



Tecnológico  
de Monterrey



# Augmented Reality RealityKit + SwiftUI

**Elvia Rosas**

Software Engineering Teacher

University Level

Apple Distinguished Educator

October 10, 2023



@elviarosasherrera



Erick Padilla



Elvia Rosas

In this session, I will guide you through creating an [AR app](#) using Reality Kit, Reality Composer, and SwiftUI with very simple steps.



@elviarosas

**RealityKit** is a framework that you can use to create Augmented Reality Apps and vision OS Apps.

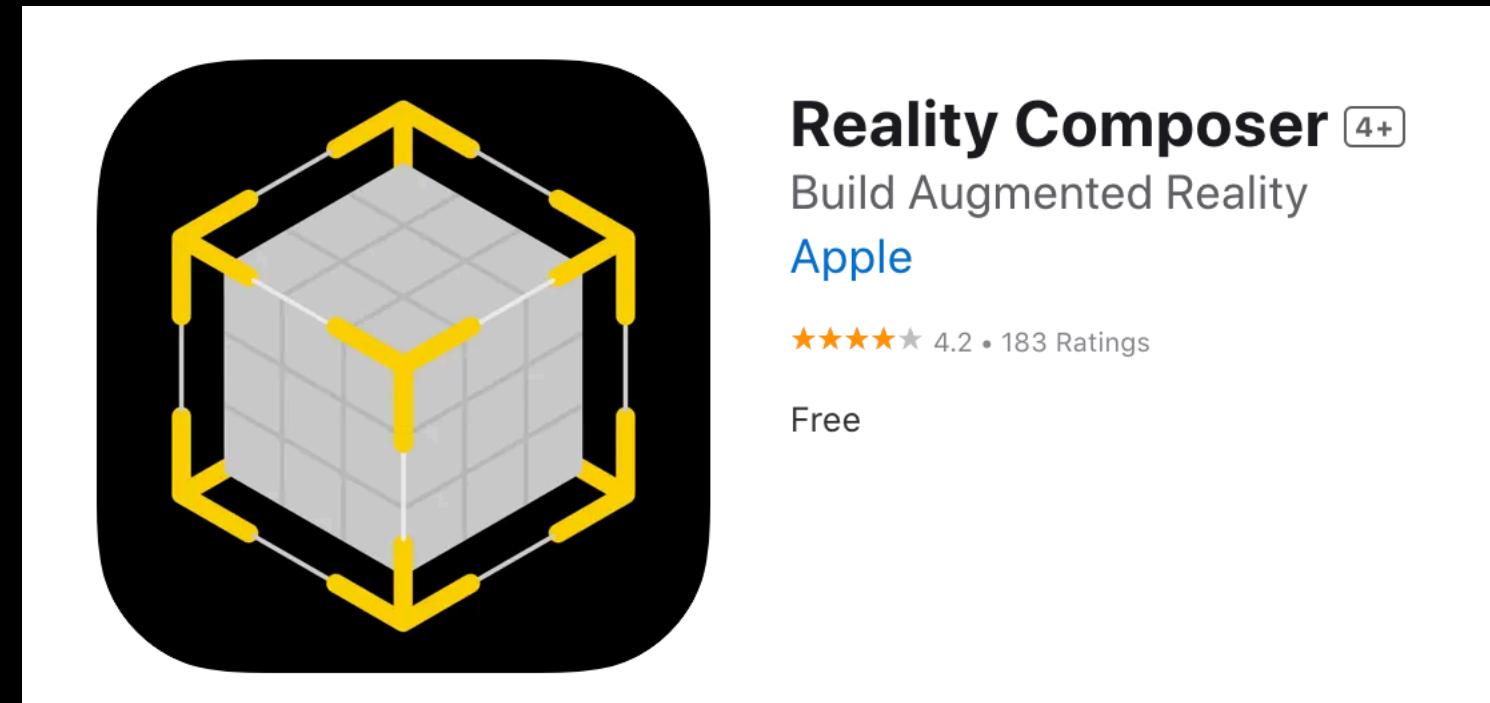
**RealityKit** uses ARKit to integrate virtual objects into the real world.

**RealityKit** makes AR development faster and easier.

<https://developer.apple.com/augmented-reality/realitykit/>

<https://developer.apple.com/documentation/realitykit/>

# Reality Composer

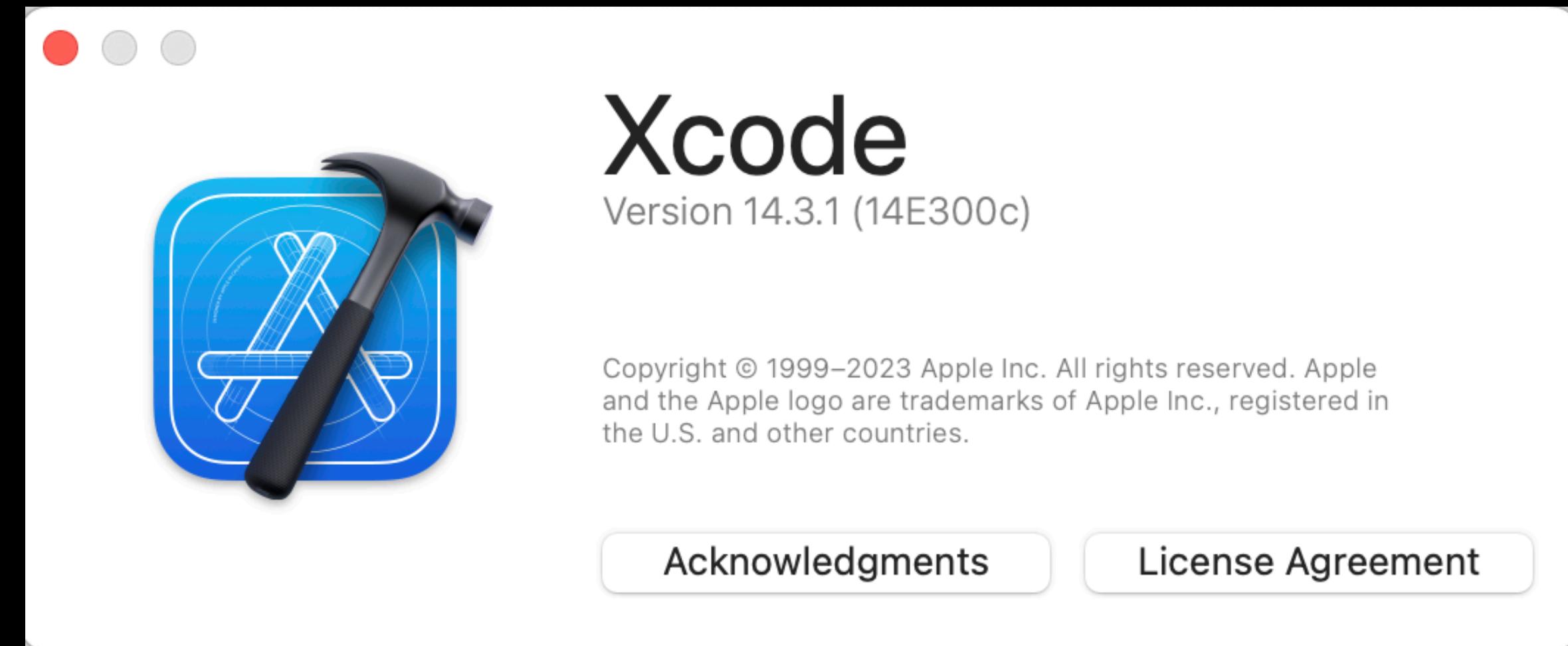


A tool that allows you to create **AR** experiences.

Once the experience is created, you can use it to create Augmented Reality Apps  
with **RealityKit**

<https://developer.apple.com/documentation/realitykit/creating-3d-content-with-reality-composer>

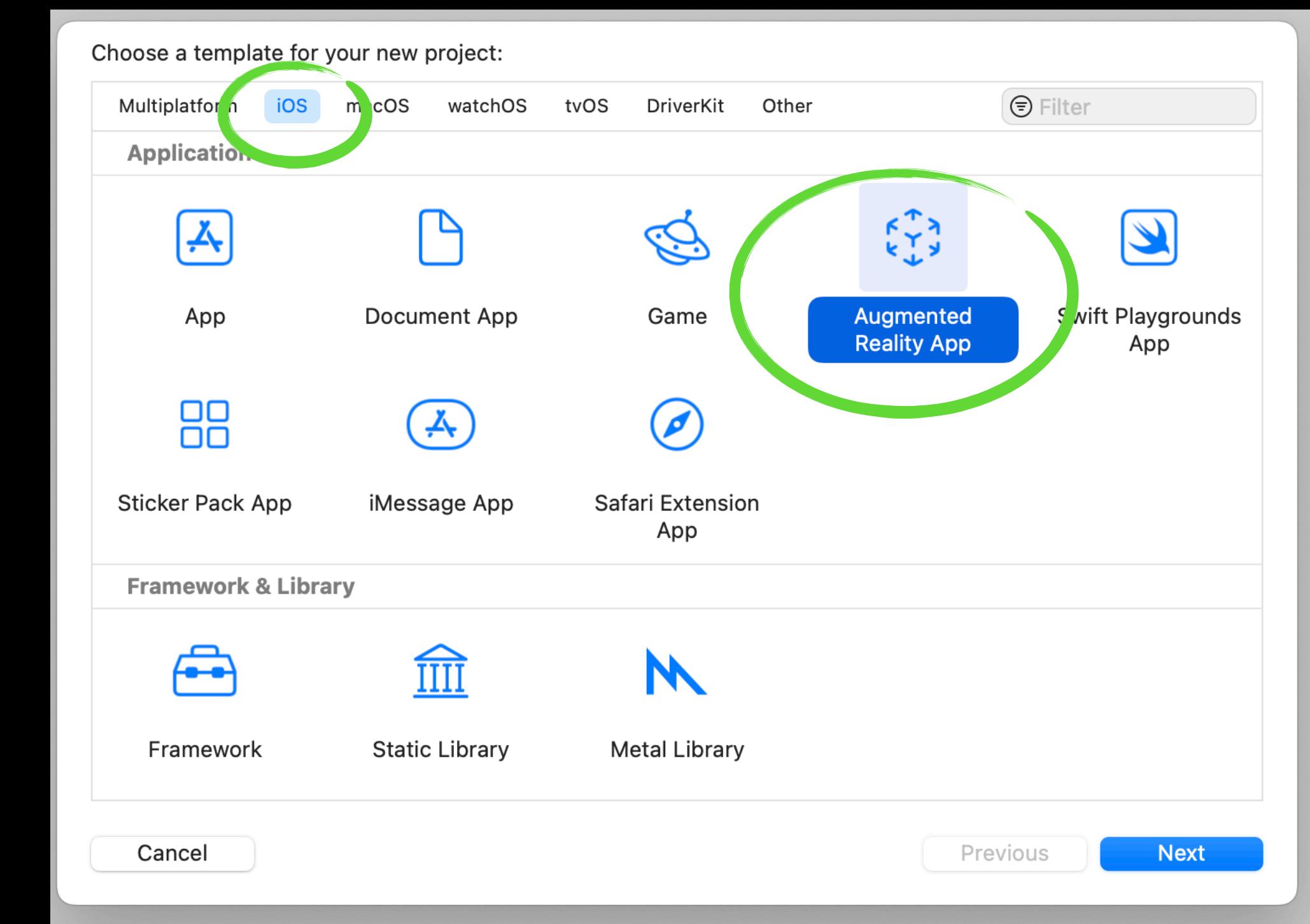
# For this exercise, we will use Xcode version 14.3.1.



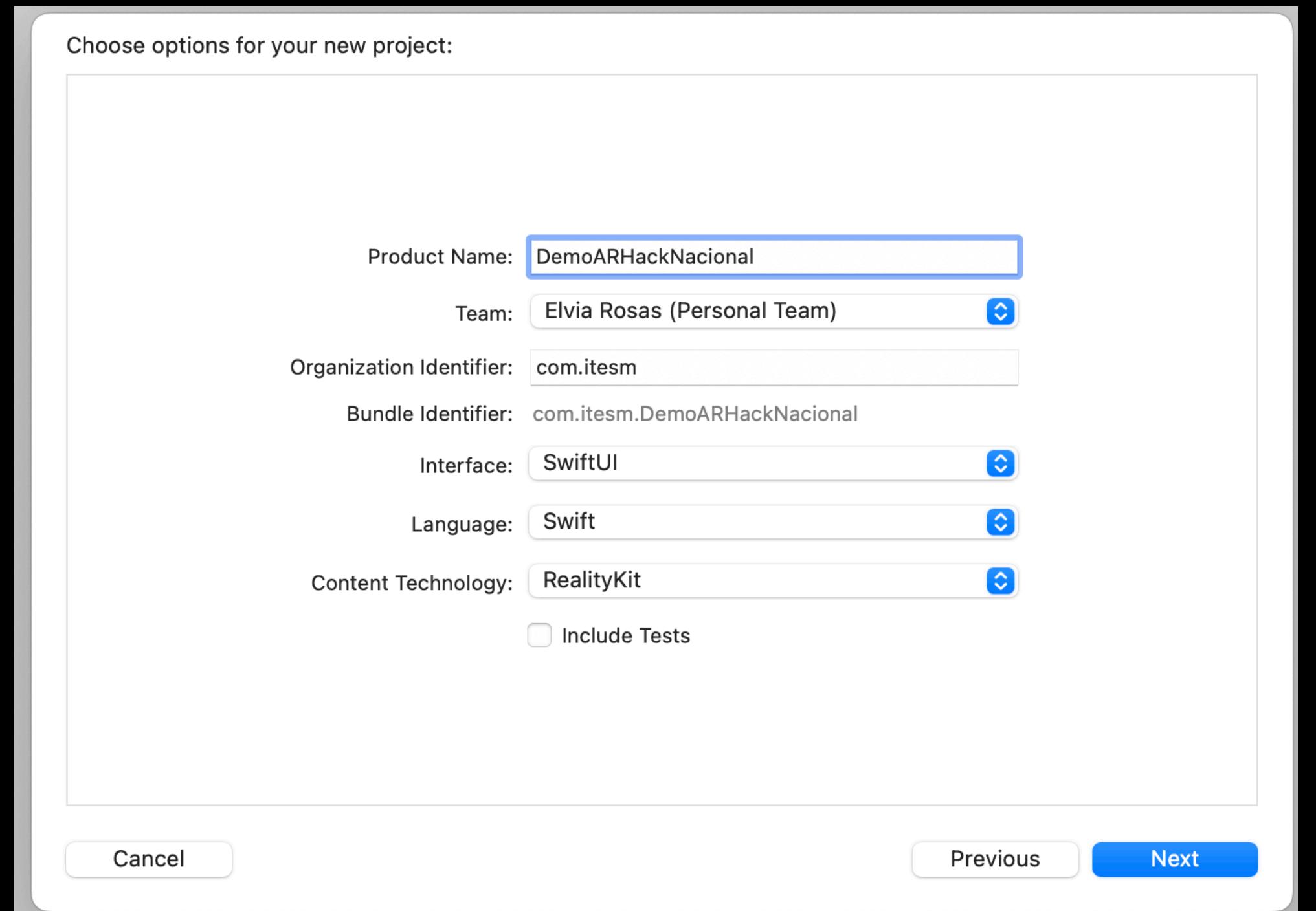
To experience the augmented reality, you'll require an iPhone or iPad.

# Let's start creating an AR Experience

Create a project with Xcode and select Augmented Reality.



# Choose a name, SwiftUI and RealityKit



# Xcode generates the necessary components to display an augmented reality scene in a View.



```
1 //  
2 // ContentView.swift  
3 // DemoARHackNacional  
4 //  
5 // Created by Elvia Rosas on 10/05/23.  
6 //  
7  
8 import SwiftUI  
9 import RealityKit  
10  
11 struct ContentView : View {  
12     var body: some View {  
13         ARViewContainer().edgesIgnoringSafeArea(.all)  
14     }  
15 }  
16  
17 struct ARViewContainer: UIViewRepresentable {  
18  
19     func makeUIView(context: Context) -> ARView {  
20  
21         let arView = ARView(frame: .zero)  
22  
23         // Load the "Box" scene from the "Experience" Reality  
24         // File  
25         let boxAnchor = try! Experience.loadBox()  
26  
27         // Add the box anchor to the scene  
28         arView.scene.anchors.append(boxAnchor)  
29  
30         return arView  
31     }  
32  
33     func updateUIView(_ uiView: ARView, context: Context) {}  
34  
35 }
```

The initial view of SwiftUI.

With a container of a UIKit View

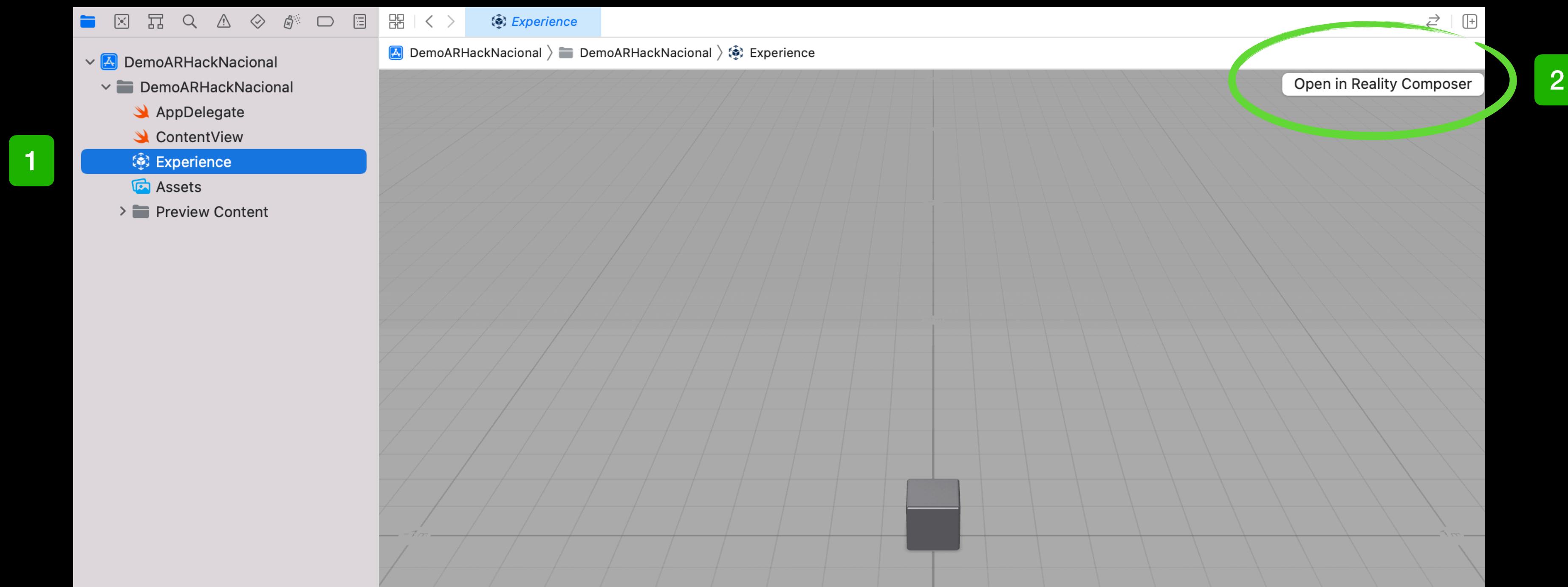
The `UIViewRepresentable` Protocol allows you to use  
A UIKit view in SwiftUI

This protocol requires 2 functions:  
One to create the view and the other to update it

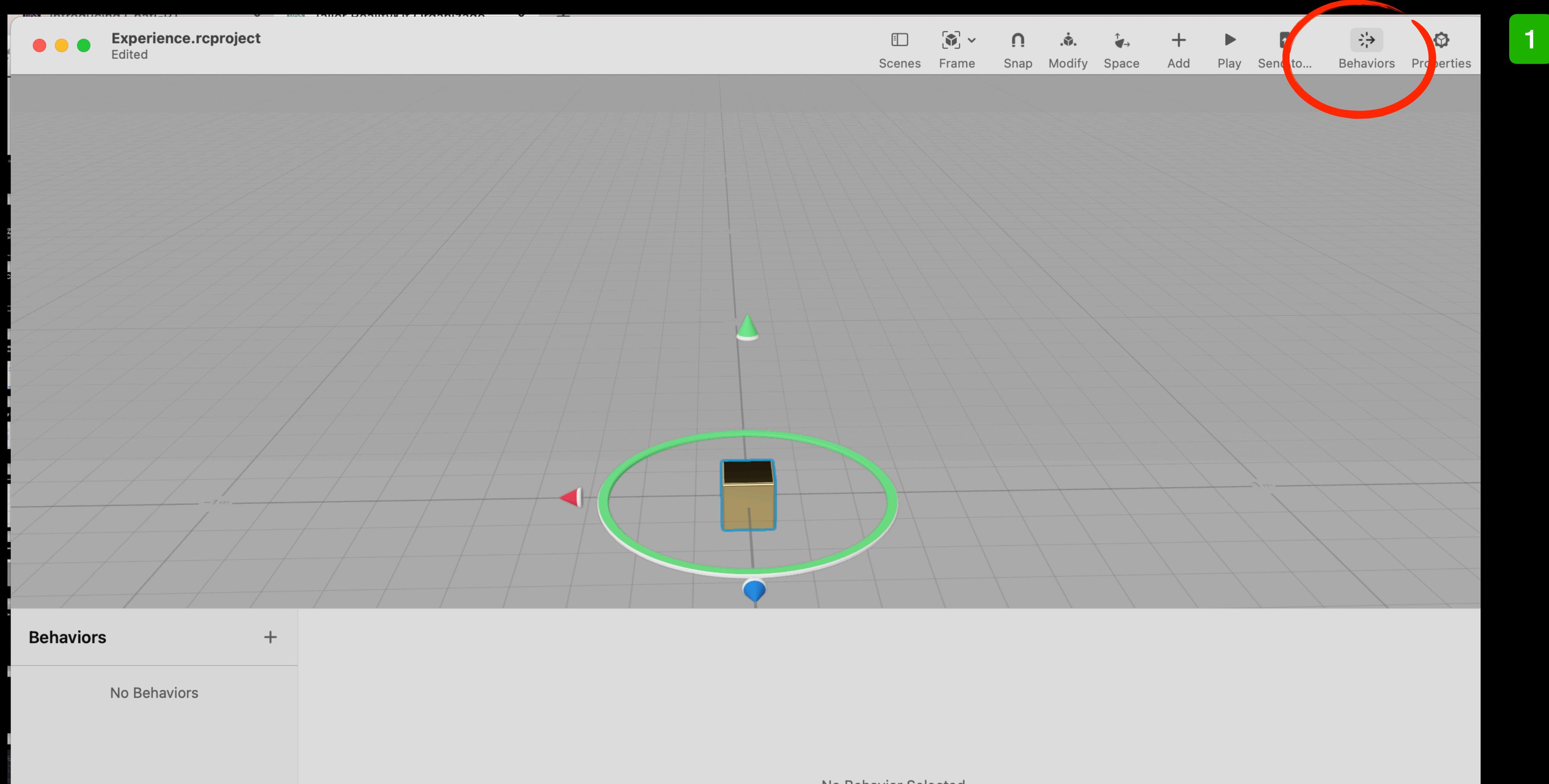
The view we are using is `ARView` from `RealityKit`

- \* A view that enables you to display an AR experience with RealityKit.
- \* It is the view that is superimposed on the camera
- \* `UIView` subclass

# Go to Experience and Open Reality Composer



# You can change the look or add some Behavior

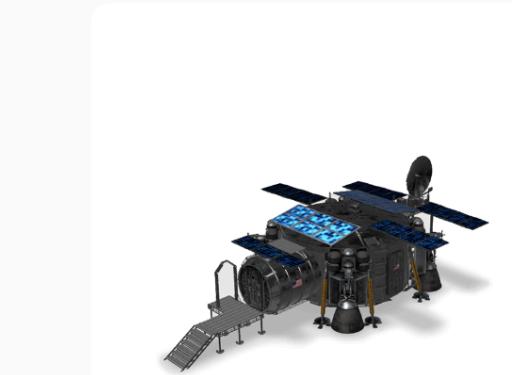


- You can also import existing Apple experiences, models or animations
- <https://developer.apple.com/augmented-reality/quick-look/>

**Augmented Reality**

AR Quick Look supports built-in behaviors and animations in the AR experiences you create.

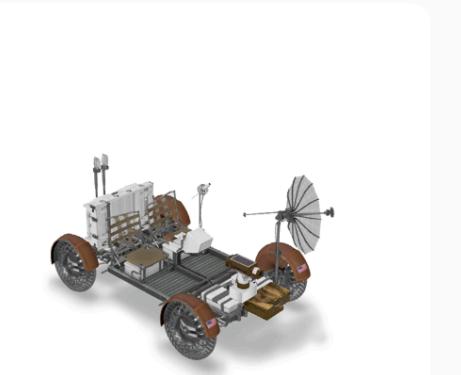
Experience the interactivity of AR Quick Look\* in the For All Mankind: Explorable Objects examples below.



The Hab



Cosmonaut



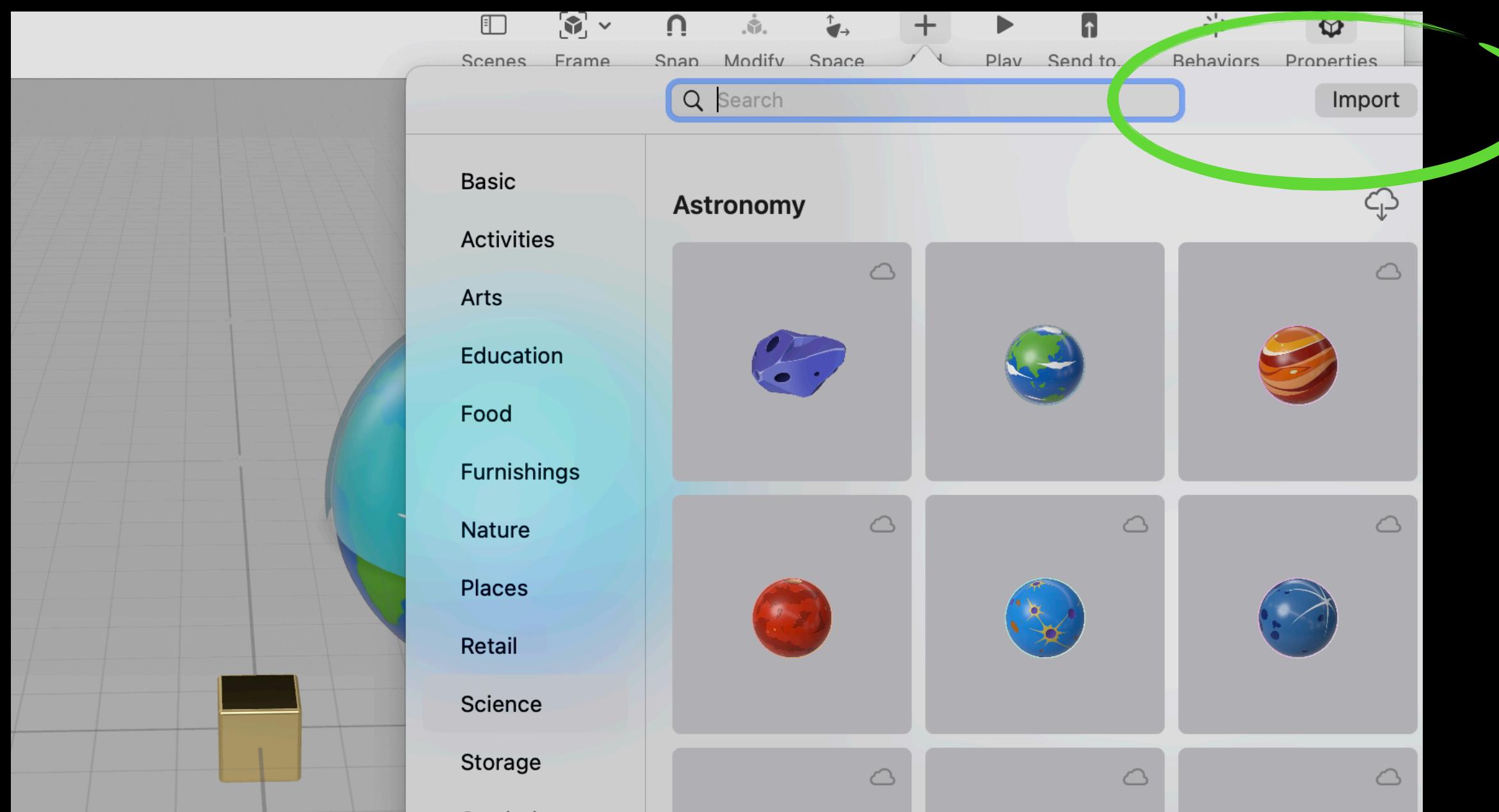
Lunar Rover

---

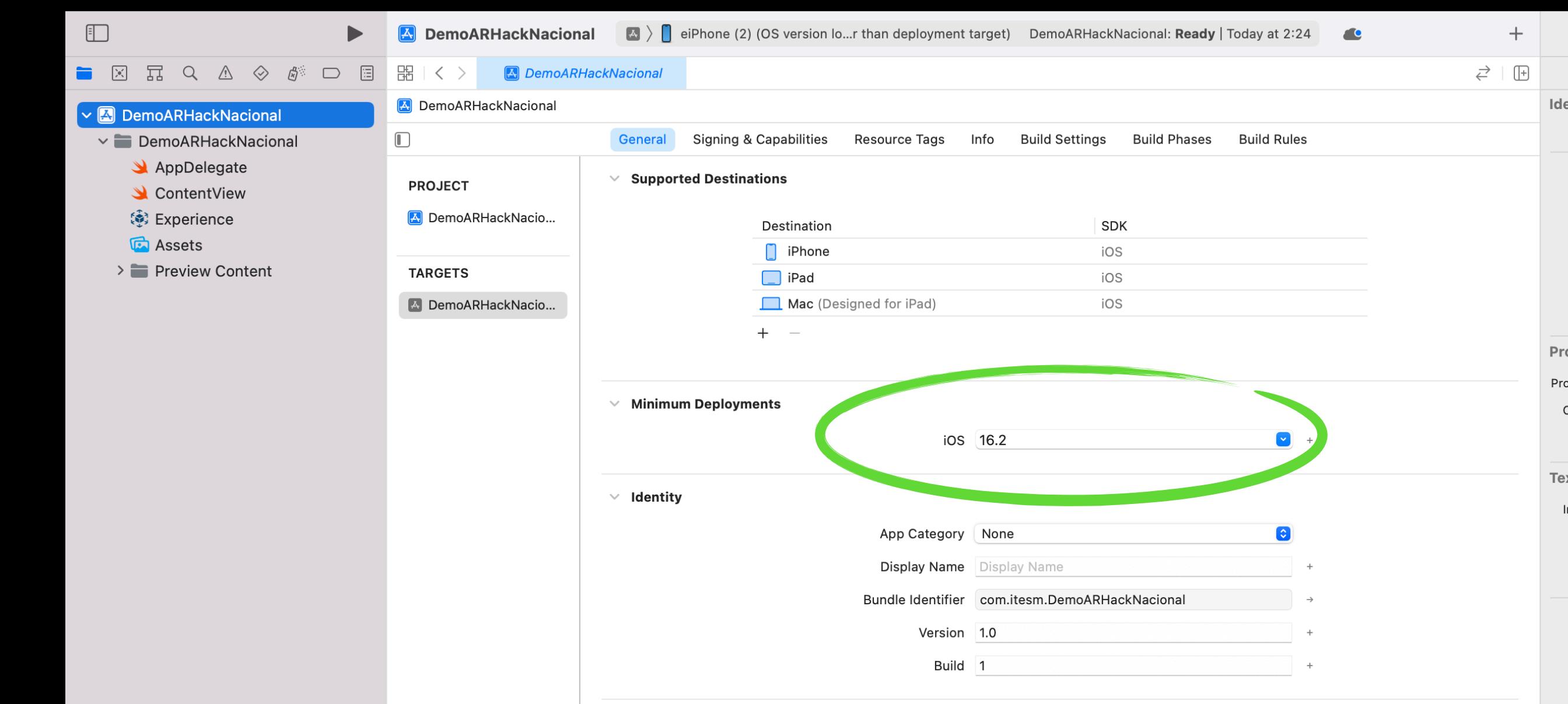
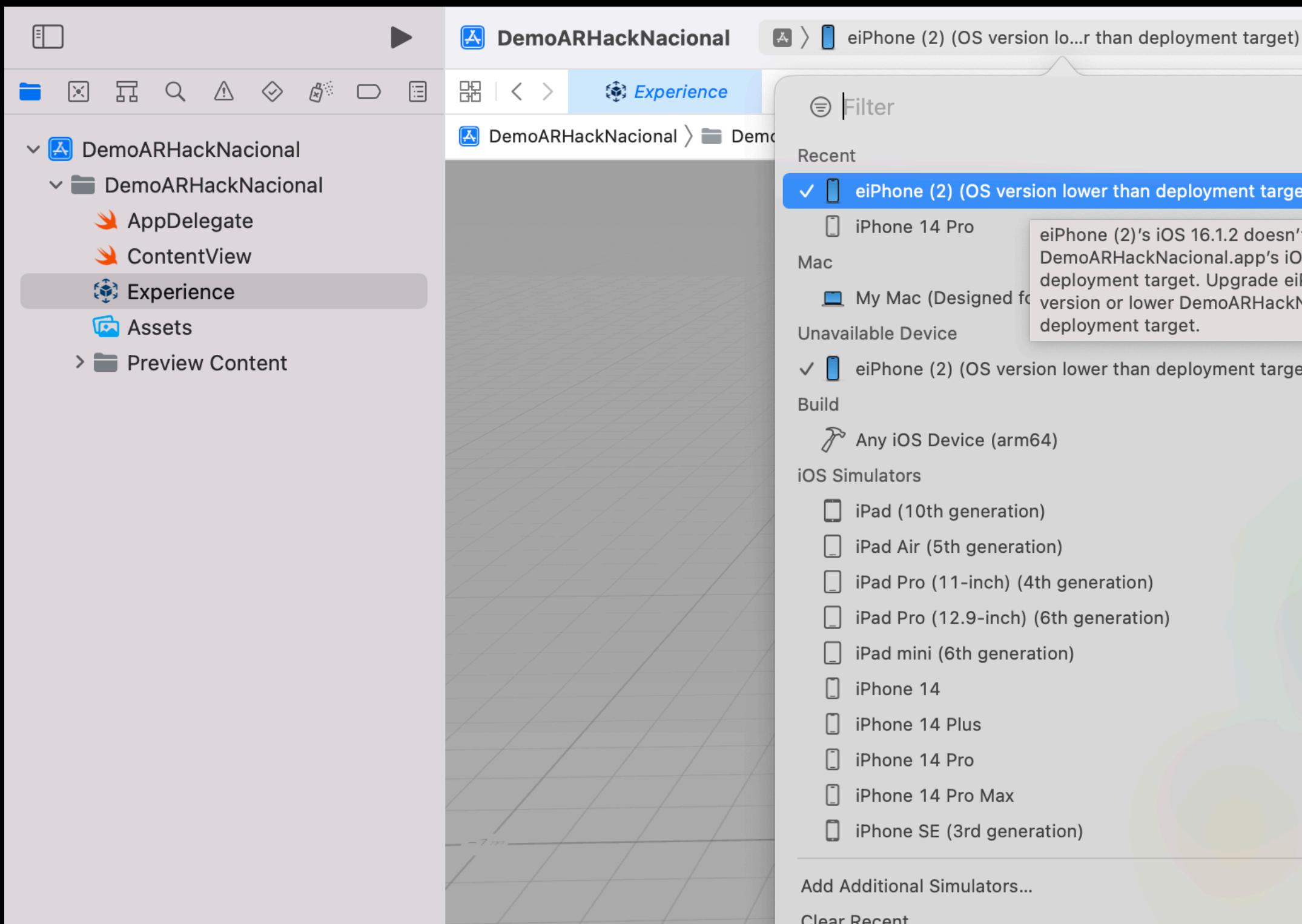
**3D models**

Tap any of the 3D models below on a iPhone or iPad to view the object and place it in AR.\*  
Or click a model on Mac to download the USDZ file.

- Import the .reality file
- Or .usdz models



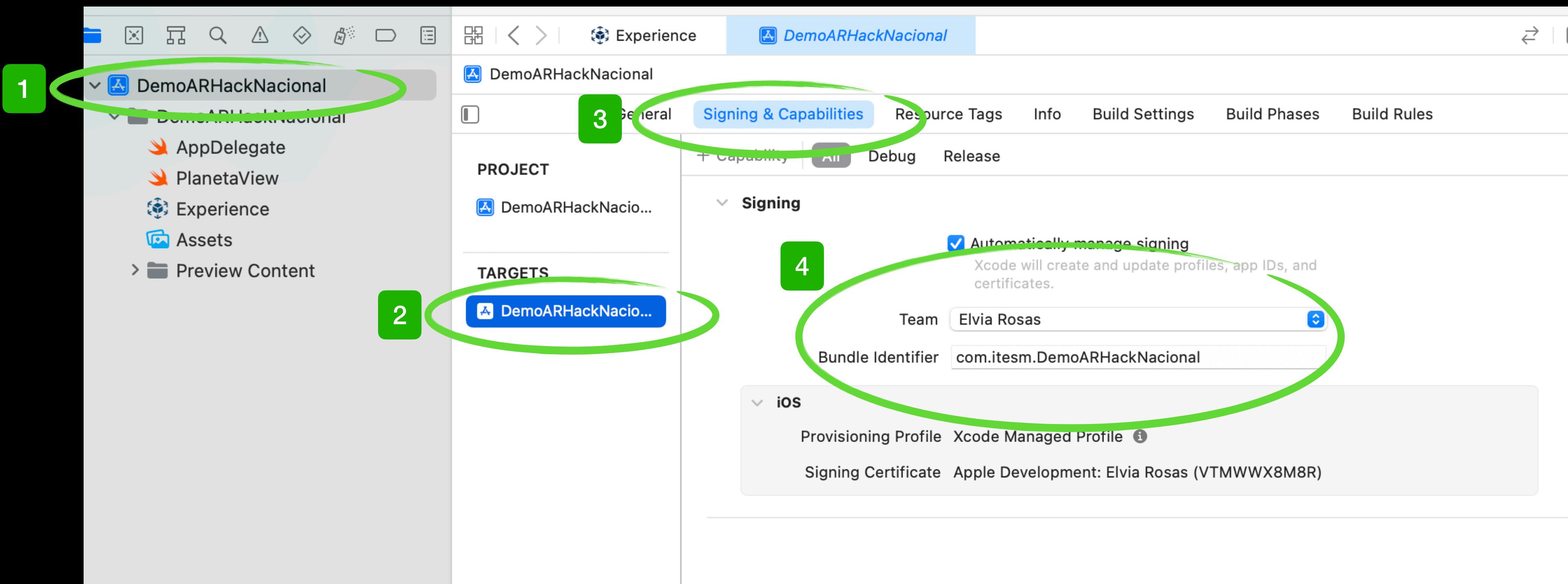
# Try your AR Experience on your iPad/iPhone



First, Ensure that the minimum deployment version in General Settings is the same or lower than that of your iPad/iPhone.

# ... Try your AR Experience on your iPad/iPhone

Next, Make sure you have selected a Team in Signing & Capabilities section

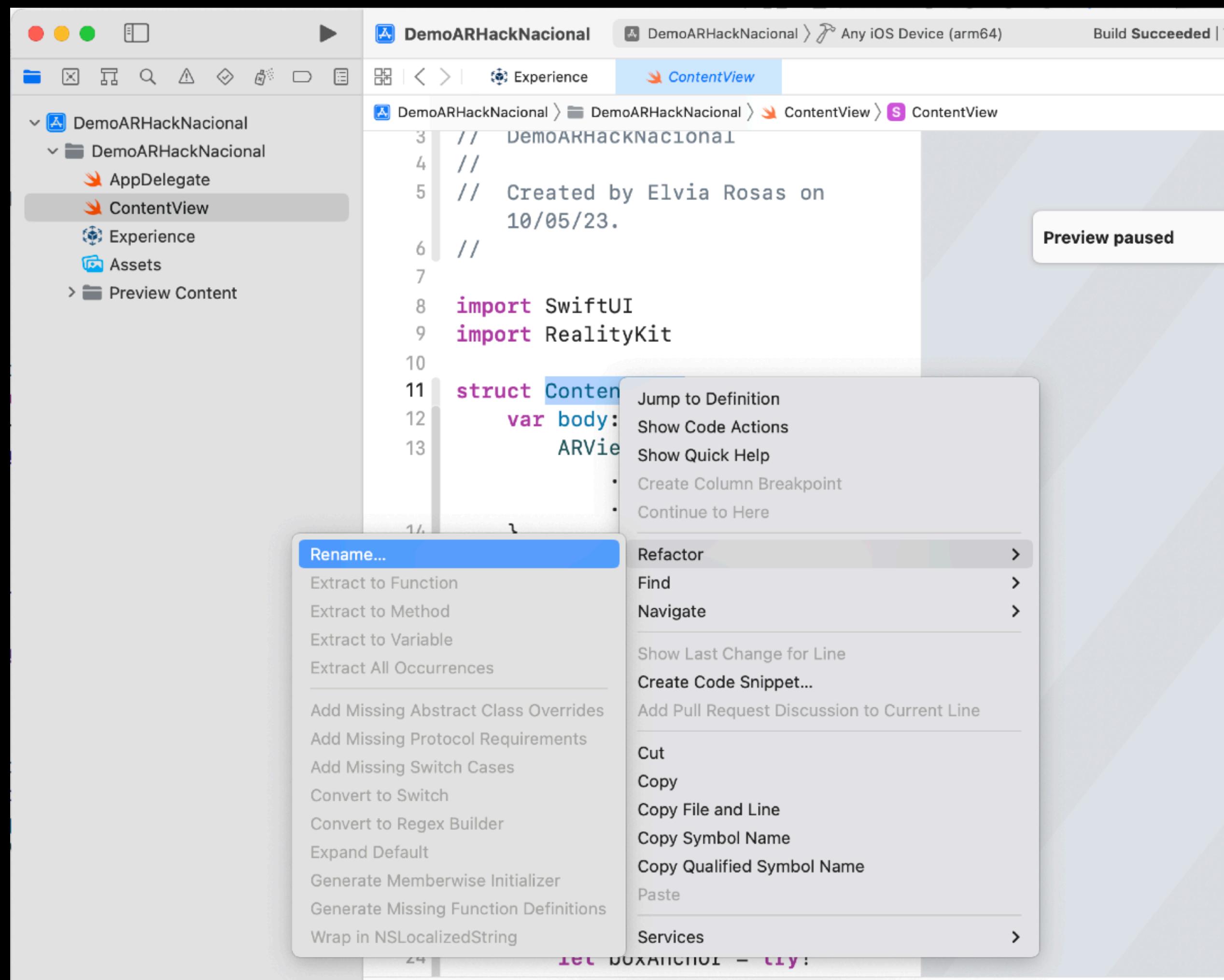


# On your iPhone/iPad

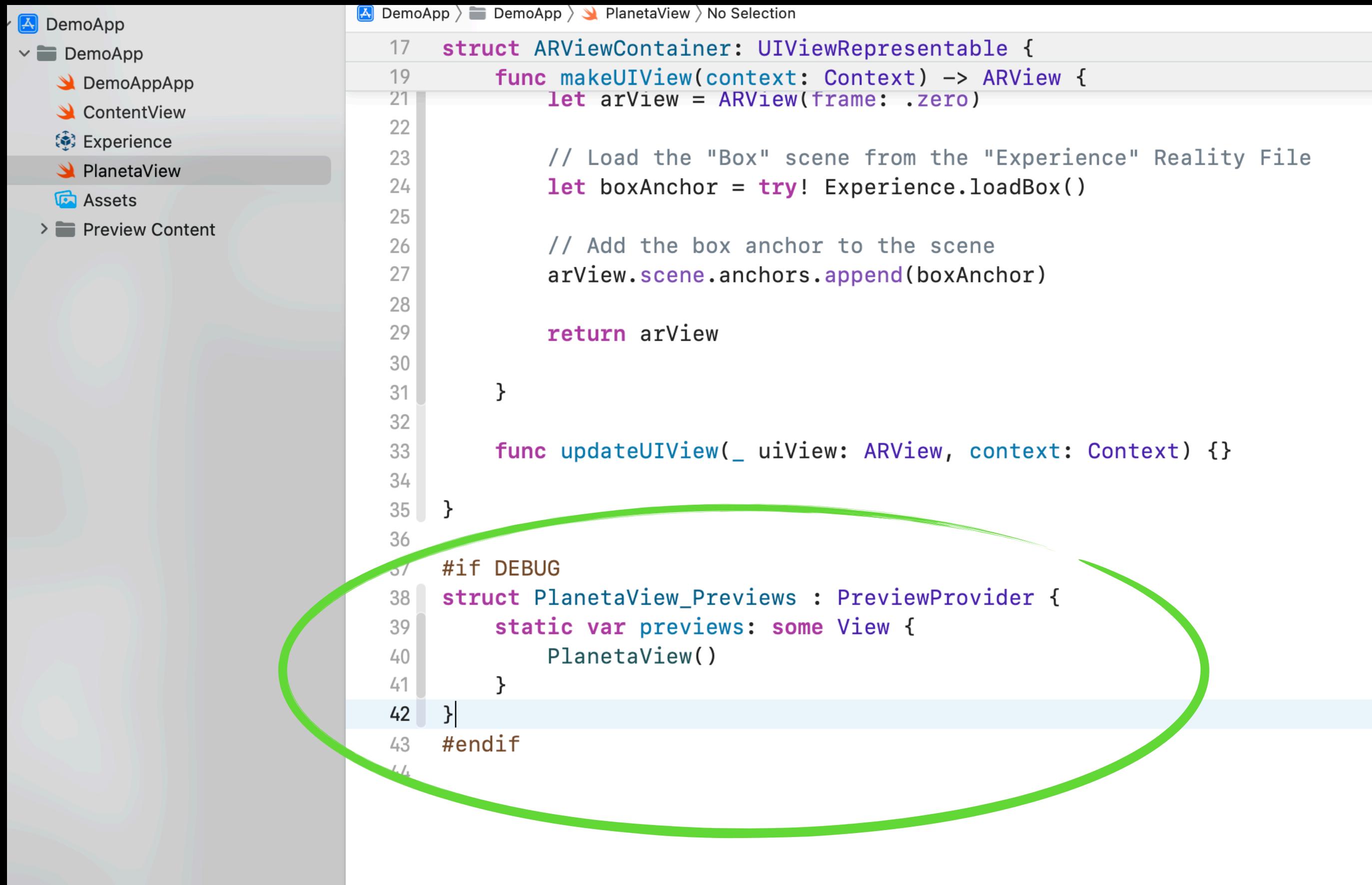
- Go to Settings
- Check that your iPad/iPhone has a registered Apple account
- Enable Developer Mode on your iPad/iPhone
  - Settings -> Privacy & Security
    - Developer Mode ON
  - Ensure to trust in your Developer Team
  - Settings -> General -> VPN & Device Management

**Export the experience to a SwiftUI App**

# First, rename the name of the View using Refactor



# For Preview, manually rename to PlanetaView\_Previews



The screenshot shows the Xcode interface with a project named "DemoApp". The left sidebar shows files like DemoAppApp, ContentView, Experience, PlanetaView, and Assets. The main editor window displays Swift code for an ARViewContainer:

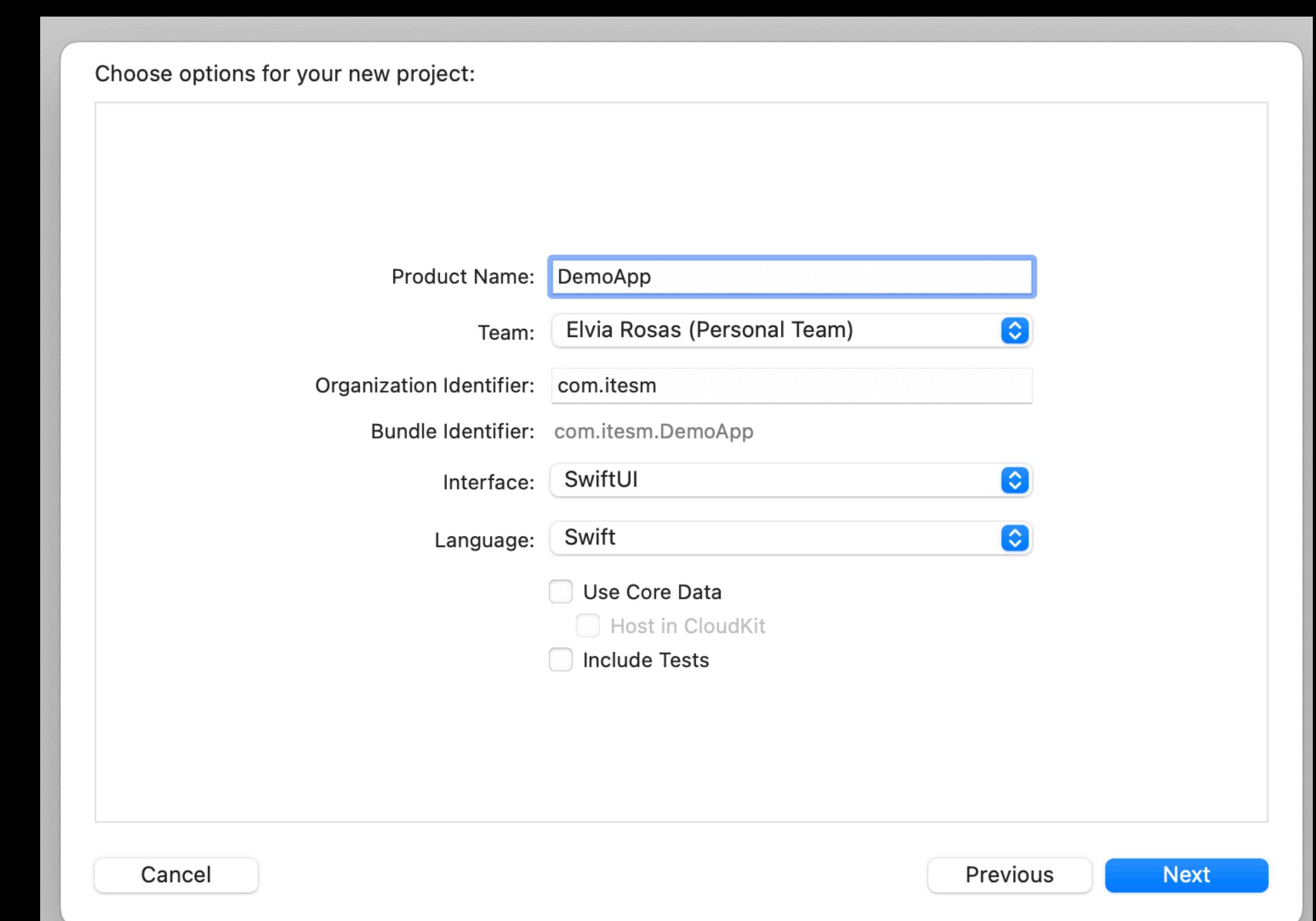
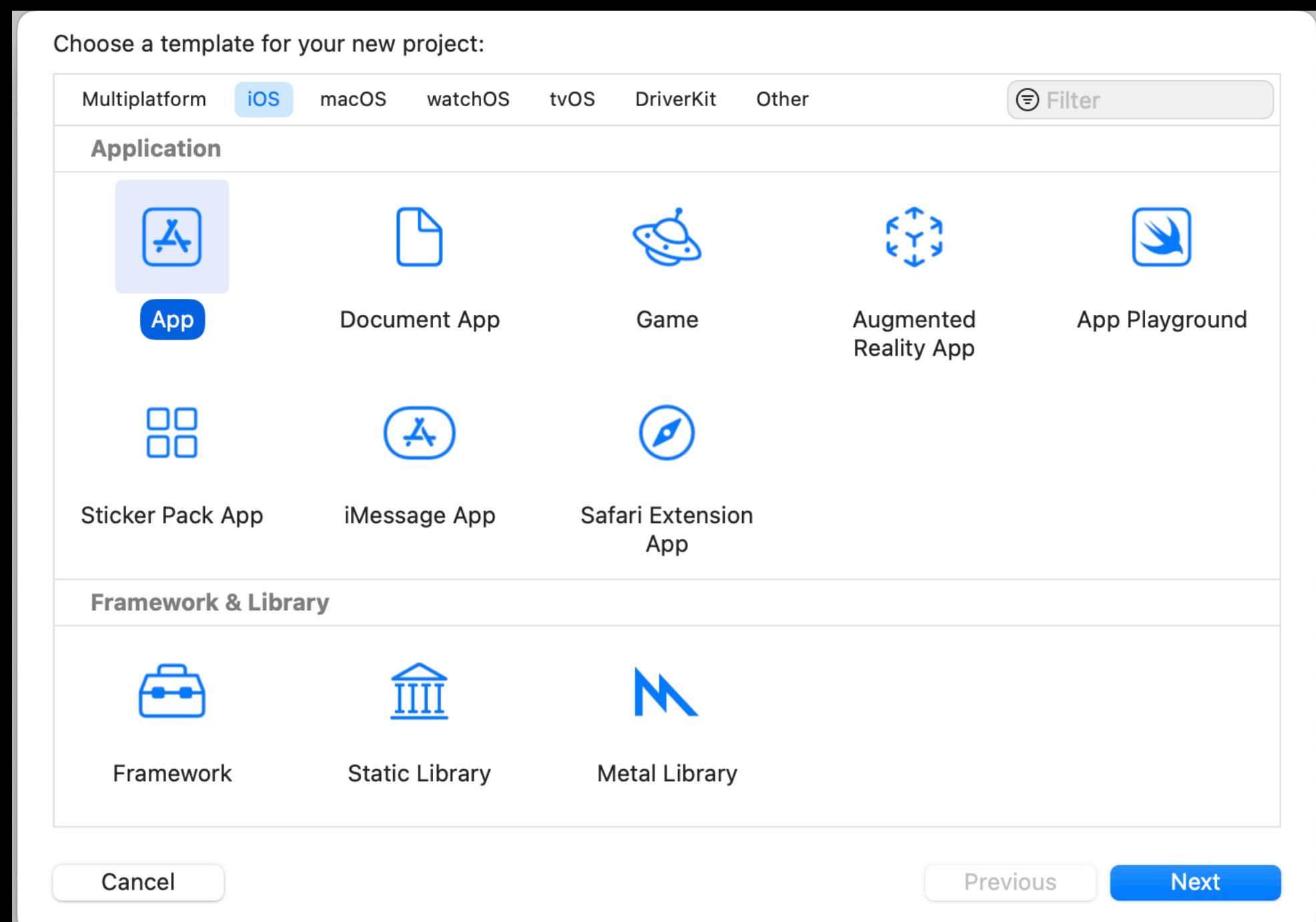
```
17 struct ARViewContainer: UIViewRepresentable {
18     func makeUIView(context: Context) -> ARView {
19         let arView = ARView(frame: .zero)
20
21         // Load the "Box" scene from the "Experience" Reality File
22         let boxAnchor = try! Experience.loadBox()
23
24         // Add the box anchor to the scene
25         arView.scene.anchors.append(boxAnchor)
26
27         return arView
28     }
29
30 }
31
32
33 func updateUIView(_ uiView: ARView, context: Context) {}
34
35 }
36
37 #if DEBUG
38 struct PlanetaView_Previews : PreviewProvider {
39     static var previews: some View {
40         PlanetaView()
41     }
42 }
43 #endif
44
```

A green oval highlights the line "#if DEBUG" and the entire definition of the `PlanetaView_Previews` struct.

## Steps:

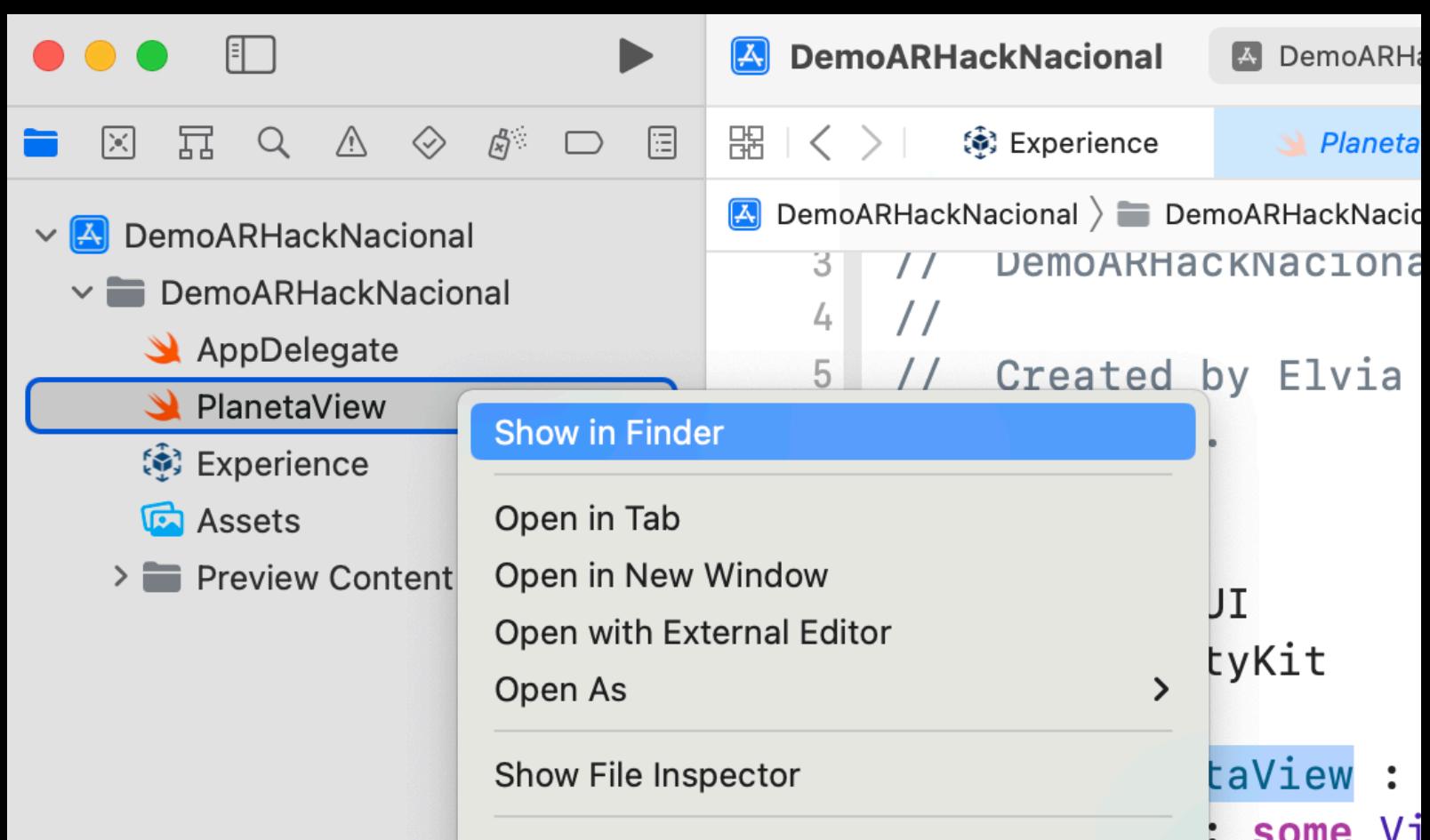
1. Create a SwiftUI Project
2. Export the AR Experiences files to your SwiftUI project
3. Add settings for the camera

# 1. Create a SwiftUI Project

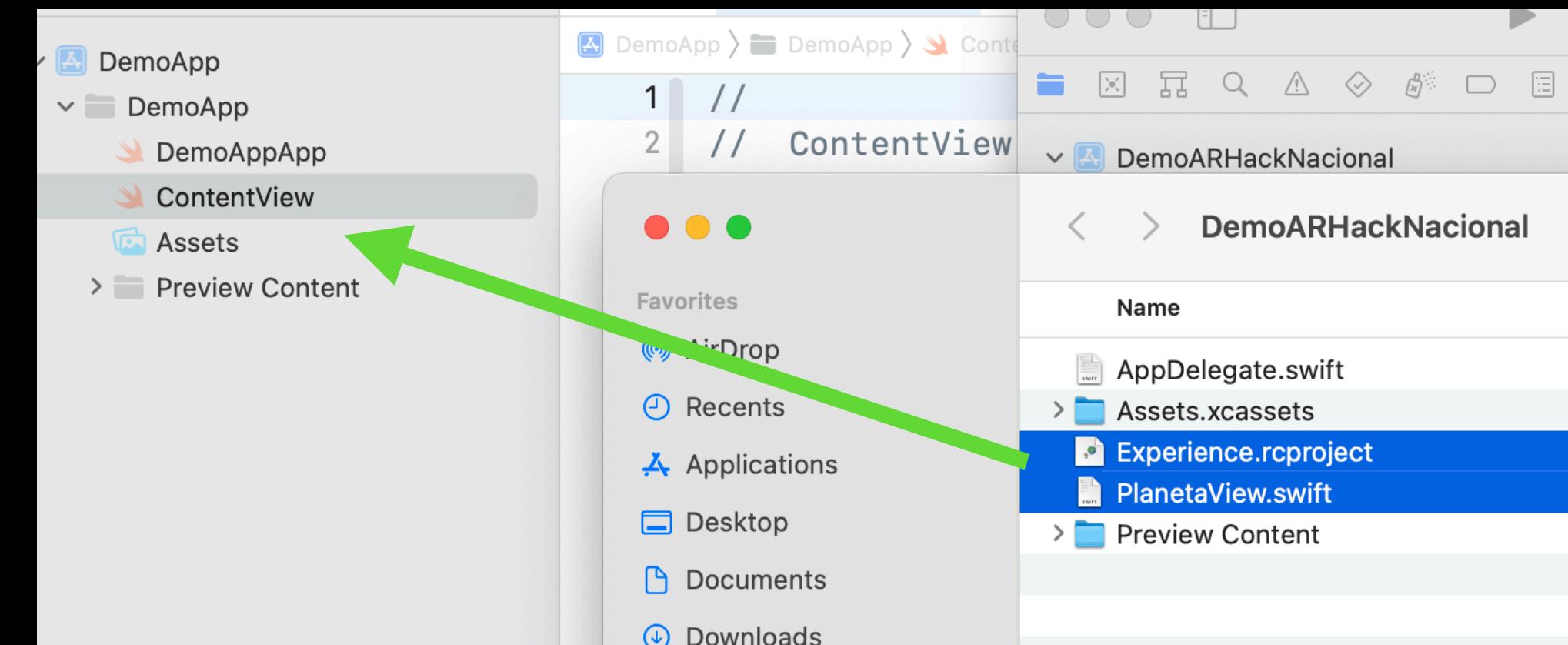


## 2. Find the two files in Finder to bring them to the new Project

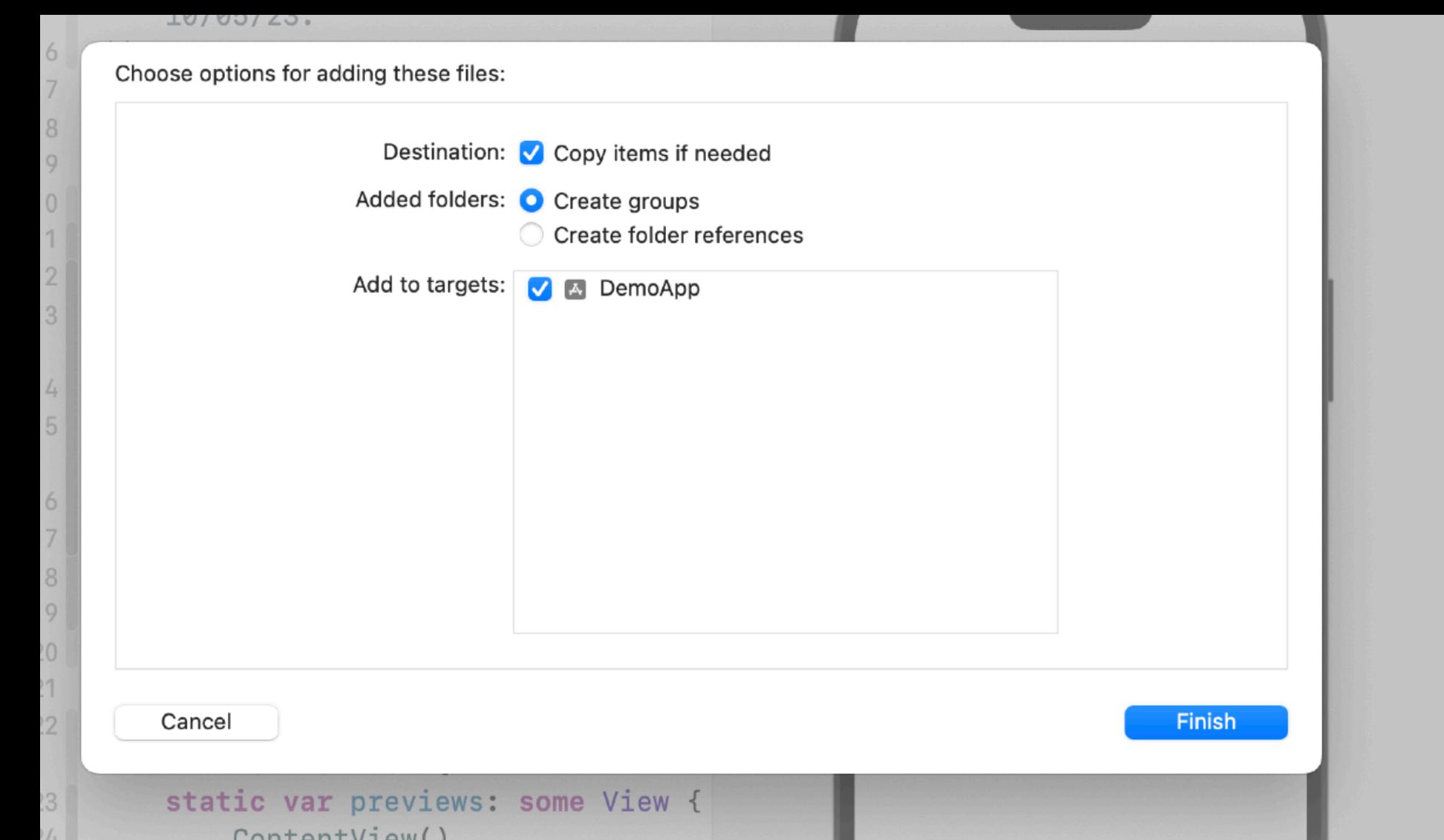
1



2



3

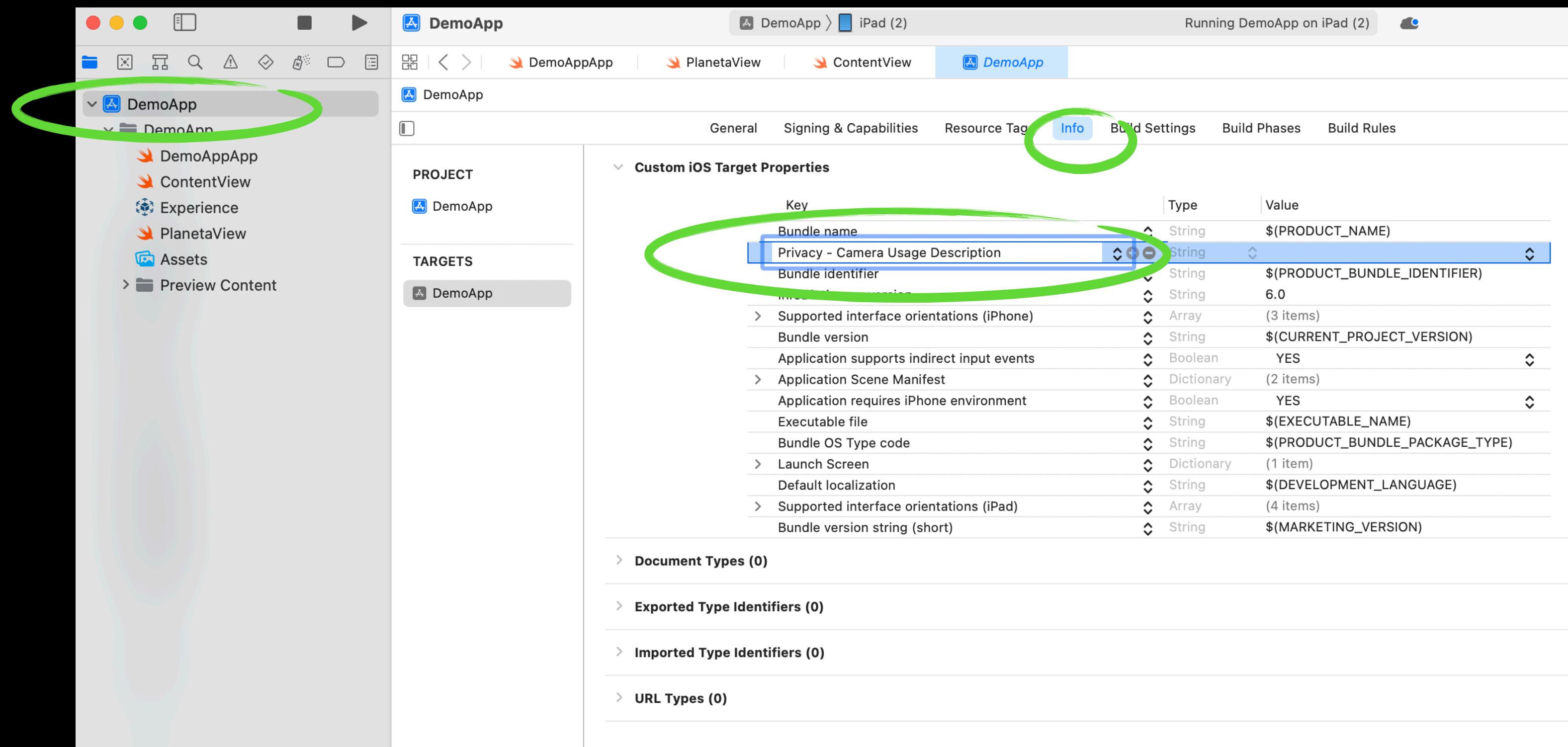


### 3.Tell the user that you are going to use the camara

Add a key in info.plist

Privacy - Camera Usage Description

It is necessary to access your camera to see the AR experience



**Ready! Run your App on your iPad/iPhone**

**Now, it's time for you to explore Reality  
Composer and RealityKit capabilities and  
build great apps.**

<https://developer.apple.com/documentation/realitykit/>

<https://developer.apple.com/augmented-reality/realitykit/>

# RealityKit vs ARKit

Tips from Erick

# General Rule

How do I know if I'm using RealityKit?

- Avoid using tutorials with  
 **import ARKit**
- Avoid using delegates for our views, the correct thing is to subscribe to events in our entities or scenes
-  **ARSessionDelegate**



**arView.scene.subscribe**

```
// RealityKit ✓  
arView.scene  
    .subscribe(  
        to: SceneEvents.DidActivateEntity.self,  
        on: planeEntity  
    ) { event in  
        code  
    }  
.store(in: &cancellables)
```

```
// ARKit ✗  
func session(_ session: ARSession,  
            didAdd anchors: [ARAnchor])  
{  
    code  
}
```



Accessibility



ShazamKit



Create ML



# What's Next?



<https://developer.apple.com/wwdc23/>

# Reality Composer Pro

Meet Reality Composer Pro

<https://developer.apple.com/videos/play/wwdc2023/10083/>

Enhance your spatial computing app with RealityKit

<https://developer.apple.com/videos/play/wwdc2023/10081/>

# Spatial Computing

Build spatial experiences with RealityKit

<https://developer.apple.com/videos/play/wwdc2023/10080/>

Create accessible spatial experiences

<https://developer.apple.com/videos/play/wwdc2023/10034/>

Get started with building apps for spatial computing

<https://developer.apple.com/videos/play/wwdc2023/10260/>

# SwiftUI

Meet SwiftUI for spatial computing

<https://developer.apple.com/videos/play/wwdc2023/10109/>

What's new in SwiftUI

<https://developer.apple.com/videos/play/wwdc2023/10148/>

Wind your way through advanced animations in SwiftUI

<https://developer.apple.com/videos/play/wwdc2023/10157/>

Build accessible apps with SwiftUI and UIKit

<https://developer.apple.com/videos/play/wwdc2023/10036/>

Design with SwiftUI

<https://developer.apple.com/videos/play/wwdc2023/10115/>

# Thank you



@elviarosasherrera



@elviarosas



# Bibliography

Thanks to Erick Padilla (student) for collaborationg on this course.  
Thanks to Joaquín Ramírez for the Augemented Reality Couse.

<https://www.apple.com/augmented-reality/>

<https://developer.apple.com/augmented-reality/>

<https://developer.apple.com/augmented-reality/quick-look/>