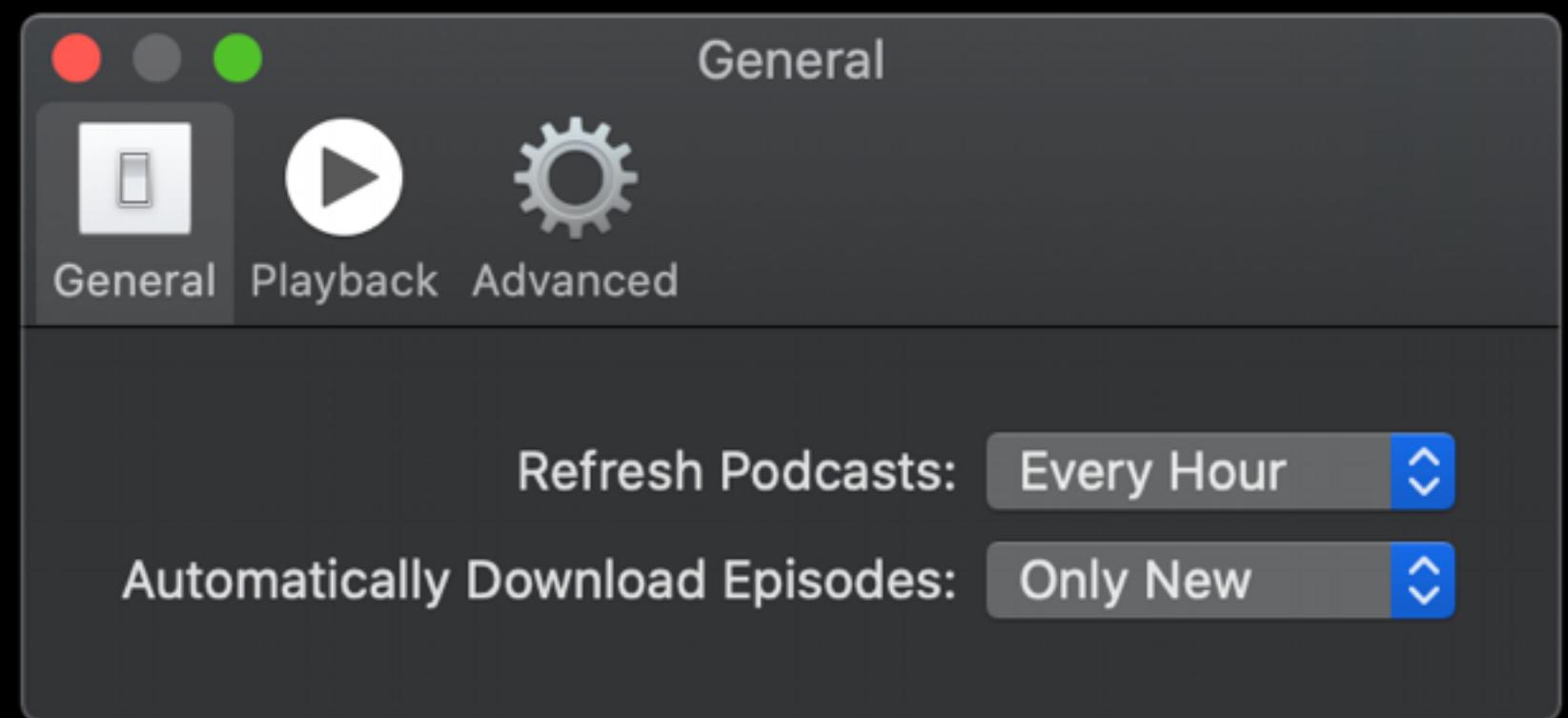


Settings.bundle



Settings.bundle

Preferences



Builds full-fledged macOS app

Distribute outside of App Store

Kinda Changes the App Lifecycle

Apps can run forever

[Huge changes, 30 min talk section at WWDC]

Disables Dynamic Type

macOS doesn't have that

[Everything is Large]

Scales UI to 77%

iOS touch targets are bigger than macOS

[Every UI element is only 77% of its size]

Default Font Size on macOS is 13.0

iOS is 17.0

New Bundle Identifier

i.e. "**com.apple.notes**" to "**uikitformac.com.apple.notes**"

iOS to macOS mappings

- A default menu bar for your app.
- Support for trackpad, mouse, and keyboard input.
- Support for window resizing and full-screen display.
- Mac-style scroll bars.
- Copy-and-paste support.
- Drag-and-drop support.
- Support for system Touch Bar controls.

macOS UIKit additions

MenuBar: UIMenu



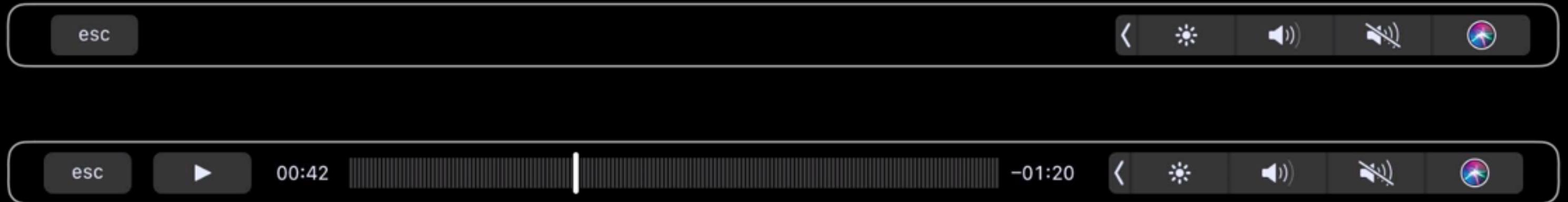
Mouse Hovering: UIHoverGestureRecognizer



Toolbar: NSToolbar



TouchBar: NSTouchBar



Notable Mentions

macOS only Code

```
#if targetEnvironment(UIKitForMac)
    let toolbar = NSToolbar(identifier: "MyToolbar")
#endif
```

iOS only Code

```
#if !targetEnvironment(UIKitForMac)
    import ARKit
#endif
```

Notes

- macOS windows resize a lot. Fast drawing / layout
- macOS windows resize in manifold ways
- Much bigger screens
- Your app can run forever. Memory Leaks!
- Mac users love ~the~ a keyboard. Add shortcuts
- Binary third party SDKs need to be recompiled
- Non-Retina displays still exist

Receipt Validation

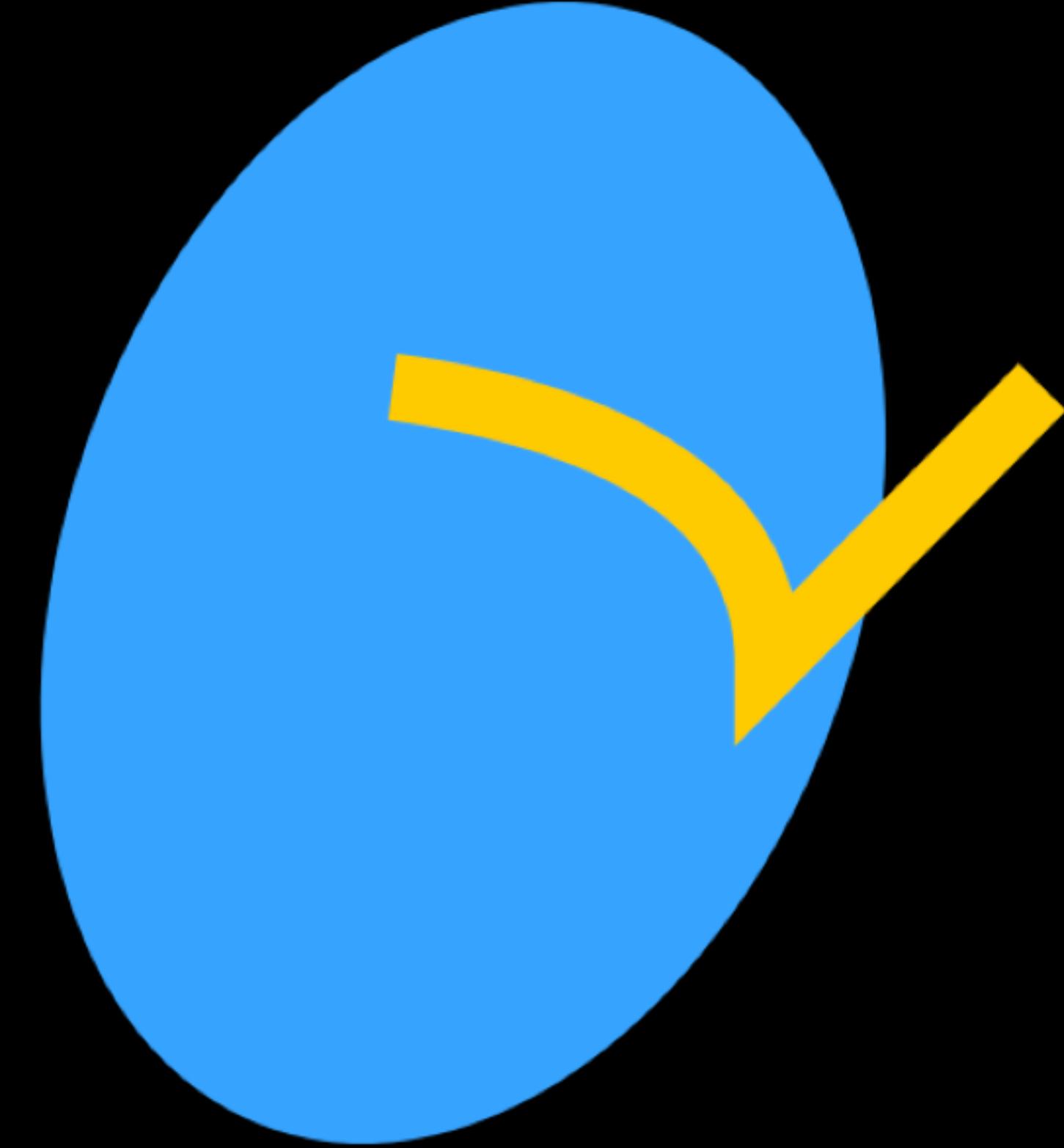
- Mac App Store doesn't prevent piracy
- <http://receigen.etiemble.com>

Demo

Workshop Task

- We all have pure codebases, right?
- This is not a nice codebase
- Added a lot of wrinkles
- Terrible code

- Take Nerau, add macOS support, and fix the compile errors
- Look at the finished Nerau code to see how to achieve what
- Pick which features you want to work on. Toolbar? Multiwindow? Menubar? You decide



The screenshot shows a macOS application window with a sidebar on the left and a main content area on the right.

Sidebar (Left):

- How do I sign the app
- How do I (programmatically) modify the menubar
- How can I insert a 'New Window' entry into my custom storyboard MainMenu?
- How do I create a toolbar** (highlighted with a blue background)
- How do I modify the looks of the Window (for example, hide the title)
- How do I add TouchBar support?
- How do I use the cool TouchBar images that Apple provides with AppKit
- How do I dynamically update the contents of my touchbar?
- How do I nest touchbars?
- How do I create a sidebar

Main Content Area (Right):

How do I create a toolbar

Preparations

Note that in Beta 2 you need to manually import a special bridging header if you want to use `NSToolbar`. [Here's how to do it.](#)

The best way to insert toolbars is to use the new `UIScene` API that Apple introduced with iOS 13. The gist is that you use a `SceneDelegate` and in the `scene:willConnectToSession:options` method, you can modify a window scene's `titlebar` by setting a new `NSToolbar` on it.

If your project doesn't have a `SceneDelegate` yet, [here's a brief primer on how to set it up..](#) This delegate is - as far as I'm aware - required to support toolbars.

Modifying the Scene

The first step is to modify the `scene(willConnectTo:options:)` method in your `SceneDelegate` to check if we're running `UIKitForMac` - after all, iOS doesn't have toolbars - and then make sure that our scene is actually a `UIWindowScene`:

```
func scene(_ scene: UIScene, willConnectTo session: UISceneSession, options: UIScene.ConnectionOptions) {  
    guard let windowScene = (scene as? UIWindowScene) else { return }  
    // ...  
}
```

The screenshot shows a pull request interface in Xcode. At the top, it says "0 of 34 tasks done". The main area displays two commits:

Benedikt Terhechte (f1f89) 29.06.19, 19:03
Catalyst: Support for multiple windows
Catalyst: Support for multiple windows

As we have a custom storyboard, we don't get the auto-generated "New Window" entry. First Responder also currently doesn't offer the correct @IBAction. It has to be defined in the storyboard in the 'User Defined Actions' on 'First Responder': "requestNewScene":

Benedikt Terhechte (d6604) 29.06.19, 19:03
Various Bugfixes and Changes
Various Bugfixes and Changes

Benedikt Terhechte (228c3) 29.06.19, 19:03
Catalyst: Support For Context Menus (right click on a detail table view result cell)
Catalyst: Support For Context Menus (right click on a detail table view result cell)

This will add context menus to each table row. As an example, we're staring the rows.
This just required adding one method to our tableView

The right side shows the code diff for **AppDelegate.swift**:

```
59 func openResult(result: TrainResult) {
60     let window = UIApplication.shared.keyWindow
61     let storyboard = UIStoryboard(name: "Result", bundle: nil)
62     guard let controller = storyboard.instantiateInitialViewController()
63     else { fatalError("Missing Settings") }
64     controller.modalPresentationStyle = .automatic
65     controller.result = result
66     window?.rootViewController?.present(controller, animated: true)
67 }
68
69 guard let wrappingController = window?.rootViewController?.presentingViewController
70 else { return }
71 wrappingController.result = result
72 window?.rootViewController?.present(wrappingController, animated: true)
73
74
75
76
77
78
79
80 }
```

Below it is the **Main.storyboard** file:

```
183 <!--Detail List Controller-->
184 <!-->
```

Demo

[https://github.com/terhechte/
CatalystMaterial](https://github.com/terhechte/CatalystMaterial)