Foundations

Introduction to SwiftUI

Hacking Spatial Computing • National University of Singapore

Welcome to Xcode

Text & Image Views

Stacks & Layouts

View Modifiers

Welcome to Xcode

Text & Image Views

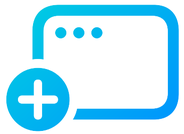
Stacks & Layouts

View Modifiers

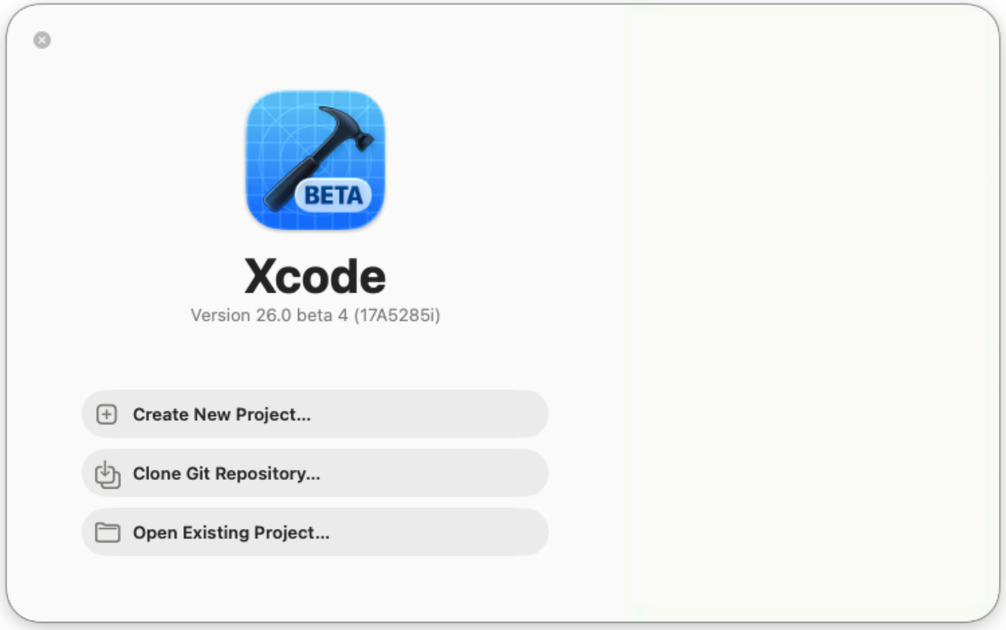
Xcode is the **integrated development environment** for creating apps for all Apple platforms.

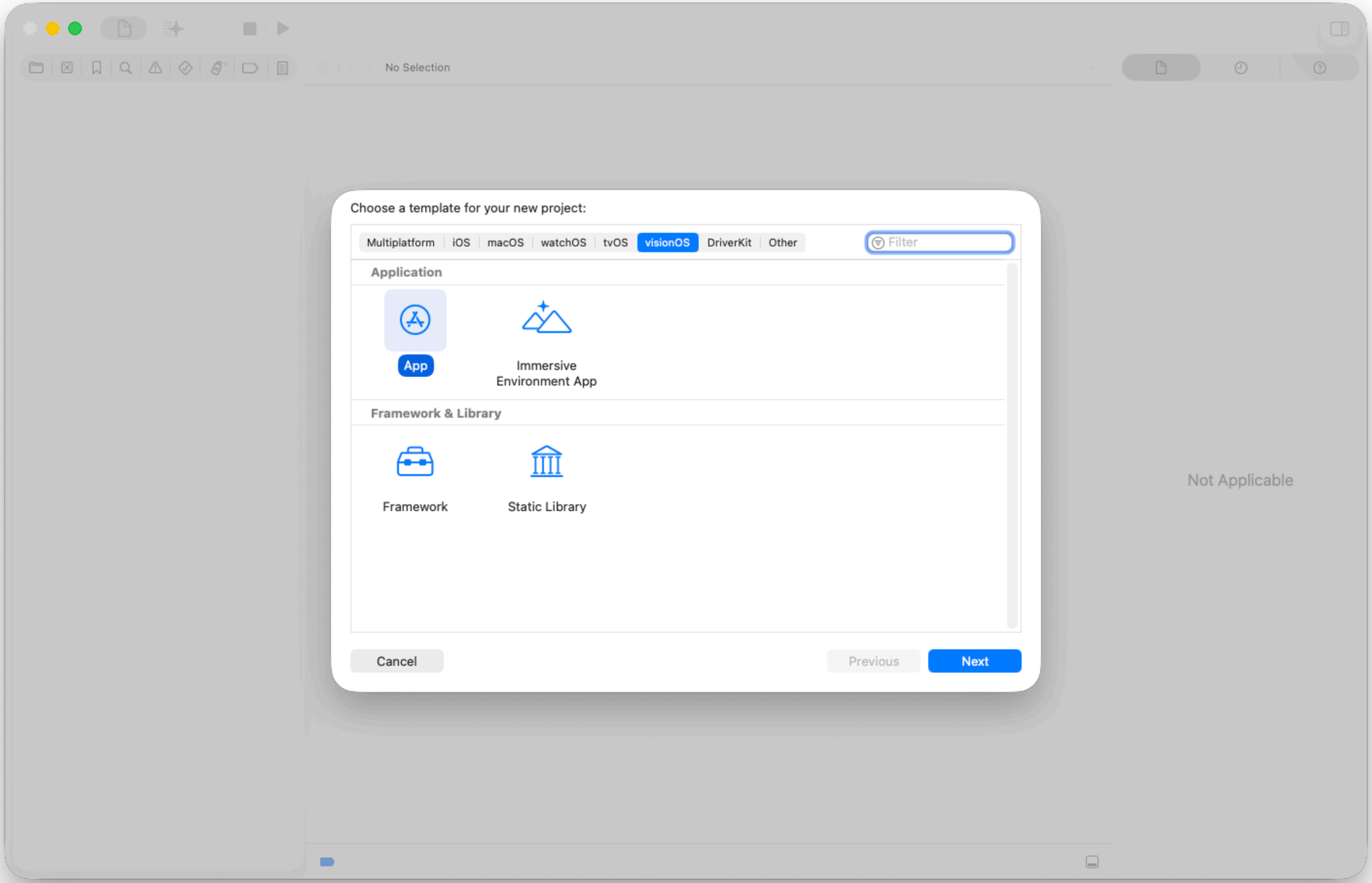
Swift is a **general-purpose programming language** that’s approachable for newcomers and powerful for experts.
It is **fast, modern, safe**, and a **joy to write**.

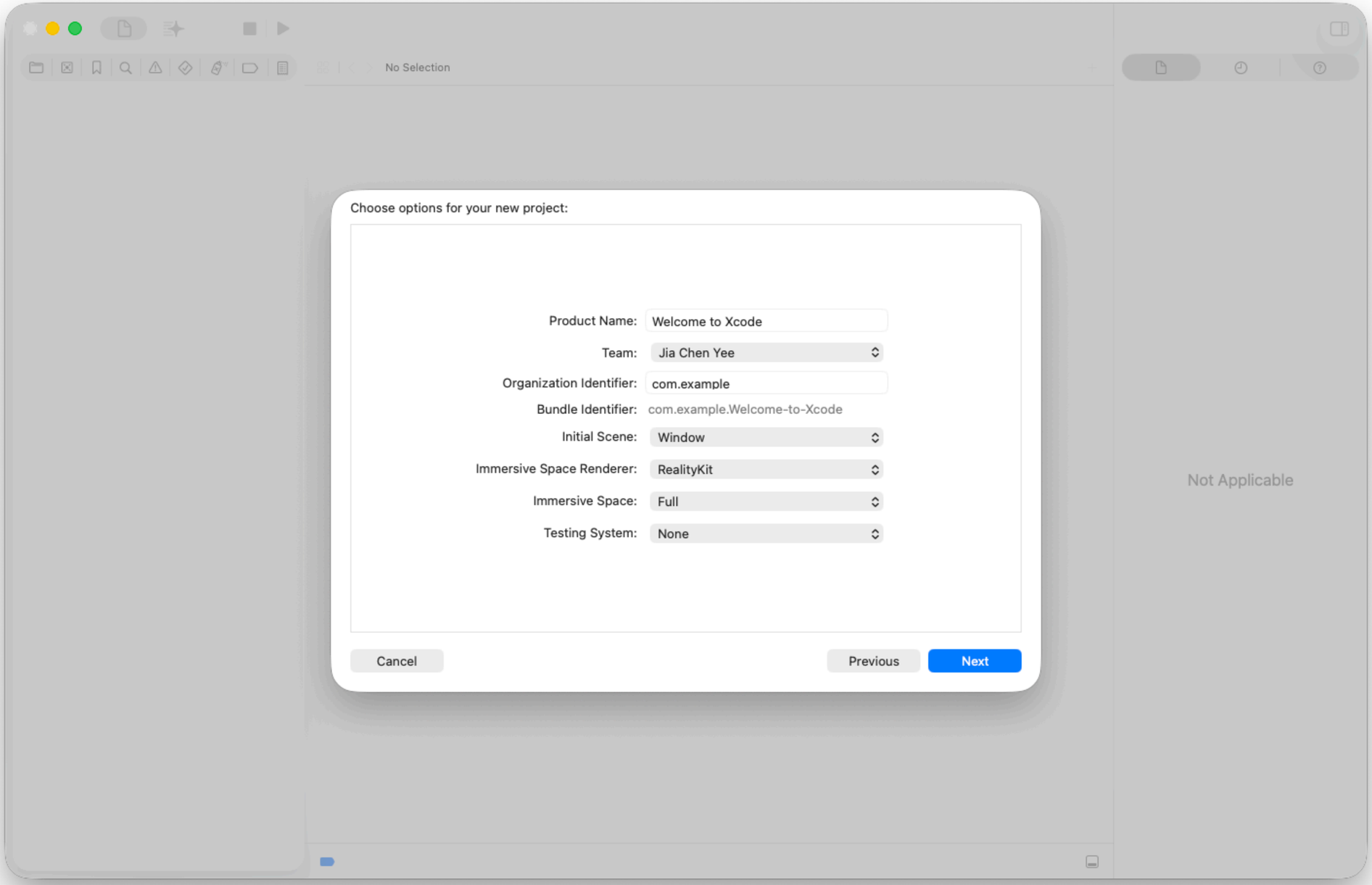
- iPhone
- iPad
- Mac
- Apple TV
- Apple Watch
- Apple Vision Pro

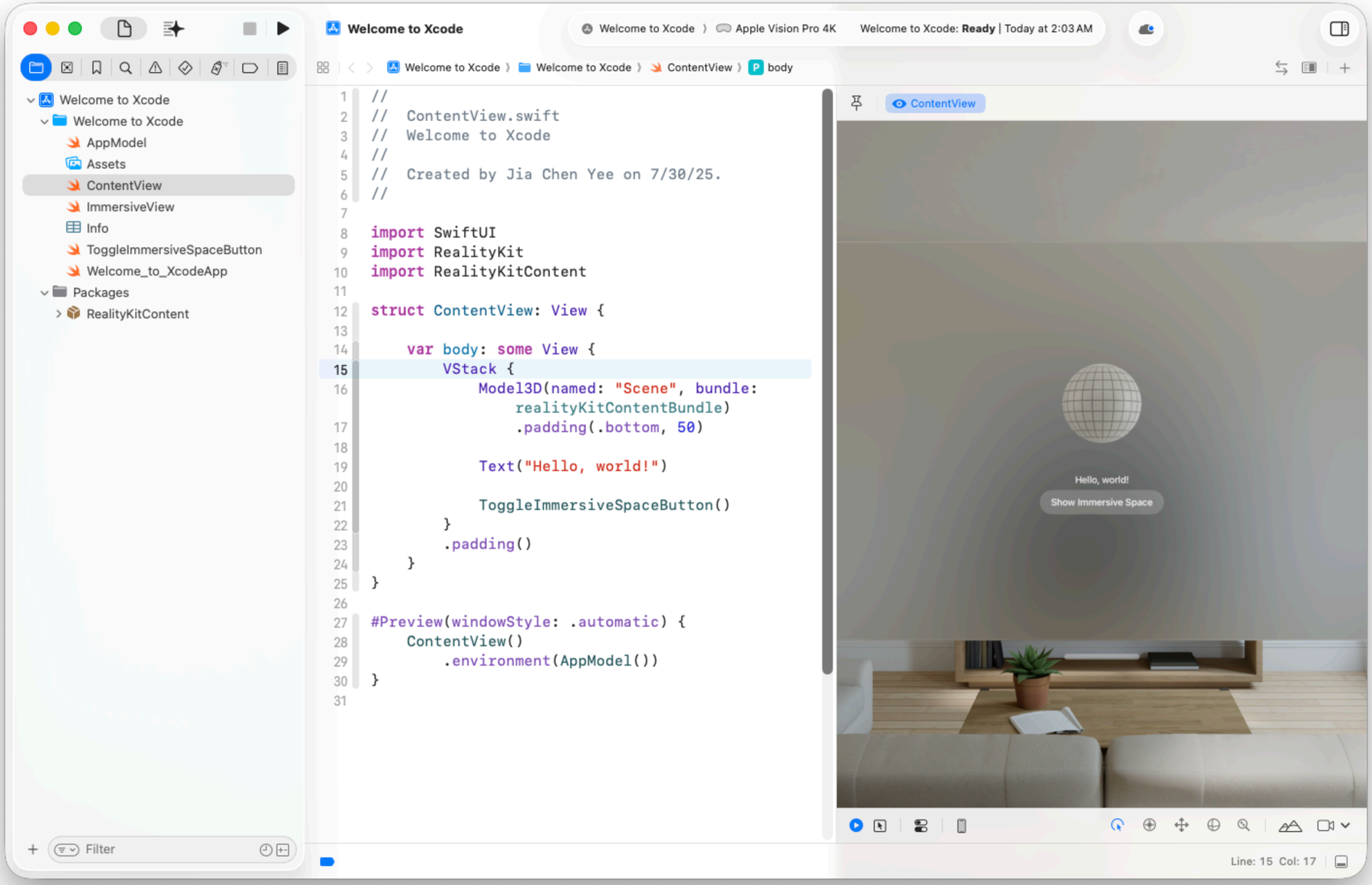


Create a visionOS Xcode Project


BETA
Xcode
Version 26.0 beta 4 (17A5285i)
Create New Project...
Clone Git Repository...
Open Existing Project...


No Selection
Choose a template for your new project:
Multiplatform
iOS
macOS
watchOS
tvOS
visionOS
DriverKit
Other
Filter
Application
App
Immersive Environment App
Framework & Library
Framework
Static Library
Cancel
Previous
Next
Not Applicable


No Selection
Choose options for your new project:
Product Name:
Welcome to Xcode
Team:
Jia Chen Yee
Organization Identifier:
com.example
Bundle Identifier:
com.example.Welcome-to-Xcode
Initial Scene:
Window
Immersive Space Renderer:
RealityKit
Immersive Space:
Full
Testing System:
None
Cancel
Previous
Next
Not Applicable


Welcome to Xcode
Welcome to Xcode
Apple Vision Pro 4K
Welcome to Xcode: Ready | Today at 2:03 AM
Welcome to Xcode
Welcome to Xcode
AppModel
Assets
ContentView
ImmersiveView
Info
ToggleImmersiveSpaceButton
Welcome_to_XcodeApp
Packages
RealityKitContent
Filter
Welcome to Xcode
Welcome to Xcode
ContentView
body
//
// ContentView.swift
// Welcome to Xcode
//
// Created by Jia Chen Yee on 7/30/25.
//
import SwiftUI
import RealityKit
import RealityKitContent
struct ContentView: View {
var body: some View {
VStack {
Model3D(named: "Scene", bundle:
realityKitContentBundle)
.padding(.bottom, 50)
Text("Hello, world!")
ToggleImmersiveSpaceButton()
}
.padding()
}
}
#Preview(windowStyle: .automatic) {
ContentView()
.environment(AppModel())
}
ContentView
Hello, world!
Show Immersive Space
Line: 15 Col: 17

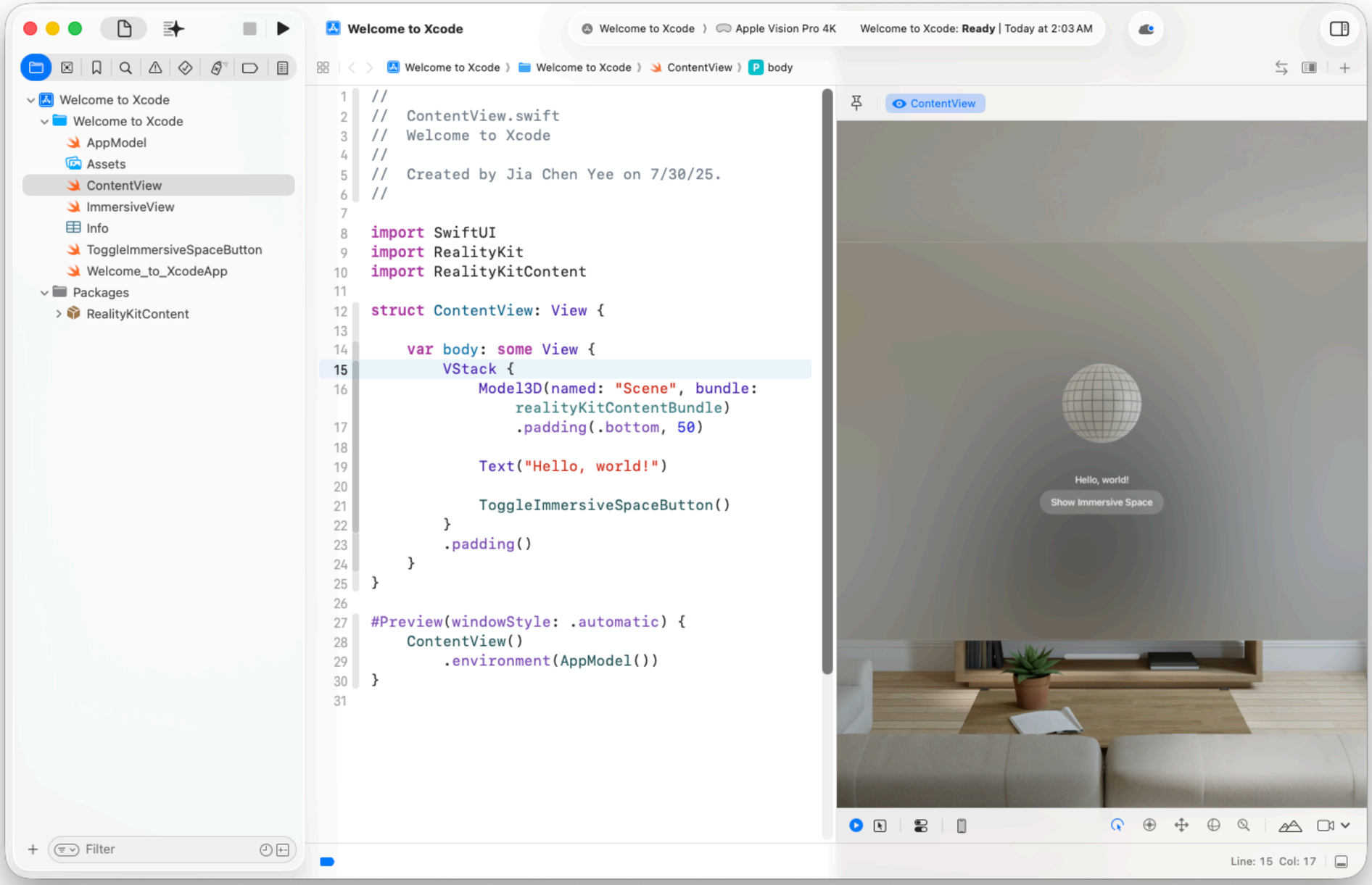
Welcome to Xcode

Create an Xcode Project

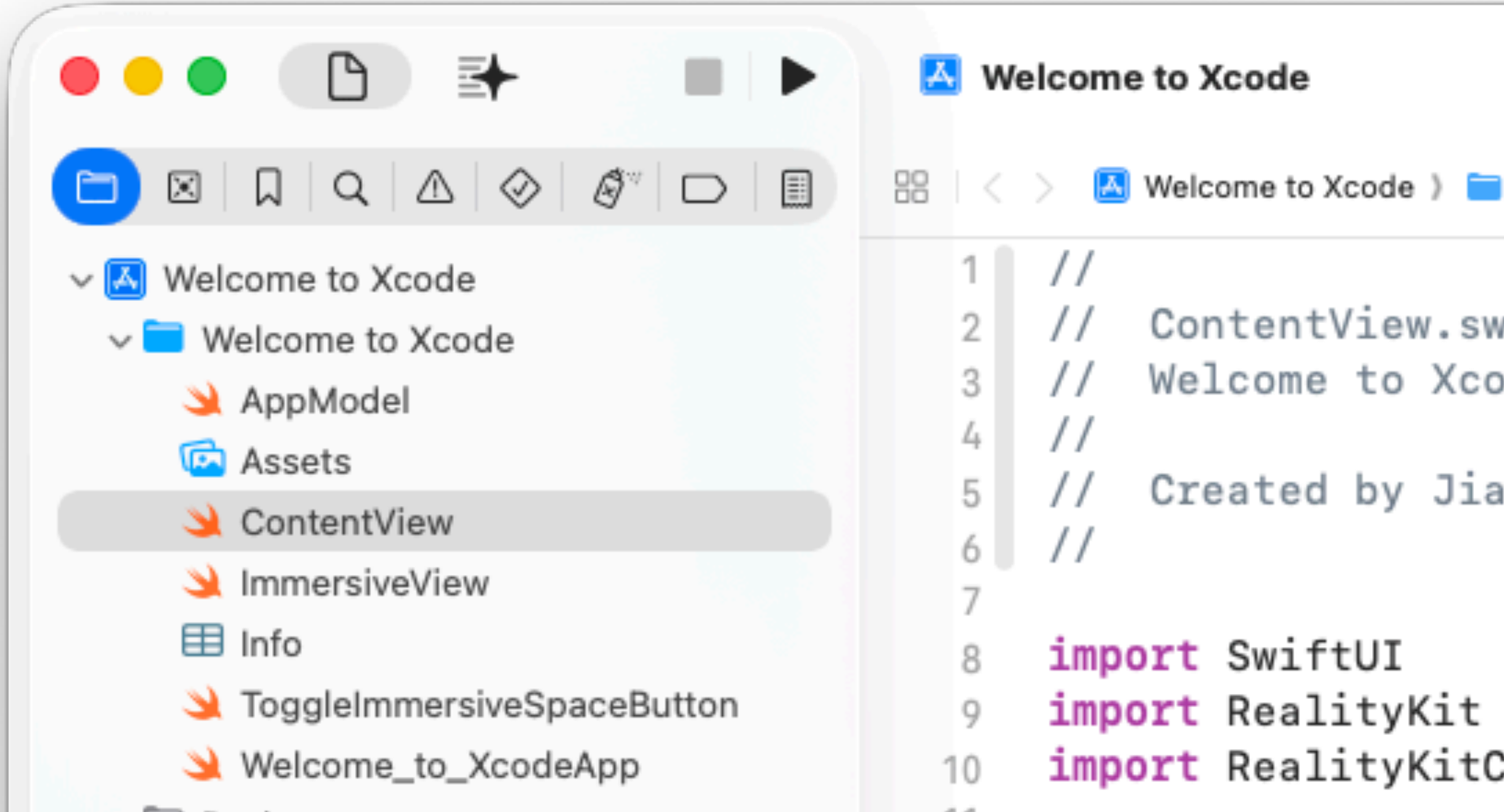
- Create a visionOS App project
- Name it “Welcome to Xcode”

5-minute timer has not started.

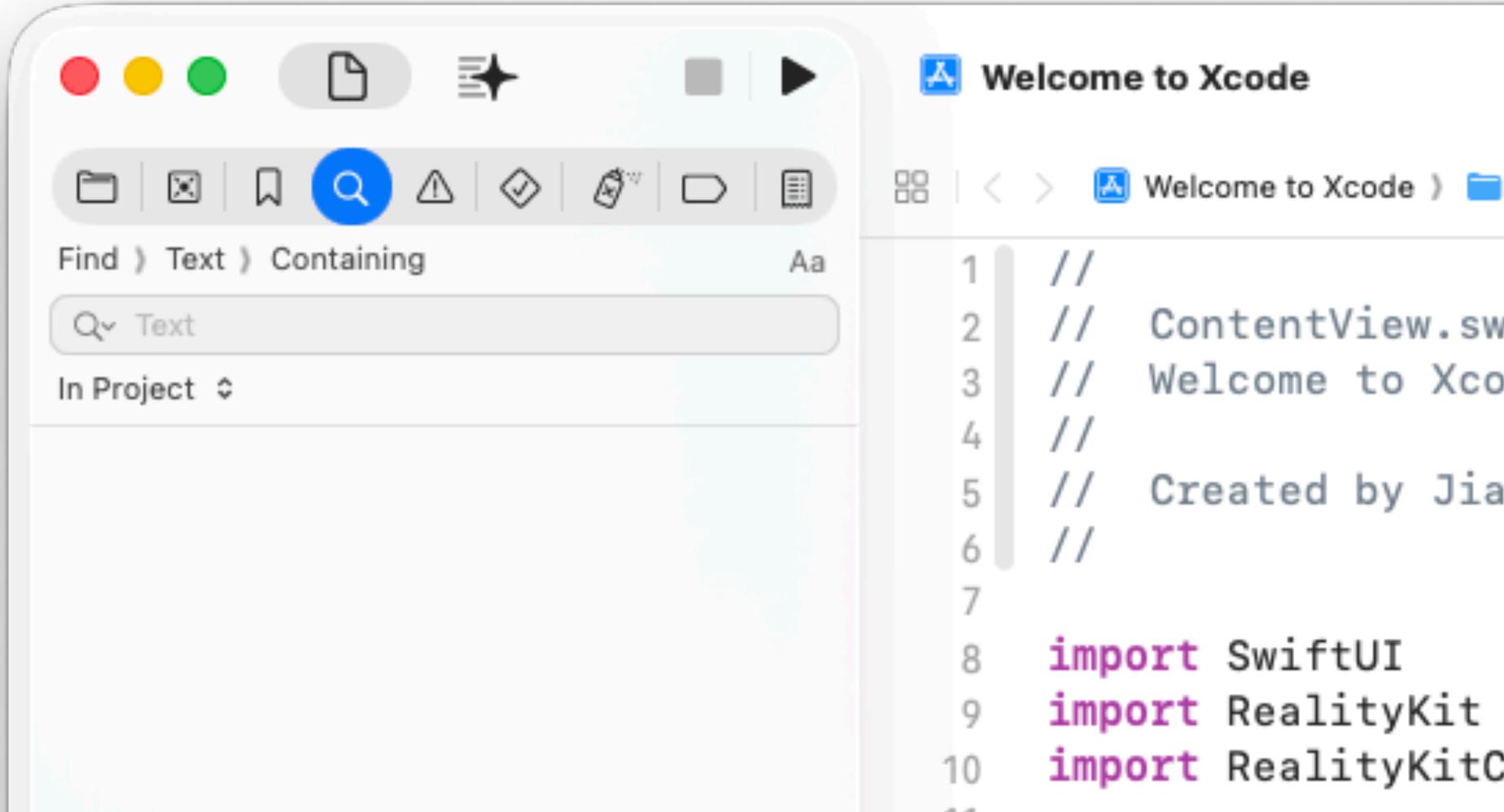
Navigating Xcode


Welcome to Xcode
Welcome to Xcode
Apple Vision Pro 4K
Welcome to Xcode: Ready | Today at 2:03 AM
Welcome to Xcode
Welcome to Xcode
AppModel
Assets
ContentView
ImmersiveView
Info
ToggleImmersiveSpaceButton
Welcome_to_XcodeApp
Packages
RealityKitContent
Filter
Welcome to Xcode
Welcome to Xcode
ContentView
body
//
// ContentView.swift
// Welcome to Xcode
//
// Created by Jia Chen Yee on 7/30/25.
//
import SwiftUI
import RealityKit
import RealityKitContent
struct ContentView: View {
var body: some View {
VStack {
Model3D(named: "Scene", bundle:
realityKitContentBundle)
.padding(.bottom, 50)
Text("Hello, world!")
ToggleImmersiveSpaceButton()
}
.padding()
}
}
#Preview(windowStyle: .automatic) {
ContentView()
.environment(AppModel())
}
ContentView
Hello, world!
Show Immersive Space
Line: 15 Col: 17

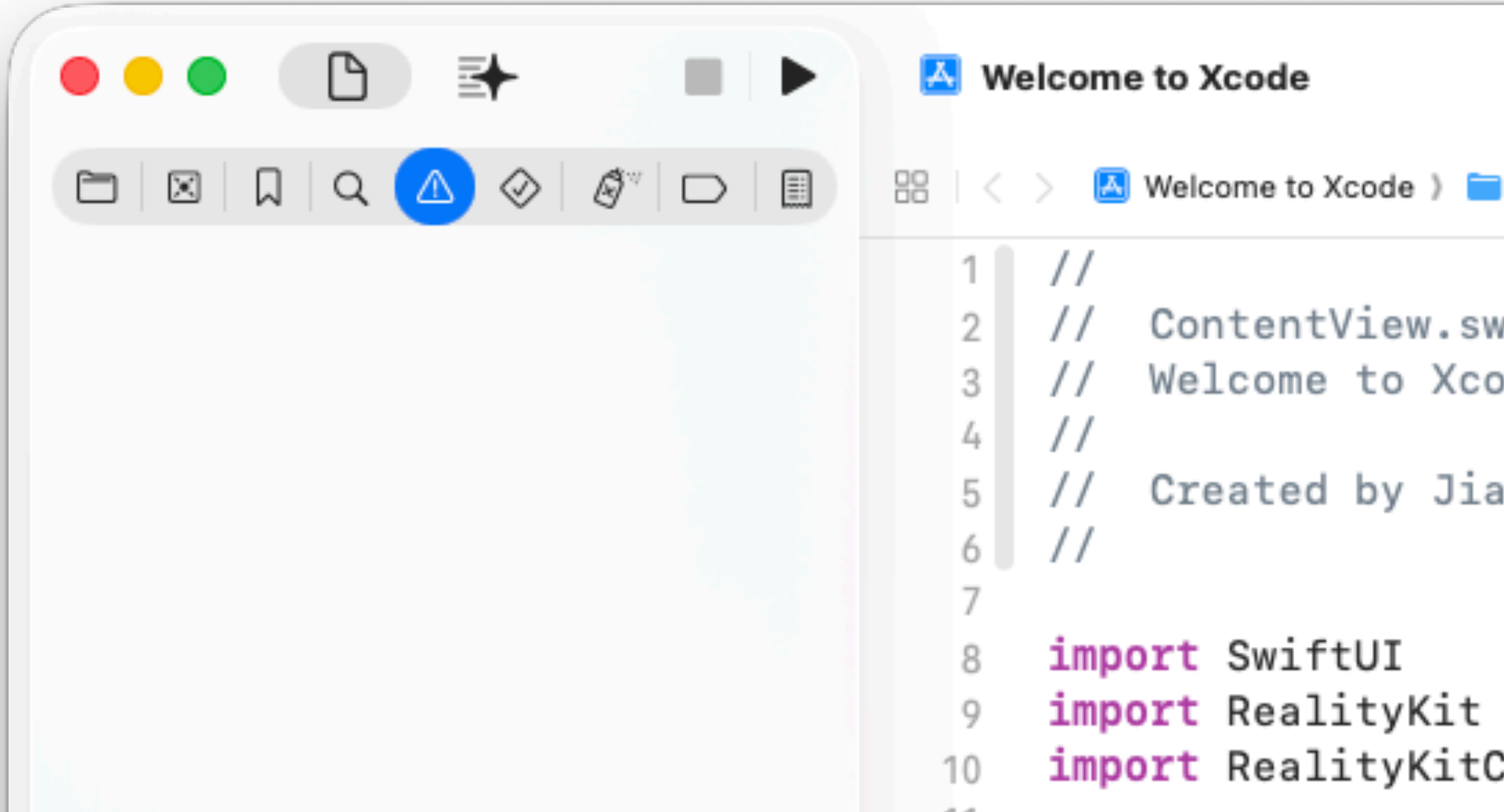
Project Navigator



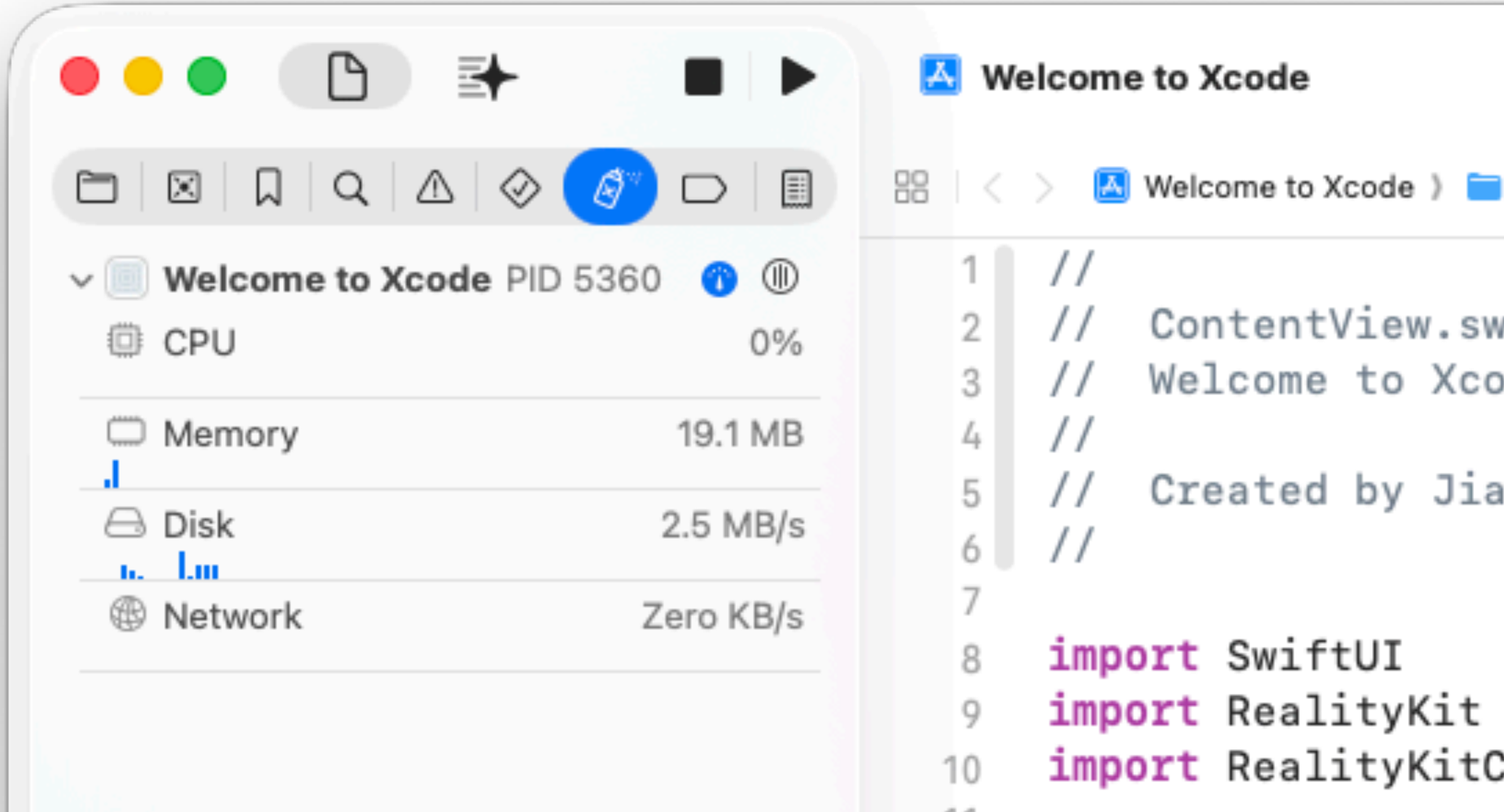
Search Entire Project



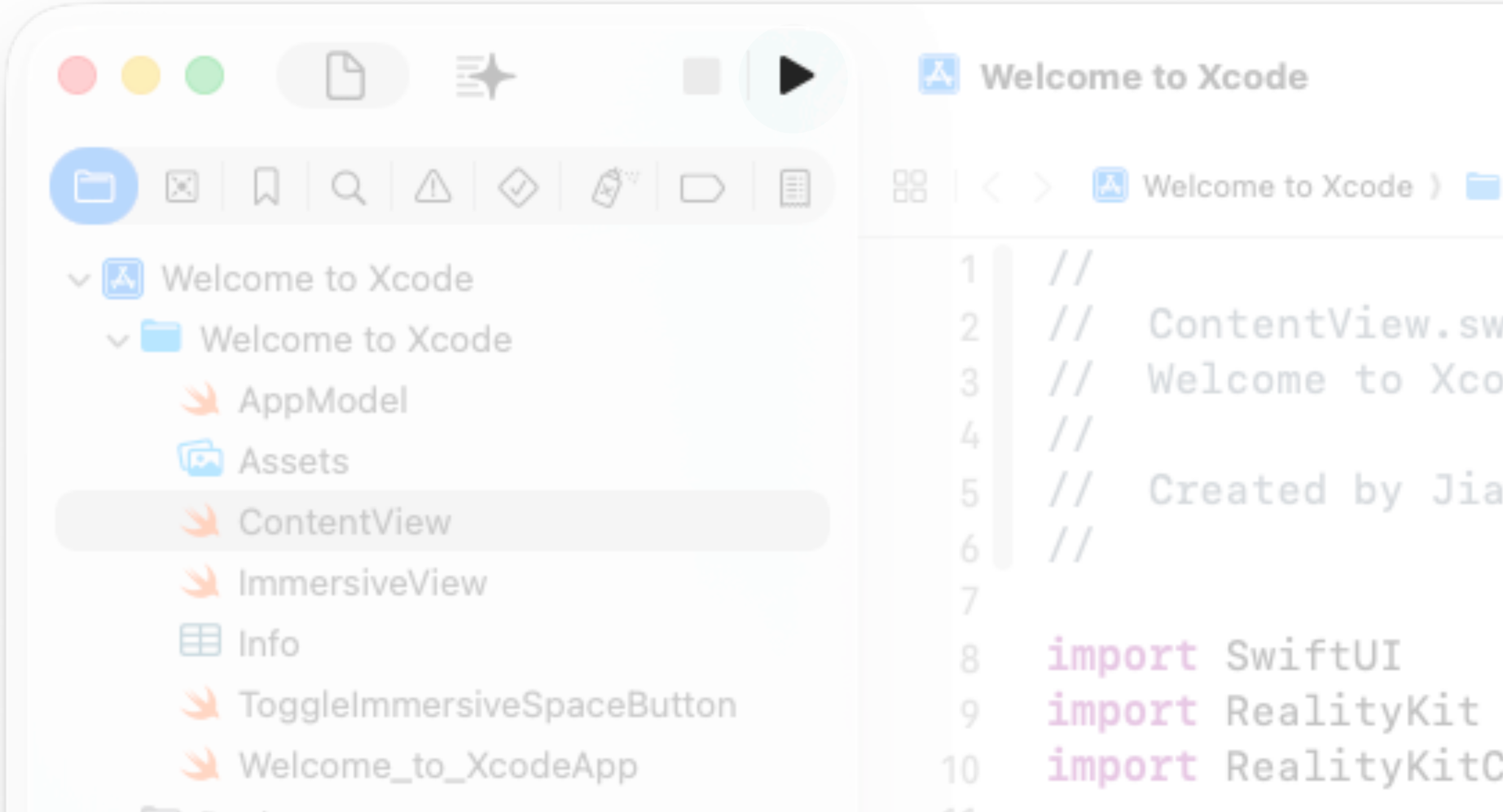
Warnings & Errors



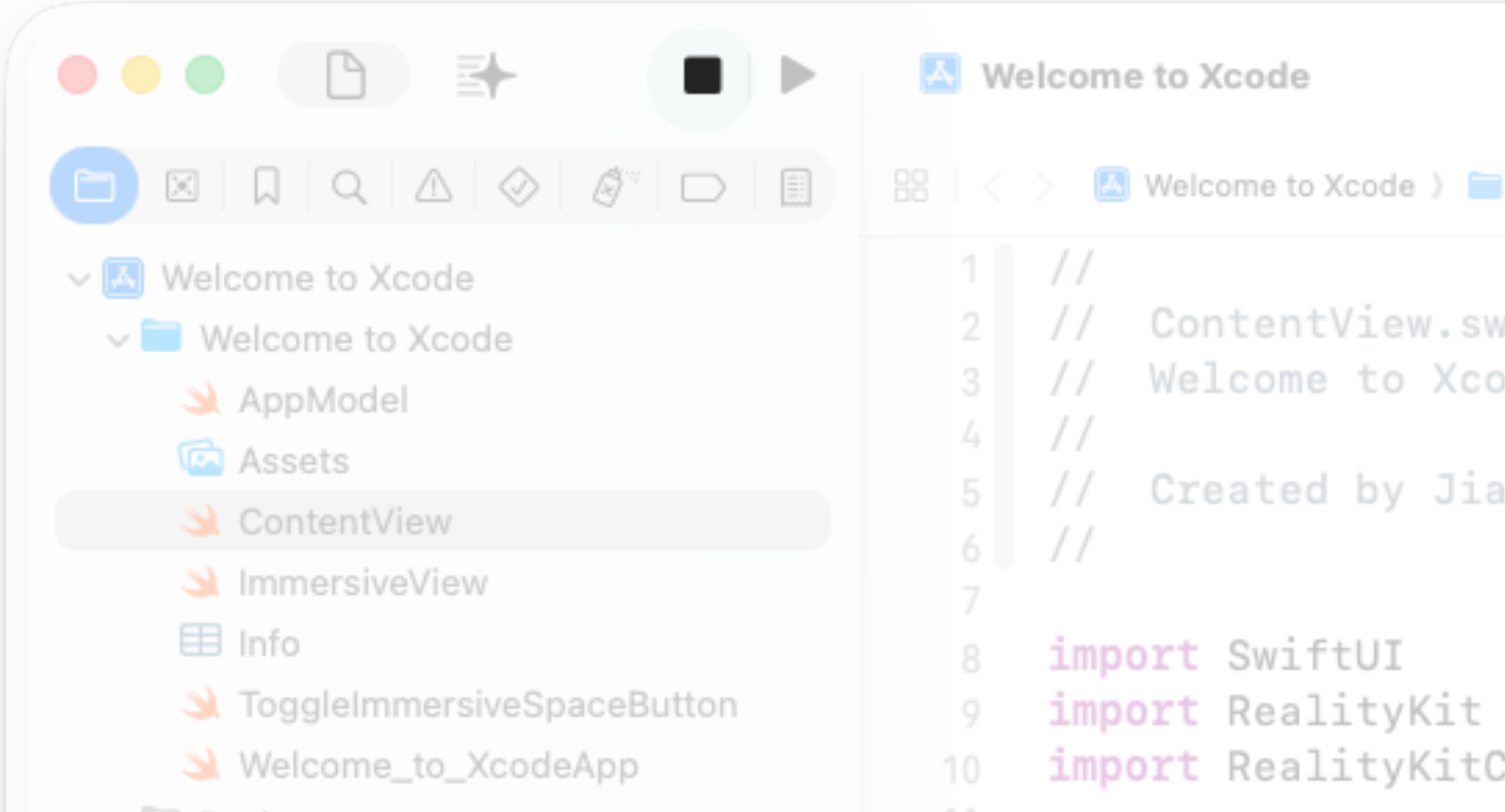
Runtime Inspector



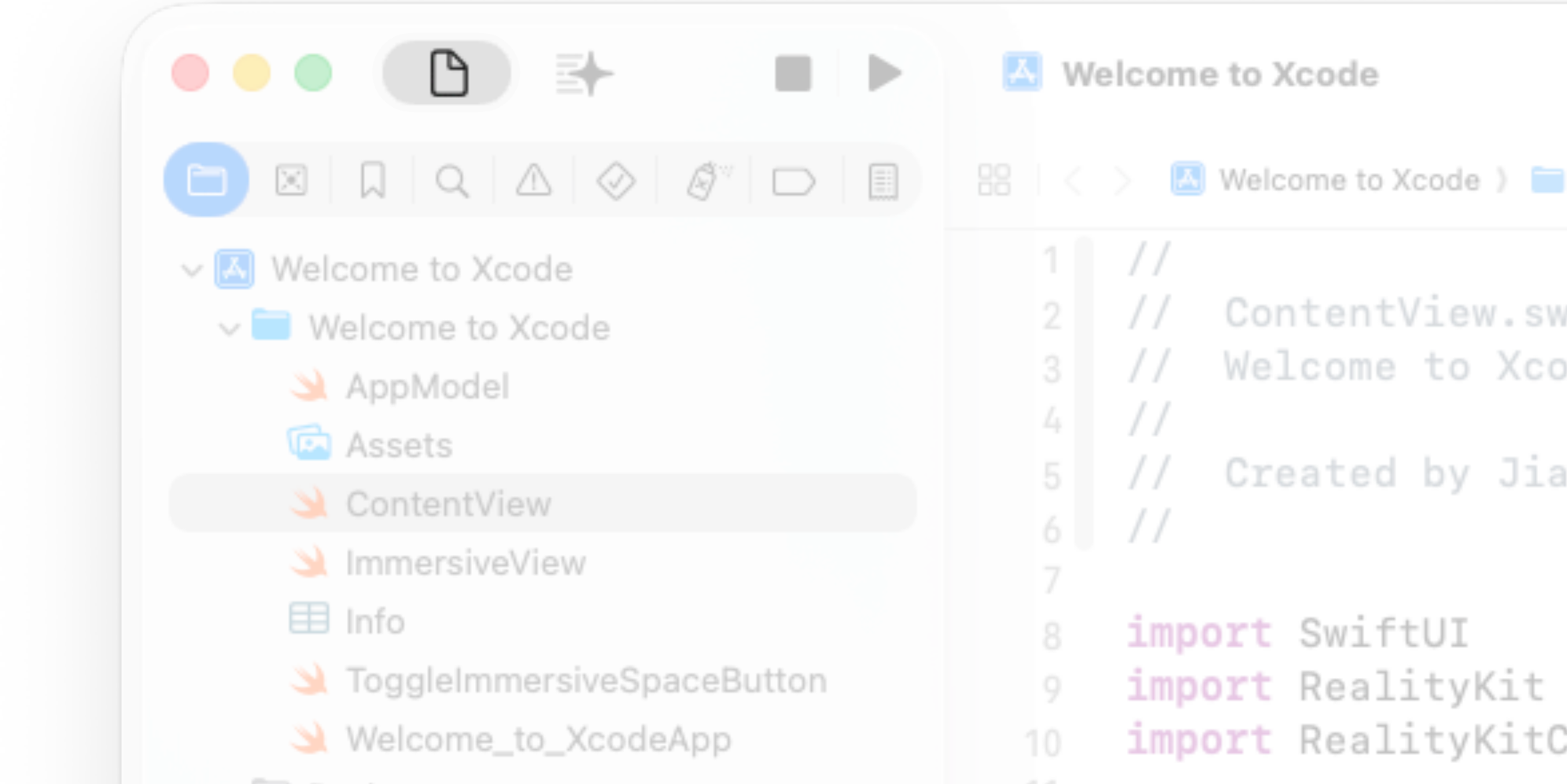
Run in Simulator



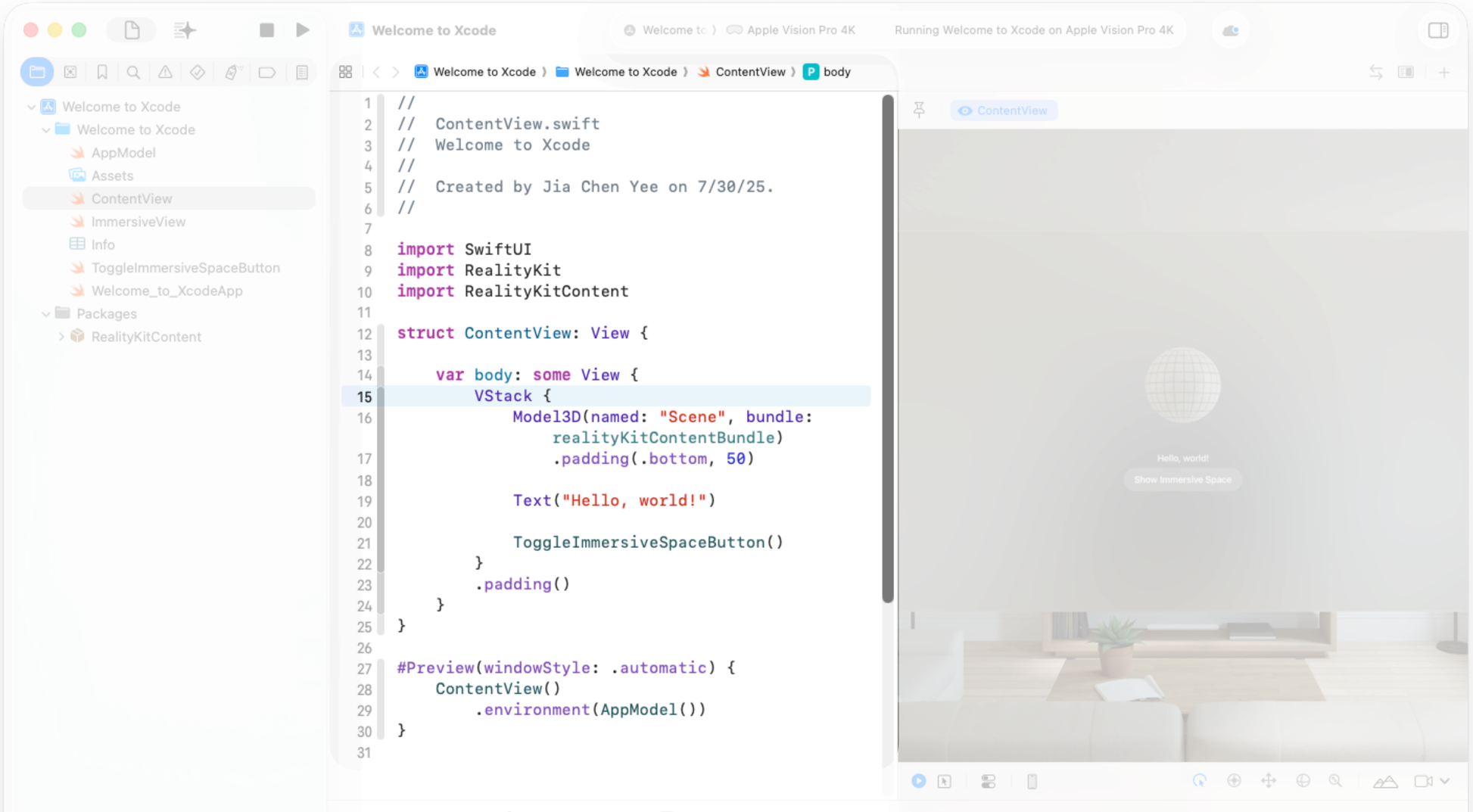
Stop Execution



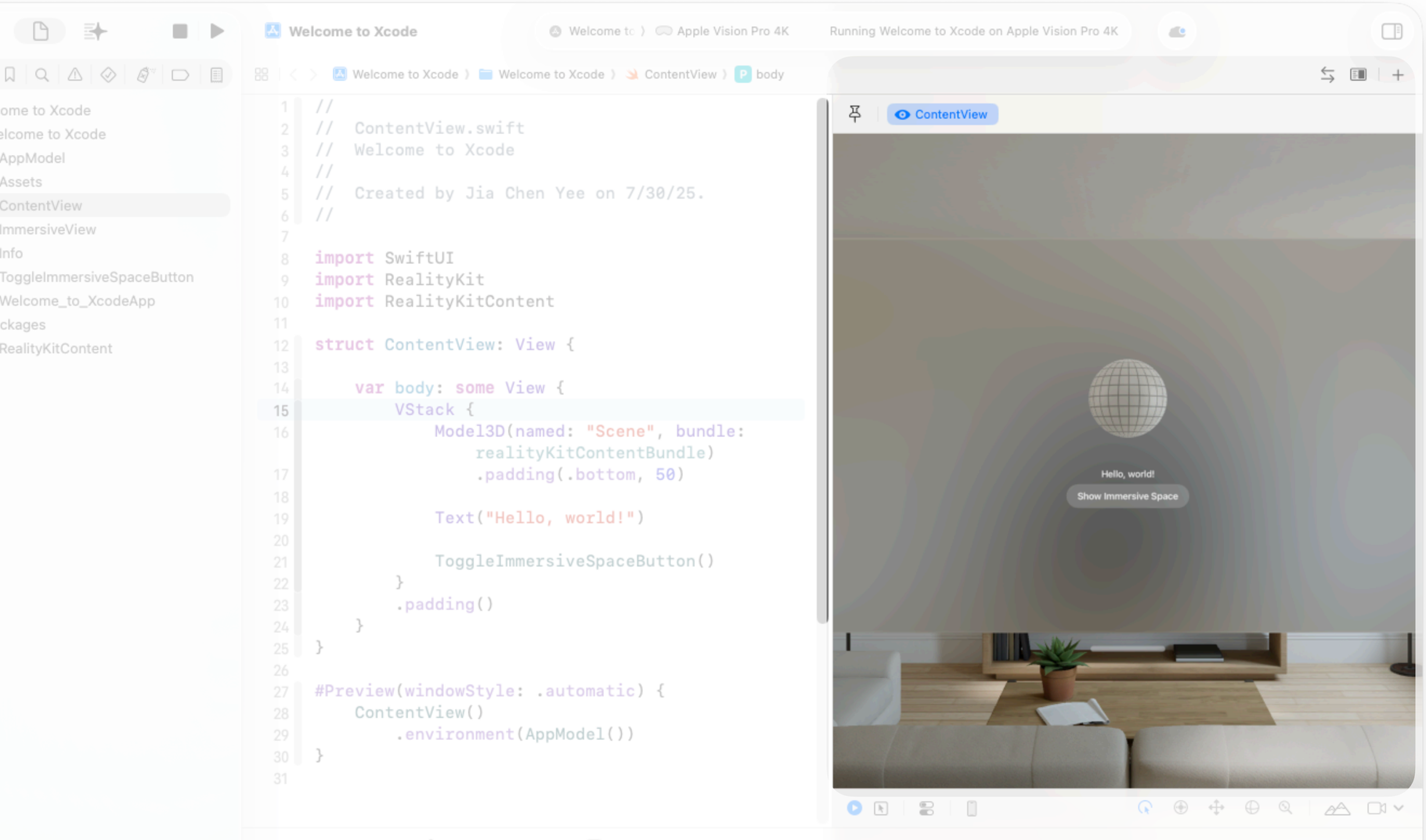
Show/Hide Project Navigator



Editor


Welcome to Xcode
Welcome to Xcode
Apple Vision Pro 4K
Running Welcome to Xcode on Apple Vision Pro 4K
Welcome to Xcode
Welcome to Xcode
ContentView
body
Welcome to Xcode
Welcome to Xcode
AppModel
Assets
ContentView
ImmersiveView
Info
ToggleImmersiveSpaceButton
Welcome_to_XcodeApp
Packages
RealityKitContent
//
// ContentView.swift
// Welcome to Xcode
//
// Created by Jia Chen Yee on 7/30/25.
//
import SwiftUI
import RealityKit
import RealityKitContent
struct ContentView: View {
var body: some View {
VStack {
Model3D(named: "Scene", bundle:
realityKitContentBundle)
.padding(.bottom, 50)
Text("Hello, world!")
ToggleImmersiveSpaceButton()
}
.padding()
}
}
#Preview(windowStyle: .automatic) {
ContentView()
.environment(AppModel())
}
ContentView
Hello, world!
Show Immersive Space

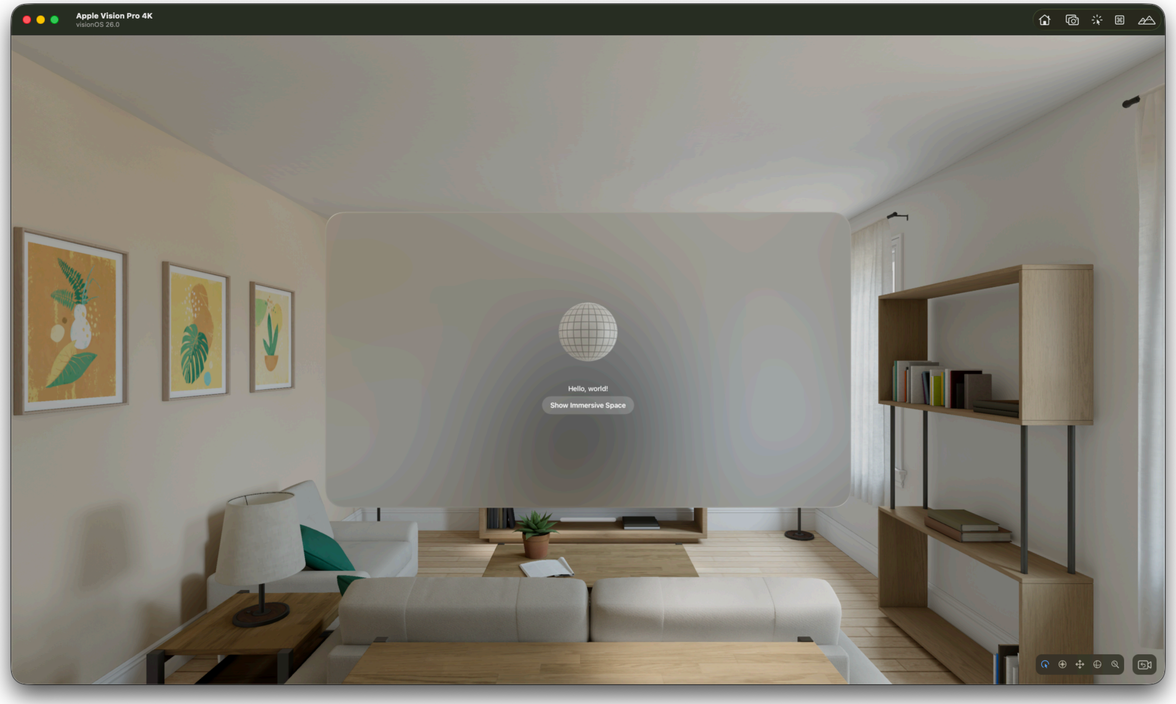
Live Preview (Canvas)



Welcome to Xcode

Run in Simulator

- Click ▶ in the toolbar to run your code
- Make sure the Simulator works!



5-minute timer has not started.

Welcome to Xcode

Text & Image Views

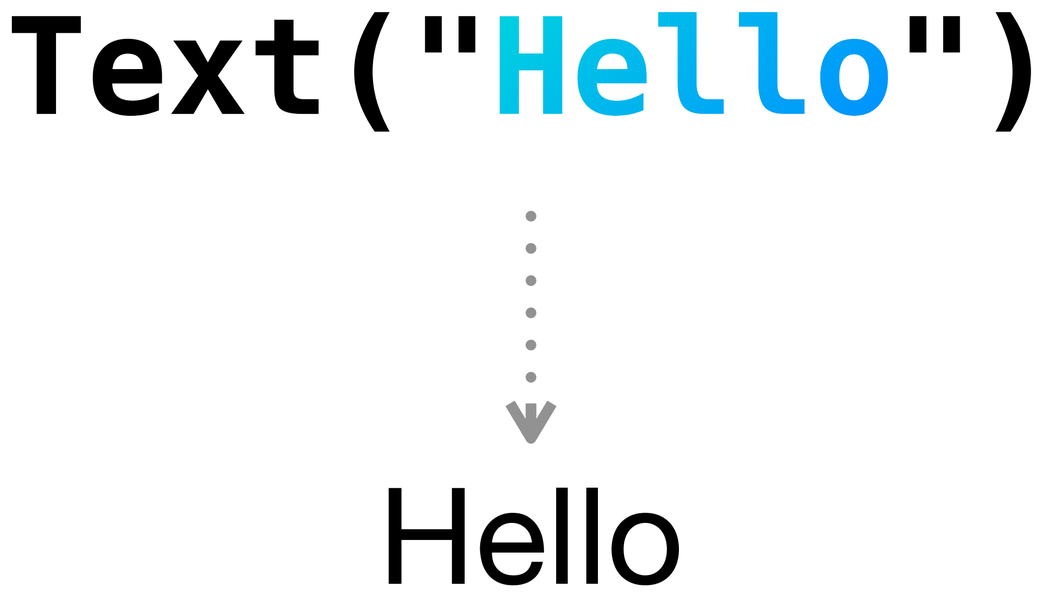
Stacks & Layouts

View Modifiers

Text Views

A text view **draws a string in your app's user interface** using a body font that's appropriate for the current platform.

```
Text("Hello")
```


Text("Hello")
Hello

```
Text("Hello!!")
```

Hello!!

Image Views

An image view **displays an image in your app's user interface**. The image can be an external image (PNG, JPG, etc.) or an SF Symbol.

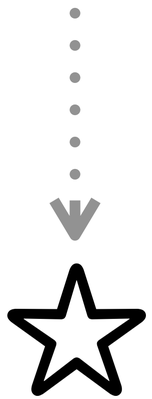


SF Symbols is a library of over 6,000 symbols that are designed to **integrate seamlessly** with the system on Apple platforms.

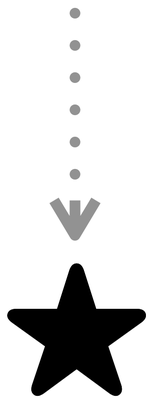


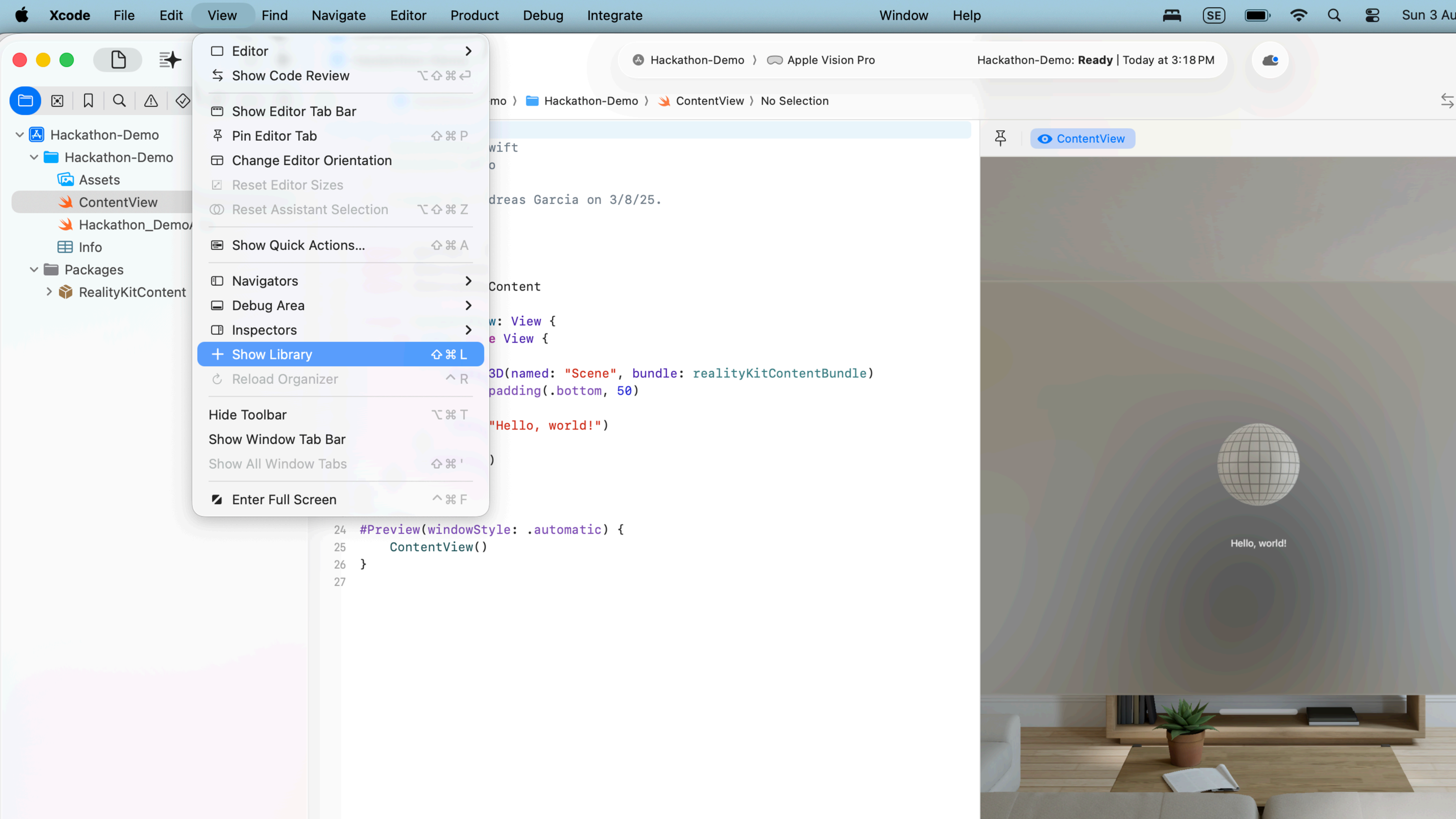

```
Image(systemName: "star")
```

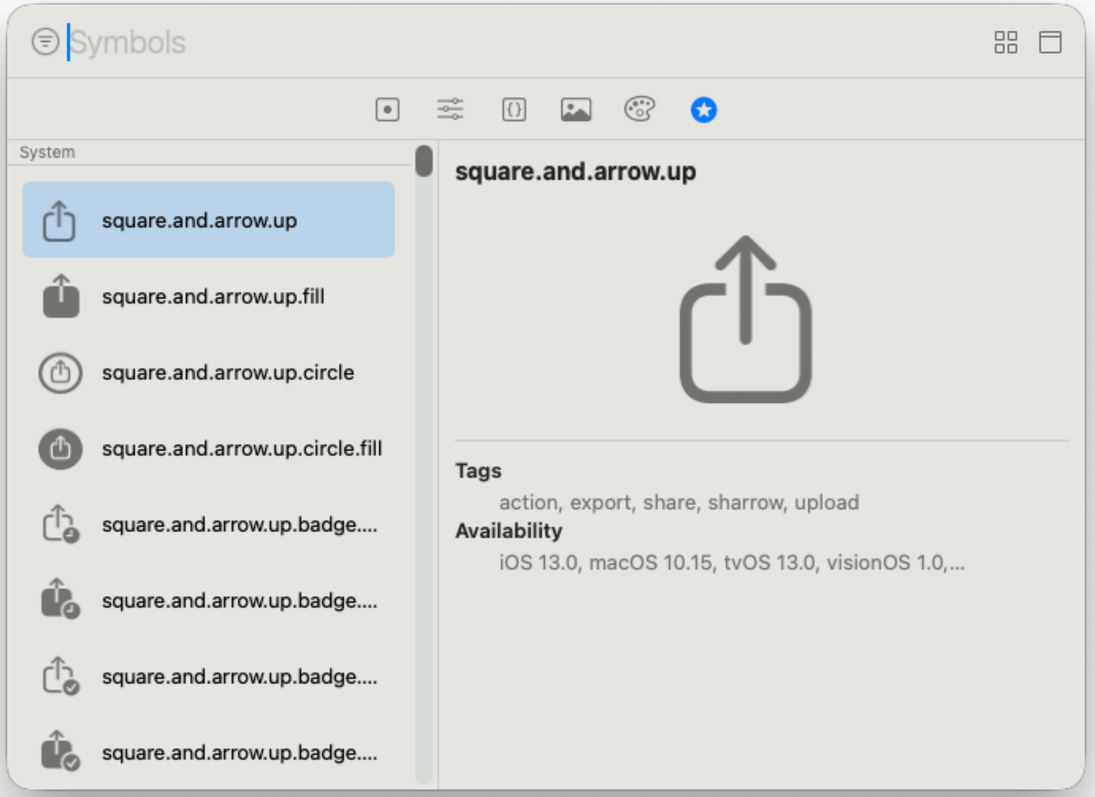
```
Image(systemName: "star")
```



```
Image(systemName: "star.fill")
```




Xcode File Edit View Find Navigate Editor Product Debug Integrate Window Help
Sun 3 Au
Editor
Show Code Review ⌥⇧⌘⏎
Show Editor Tab Bar
Pin Editor Tab ⇧⌘P
Change Editor Orientation
Reset Editor Sizes
Reset Assistant Selection ⌥⇧⌘Z
Show Quick Actions... ⇧⌘A
Navigators
Debug Area
Inspectors
Show Library ⇧⌘L
Reload Organizer ^R
Hide Toolbar ⌥⌘T
Show Window Tab Bar
Show All Window Tabs ⇧⌘'
Enter Full Screen ^⌘F
Hackathon-Demo 〉 Apple Vision Pro
Hackathon-Demo: Ready | Today at 3:18 PM
Hackathon-Demo
Hackathon-Demo
Assets
ContentView
Hackathon_Demo
Info
Packages
RealityKitContent
mo 〉 Hackathon-Demo 〉 ContentView 〉 No Selection
wift
o
dreas Garcia on 3/8/25.
Content
w: View {
e View {
3D(named: "Scene", bundle: realityKitContentBundle)
padding(.bottom, 50)
"Hello, world!")
)
24 #Preview(windowStyle: .automatic) {
25 ContentView()
26 }
27
ContentView
Hello, world!


Symbols
System
square.and.arrow.up
square.and.arrow.up.fill
square.and.arrow.up.circle
square.and.arrow.up.circle.fill
square.and.arrow.up.badge....
square.and.arrow.up.badge....
square.and.arrow.up.badge....
square.and.arrow.up.badge....
square.and.arrow.up
Tags
action, export, share, sharrow, upload
Availability
iOS 13.0, macOS 10.15, tvOS 13.0, visionOS 1.0,...

Change the Text and Image

- Change the Text to your name
- Change the Image icon to something else! Any icon you can find in the ✪ *Symbol Explorer*.

5-minute timer has not started.

Welcome to Xcode

Text & Image Views

Stacks & Layouts

View Modifiers

Vertical Stack

A view that arranges its subviews in **a vertical line**.

```swift
struct ContentView: View {
    var body: some View {
        VStack {
            Image(systemName: "face.smiling")
                .imageScale(.large)
                .foregroundStyle(.tint)
            Text("Jia Chen")
        }
        .padding()
    }
}
```

ContentView.swift

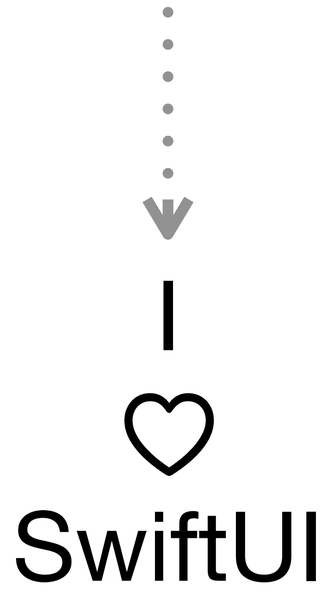
Contents of the VStack

```
struct ContentView: View {
    var body: some View {
        VStack {
            Image(systemName: "face.smiling")
                .imageScale(.large)
                .foregroundStyle(.tint)
            Text("Jia Chen")
        }
        .padding()
    }
}
```

ContentView.swift

```
VStack {
    Text("I")
    Image(systemName: "heart")
    Text("SwiftUI")
}
```

```
VStack {
    Text("I")
    Image(systemName: "heart")
    Text("SwiftUI")
}
```



Horizontal Stack

A view that arranges its subviews in **a horizontal line**.

```swift
struct ContentView: View {
    var body: some View {
        HStack {
            Image(systemName: "face.smiling")
                .imageScale(.large)
                .foregroundStyle(.tint)
            Text("Jia Chen")
        }
        .padding()
    }
}
```

ContentView.swift

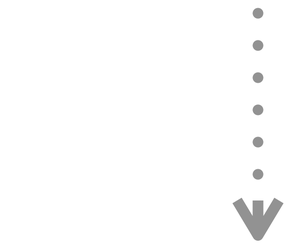
Contents of the HStack

```
struct ContentView: View {
    var body: some View {
        HStack {
            Image(systemName: "face.smiling")
                .imageScale(.large)
                .foregroundStyle(.tint)
            Text("Jia Chen")
        }
        .padding()
    }
}
```

ContentView.swift

```
HStack {
    Text("I")
    Image(systemName: "heart")
    Text("SwiftUI")
}
```

```
HStack {
    Text("I")
    Image(systemName: "heart")
    Text("SwiftUI")
}
```



I ♡ SwiftUI

Z-Axis Stack

A view that **overlays its subviews**, aligning them in both axes.

```swift
struct ContentView: View {
    var body: some View {
        ZStack {
            Image(systemName: "face.smiling")
                .imageScale(.large)
                .foregroundStyle(.tint)
            Text("Jia Chen")
        }
        .padding()
    }
}
```

ContentView.swift

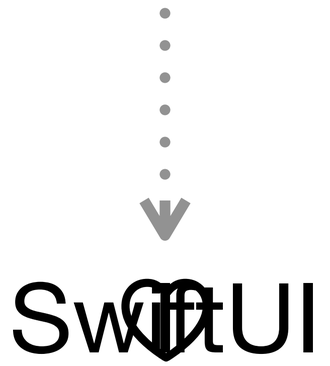
Contents of the ZStack

```
struct ContentView: View {
    var body: some View {
        ZStack {
            Image(systemName: "face.smiling")
                .imageScale(.large)
                .foregroundStyle(.tint)
            Text("Jia Chen")
        }
        .padding()
    }
}
```

ContentView.swift

```
ZStack {
    Text("I")
    Image(systemName: "heart")
    Text("SwiftUI")
}
```

```
ZStack {
    Text("I")
    Image(systemName: "heart")
    Text("SwiftUI")
}
```

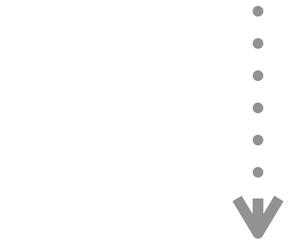


Spacer

A **flexible, invisible** view that expands to fill available space in the direction of its containing stack, pushing sibling views apart.

```
HStack {
    Text("I")
    Image(systemName: "heart")
    Text("SwiftUI")
}
```

```
HStack {
    Text("I")
    Image(systemName: "heart")
    Text("SwiftUI")
}
```



I ♡ SwiftUI

```
HStack {
    Text("I")
    Image(systemName: "heart")
    Spacer()
    Text("SwiftUI")
}
```

I ♡ SwiftUI

Welcome to Xcode

Text & Image Views

Stacks & Layouts

View Modifiers

View Modifiers allow you to **customise the appearance and behaviour** of your app's views.

```swift
struct ContentView: View {
    var body: some View {
        HStack {
            Image(systemName: "face.smiling")
          Text("Jia Chen")
        }
    }
}
```

ContentView.swift

```swift
struct ContentView: View {
    var body: some View {
        HStack {
            Image(systemName: "face.smiling")
          Text("Jia Chen")
        }
    }
}
```

ContentView.swift

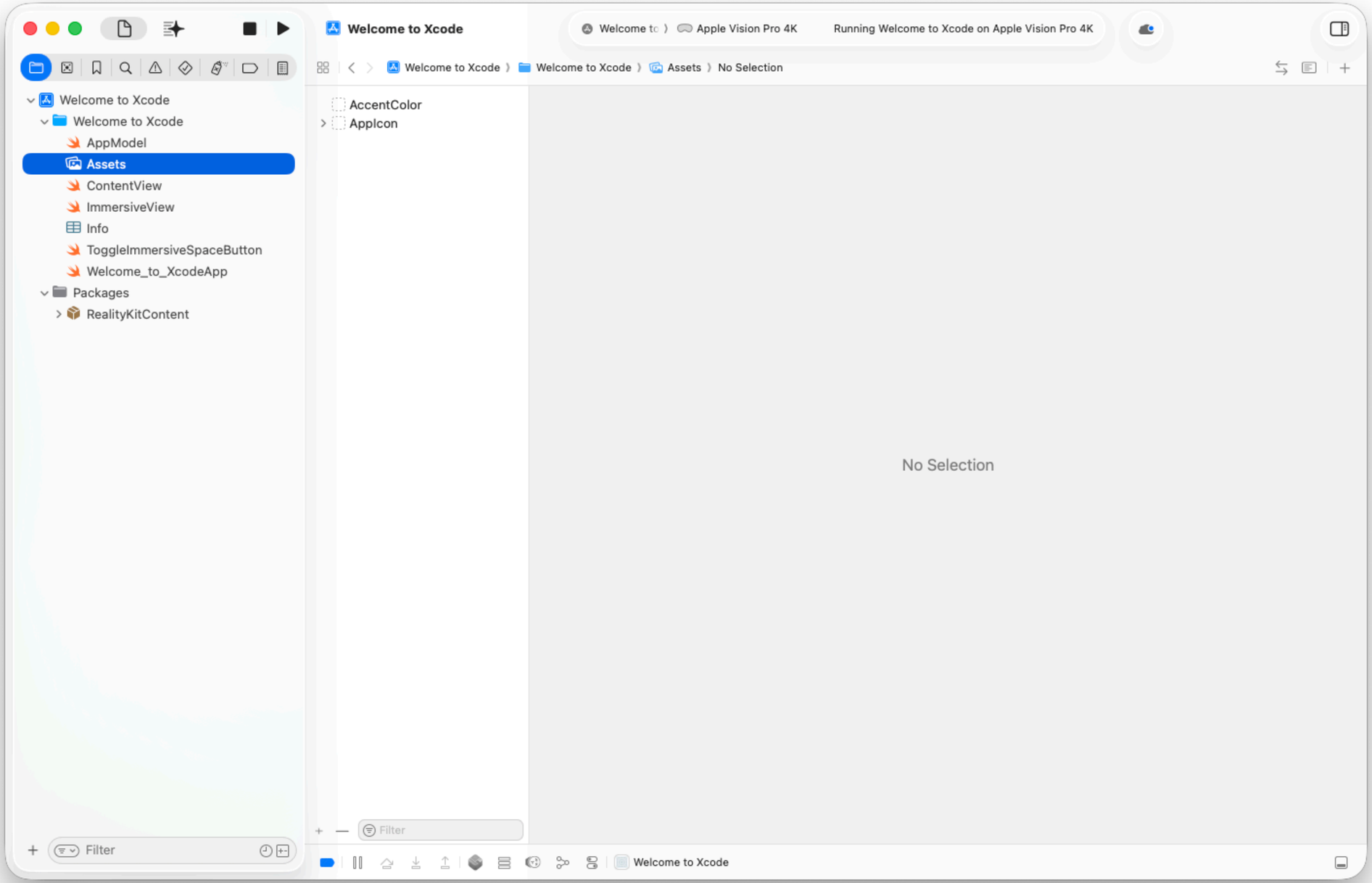
```swift
struct ContentView: View {
    var body: some View {
        HStack {
            Image(systemName: "face.smiling")
                .imageScale(.large)
                .foregroundStyle(.tint)
            Text("Jia Chen")
        }
        .padding()
        .background(.red)
    }
}
```

ContentView.swift

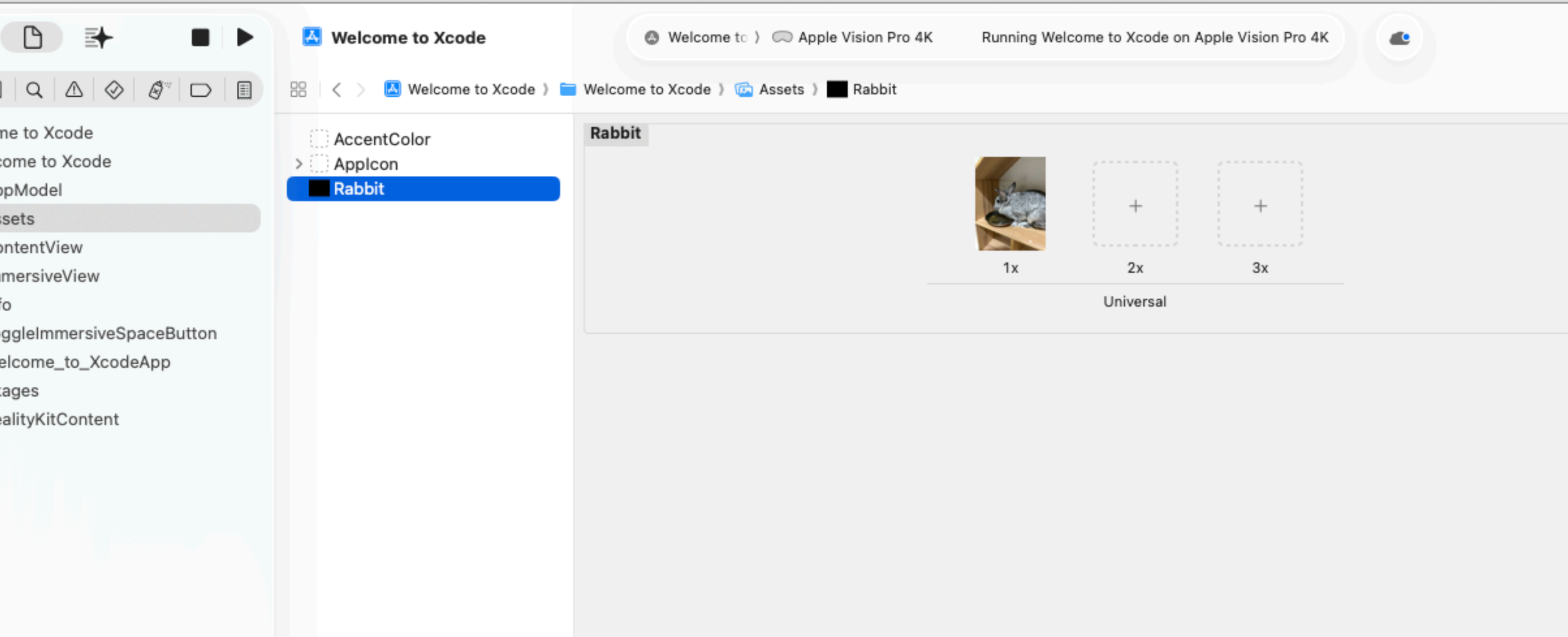
```swift
struct ContentView: View {
    var body: some View {
        HStack {
            Image(systemName: "face.smiling")
                .imageScale(.large)
                .foregroundStyle(.tint)
            Text("Jia Chen")
        }
        .padding()
        .background(.red)
    }
}
```

ContentView.swift

Image Modifiers


Welcome to Xcode
Welcome to Xcode
AppModel
Assets
ContentView
ImmersiveView
Info
ToggleImmersiveSpaceButton
Welcome_to_XcodeApp
Packages
RealityKitContent
Filter
Welcome to Xcode
Apple Vision Pro 4K
Running Welcome to Xcode on Apple Vision Pro 4K
Welcome to Xcode
Welcome to Xcode
Assets
No Selection
AccentColor
AppIcon
Filter
No Selection
Welcome to Xcode

Drag in images to import



```
Image("Rabbit")
```

```
Image("Rabbit")
```

```
Image("Rabbit")
    .resizable()
```

```
Image("Rabbit")
    .resizable()
    .scaledToFit()
```

```
Image("Rabbit")
    .resizable()
    .scaledToFill()
```

View Modifiers

Chaining Modifiers

Chaining Modifiers

```
.bold()
```

Chaining Modifiers

```
.bold()
.background(.yellow)
```

Chaining Modifiers

```
.bold()
.background(.yellow)
.padding()
```

Chaining Modifiers

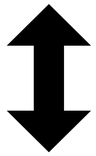
```
.bold()
.background(.yellow)
.padding()
.background(.black)
```

Chaining Modifiers

```
.bold()
.background(.yellow)
.padding()
.background(.black)
.cornerRadius(16)
```

Chaining Modifiers

Order matters



```
.bold()
.padding()
.background(.yellow)
.padding()
.background(.black)
.cornerRadius(16)
```

What can **View Modifiers** do?

```
.rotationEffect(.degrees(10))
```

```
.shadow(radius: 10)
```

```
.border(.red)
```

```
.colorInvert()
```

```
.cornerRadius(16)
```

```
.opacity(0.5)
```

```
.padding(16)
```

```
.mask {
    Circle()
}
```

What can **View Modifiers** do?

```
.blur(radius: 3)
```

```
.bold()
```

```
.font(.system(size: 20
              weight: .bold,
              design: .monospaced))
```

Font Weights

```
.background(.teal)
```

```
.frame(width: 100
       height: 100)
```

100

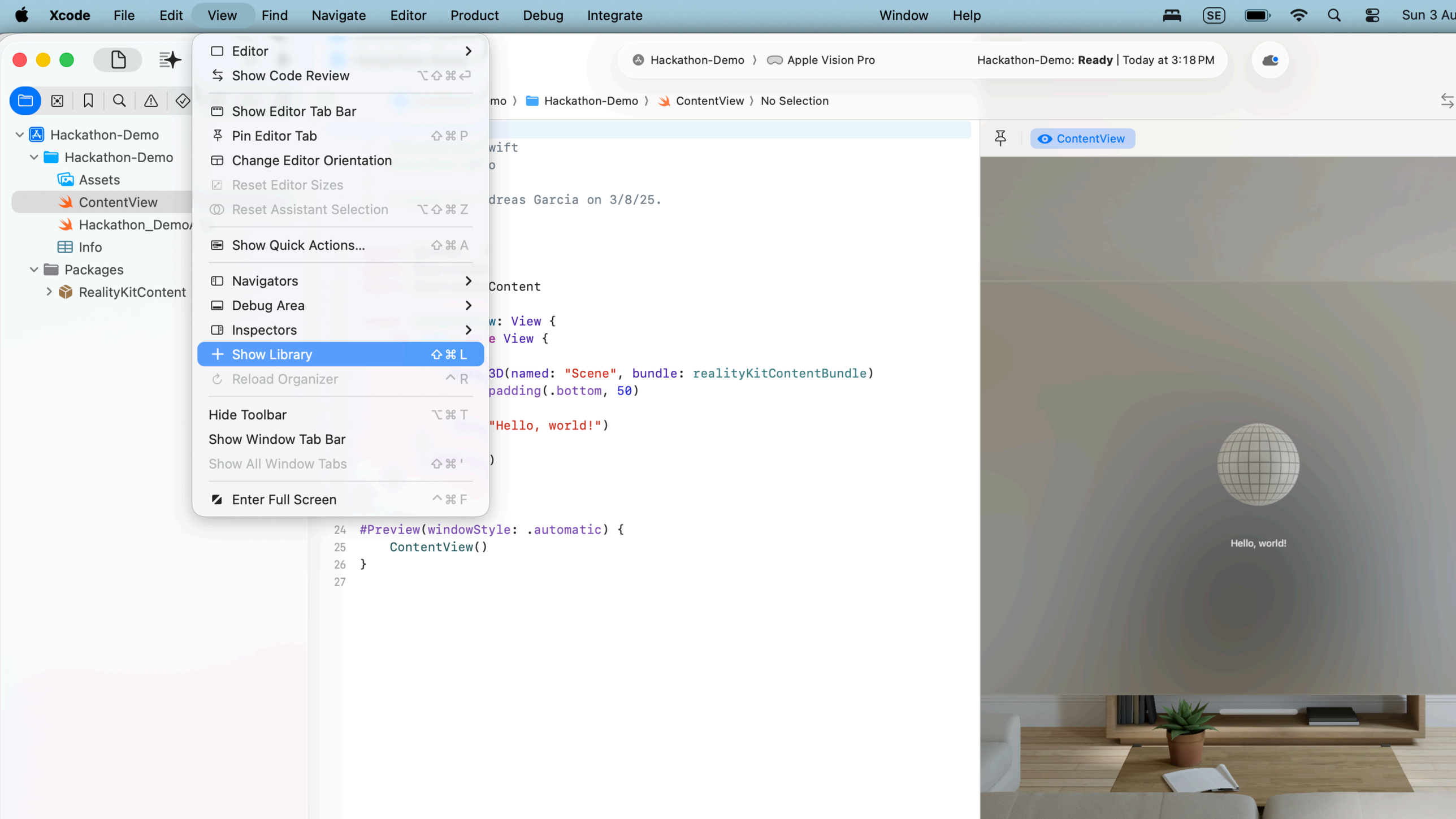
100

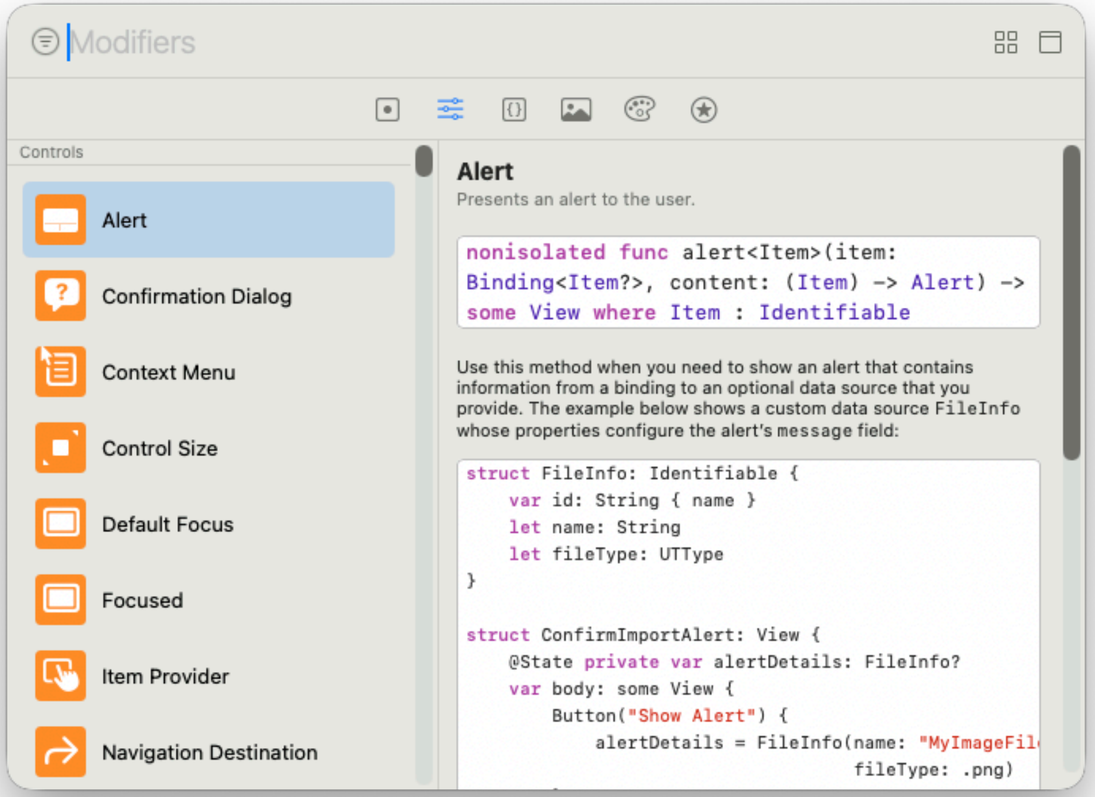
```
.foregroundStyle(.green)
```

```
.overlay {
    Text("Hello")
}
```

Hello

Modifiers Library


Xcode File Edit View Find Navigate Editor Product Debug Integrate Window Help
Sun 3 Au
Editor
Show Code Review ⌥⇧⌘⏎
Show Editor Tab Bar
Pin Editor Tab ⇧⌘P
Change Editor Orientation
Reset Editor Sizes
Reset Assistant Selection ⌥⇧⌘Z
Show Quick Actions... ⇧⌘A
Navigators
Debug Area
Inspectors
Show Library ⇧⌘L
Reload Organizer ^R
Hide Toolbar ⌥⌘T
Show Window Tab Bar
Show All Window Tabs ⇧⌘'
Enter Full Screen ^⌘F
Hackathon-Demo 〉 Apple Vision Pro
Hackathon-Demo: Ready | Today at 3:18 PM
Hackathon-Demo
Hackathon-Demo
Assets
ContentView
Hackathon_Demo
Info
Packages
RealityKitContent
mo 〉 Hackathon-Demo 〉 ContentView 〉 No Selection
wift
o
dreas Garcia on 3/8/25.
Content
w: View {
e View {
3D(named: "Scene", bundle: realityKitContentBundle)
padding(.bottom, 50)
"Hello, world!")
)
24 #Preview(windowStyle: .automatic) {
25 ContentView()
26 }
27
ContentView
Hello, world!


Modifiers
Controls
Alert
Confirmation Dialog
Context Menu
Control Size
Default Focus
Focused
Item Provider
Navigation Destination
Alert
Presents an alert to the user.
nonisolated func alert<Item>(item: Binding<Item?>, content: (Item) -> Alert) -> some View where Item : Identifiable
Use this method when you need to show an alert that contains information from a binding to an optional data source that you provide. The example below shows a custom data source FileInfo whose properties configure the alert's message field:
struct FileInfo: Identifiable {
var id: String { name }
let name: String
let fileType: UTType
}
struct ConfirmImportAlert: View {
@State private var alertDetails: FileInfo?
var body: some View {
Button("Show Alert") {
alertDetails = FileInfo(name: "MyImageFil
fileType: .png)

Fitting & Filling

Some views are fitting, they take up **as little space as possible**;
other views are filling, they take up **as much space as possible**.

Fitting Views

Takes up as little space as possible.

```
Text("Hello")
```

→


```
Image(systemName: "hand.wave")
```

→

```
VStack {
    Image(systemName: "hand.wave")
    Text("Hello")
}
```

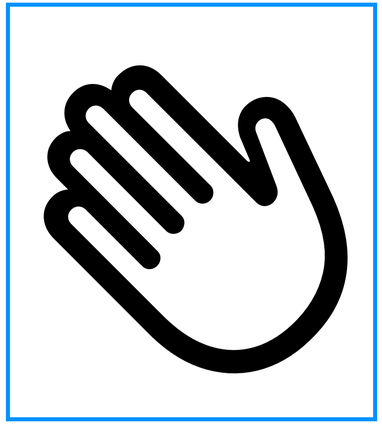
→

Filling Views

Takes up as much space as possible.

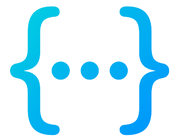
```
Image(systemName: "hand.wave")
    .resizable()
    .scaledToFit()
```

→



```
HStack {
    Image(systemName: "hand.wave")
    Spacer()
}
```

→



Activity

Activity

Introduce Yourself

- The app should…
  - share about yourself and one fun fact about you
  - include an image
  - use at least 3 view modifiers

15-minute timer has not started.

Andreas Garcia

”All things Swift”