SwiftCrossUI

Swift Apps - Everywhere

by Mia Koring & moreSwift

whoami

- Mia
- Active contributor to SwiftCrossUI
- Other open source projects, mostly in Swift
- Amethyst Browser & Authenticator

What is SwiftCrossUI

- declarative
- reactive
- easy to use UI Framework
- wrapping native imperative frameworks
- basically SwiftUI but everywhere

Logo by @sirlan

Supported Platforms

& used Frameworks

UIKit & AppKit

WinUI 3

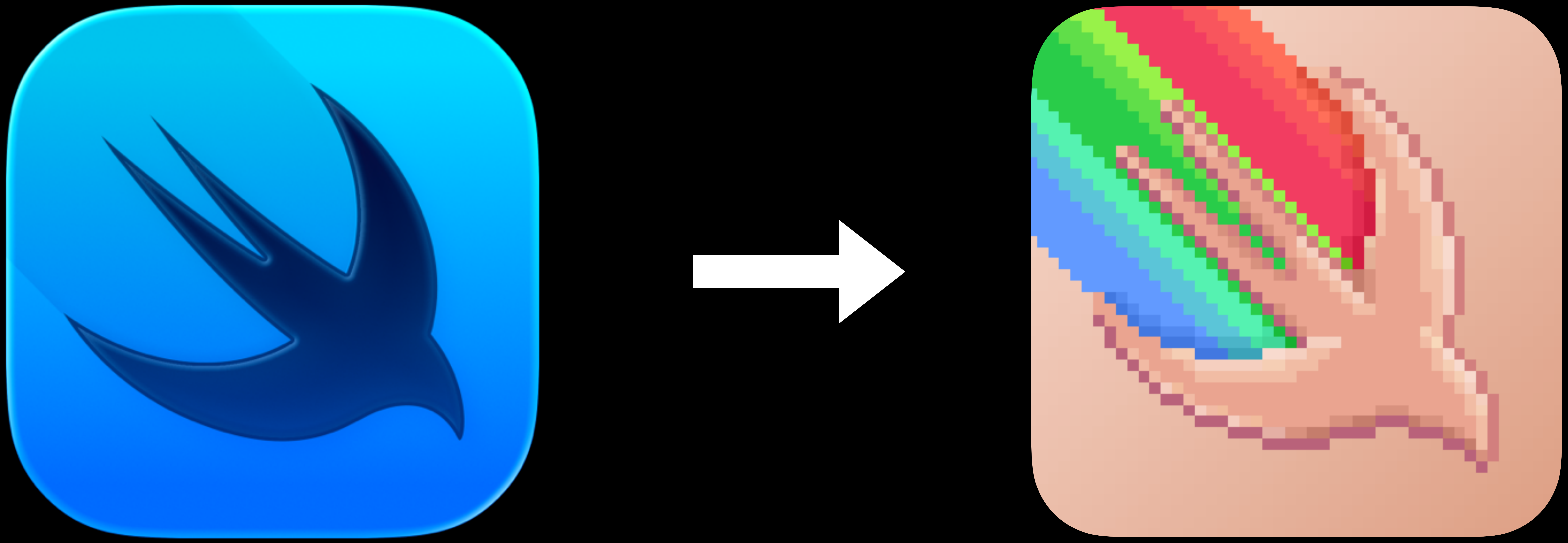
Gtk 4

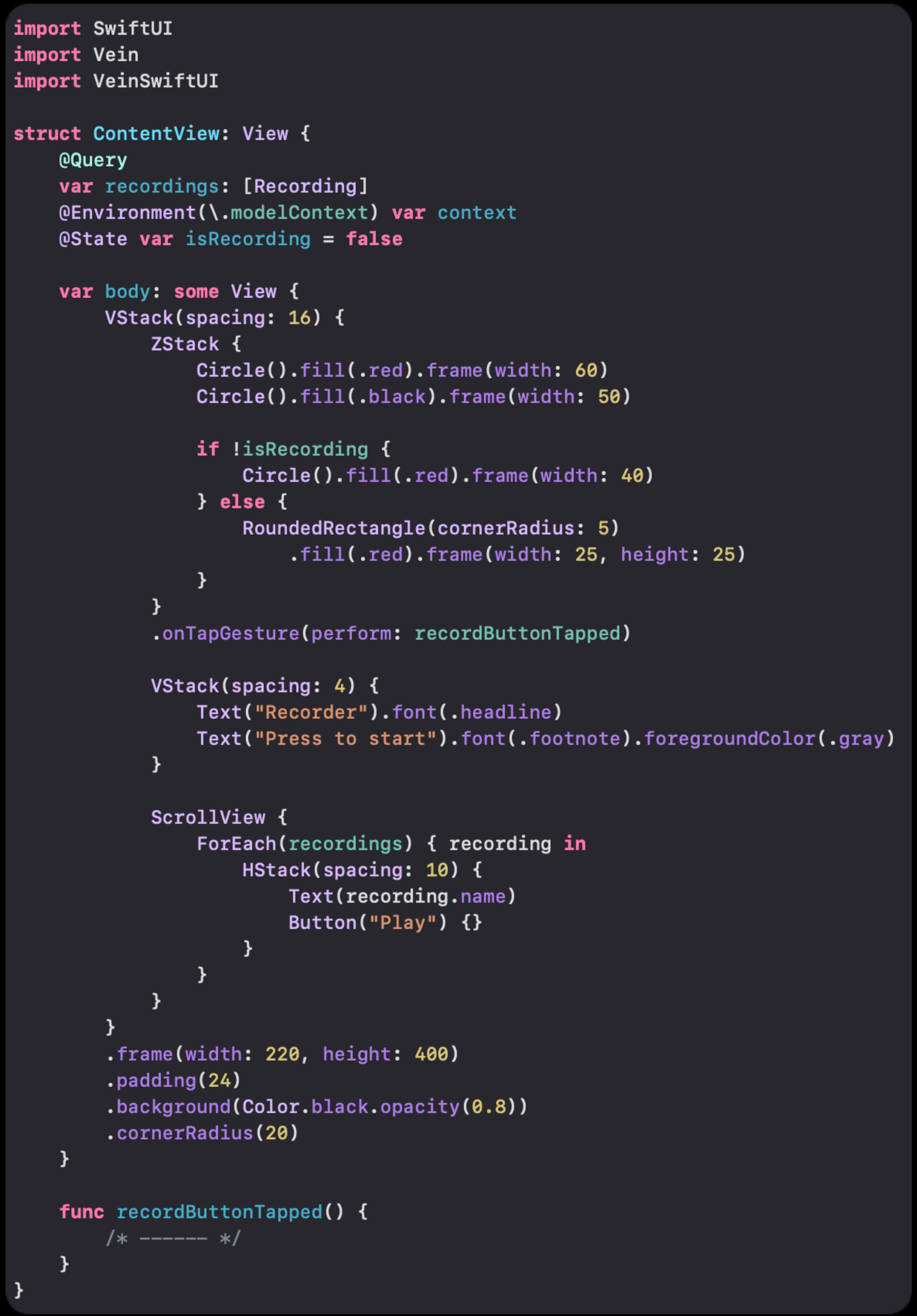
Android Views

(coming soon)

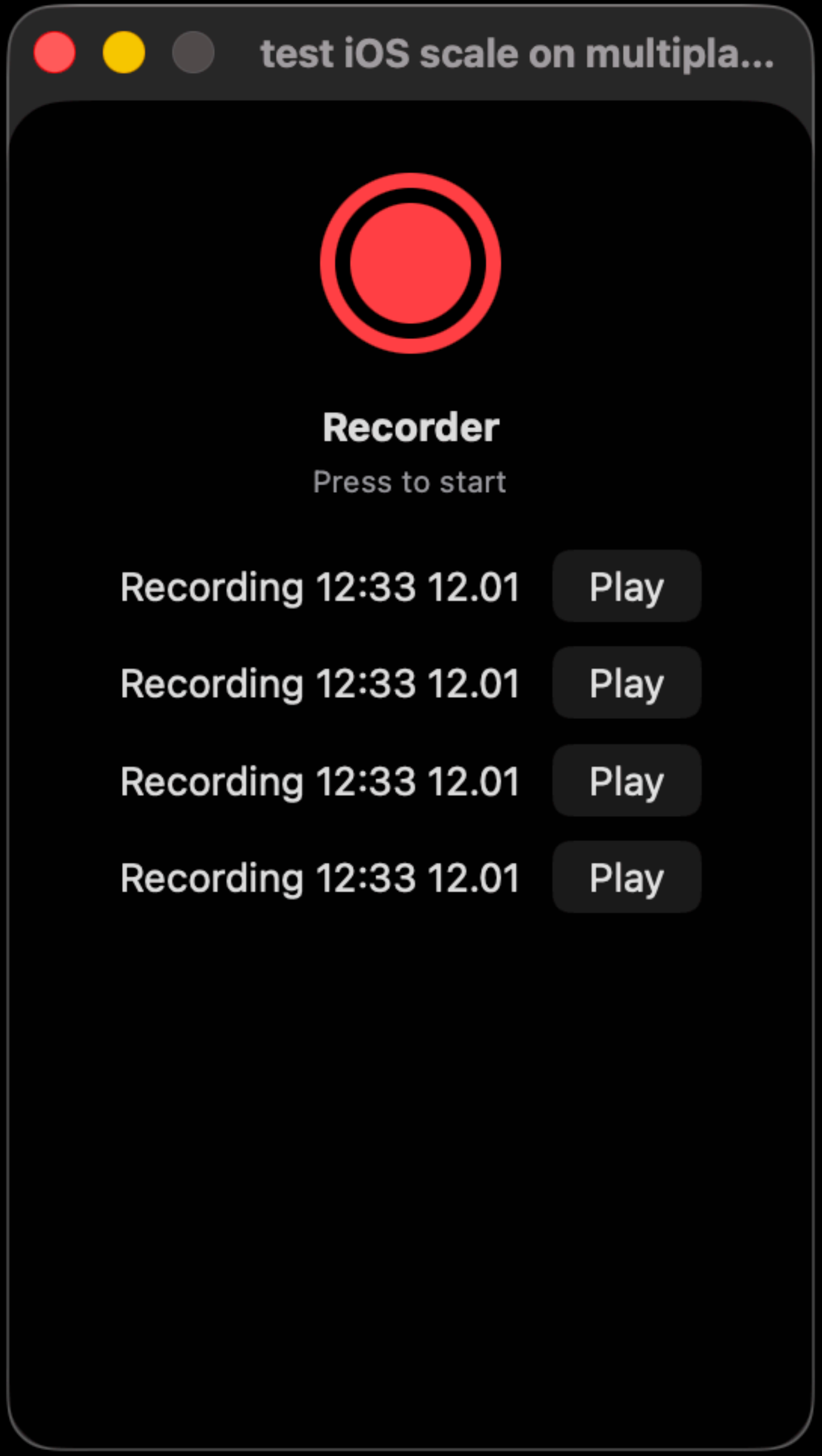
& more

Lets port some SwiftUI Code





```swift
import SwiftUI
import Vein
import VeinSwiftUI

struct ContentView: View {
    @Query
    var recordings: [Recording]
    @Environment(\.modelContext) var context
    @State var isRecording = false

    var body: some View {
        VStack(spacing: 16) {
            ZStack {
                Circle().fill(.red).frame(width: 60)
                Circle().fill(.black).frame(width: 50)

                if !isRecording {
                    Circle().fill(.red).frame(width: 40)
                } else {
                    RoundedRectangle(cornerRadius: 5)
                        .fill(.red).frame(width: 25, height: 25)
                }
            }
            .onTapGesture(perform: recordButtonTapped)

            VStack(spacing: 4) {
                Text("Recorder").font(.headline)
                Text("Press to start").font(.footnote).foregroundColor(.gray)
            }

            ScrollView {
                ForEach(recordings) { recording in
                    HStack(spacing: 10) {
                        Text(recording.name)
                        Button("Play") {}
                    }
                }
            }
        }
        .frame(width: 220, height: 400)
        .padding(24)
        .background(Color.black.opacity(0.8))
        .cornerRadius(20)
    }

    func recordButtonTapped() {
        /* ------ */
    }
}
```



test iOS scale on multipla...

Recorder

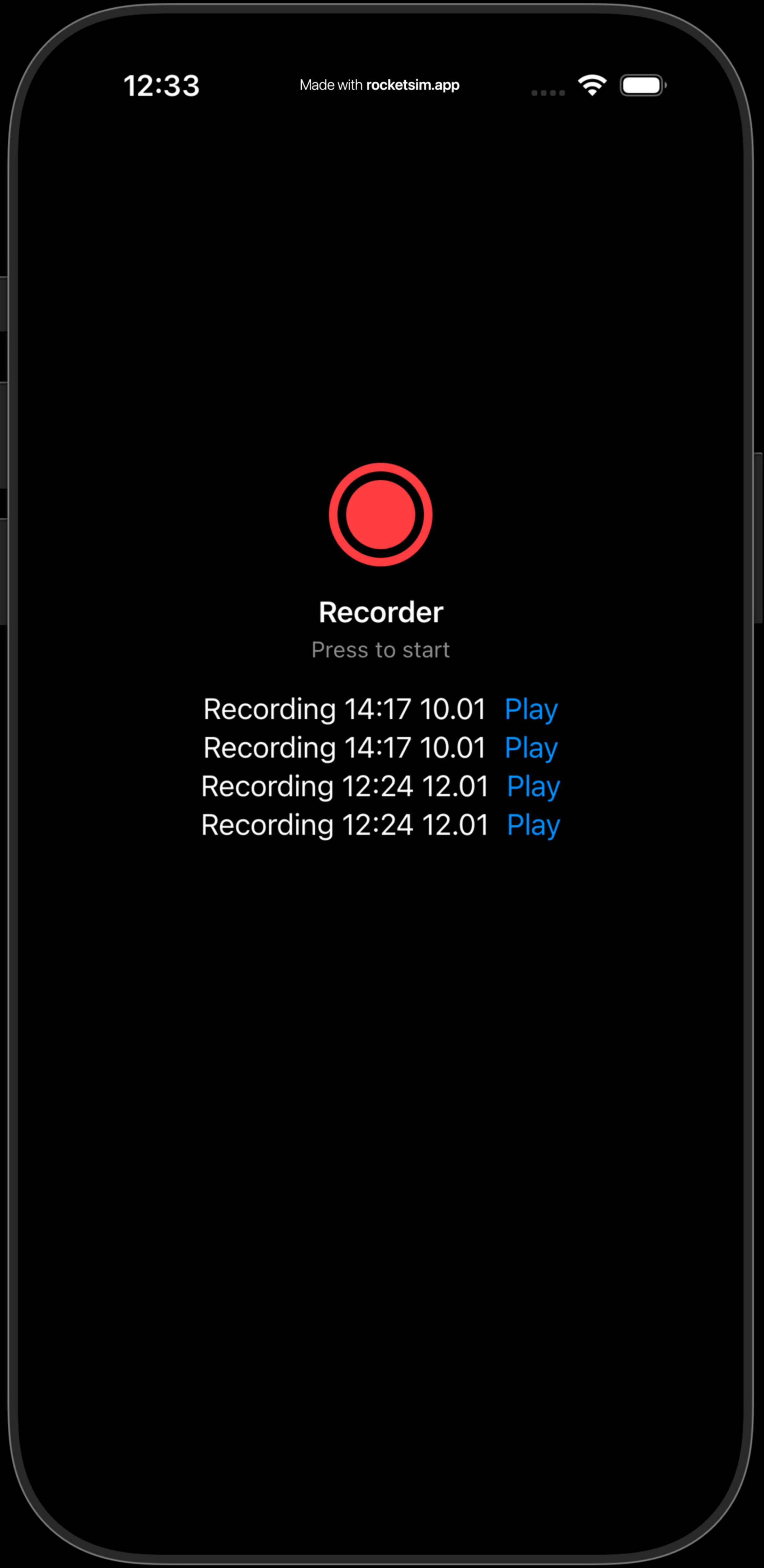
Press to start

Recording 12:33 12.01 Play

Recording 12:33 12.01 Play

Recording 12:33 12.01 Play

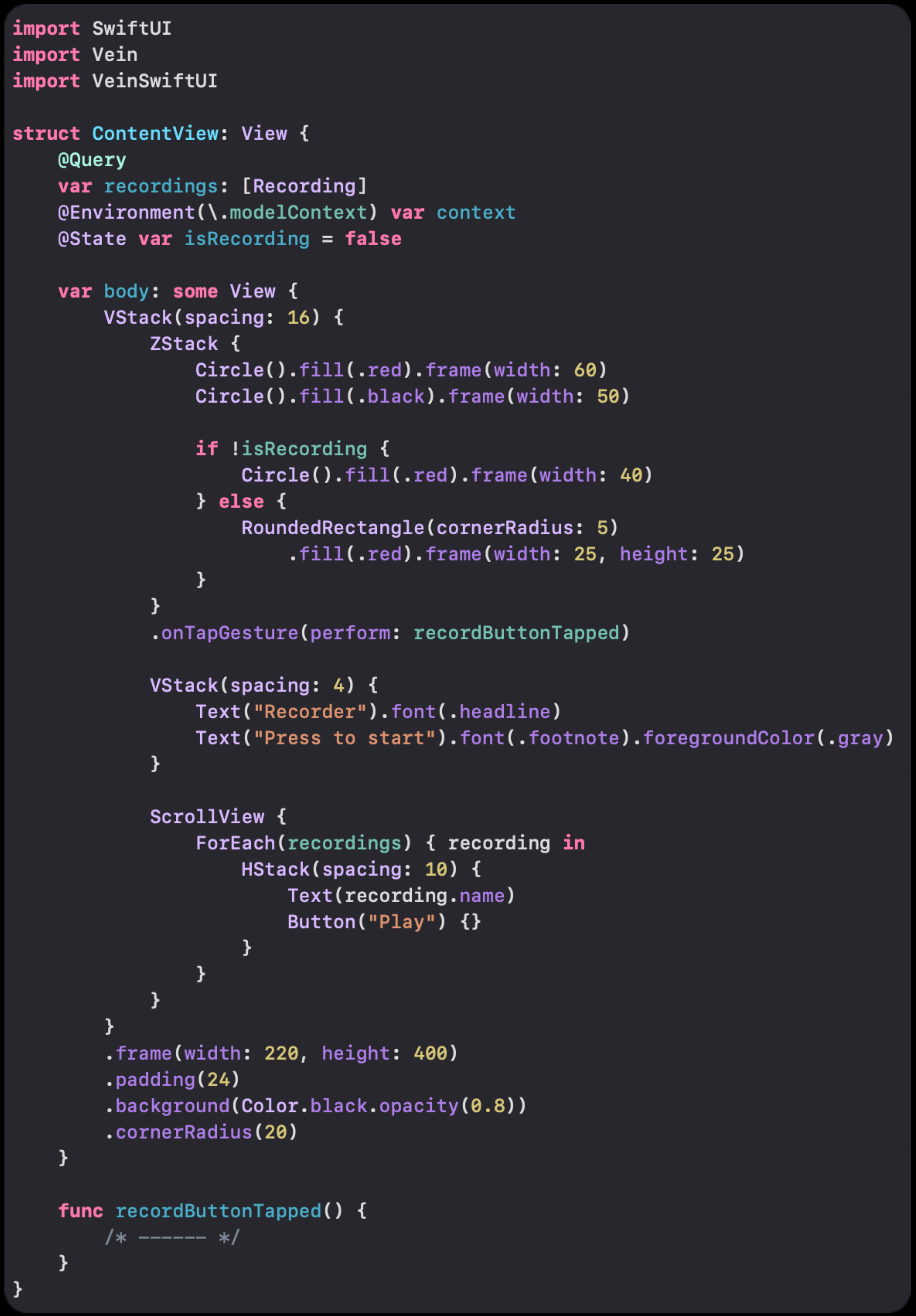
Recording 12:33 12.01 Play



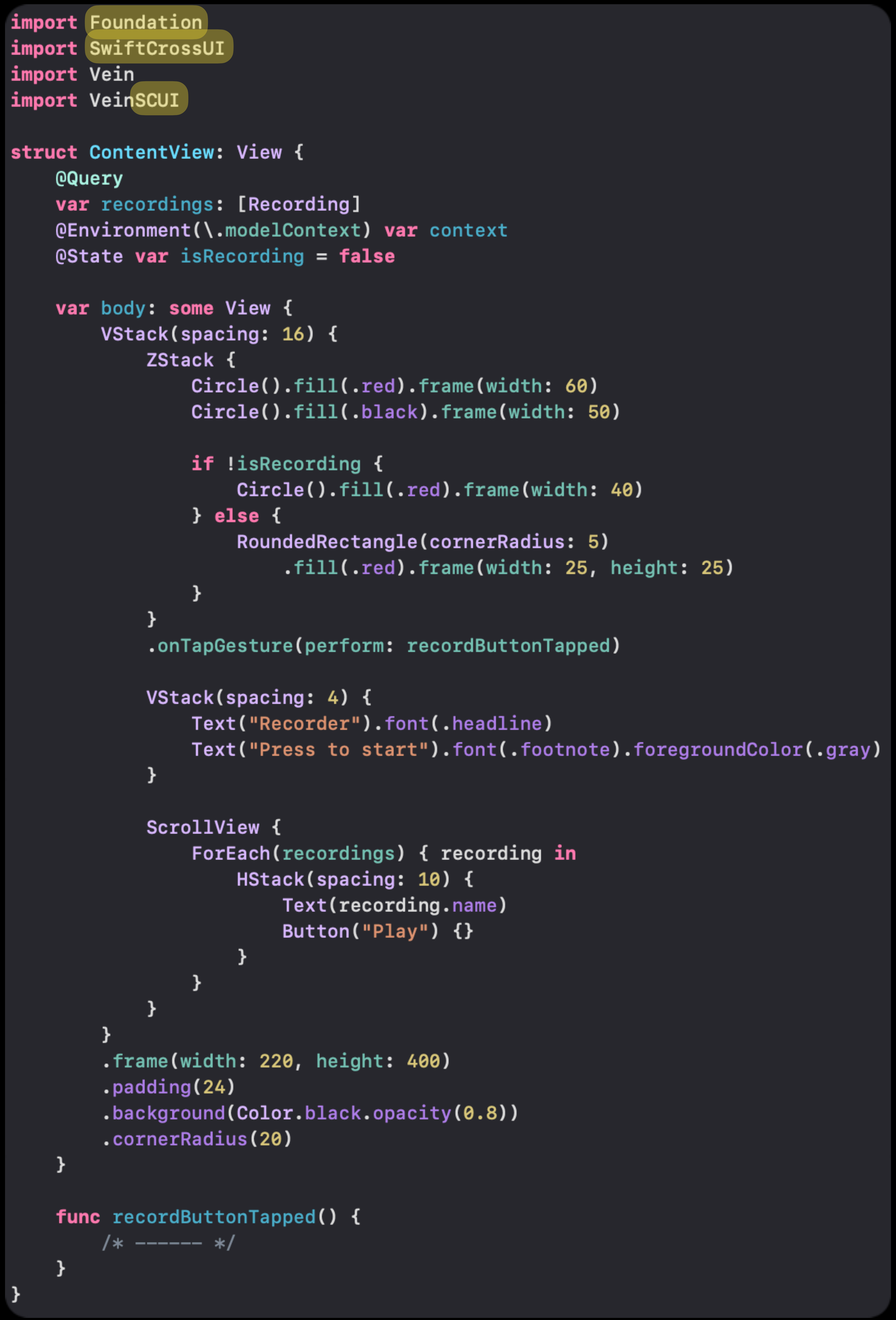
Recorder

Press to start

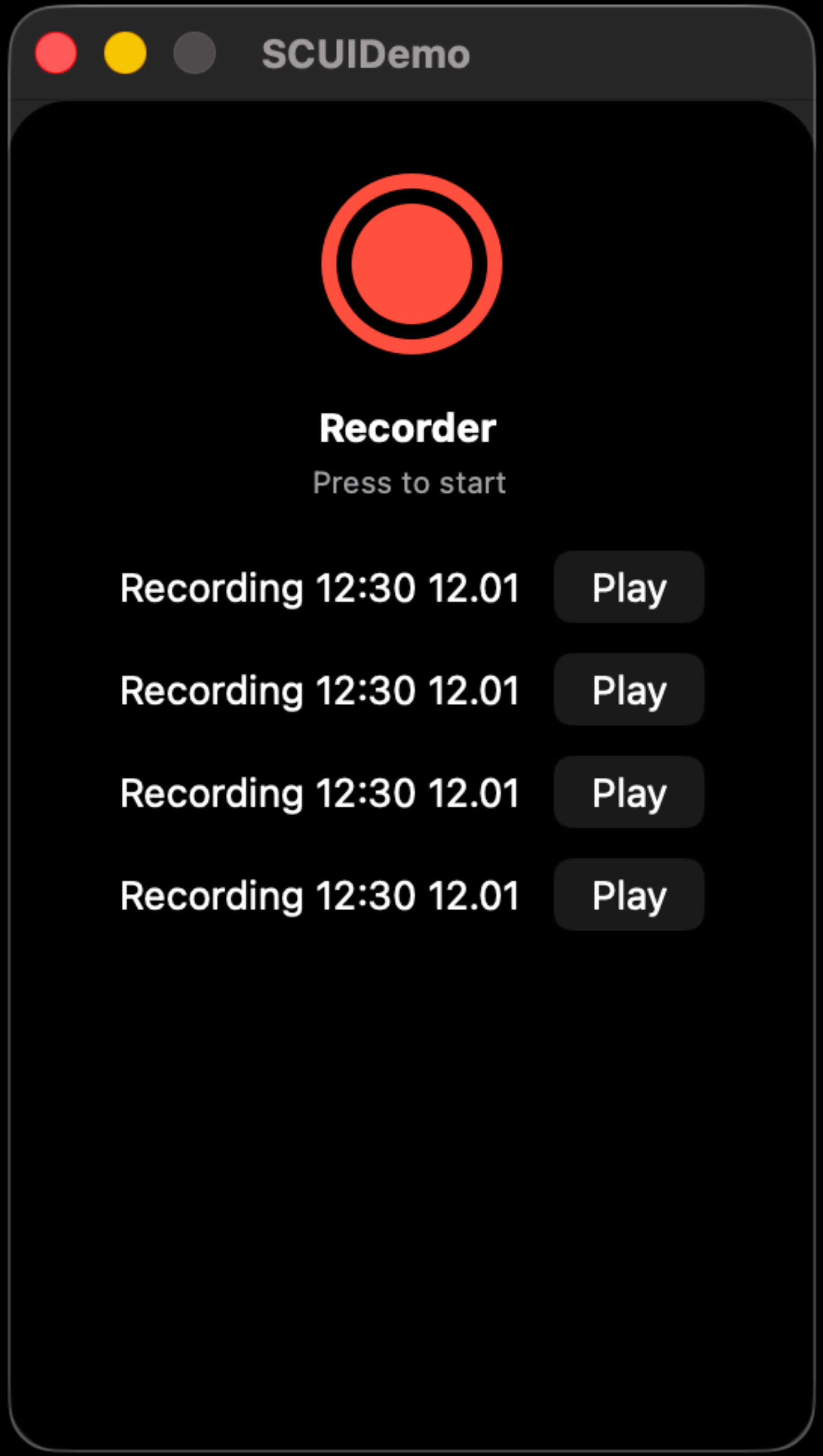
Recording 14:17 10.01 Play
Recording 14:17 10.01 Play
Recording 12:24 12.01 Play
Recording 12:24 12.01 Play



```swift
import SwiftUI
import Vein
import VeinSwiftUI

struct ContentView: View {
    @Query
    var recordings: [Recording]
    @Environment(\.modelContext) var context
    @State var isRecording = false

    var body: some View {
        VStack(spacing: 16) {
            ZStack {
                Circle().fill(.red).frame(width: 60)
                Circle().fill(.black).frame(width: 50)

                if !isRecording {
                    Circle().fill(.red).frame(width: 40)
                } else {
                    RoundedRectangle(cornerRadius: 5)
                        .fill(.red).frame(width: 25, height: 25)
                }
            }
            .onTapGesture(perform: recordButtonTapped)

            VStack(spacing: 4) {
                Text("Recorder").font(.headline)
                Text("Press to start").font(.footnote).foregroundColor(.gray)
            }

            ScrollView {
                ForEach(recordings) { recording in
                    HStack(spacing: 10) {
                        Text(recording.name)
                        Button("Play") {}
                    }
                }
            }
        }
        .frame(width: 220, height: 400)
        .padding(24)
        .background(Color.black.opacity(0.8))
        .cornerRadius(20)
    }

    func recordButtonTapped() {
        /* ------ */
    }
}
```



```swift
import Foundation
import SwiftCrossUI
import Vein
import VeinSCUI

struct ContentView: View {
    @Query
    var recordings: [Recording]
    @Environment(\.modelContext) var context
    @State var isRecording = false

    var body: some View {
        VStack(spacing: 16) {
            ZStack {
                Circle().fill(.red).frame(width: 60)
                Circle().fill(.black).frame(width: 50)

                if !isRecording {
                    Circle().fill(.red).frame(width: 40)
                } else {
                    RoundedRectangle(cornerRadius: 5)
                        .fill(.red).frame(width: 25, height: 25)
                }
            }
            .onTapGesture(perform: recordButtonTapped)

            VStack(spacing: 4) {
                Text("Recorder").font(.headline)
                Text("Press to start").font(.footnote).foregroundColor(.gray)
            }

            ScrollView {
                ForEach(recordings) { recording in
                    HStack(spacing: 10) {
                        Text(recording.name)
                        Button("Play") {}
                    }
                }
            }
        }
        .frame(width: 220, height: 400)
        .padding(24)
        .background(Color.black.opacity(0.8))
        .cornerRadius(20)
    }

    func recordButtonTapped() {
        /* ------ */
    }
}
```



Recorder

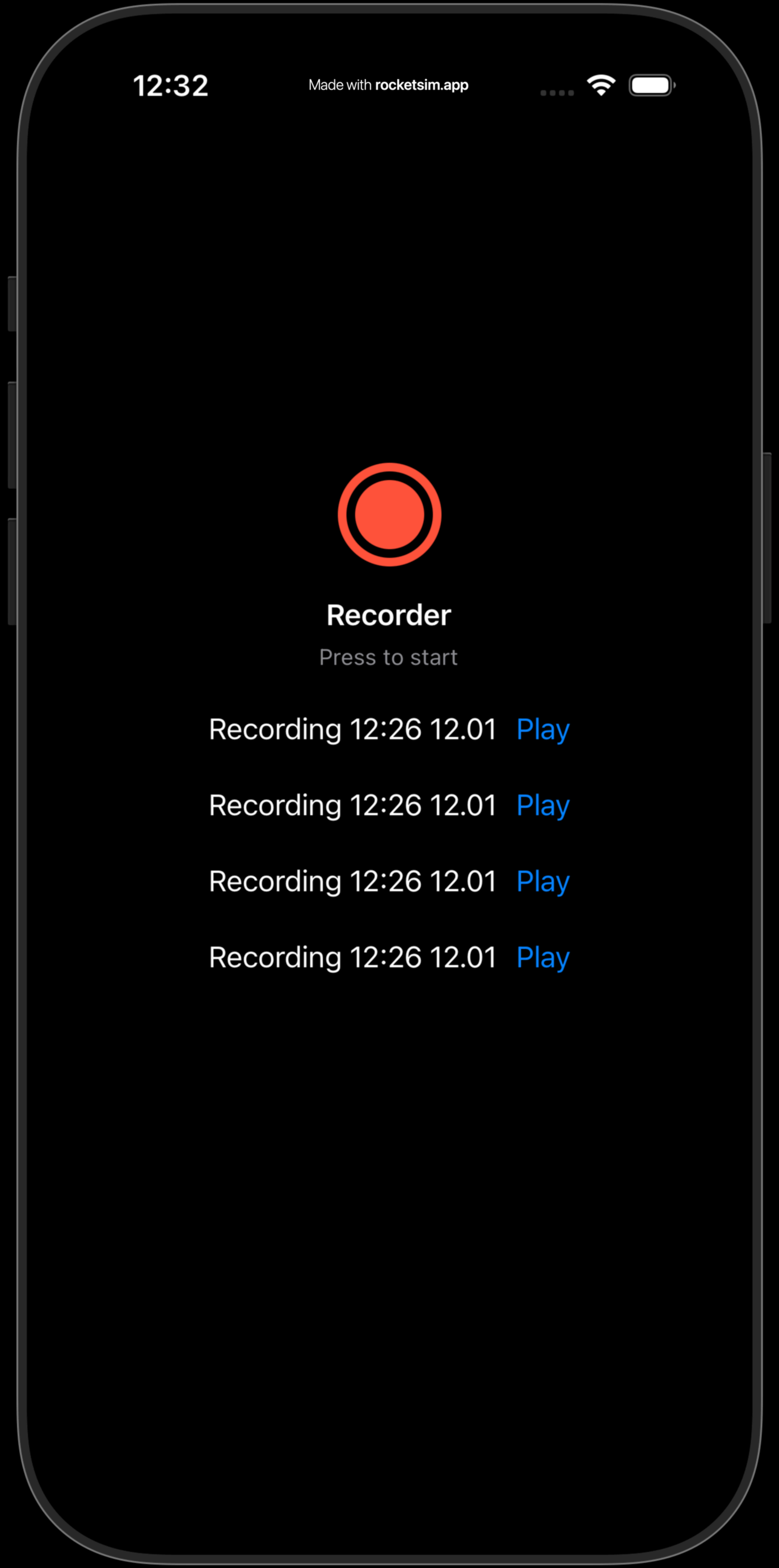
Press to start

Recording 12:30 12.01 Play

Recording 12:30 12.01 Play

Recording 12:30 12.01 Play

Recording 12:30 12.01 Play



Recorder

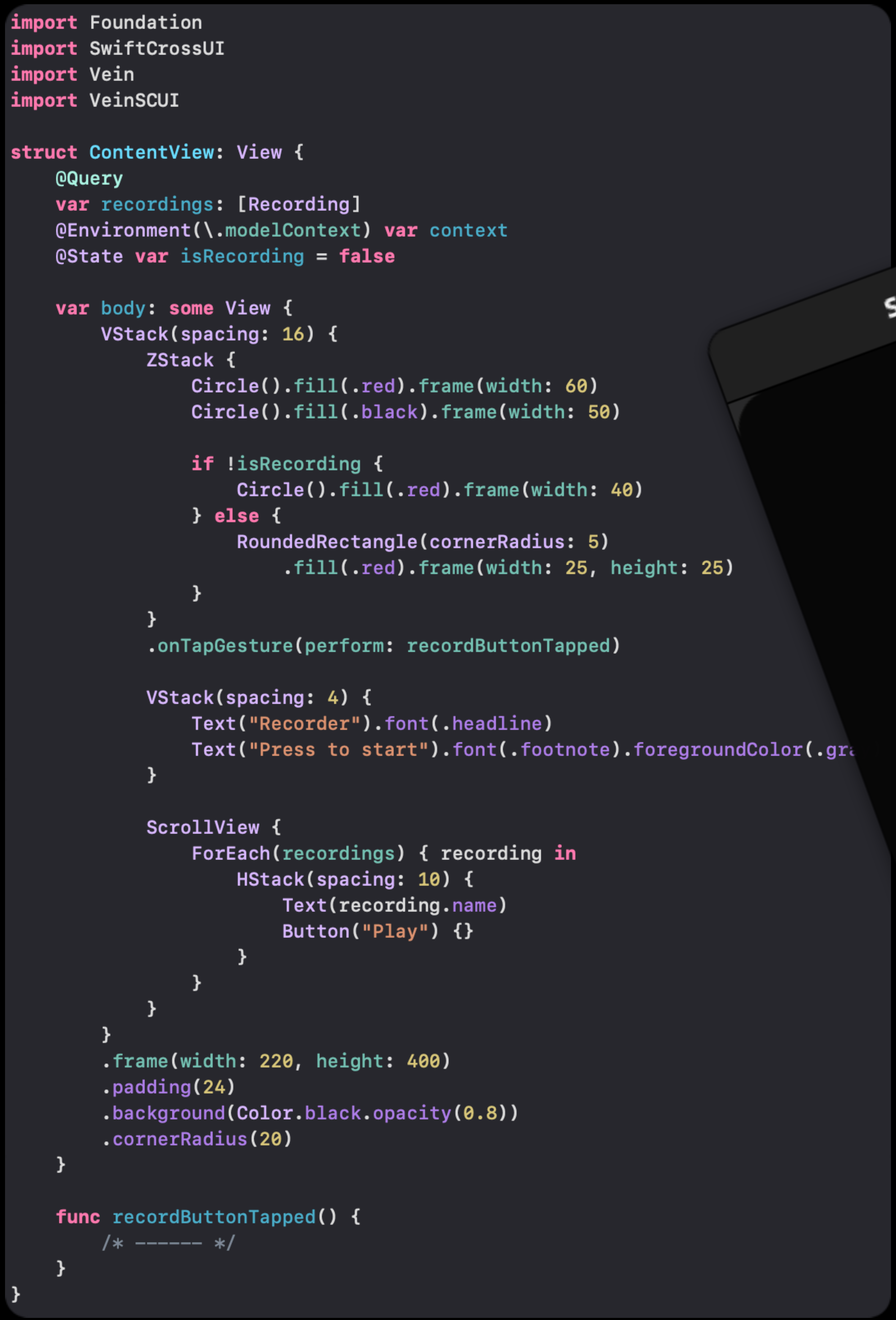
Press to start

Recording 12:26 12.01 Play

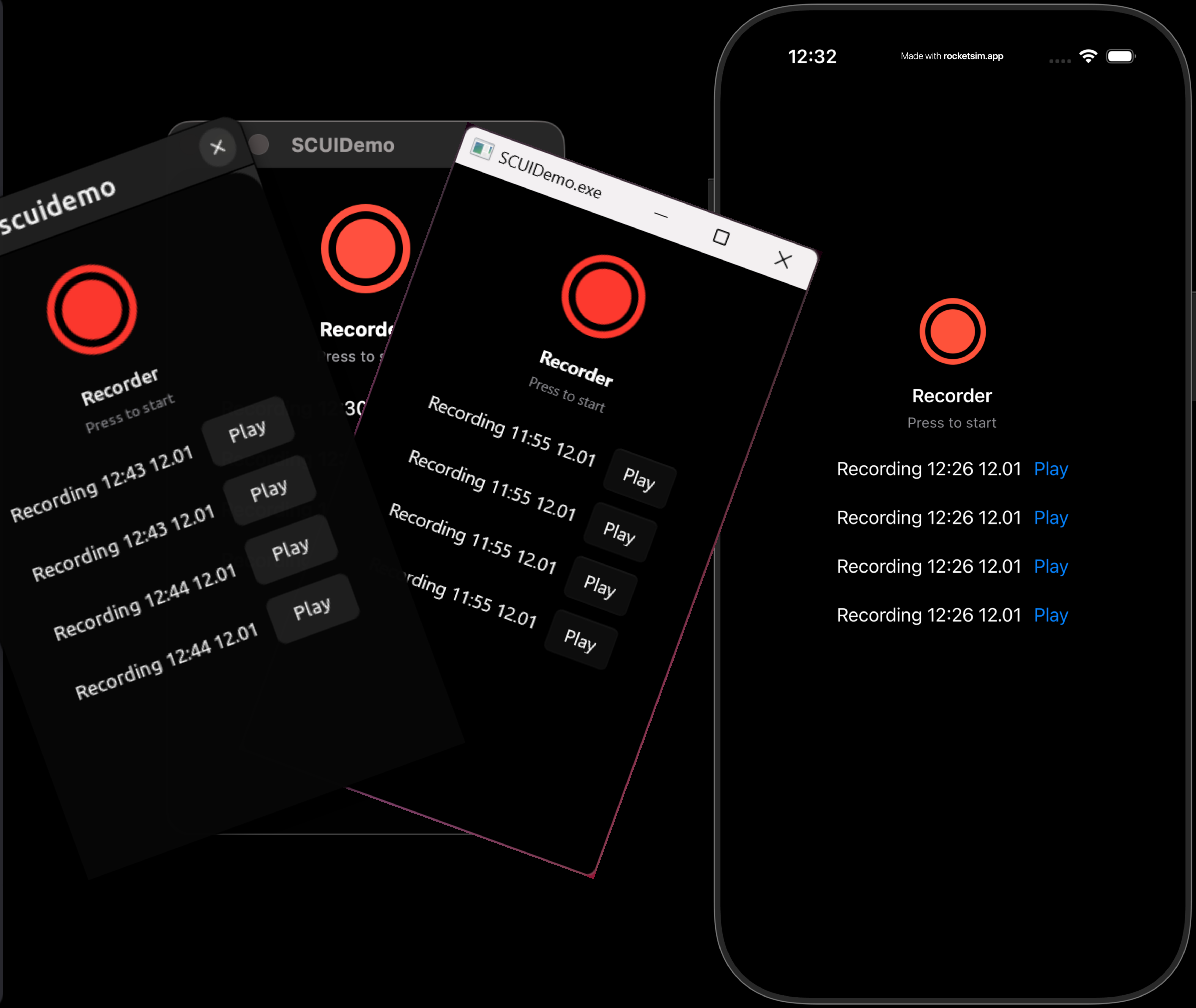
Recording 12:26 12.01 Play

Recording 12:26 12.01 Play

Recording 12:26 12.01 Play



```swift
import Foundation
import SwiftCrossUI
import Vein
import VeinSCUI

struct ContentView: View {
    @Query
    var recordings: [Recording]
    @Environment(\.modelContext) var context
    @State var isRecording = false

    var body: some View {
        VStack(spacing: 16) {
            ZStack {
                Circle().fill(.red).frame(width: 60)
                Circle().fill(.black).frame(width: 50)

                if !isRecording {
                    Circle().fill(.red).frame(width: 40)
                } else {
                    RoundedRectangle(cornerRadius: 5)
                        .fill(.red).frame(width: 25, height: 25)
                }
            }
            .onTapGesture(perform: recordButtonTapped)

            VStack(spacing: 4) {
                Text("Recorder").font(.headline)
                Text("Press to start").font(.footnote).foregroundColor(.gra
            }

            ScrollView {
                ForEach(recordings) { recording in
                    HStack(spacing: 10) {
                        Text(recording.name)
                        Button("Play") {}
                    }
                }
            }
        }
        .frame(width: 220, height: 400)
        .padding(24)
        .background(Color.black.opacity(0.8))
        .cornerRadius(20)
    }

    func recordButtonTapped() {
        /* ------ */
    }
}
```

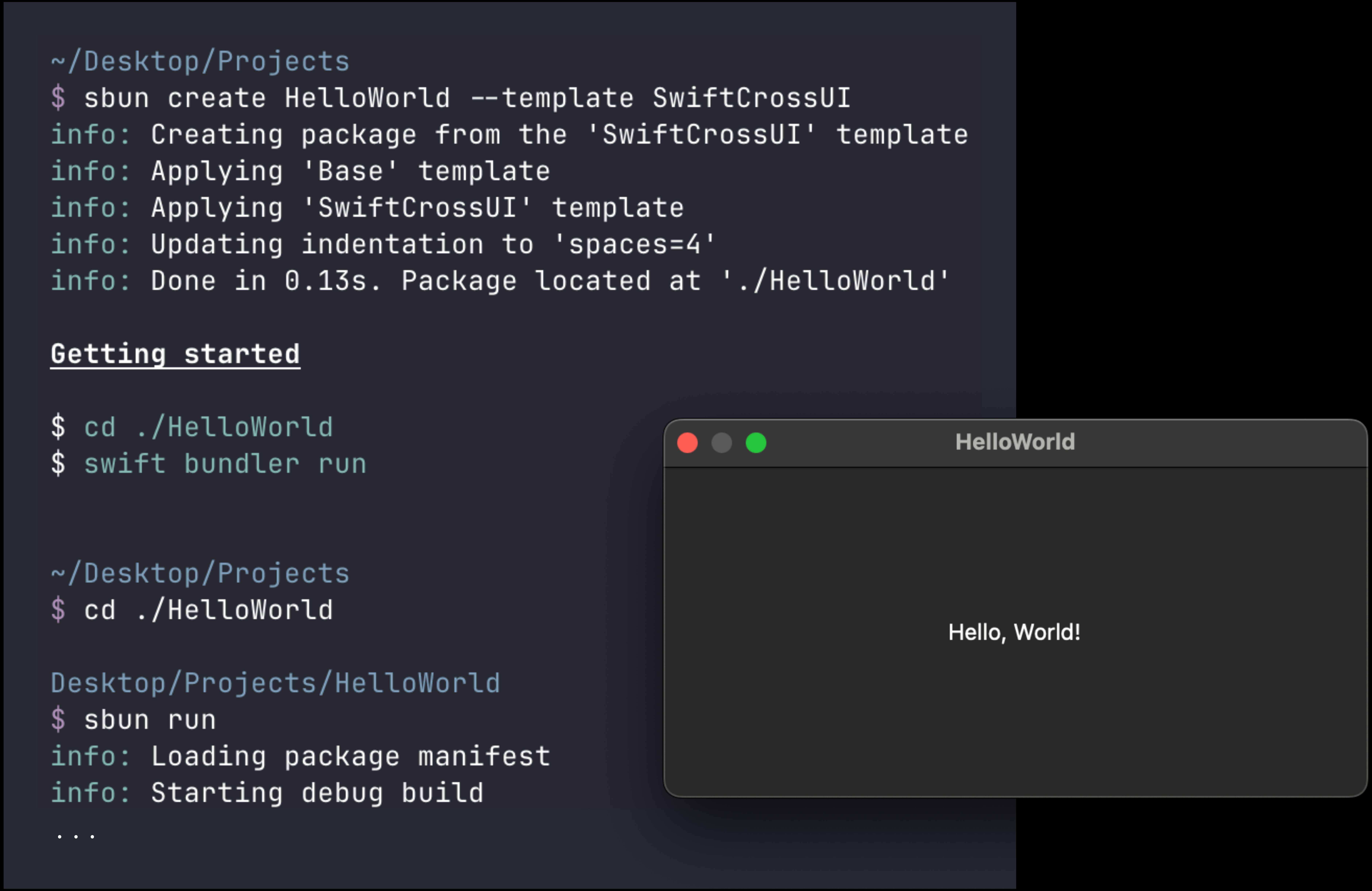

scuidemo
Recorder
Press to start
Recording 12:43 12.01 Play
Recording 12:43 12.01 Play
Recording 12:44 12.01 Play
Recording 12:44 12.01 Play
SCUIDemo
Record
ress to
SCUIDemo.exe
Recorder
Press to start
Recording 11:55 12.01 Play
Recording 11:55 12.01 Play
Recording 11:55 12.01 Play
rding 11:55 12.01 Play
12:32
Made with rocketsim.app
Recorder
Press to start
Recording 12:26 12.01 Play
Recording 12:26 12.01 Play
Recording 12:26 12.01 Play
Recording 12:26 12.01 Play

You know SwiftUI?
You know SwiftCrossUI.

Swift Bundler

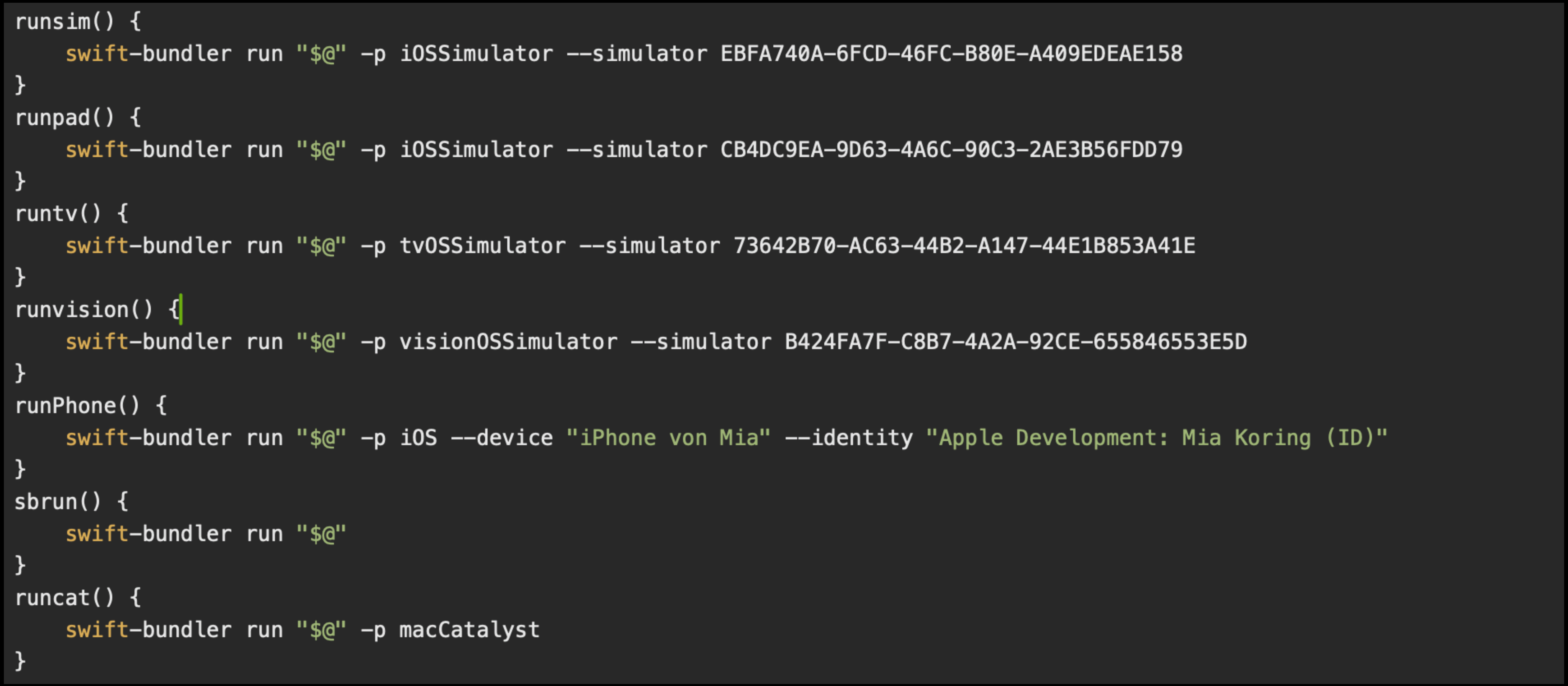
- A tool for building and distributing cross-platform Swift apps
- Supports macOS, Linux, Windows, iOS, tvOS and visionOS, (soon Android)
- Targets .app, .rpm, .AppImage and .msi, (soon .apk)
- Simple to use in CI

Logo by Syifa Arlin



```
~/Desktop/Projects
$ sbun create HelloWorld --template SwiftCrossUI
info: Creating package from the 'SwiftCrossUI' template
info: Applying 'Base' template
info: Applying 'SwiftCrossUI' template
info: Updating indentation to 'spaces=4'
info: Done in 0.13s. Package located at './HelloWorld'

Getting started

$ cd ./HelloWorld
$ swift bundler run


~/Desktop/Projects
$ cd ./HelloWorld

Desktop/Projects/HelloWorld
$ sbun run
info: Loading package manifest
info: Starting debug build
...
```

Hello, World!



```
runsim() {
    swift-bundler run "$@" -p iOSSimulator --simulator EBFA740A-6FCD-46FC-B80E-A409EDEAE158
}
runpad() {
    swift-bundler run "$@" -p iOSSimulator --simulator CB4DC9EA-9D63-4A6C-90C3-2AE3B56FDD79
}
runtv() {
    swift-bundler run "$@" -p tvOSSimulator --simulator 73642B70-AC63-44B2-A147-44E1B853A41E
}
runvision() {
    swift-bundler run "$@" -p visionOSSimulator --simulator B424FA7F-C8B7-4A2A-92CE-655846553E5D
}
runPhone() {
    swift-bundler run "$@" -p iOS --device "iPhone von Mia" --identity "Apple Development: Mia Koring (ID)"
}
sbrun() {
    swift-bundler run "$@"
}
runcat() {
    swift-bundler run "$@" -p macCatalyst
}
```

Nice, but Previews?

Terminal
Shell
Edit
View
Window
Help
69%
Mon 11:45
fosdem-scui-demo
master
SCUI
iPhone 17 Pro
Build Succeeded | 09.01.26 at 15:40
fosdem-scui-demo › Sources › SCUIDemo › ContentView.swift › body
Line: 43 Col: 32
72 %
99 %

```
fosdem-scui-demo — miakoring@MacBook-Pro-von-Mia — ..dem-scui-demo — -zsh — 80×24
→ fosdem-scui-demo git:(master) ✗
```

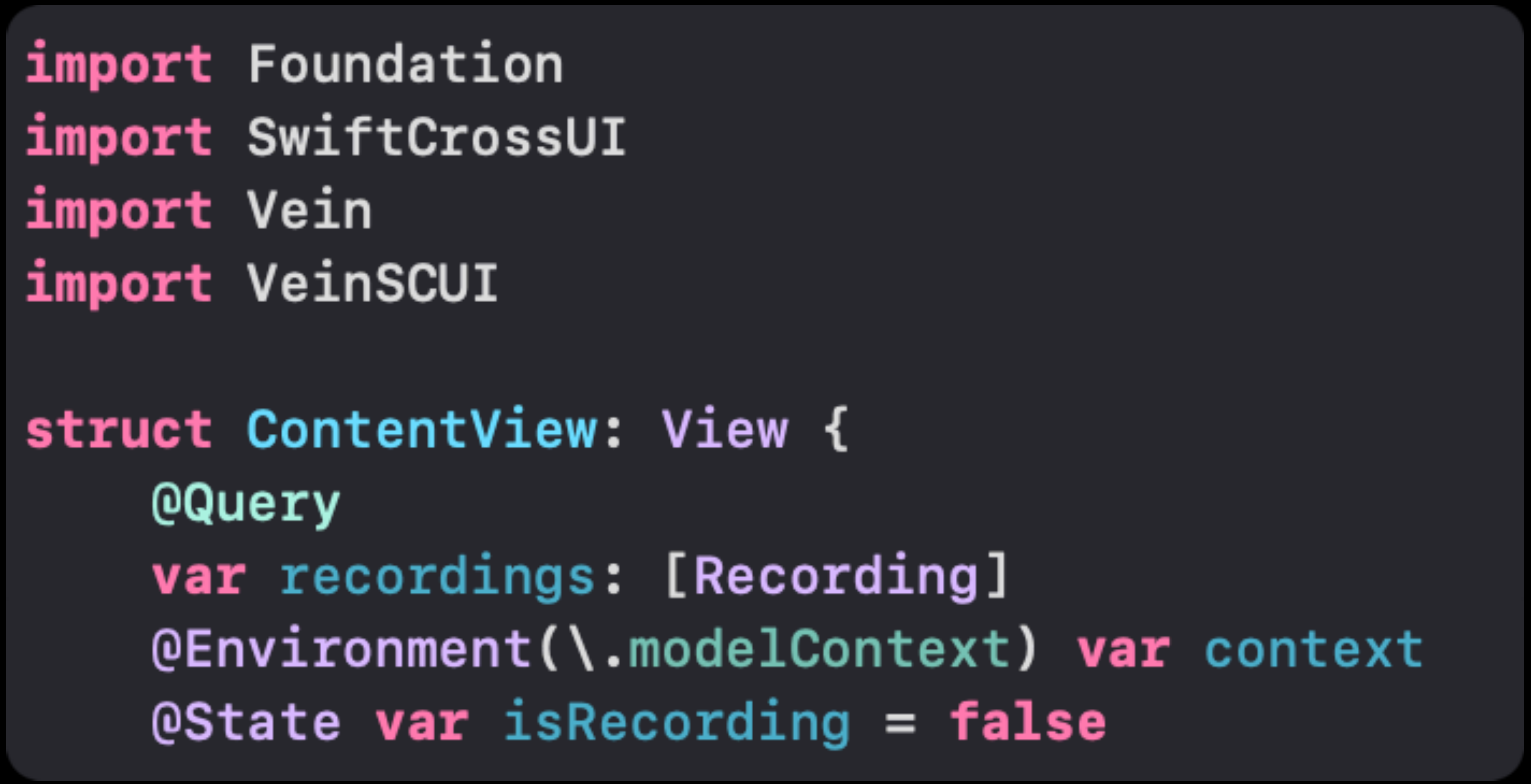
```swift
struct ContentView: View {
    var body: some View {
                    Circle().fill(.red).frame(width: 40)
                } else {
                    RoundedRectangle(cornerRadius: 5)
                        .fill(.red).frame(width: 25, height: 25)
                }
            }
            .onTapGesture(perform: recordButtonTapped)

            VStack(spacing: 4) {
                Text("Recorder").font(.headline)
                Text("Press to start").font(.footnote).foregroundColor(.gray)
            }

            ScrollView {
                ForEach(recordings) { recording in
                    HStack(spacing: 10) {
                        Text(recording.name)
                        Button("Play") {}
                    }
                }
            }
        }
        .frame(width: 220, height: 400)
        .padding(24)
        .background(Color.black.opacity(0.8))
        .cornerRadius(20)
    }

    func recordButtonTapped() {
        isRecording.toggle()
        if !isRecording {
            let formatter = DateFormatter()
            formatter.dateFormat = "HH:mm dd.MM"
            try? context.insert(
                Recording(
                    name: "Recording \(formatter.string(from: .now))",
                    audio: Data()
                )
            )
        }
    }
```

moreSwift - Roadmap

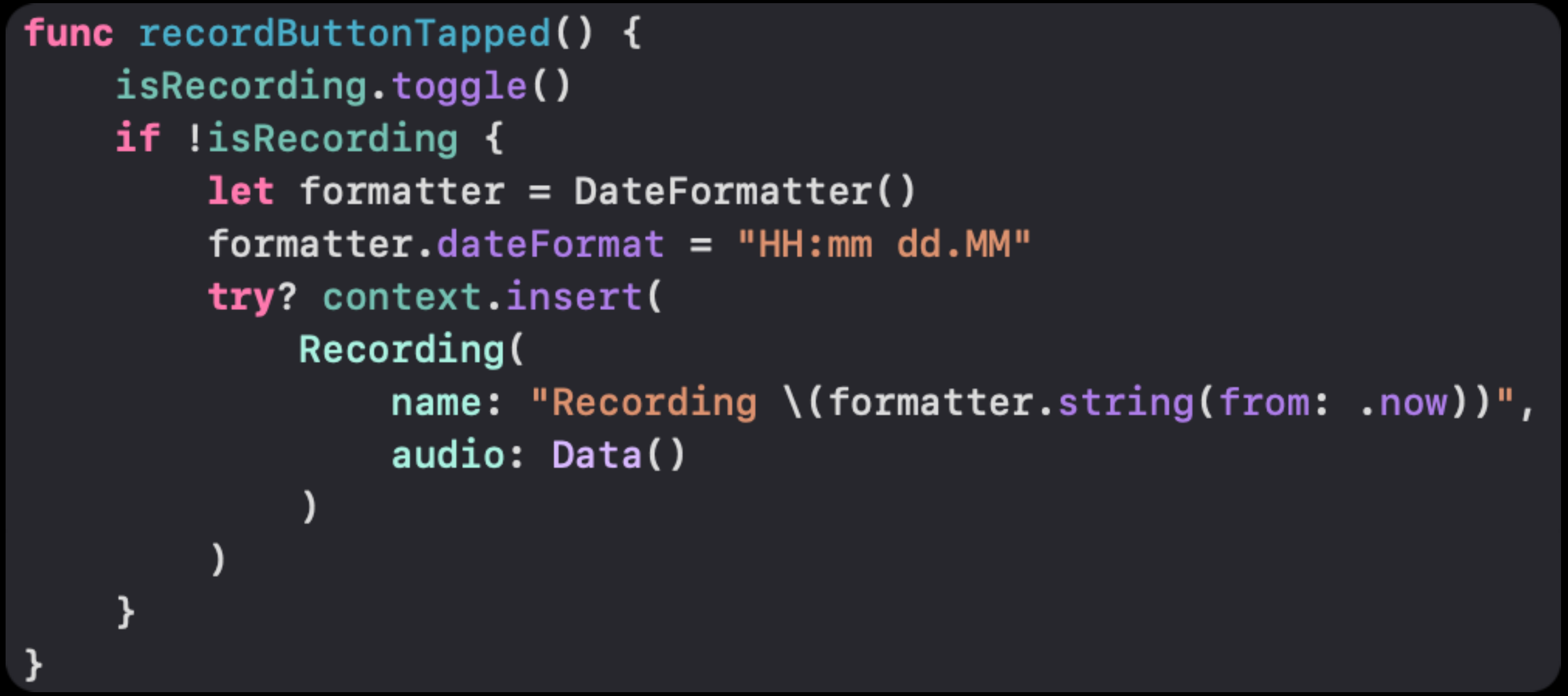
- Android Support (coming soon)
- Keyboard navigation and shortcuts
- Improvements to feature parity across platforms
- Cross-platform Graphics and Audio Frameworks
- and more

Amethyst Vein

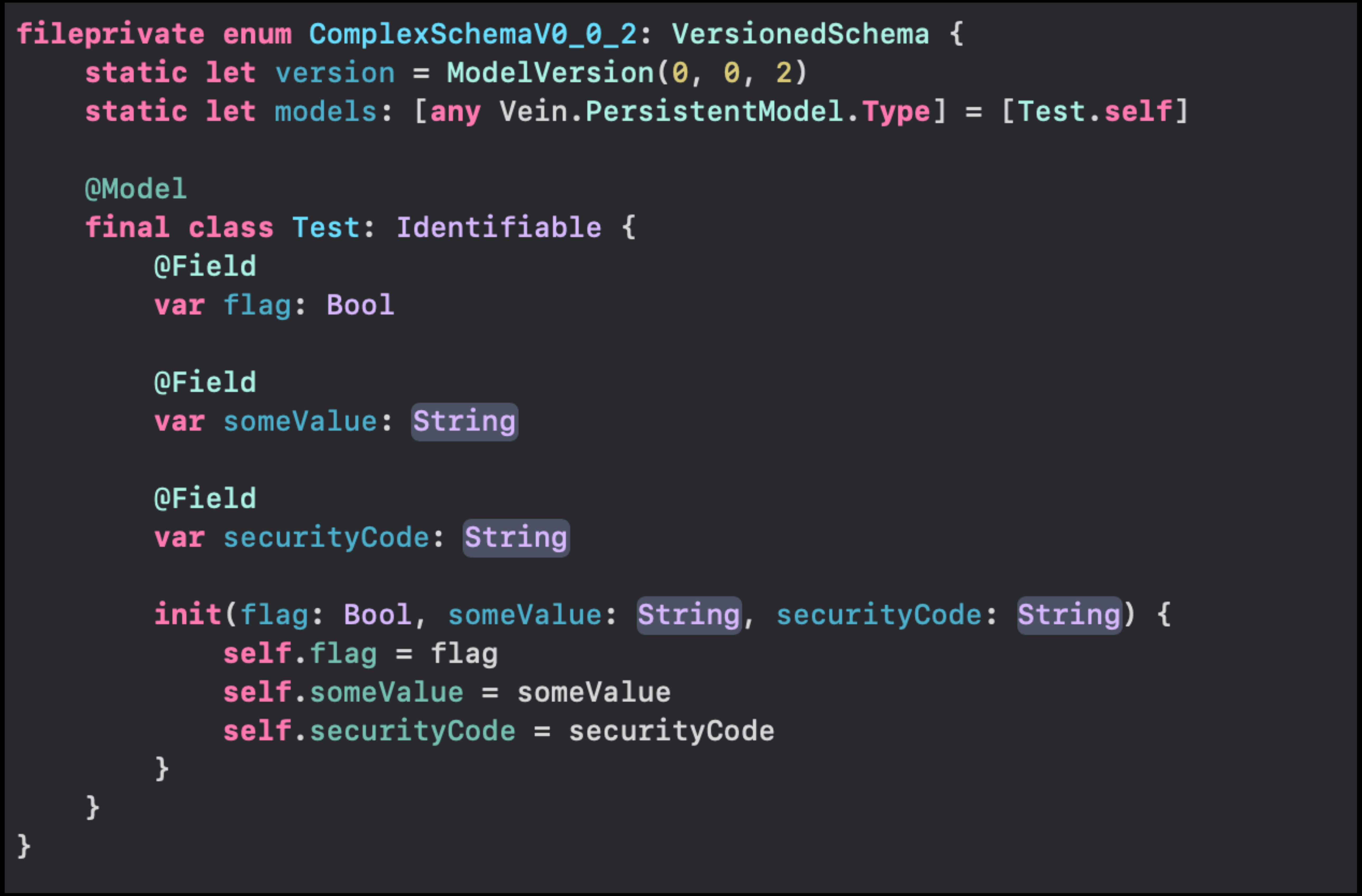
- Supports SwiftUI & SwiftCrossUI
- Declarative & Dynamic
- Easy to use
- Control over time of field load
- Wraps SQLite.swift



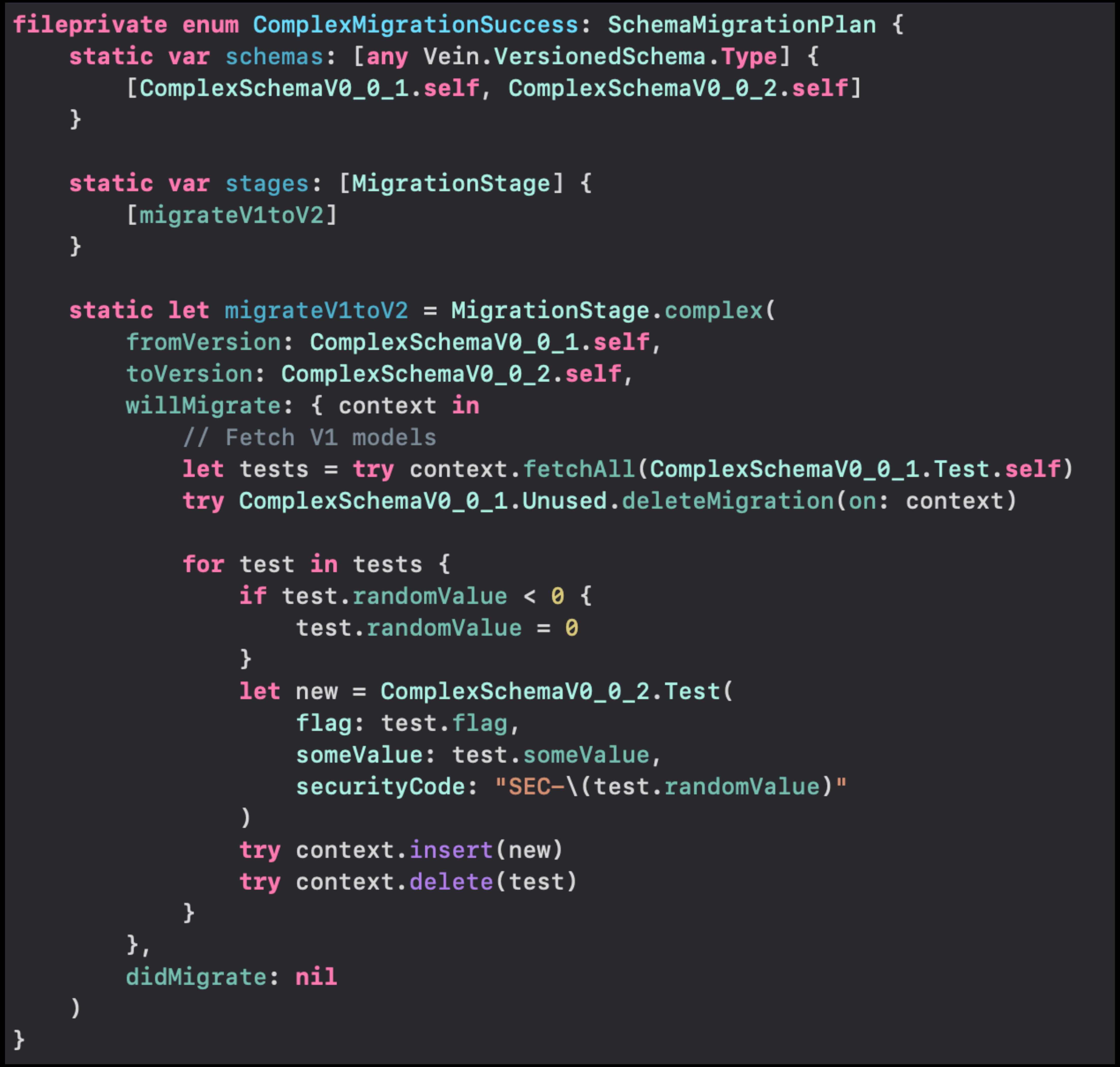
```swift
import Foundation
import SwiftCrossUI
import Vein
import VeinSCUI

struct ContentView: View {
    @Query
    var recordings: [Recording]
    @Environment(\.modelContext) var context
    @State var isRecording = false
```



```swift
func recordButtonTapped() {
    isRecording.toggle()
    if !isRecording {
        let formatter = DateFormatter()
        formatter.dateFormat = "HH:mm dd.MM"
        try? context.insert(
            Recording(
                name: "Recording \(formatter.string(from: .now))",
                audio: Data()
            )
        )
    }
}
```



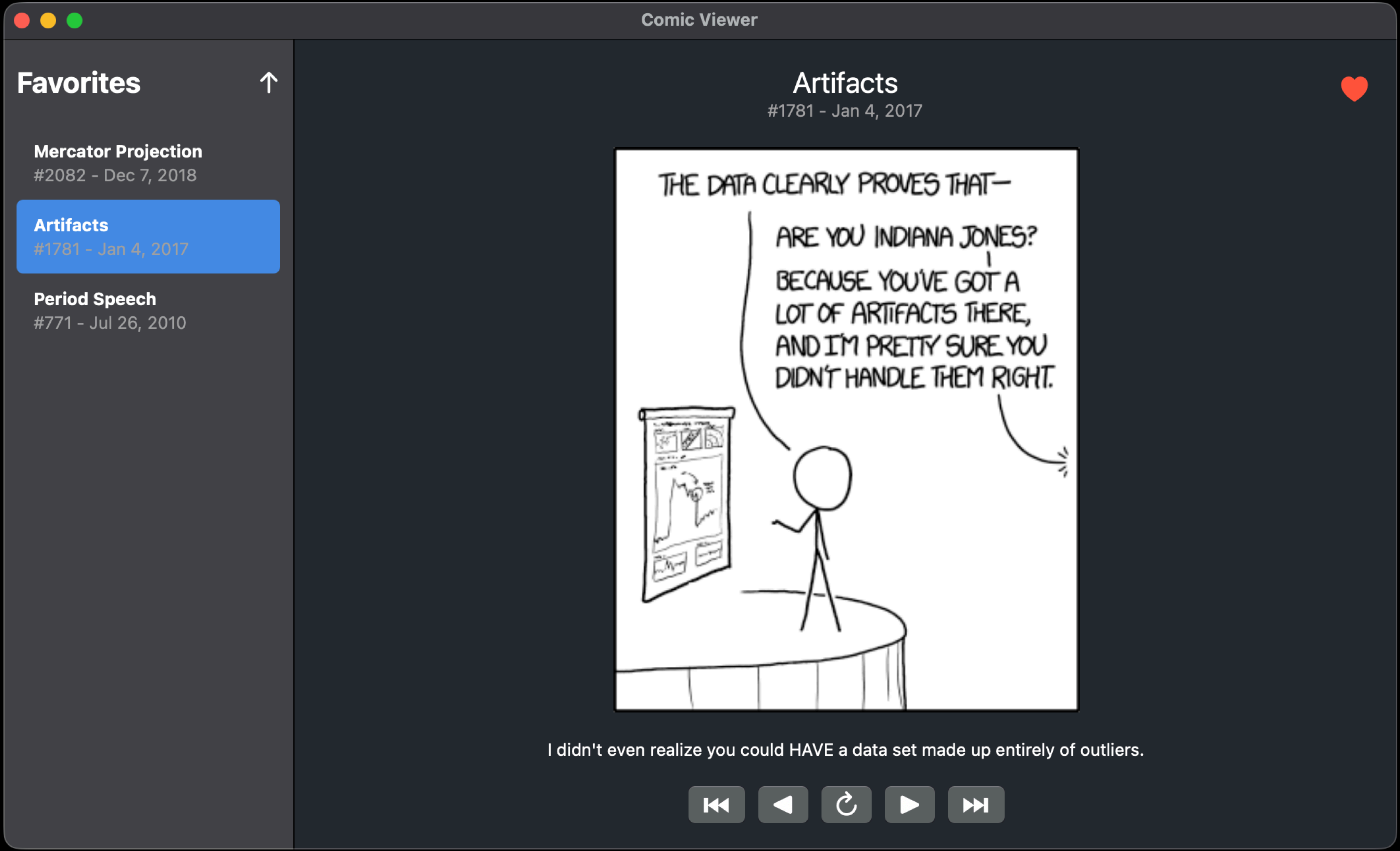
```swift
fileprivate enum ComplexSchemaV0_0_2: VersionedSchema {
    static let version = ModelVersion(0, 0, 2)
    static let models: [any Vein.PersistentModel.Type] = [Test.self]

    @Model
    final class Test: Identifiable {
        @Field
        var flag: Bool

        @Field
        var someValue: String

        @Field
        var securityCode: String

        init(flag: Bool, someValue: String, securityCode: String) {
            self.flag = flag
            self.someValue = someValue
            self.securityCode = securityCode
        }
    }
}
```



```swift
fileprivate enum ComplexMigrationSuccess: SchemaMigrationPlan {
    static var schemas: [any Vein.VersionedSchema.Type] {
        [ComplexSchemaV0_0_1.self, ComplexSchemaV0_0_2.self]
    }

    static var stages: [MigrationStage] {
        [migrateV1toV2]
    }

    static let migrateV1toV2 = MigrationStage.complex(
        fromVersion: ComplexSchemaV0_0_1.self,
        toVersion: ComplexSchemaV0_0_2.self,
        willMigrate: { context in
            // Fetch V1 models
            let tests = try context.fetchAll(ComplexSchemaV0_0_1.Test.self)
            try ComplexSchemaV0_0_1.Unused.deleteMigration(on: context)

            for test in tests {
                if test.randomValue < 0 {
                    test.randomValue = 0
                }
                let new = ComplexSchemaV0_0_2.Test(
                    flag: test.flag,
                    someValue: test.someValue,
                    securityCode: "SEC-\(test.randomValue)"
                )
                try context.insert(new)
                try context.delete(test)
            }
        },
        didMigrate: nil
    )
}
```

Amethyst Vein - Roadmap

- Relationships
- End to end encrypted sync through selfhostable backend
- Cross-Platform Keychain wrapper & sync



Comic Viewer

Favorites

Mercator Projection
#2082 - Dec 7, 2018

Artifacts
#1781 - Jan 4, 2017

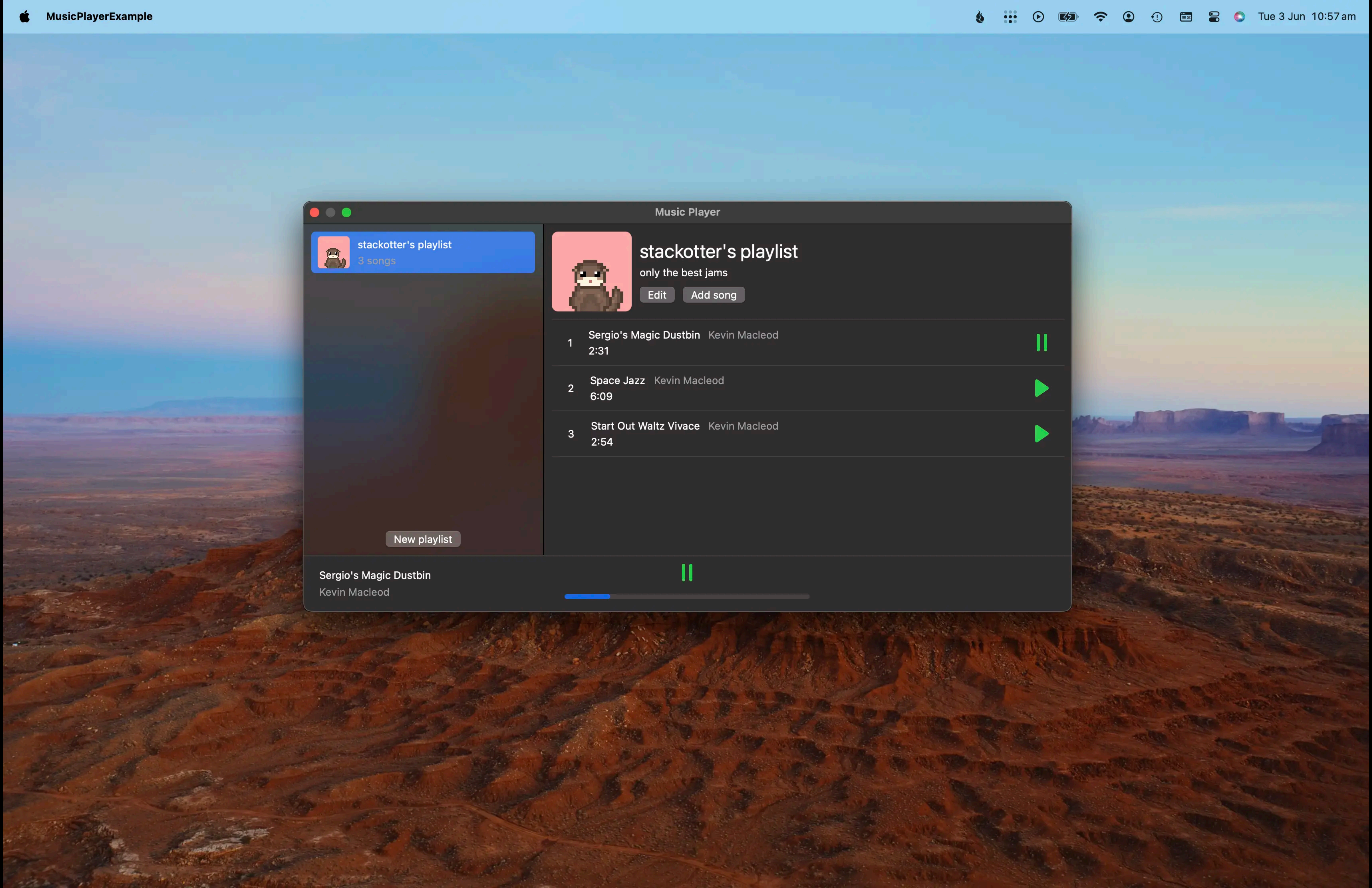
Period Speech
#771 - Jul 26, 2010

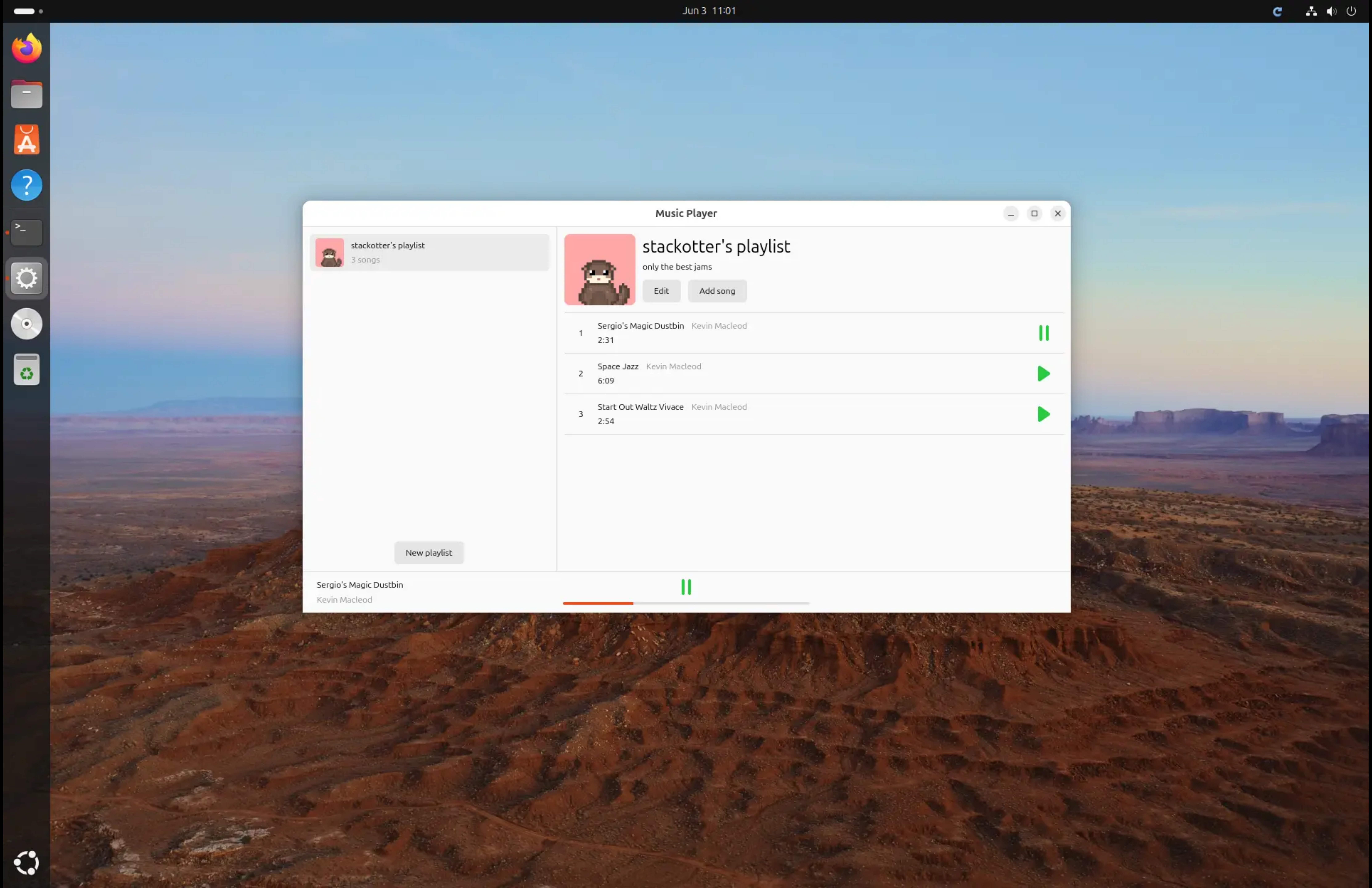
Artifacts
#1781 - Jan 4, 2017

THE DATA CLEARLY PROVES THAT—
ARE YOU INDIANA JONES?
BECAUSE YOU'VE GOT A LOT OF ARTIFACTS THERE, AND I'M PRETTY SURE YOU DIDN'T HANDLE THEM RIGHT.
⏮ ◀ ↻ ▶ ⏭

I didn't even realize you could HAVE a data set made up entirely of outliers.

@pwsacademy@mastodon.world


MusicPlayerExample
Tue 3 Jun 10:57 am
Music Player
stackotter's playlist
3 songs
New playlist
stackotter's playlist
only the best jams
Edit
Add song
1 Sergio's Magic Dustbin Kevin Macleod
2:31
2 Space Jazz Kevin Macleod
6:09
3 Start Out Waltz Vivace Kevin Macleod
2:54
Sergio's Magic Dustbin
Kevin Macleod



Jun 3 11:01

Music Player

stackotter's playlist
3 songs

New playlist

stackotter's playlist
only the best jams

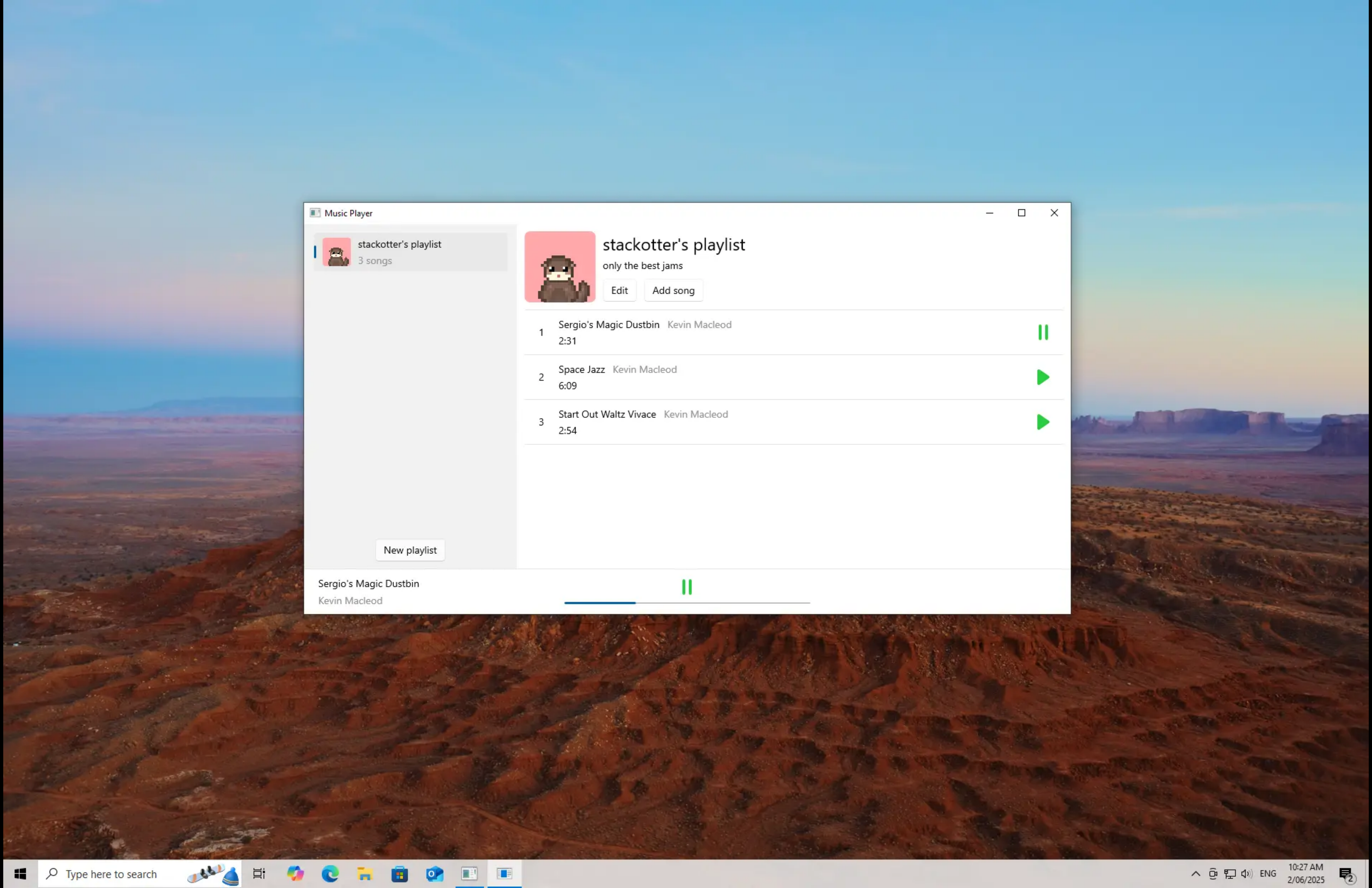
Edit
Add song

1 Sergio's Magic Dustbin Kevin Macleod
2:31

2 Space Jazz Kevin Macleod
6:09

3 Start Out Waltz Vivace Kevin Macleod
2:54

Sergio's Magic Dustbin
Kevin Macleod



stackotter's playlist
3 songs

stackotter's playlist

only the best jams

1 Sergio's Magic Dustbin Kevin Macleod
2:31

2 Space Jazz Kevin Macleod
6:09

3 Start Out Waltz Vivace Kevin Macleod
2:54

Sergio's Magic Dustbin
Kevin Macleod

If you want to know more -
Please ask me

moreSwift - call for sponsors

- moreSwift is the organisation housing SwiftCrossUI and Swift Bundler
- stackotter is working on moreSwift's projects full time (currently off savings and $100 USD per month in GitHub Sponsors)
- Sponsors will be necessary for stackotter to continue working on the projects in a full time capacity, and for moreSwift to eventually be able to pay core external contributors for their valuable time
- Avenues for accepting sponsorship as moreSwift are being explored
- Contact stackotter if interested in sponsoring moreSwift's work ❤️

sponsor page & more

Discord: @mia.codes

Email: mia.koring@amethystsoft.de

Website: https://miakoring.de

GitHub: https://github.com/miakoring

map.miakoring.de

SwiftCrossUI: https://github.com/moreSwift/swift-cross-ui

Swift Bundler: https://github.com/moreSwift/swift-bundler

Amethyst Vein: https://github.com/amethystsoft/vein