

**It's time to build
for watchOS**



NOT CONFIDENTIAL



I'm Hidden

@hiddevdploeg An Indie Software Maker based in Manchester



Vekt



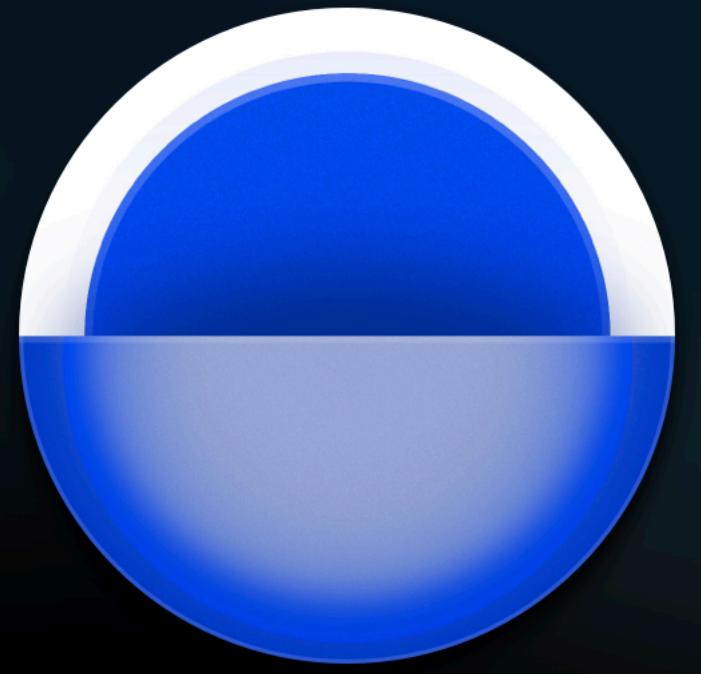
Brush



NowPlaying



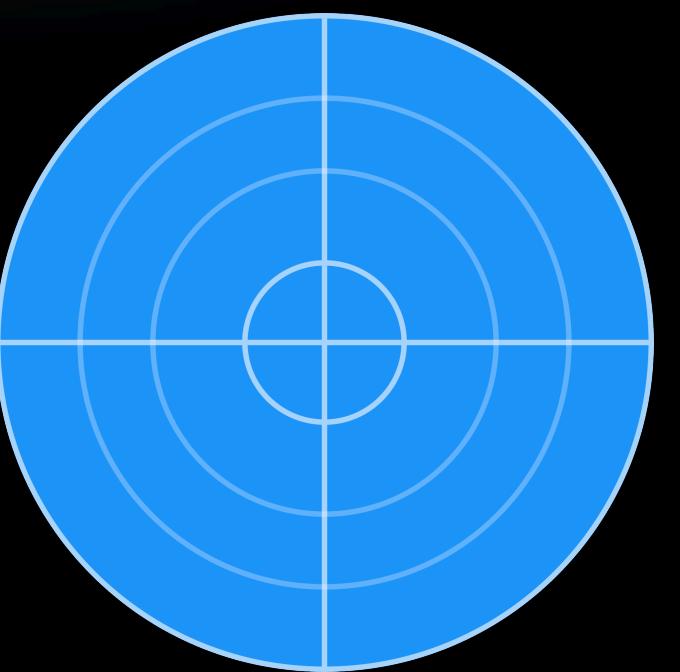
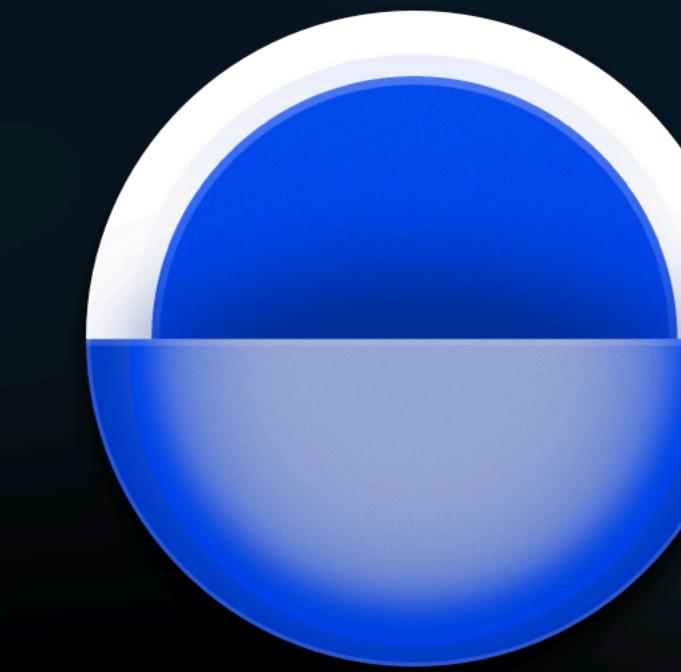
Volo



Gola



Petey





Get ready for watchOS 10.

Reimagine your apps and make them even more dynamic, colorful, and glanceable with new SwiftUI APIs and widgets for the Smart Stack. Take workouts to the next level by creating and scheduling custom workouts that can be imported into the Workout app and with access to new cycling metrics, high frequency motion data, and validated calories for golf.



Fresh new design options

Make your app fit right in with the exciting, modern design of watchOS 10. Take advantage of the latest SwiftUI APIs for system materials, tab views, split views, and more.

Enough about me

Let's Start...



The State Of watchOS

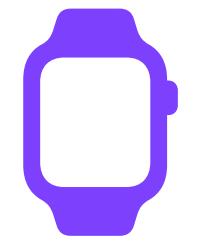
Where are we today

100M+

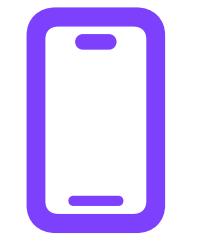
That's a lot of people



7 Devices & 5 Screen sizes



7 Devices & 5 Screen sizes



22 Devices & 10 Screen sizes



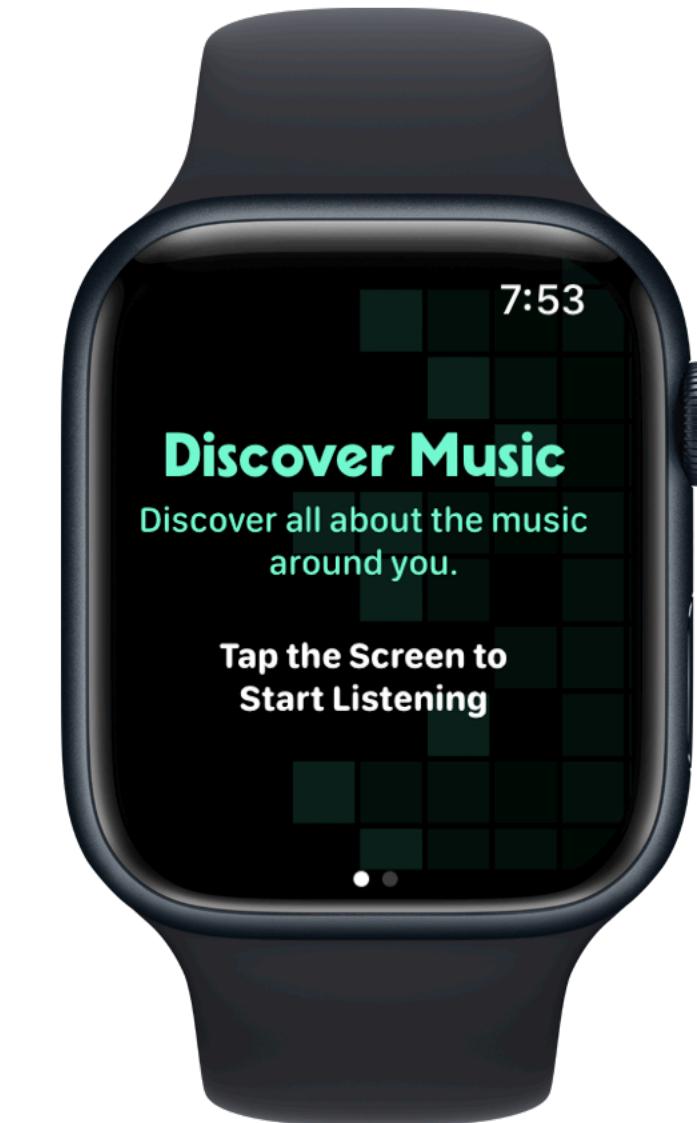
40mm
162 × 197



41mm
176 × 215



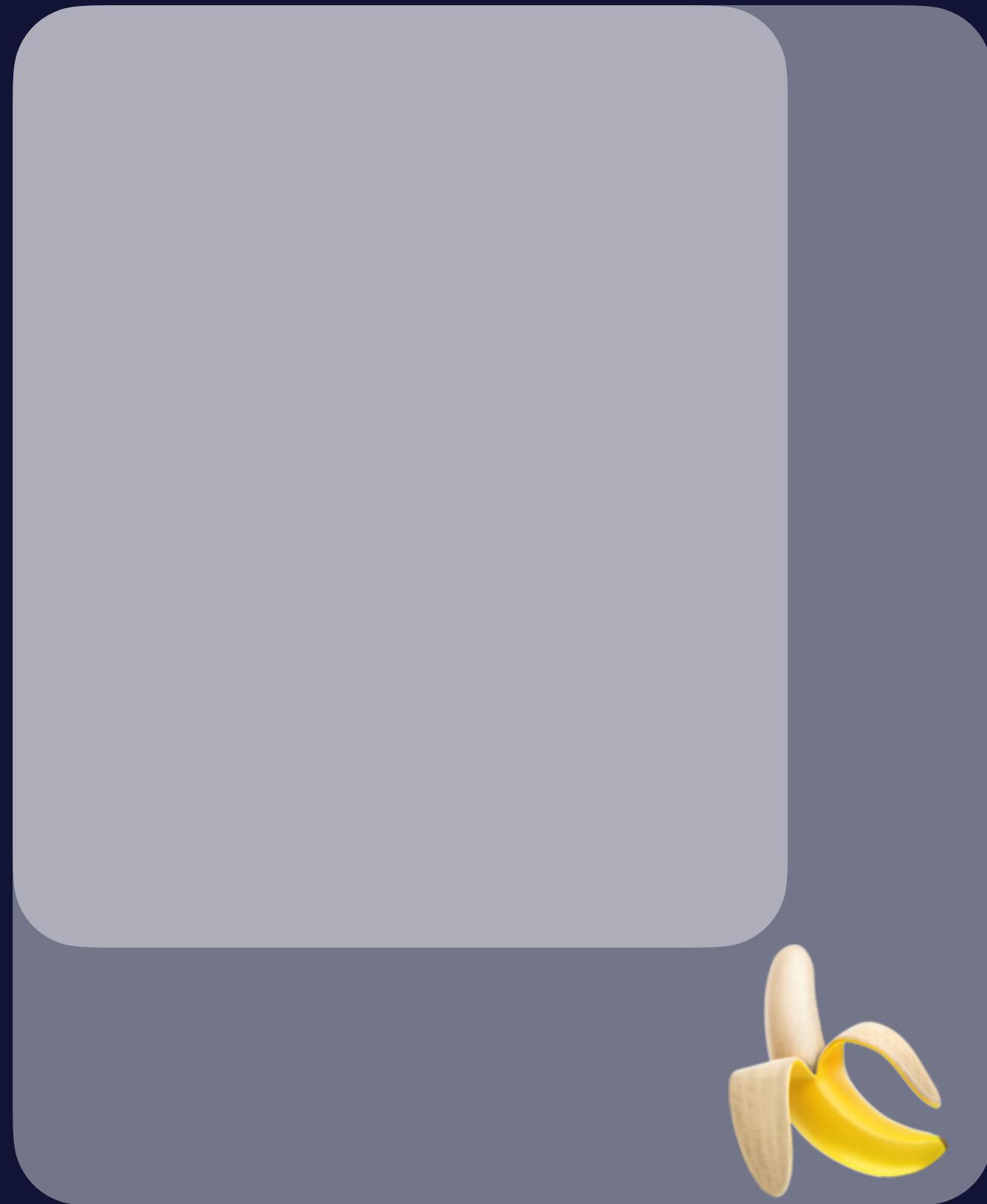
44mm
184 × 224



45mm
198 × 242



49mm
205 × 251



**API Availability has grown
A lot since watchOS 1**



SwiftCharts

CallKit

iCloud

AppIntents

Contacts

MusicKit

ShazamKit

CoreAudio

CoreLocation

DeviceDiscovery

GameKit

CoreML

CoreMotion

HomeKit

StoreKit

WeatherKit

PhotoKit

HealthKit

MapKit

PassKit

CoreData

iCloud

MusicKit

CoreML

StoreKit

HealthKit

SwiftCharts

AppIntents

ShazamKit

CoreMotion

WeatherKit

MapKit

CallKit

Contacts

CoreAudio

HomeKit

PhotoKit

PassKit

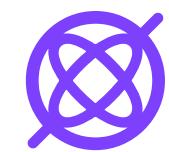
All the cool kids are here..

Some APIs are better on
watchOS than they are on **iOS**.

There are a lot of sensors...

 Accelerometer

 Magnetometer

 Gyroscope

 Blood Oxygen

 HeartRate

 Altimeter

 GPS

 Water Submersion

 Water Temperature

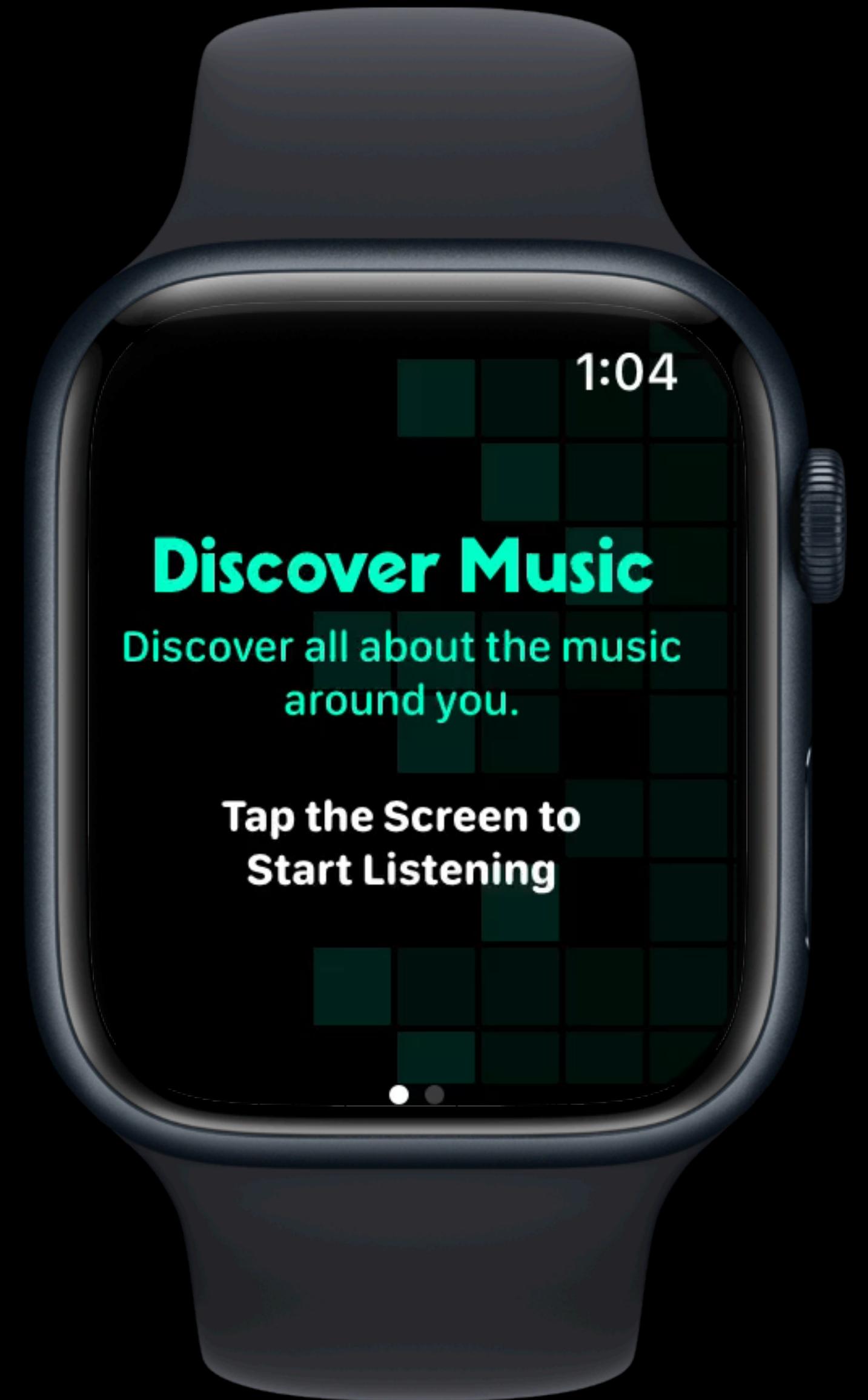
 Pedometer

**Now we know what we're
dealing with...**

**How do you build & design a
great watchOS app?**

One quick design tip...

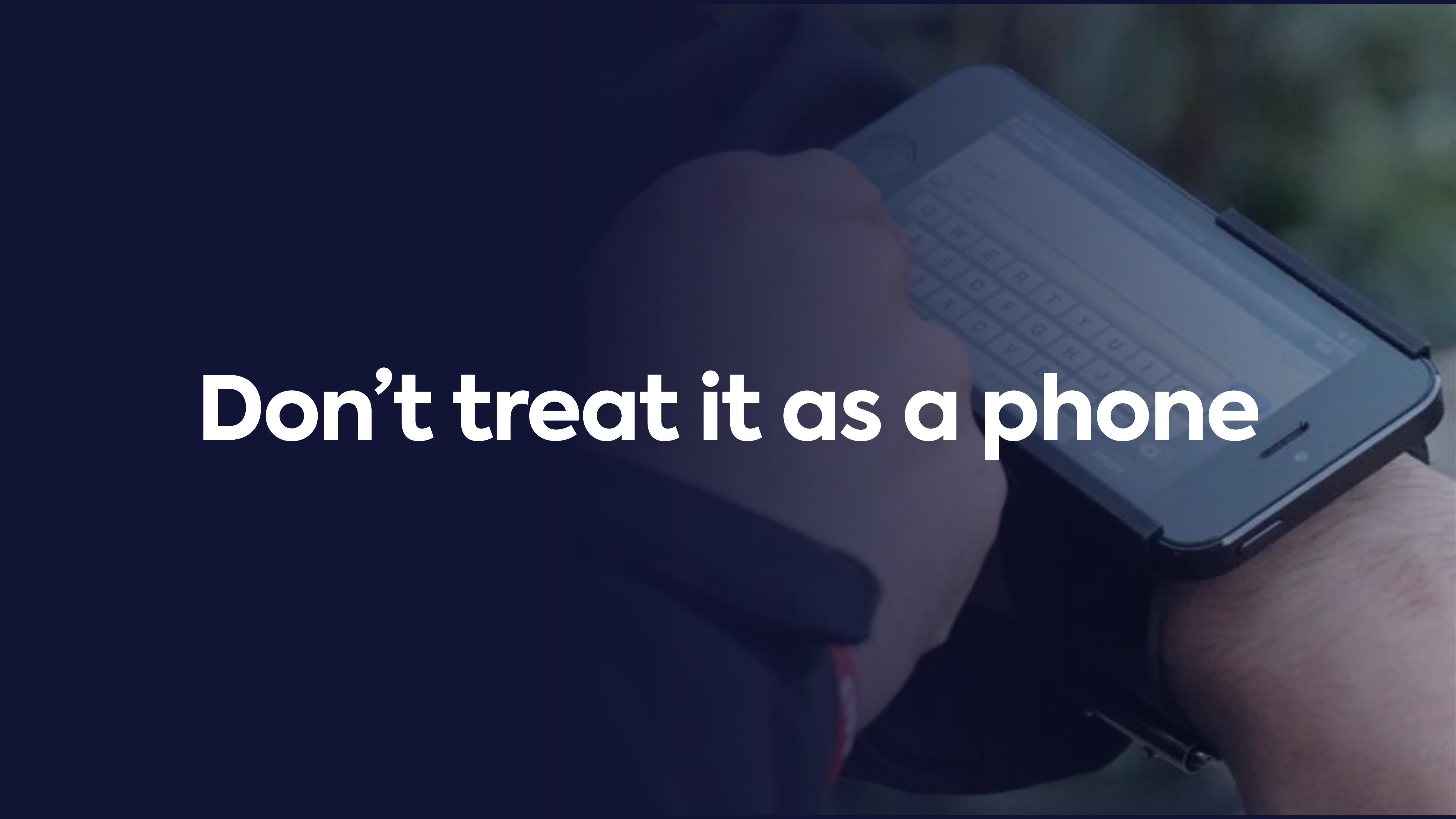
**What's the main goal of this
View/App?**



KEEP FOCUS



Seconds, not minutes



Don't treat it as a phone

What is Glanceable?

Consistent Relevant information

Highly Specialised Short Interactions

Focused Timely

Add Complications



Even if it just opens your app

If you support lock-screen
widgets today, consider
building a watch app with the
same functionality

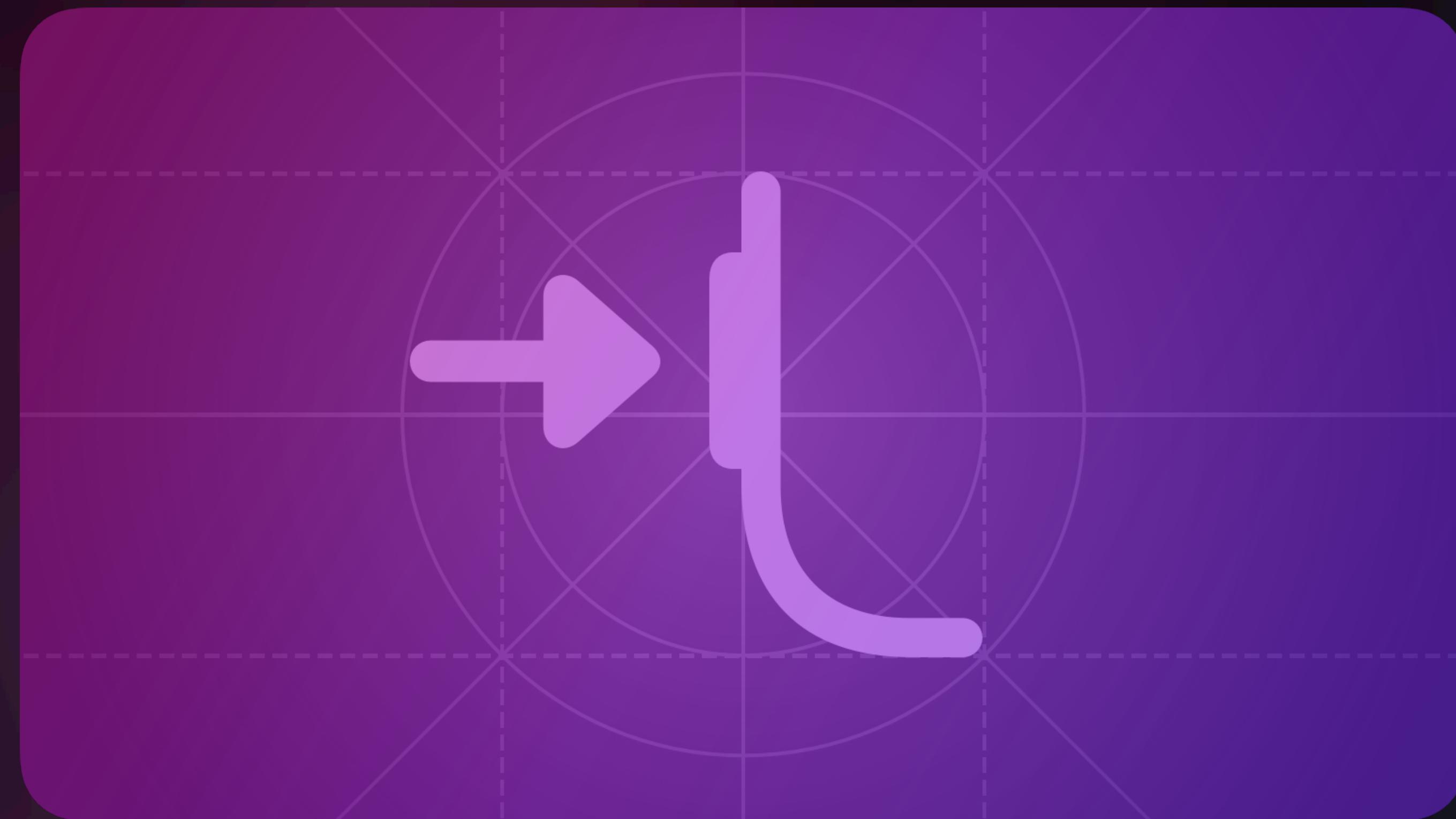


⌚ Digital Crown

Consider the digital crown
can enable fun hardware
interactions



The haptics just feel soooooo good



The Action Button

```
import AppIntents
```

`StartWorkoutIntent`

`PauseWorkoutIntent`

`ResumeWorkoutIntent`

```
struct WorkoutIntent : StartWorkoutIntent {

    @Parameter(title: "Workout Type")
    var workoutStyle: WorkoutType

    static var title : LocalizedStringResource = "Start Volo"

    static var suggestedWorkouts: [Self] = [
        WorkoutIntent(style: .kiteSurfing),
        WorkoutIntent(style: .wingSurfing)
    ]

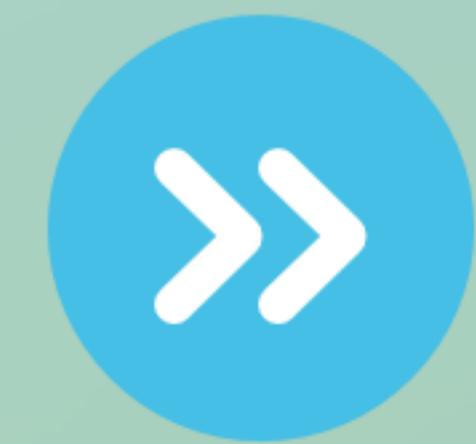
    typealias WorkoutStyle = WorkoutType

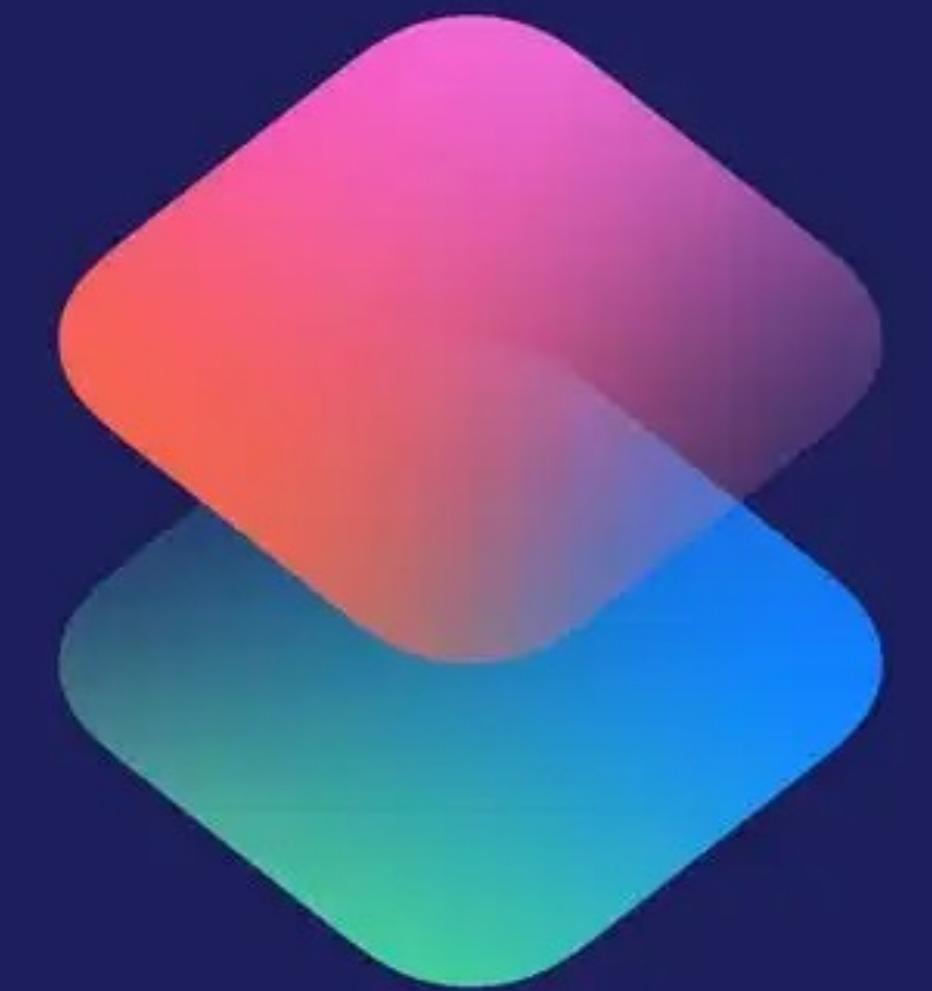
    var displayRepresentation: DisplayRepresentation {
        return workoutStyle.title
    }

    @MainActor
    func perform() async throws -> some IntentResult {
        await SessionStore.shared.startSurfing(type: workoutStyle)
        return .result(actionButtonIntent: ChangeLayoutIntent())
    }

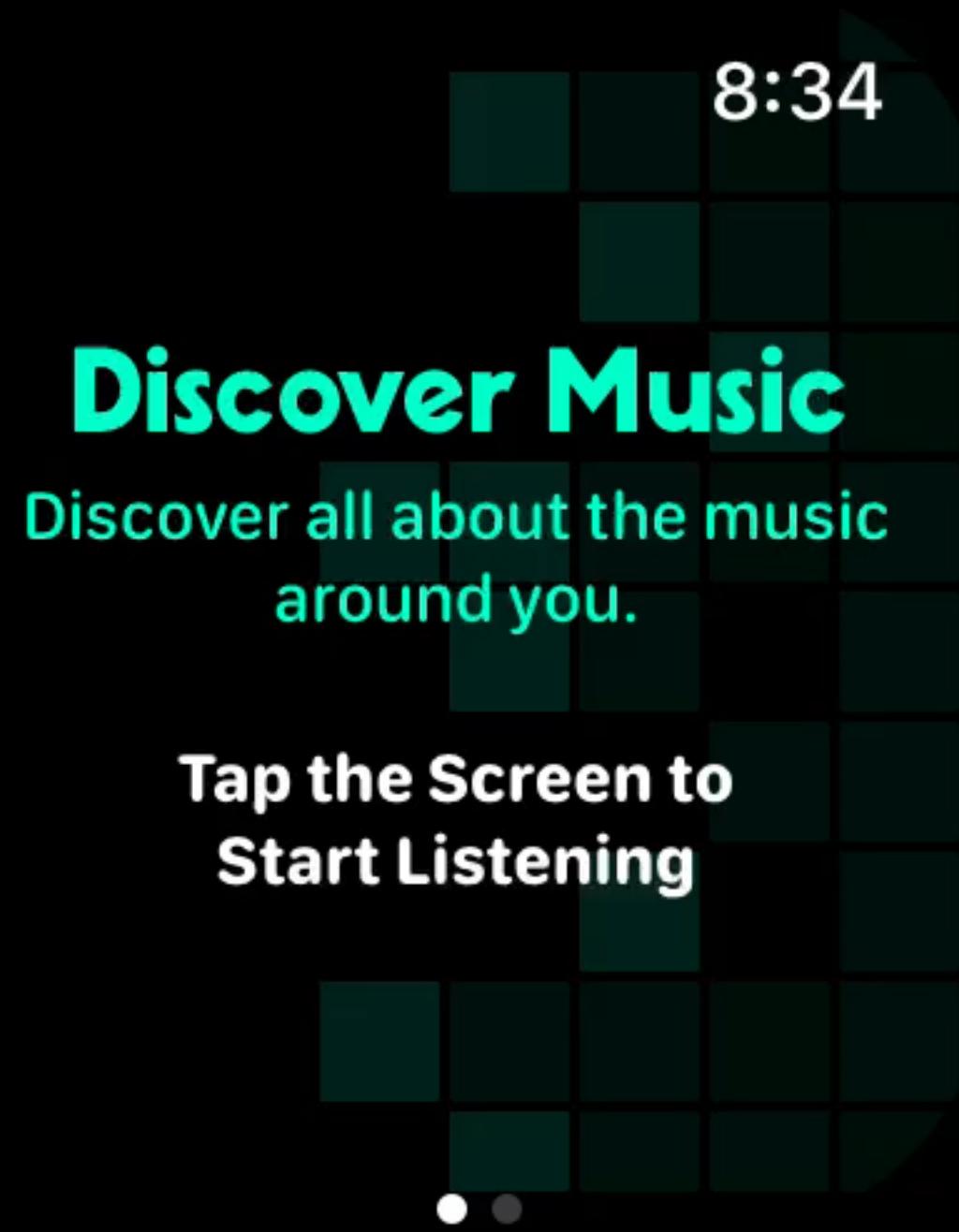
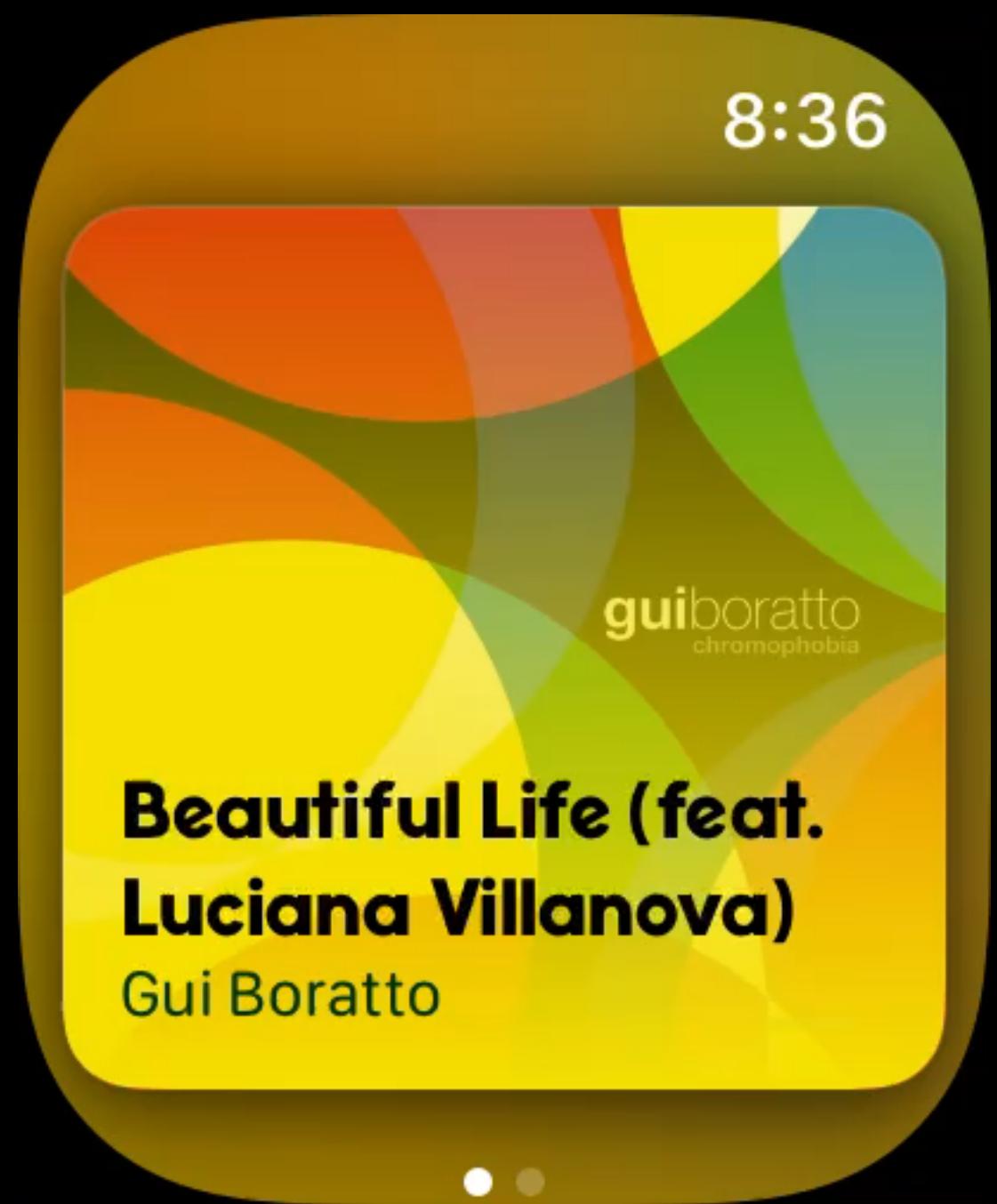
    static var parameterSummary : some ParameterSummary {
        Summary("Start \$(workoutStyle)")
    }
}
```

```
var displayRepresentation: DisplayRepresentation {  
    return workoutStyle.title  
}  
  
@MainActor  
func perform() async throws -> some IntentResult {  
    await SessionStore.shared.startGaming(type: workoutStyle)  
    return .result(actionButtonIntent: ChangeLayoutIntent())  
}  
  
static var parameterSummary : some ParameterSummary {  
    Summary("Start \$(\\.workoutStyle)")  
}  
}
```





Keep the Always On display in mind



It's on iOS now too!

```
@Environment(\.isLuminanceReduced) var isReduced
```

. scenePadding()





<

10:09

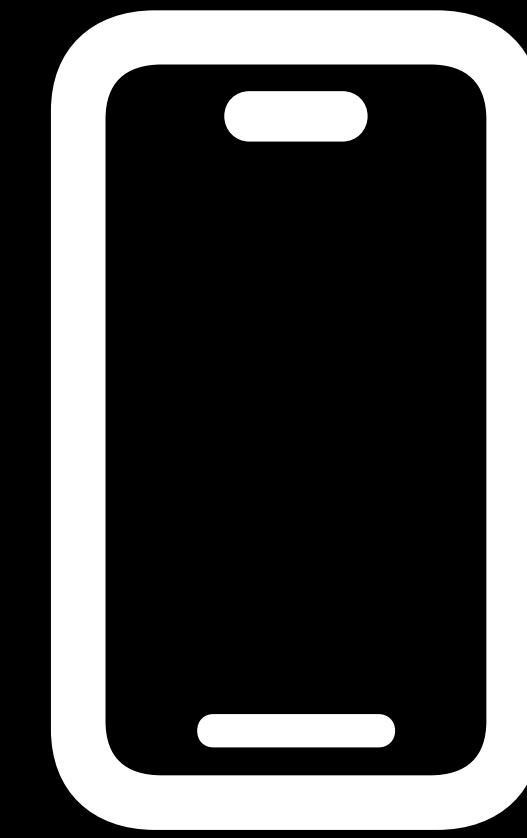
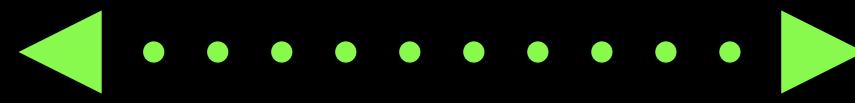
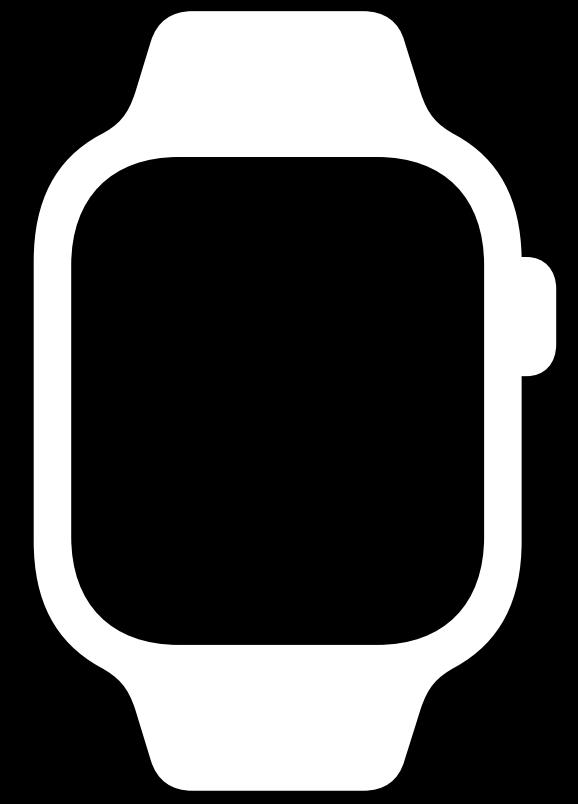
Hello World

Scene padding

Regular padding

Text with no padding

Button



The right tool for the right job

Source, Destination	Relies on companion iPhone	Supports Family Setup	Best For	
iCloud Keychain synchronization	All devices	No	Yes	Infrequently changing data
Core Data with CloudKit	All devices and iCloud	No	Yes	Structured data
Watch Connectivity	Paired iPhone and Watch	Yes	No	Optimization
URL Sessions	Server	No	Yes	Most server communication
Sockets	Server	No	Yes	Streaming audio

Persistence across platforms

Note: watchOS does find whenever they think is best to start syncing

- Make sure that when the user does install your app on one of their other devices, that data is synced over.
- iCloud (KeyValStore, CoreData, Keychain) is the way to go

NSUbiquitousKeyValueStore()

NSUbiquitousKeyValueStore()

Who named this thing?!

NSUbiquitousKeyValueStore

An iCloud-based container of key-value pairs you use to share data among instances of your app running on a user's connected devices.

iOS 5.0+

macOS 10.7+

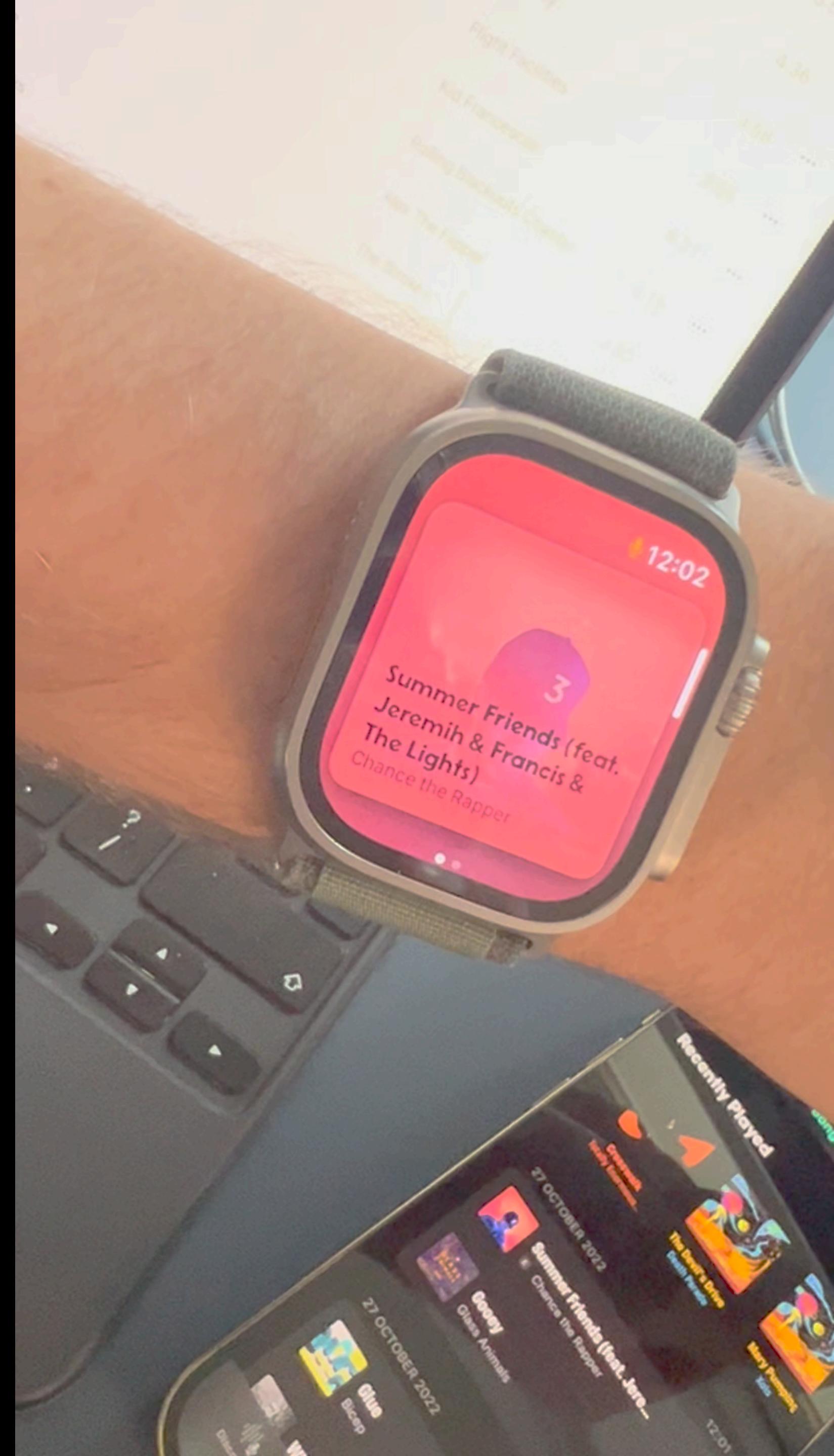
Mac Catalyst 13.1+

tvOS 9.0+

watchOS 9.0+

**What if I want my data to sync
directly?**

WatchConnectivity is your (old) friend



```
session.sendMessage(["songToSave": track.songID,  
                    replyHandler: nil])
```

Let's build something!

We're building a HeartRate Monitor!

1. Setup Project
2. Setup Navigation and View
3. Data & Persistence
4. Haptics
5. Complications
6. Update it for watchOS 10

We're building a HeartRate Monitor!

1. Setup Project
2. Setup Navigation and View
3. Data & Persistence
4. Haptics
5. Complications
6. Update it for watchOS 10
7. Action Button support
8. Digital Crown

A photograph showing a person's hands working on a complex electronic assembly. The hands are positioned over a dark-colored breadboard or circuit board, with various wires and components attached. In the background, there is a shelf filled with numerous books, suggesting a library or a technical resource center.

Let's code!

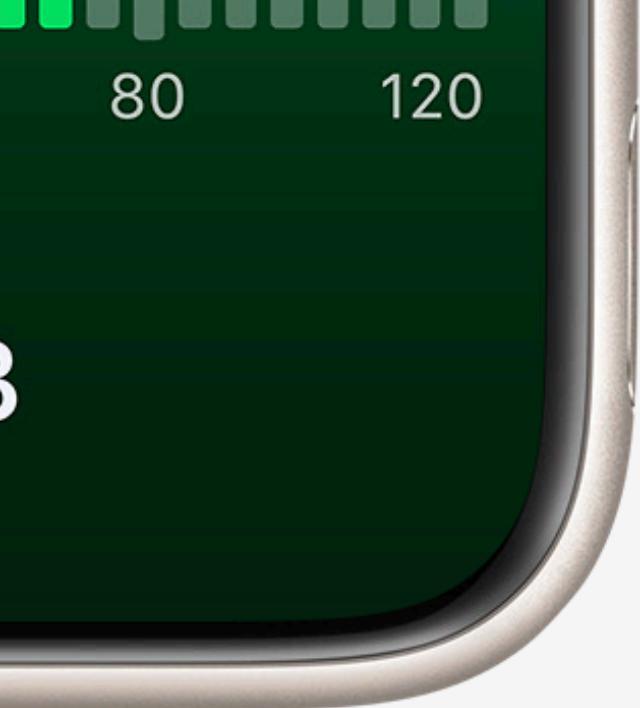
watchOS 10

So, what's new this year?

100%

more Snoopy & Woodstock







1. New Visual Lanuage
2. New Navigation Patterns
3. Smart Stack Widgets





Default



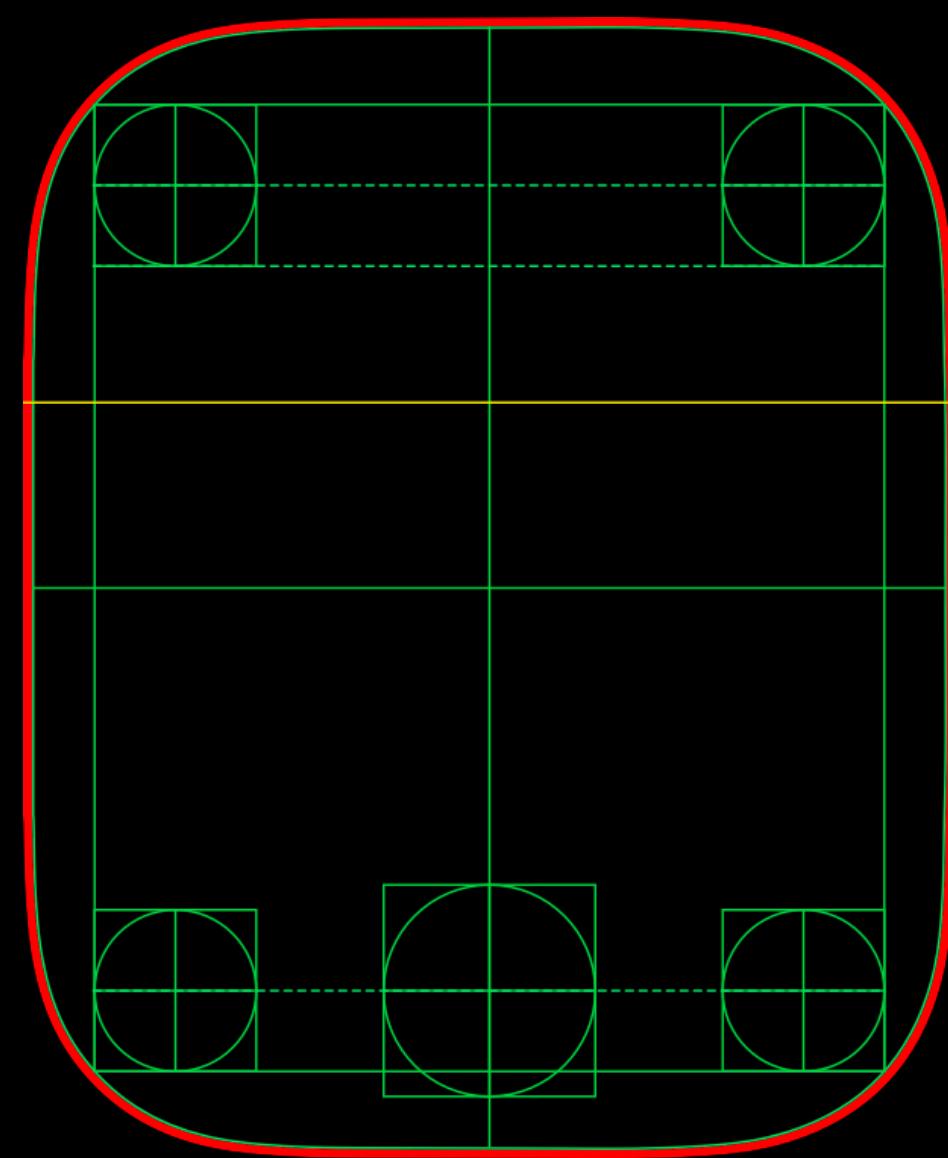
Large



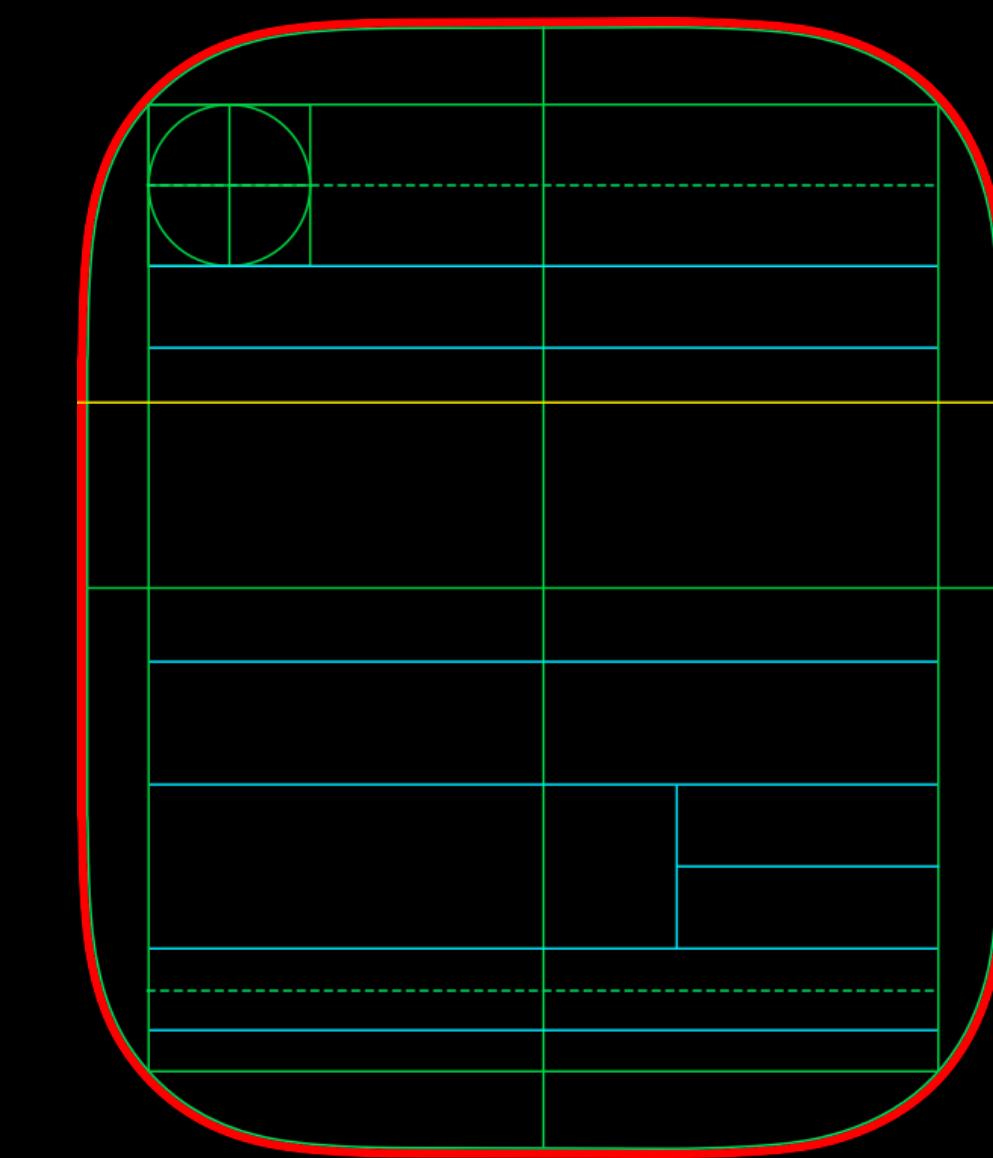
Child View



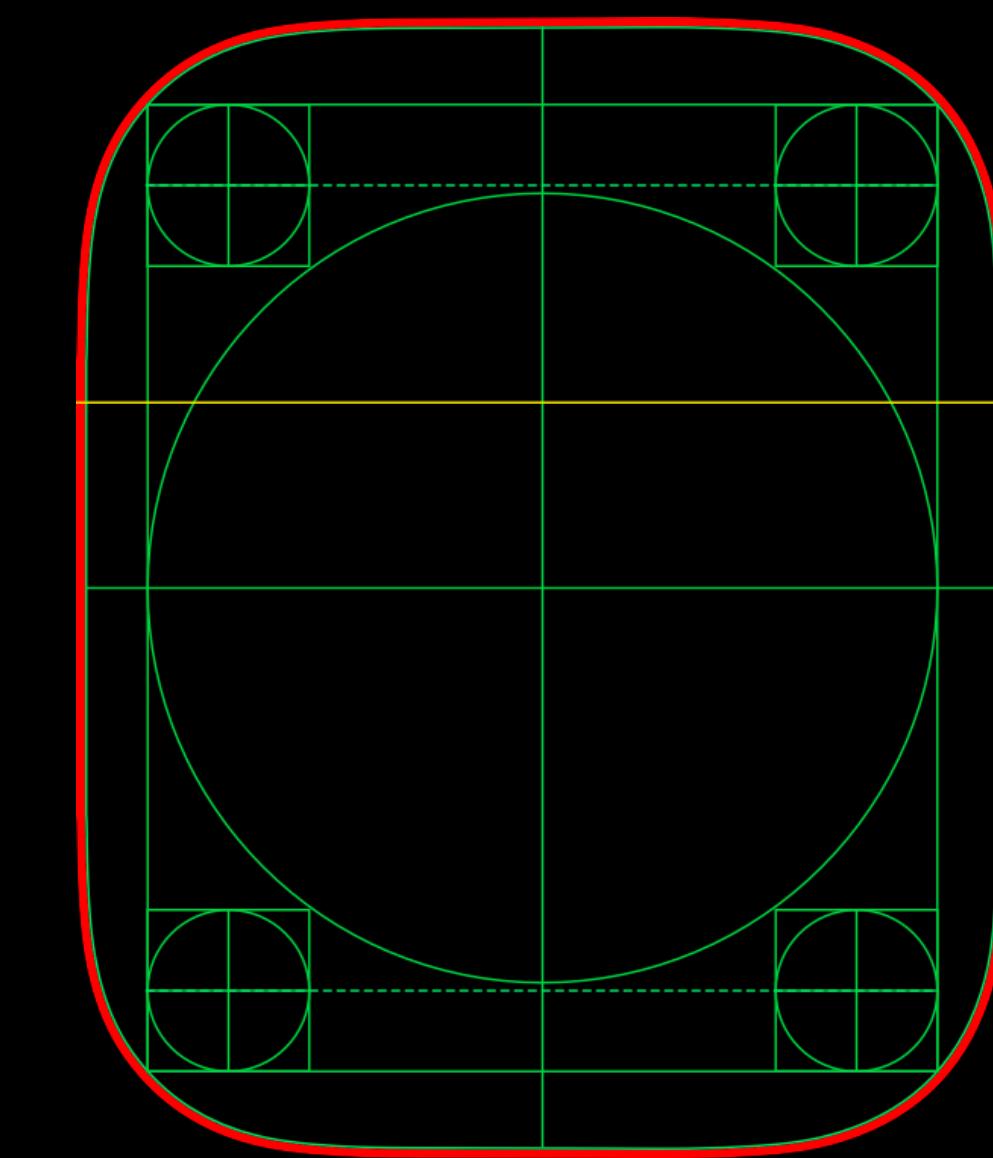
Center



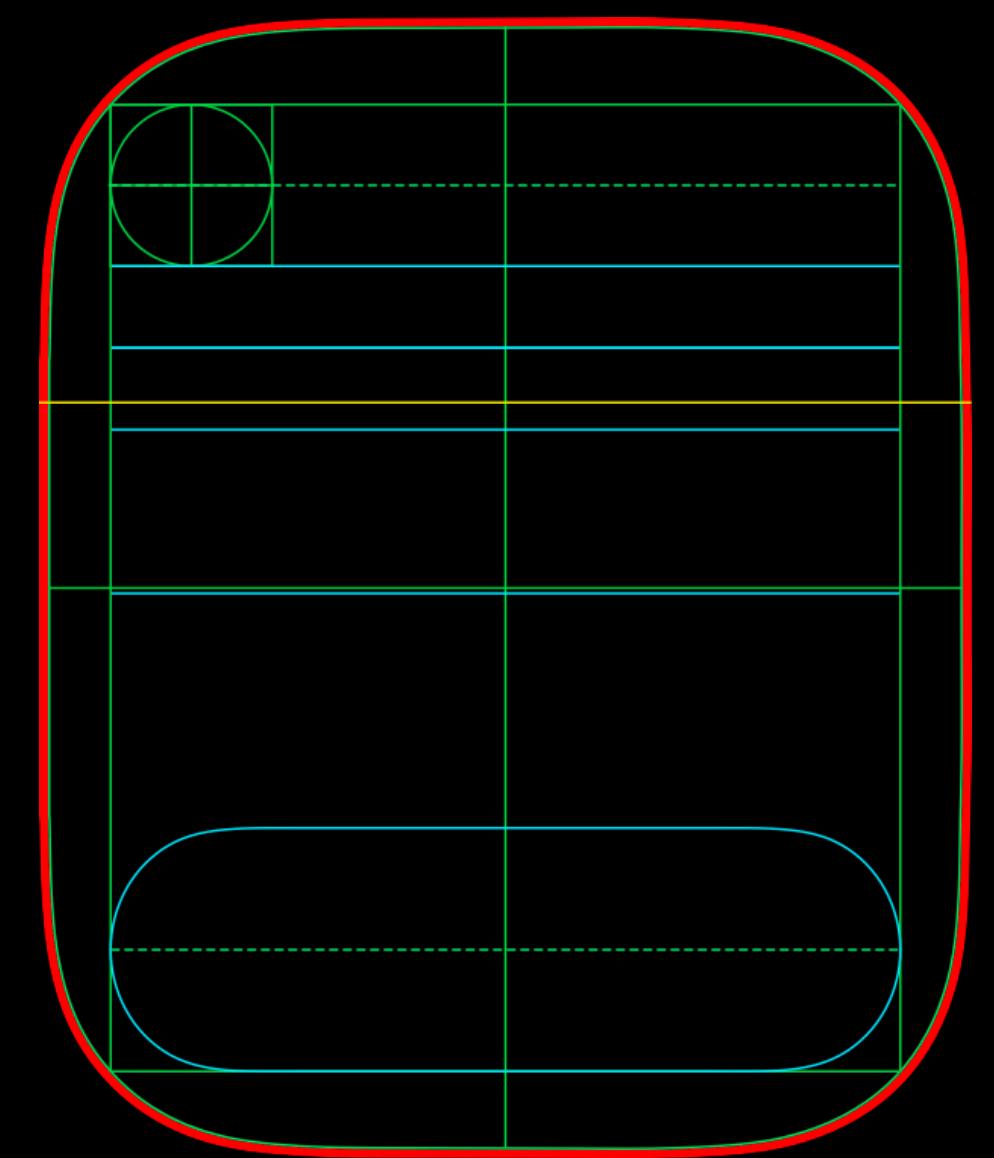
Bottom Controls



Info



Dial



Pill



Let's update!

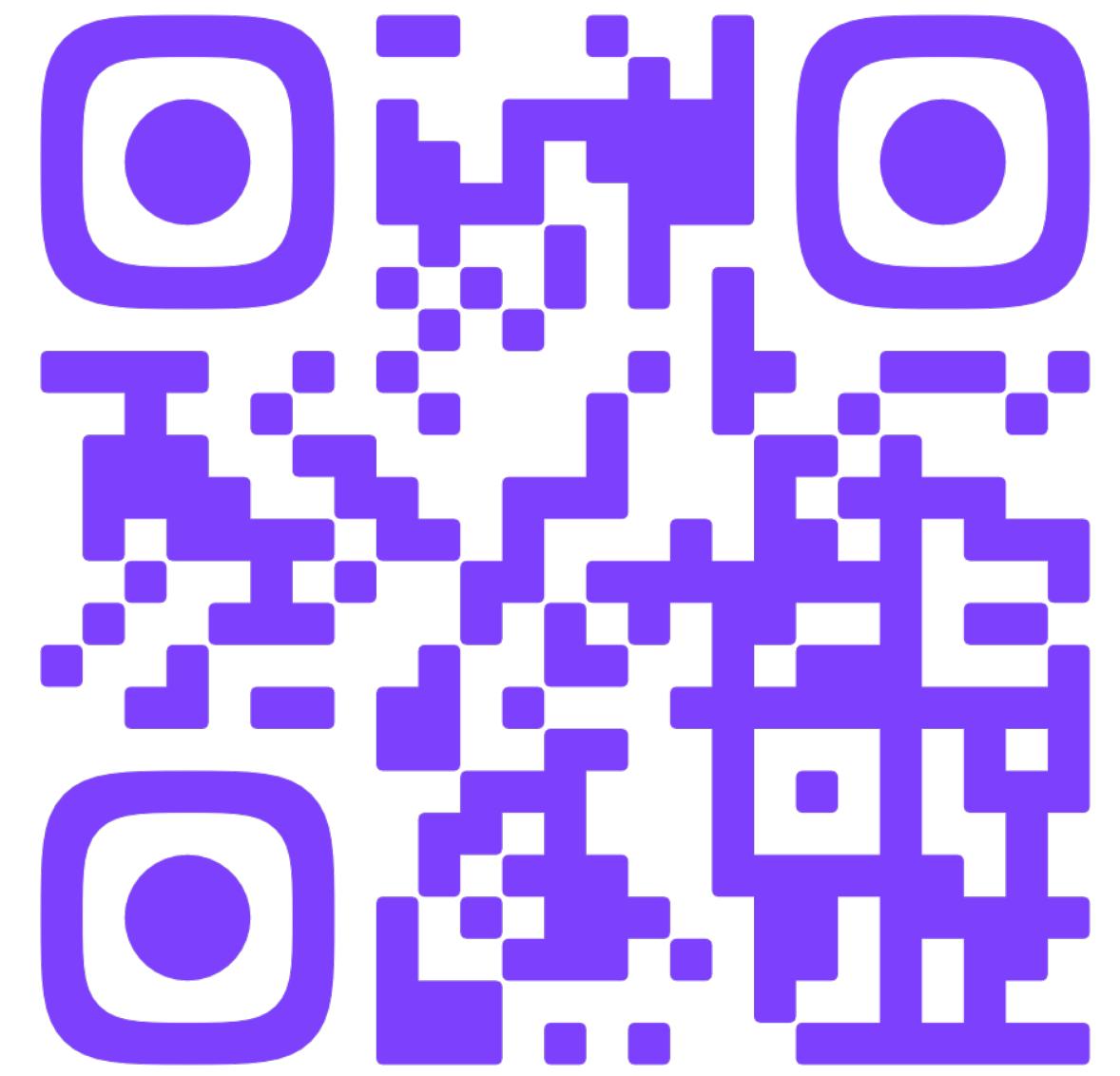
Xcode 15 required from here on...



Have fun!

@hiddevdploege

Thank you & tot ziens! 🙌



That's my website