

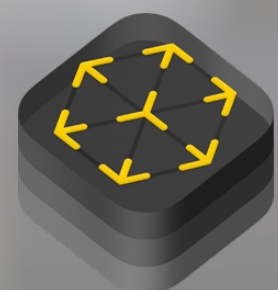
AR是.....

So the AR is?

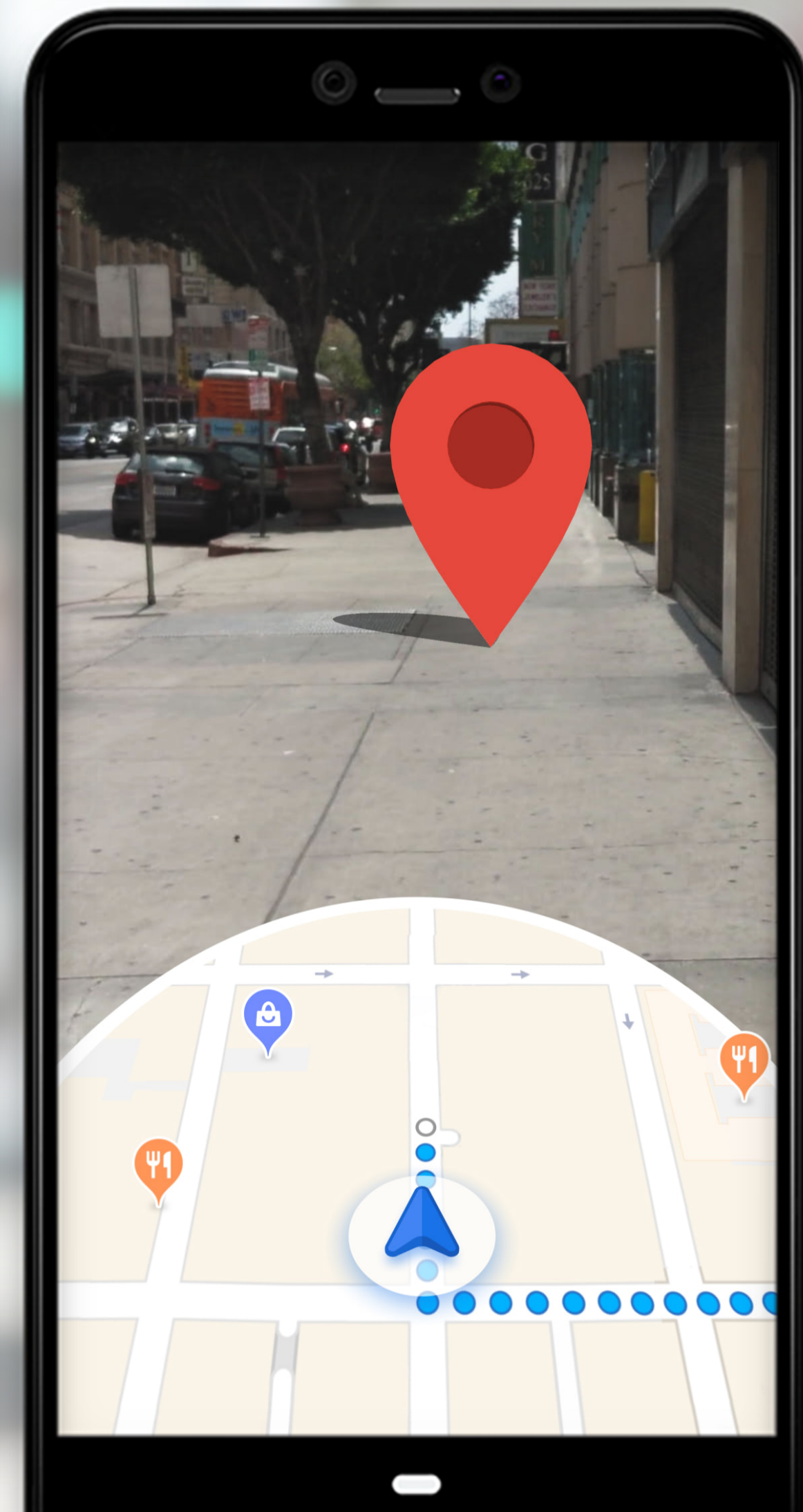


浙江大学
ZHEJIANG UNIVERSITY

- 增强现实 (Augmented Reality)
是将虚拟的信息叠加到现实世界中，实现虚拟场景与现实世界巧妙融合的效果。



iOS移动平台AR基础



1

iOS平台AR开发基础——基于Xcode 与 Playground

回顾之前的SwiftUI中如何使用UIKit视图框架，介绍两种开发模式

AR开发设备支持

Dev env requirement

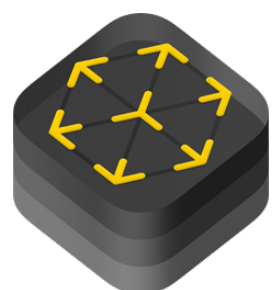


浙江大学
ZHEJIANG UNIVERSITY

ARKit对开发平台、设备的要求

iOS 11 及以上

A9芯片及以上



iOS移动平台AR基础

AR开发设备支持

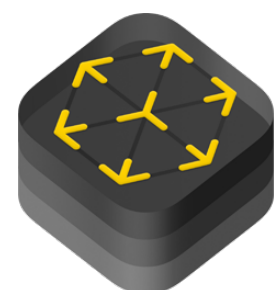
Dev env requirement



浙江大学
ZHEJIANG UNIVERSITY

RealityKit对开发的要求

iOS 13及以上



iOS移动平台AR基础

AR开发设备支持 —— ARKit & RealityKit

Dev env requirement

iPhone 6s / plus	iPhone 11	iPad mini 4 and after
iPhone 7 / Plus	iPhone 11 Pro	iPad Air serial
iPhone 8 / 8 Plus	iPhone 11 Pro Max	iPad serial
iPhone X	iPhone 12 mini	iPad Pro 10.5
iPhone XR	iPhone 12	iPad Pro 12.9
iPhone Xs / Max	iPhone 13 mini	iPad Pro 11
iPhone se serial	iPhone 13	iPod touch 7th



iOS移动平台AR基础



Device with LiDAR

iPhone 12 Pro
iPhone 12 Pro Max
iPhone 13 Pro
iPhone 13 Pro Max
iPhone 14 Pro
iPhone 14 Pro Max
iPad Pro 11 with M1
iPad Pro 12.9 with M1

LiDAR是什么

What the heck is LiDAR?

LiDAR Scanner



iOS移动平台AR基础

LiDAR是什么

What the heck is LiDAR?

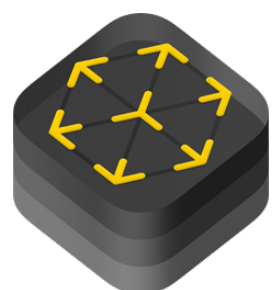


浙江大学
ZHEJIANG UNIVERSITY



2D image

Depth map

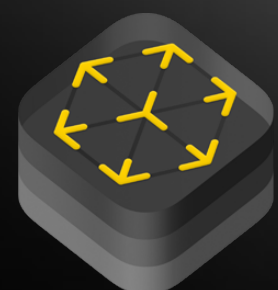
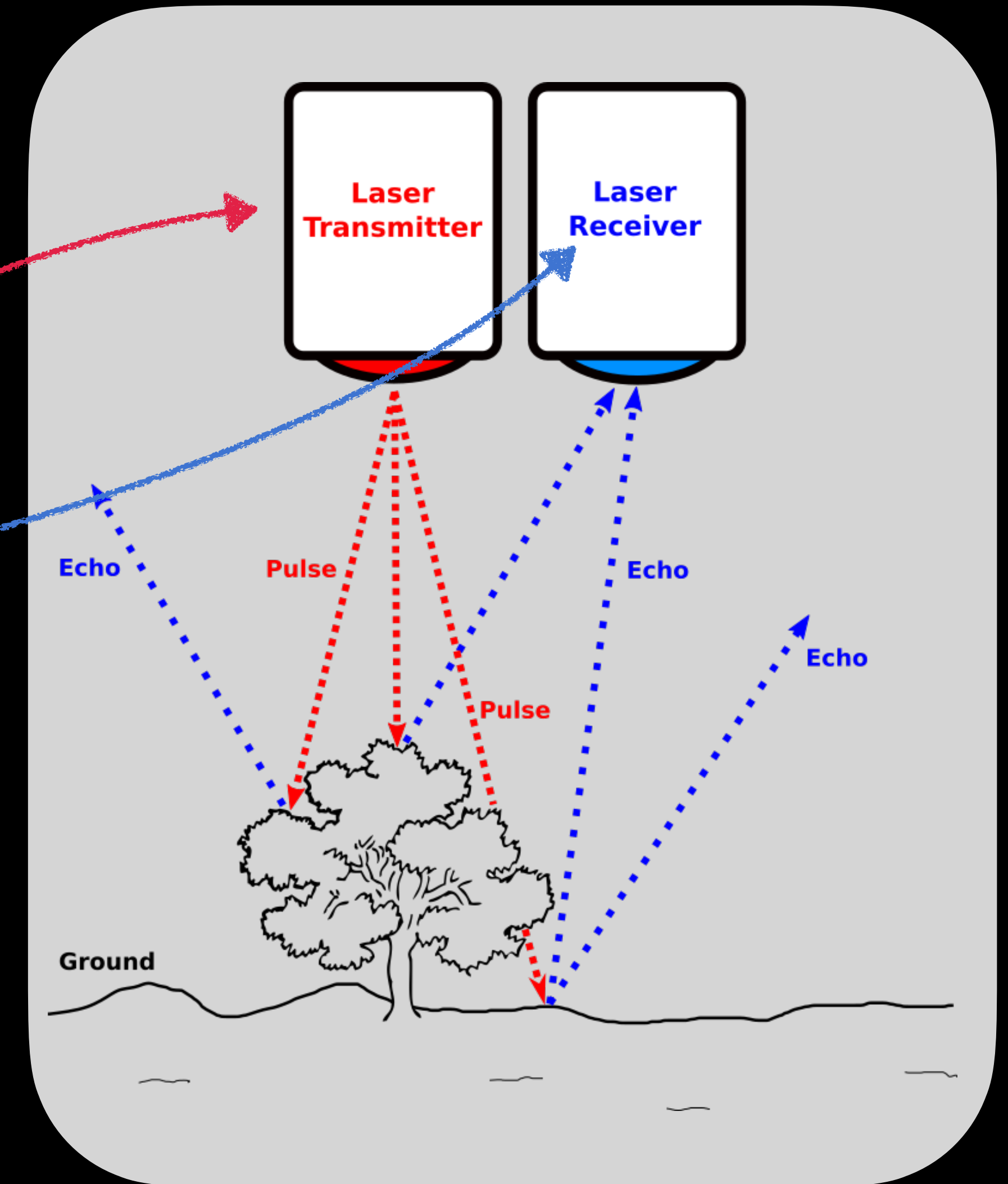
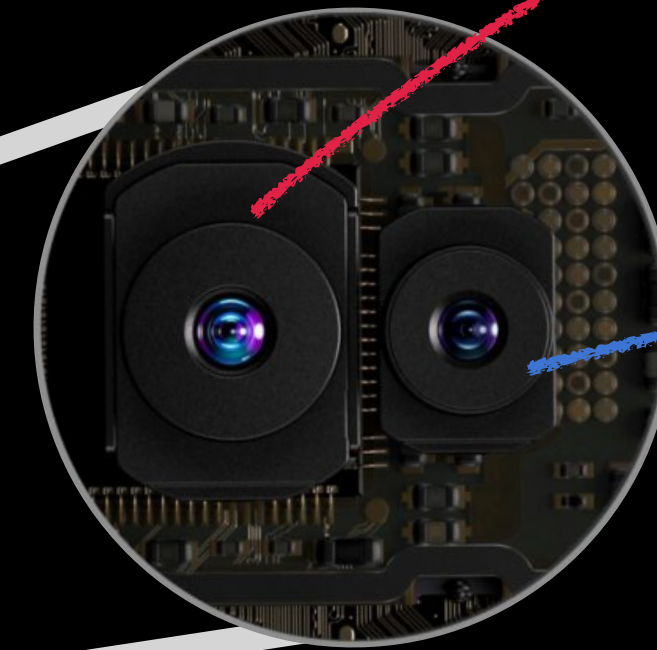


LiDAR是什么

What the heck is LiDAR?



浙江大学
ZHEJIANG UNIVERSITY



ARKit与RealityKit

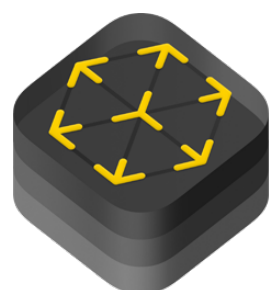
ARkit and RealityKit



浙江大学
ZHEJIANG UNIVERSITY

🤔大家知道 **ARKit** 与 **RealityKit** 之间是什么关系吗?

(除了刚提及的iOS版本区别)



iOS移动平台AR基础

ARKit与RealityKit

ARkit and RealityKit



浙江大学
ZHEJIANG UNIVERSITY

SceneKit

iOS 8 +

Swift & Objective-C

基于OpenGL

不能跨平台

ARKit

iOS 9 及 A9芯片 +

Swift & Objective-C

绘制引擎

RealityKit

iOS 13 +

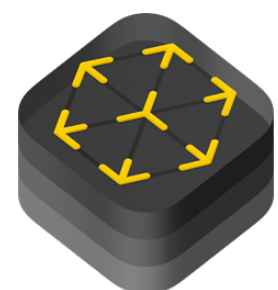
Swift

基于Metal

基于物理的真实感渲染

可跨平台

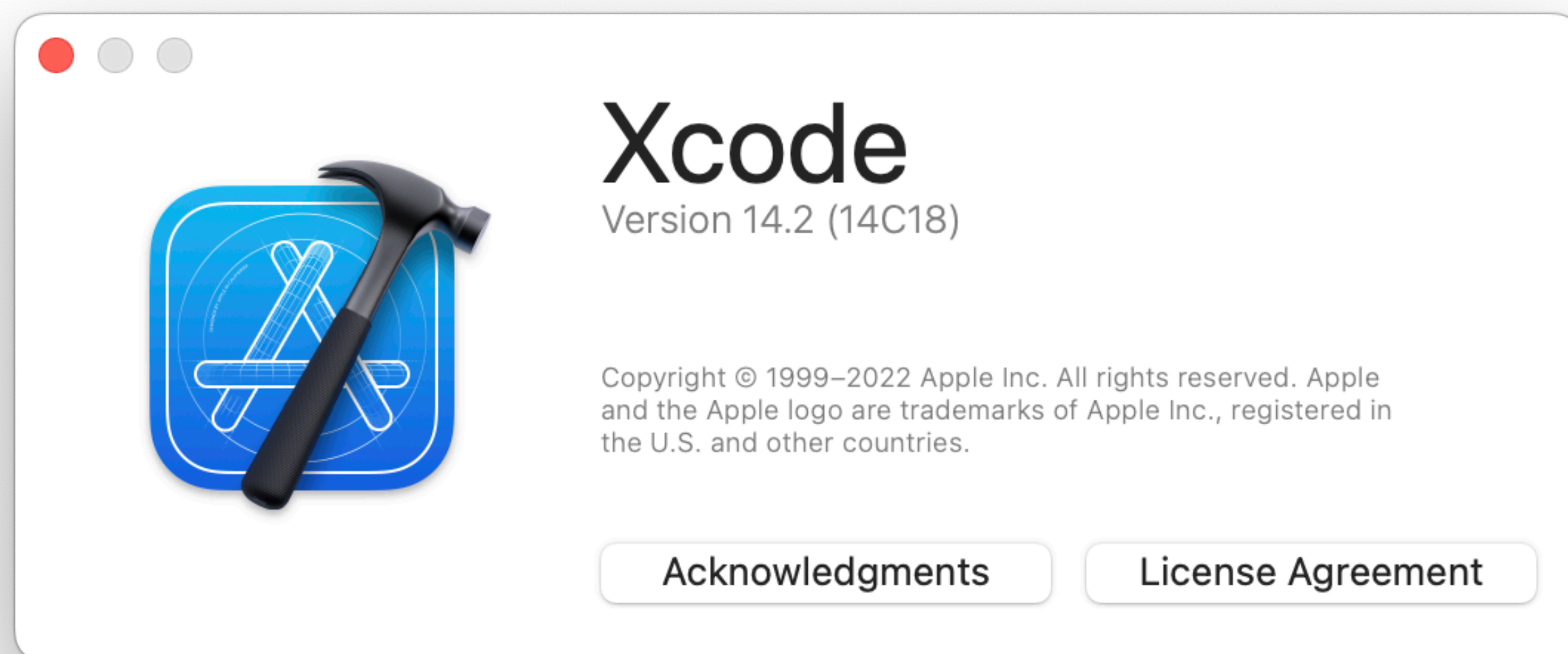
(与Unity、Unreal抗衡)



iOS移动平台AR基础

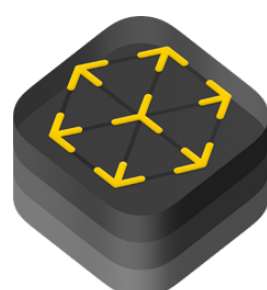
基于RealityKit和SwiftUI的AR开发

AR development based on RealityKit and SwiftUI



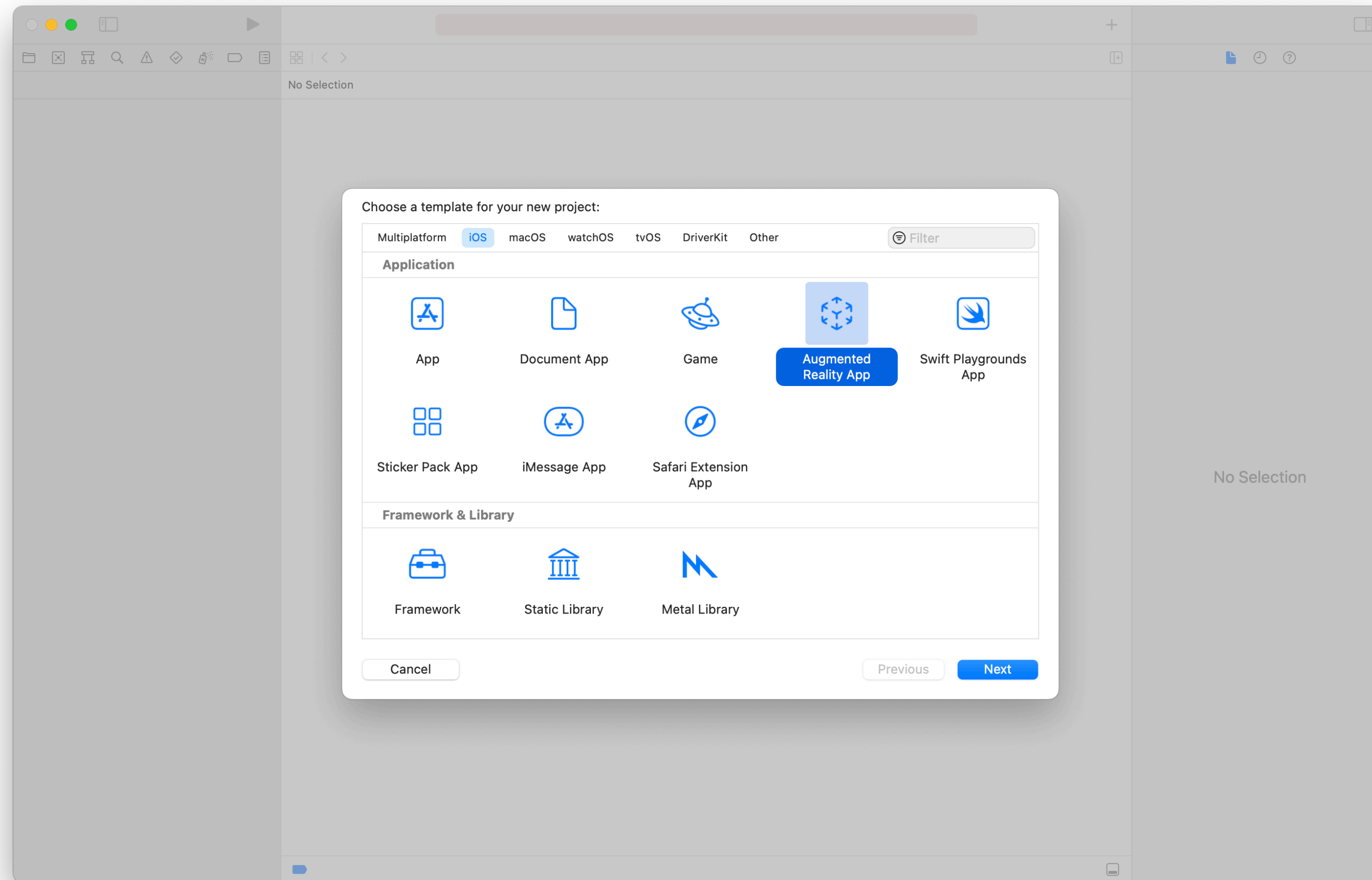
确定Xcode的版本

课程使用的是15.2版本的Xcode



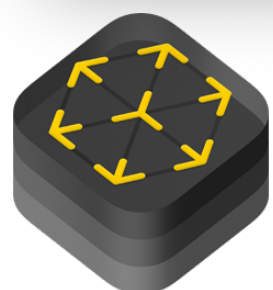
基于RealityKit和SwiftUI的AR开发

AR development based on RealityKit and SwiftUI



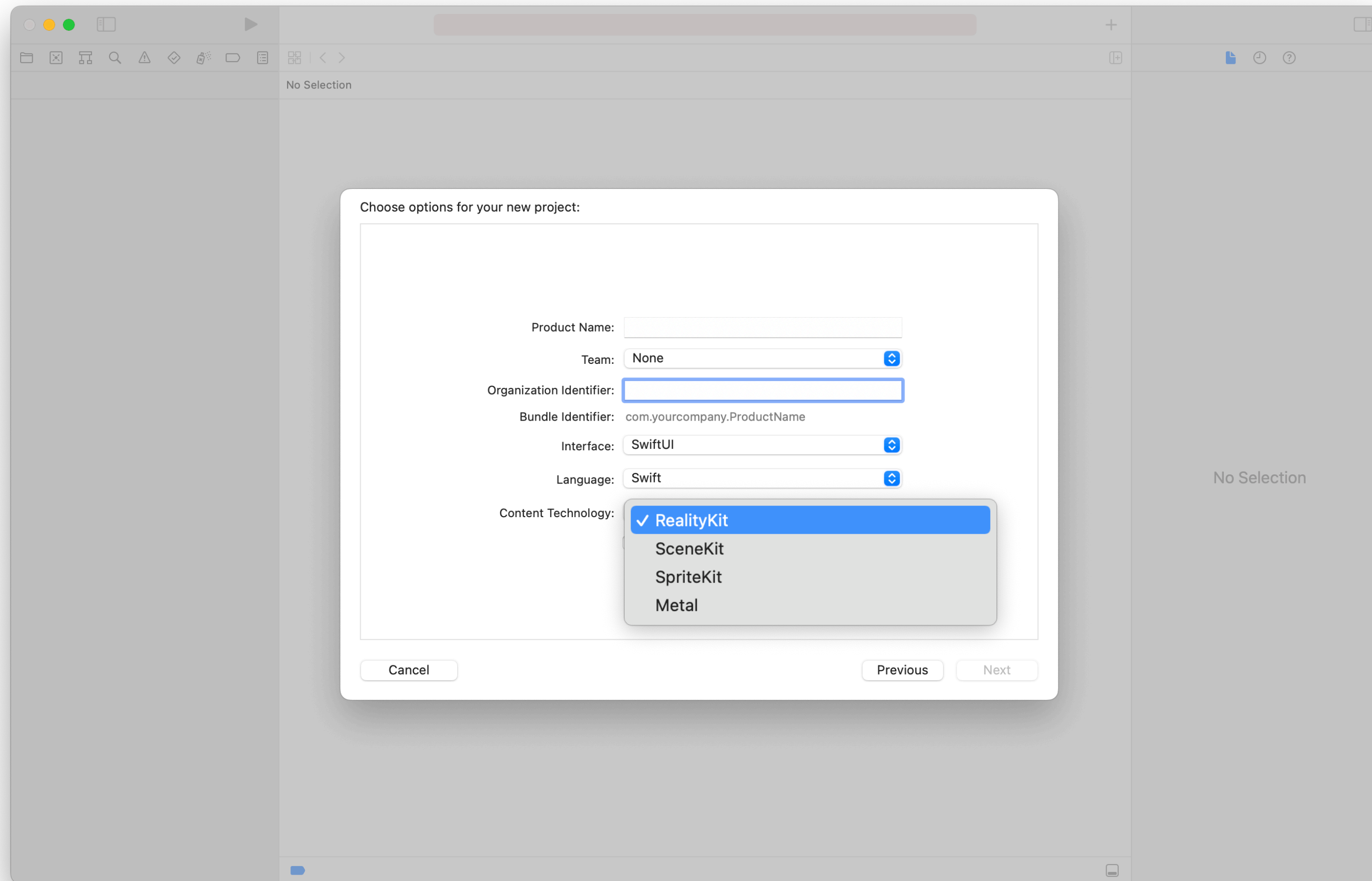
新建一个Xcode工程

选择AR App模版

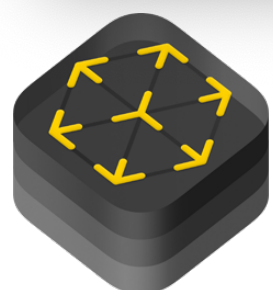


基于RealityKit和SwiftUI的AR开发

AR development based on RealityKit and SwiftUI

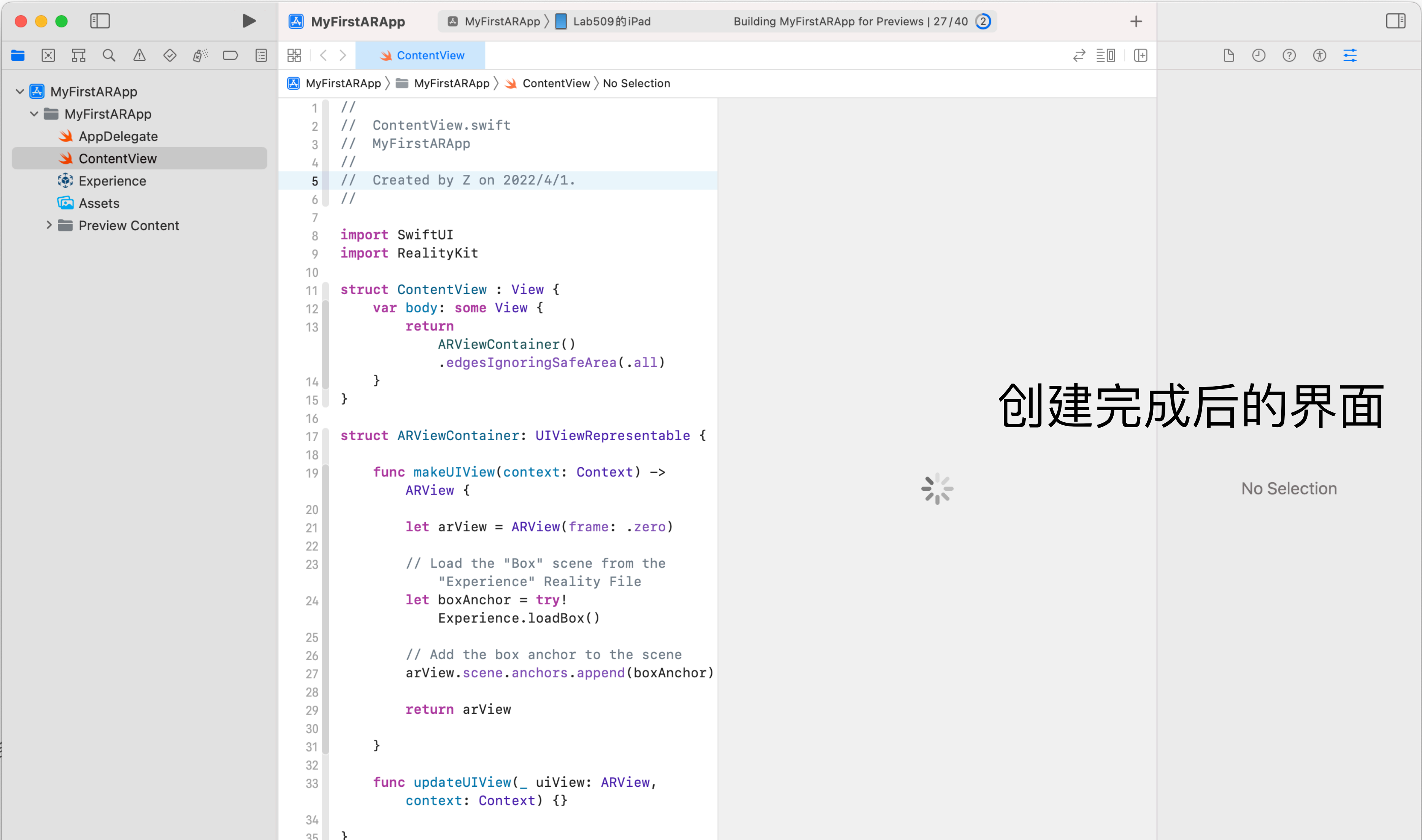


选择RealityKit作为绘制工具
编程语言Swift, UI选择SwiftUI

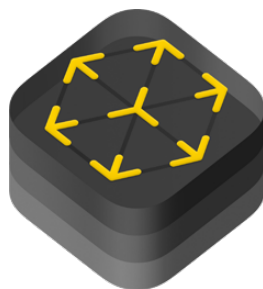


基于RealityKit和SwiftUI的AR开发

AR development based on RealityKit and SwiftUI

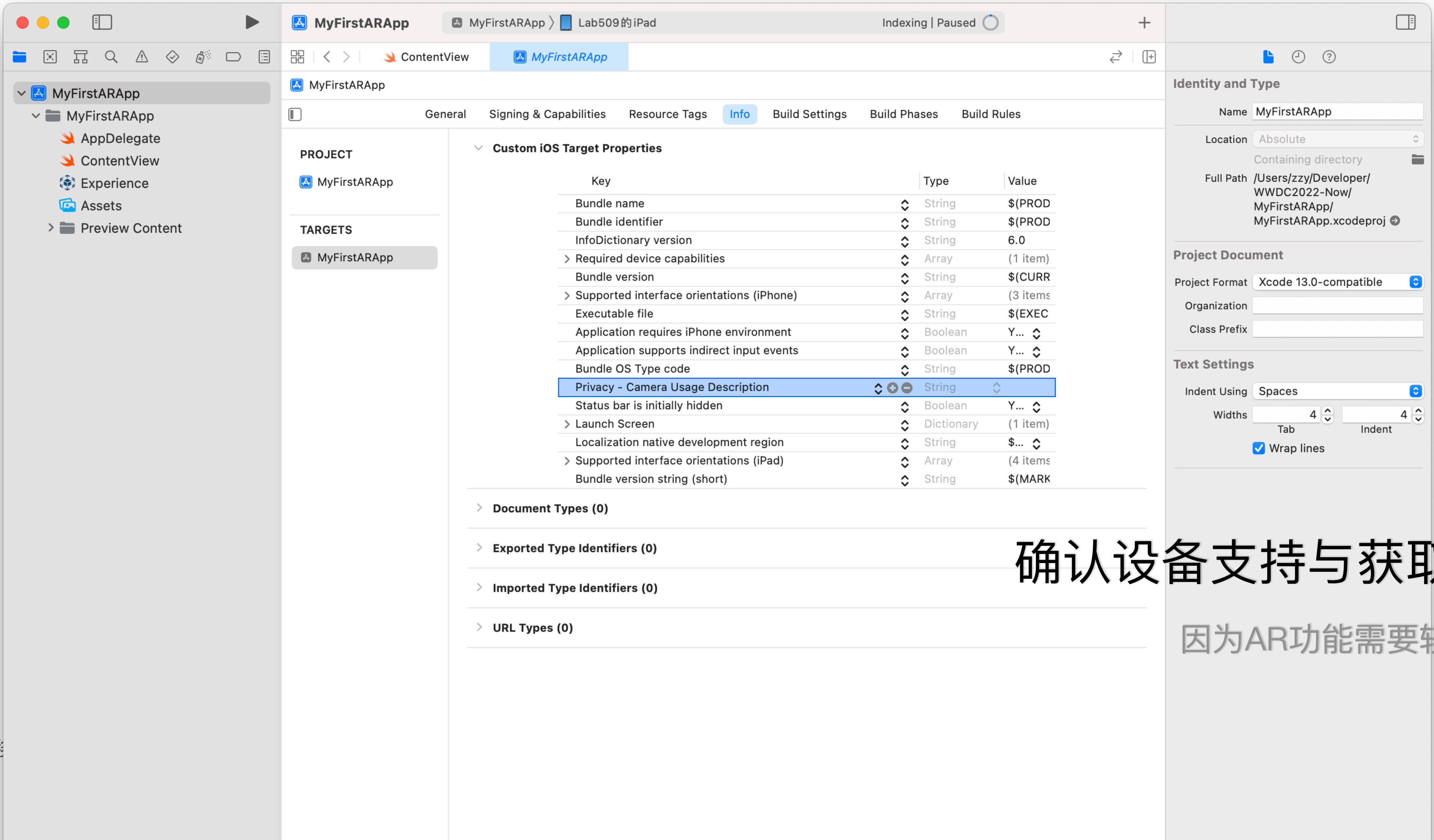


创建完成后的界面



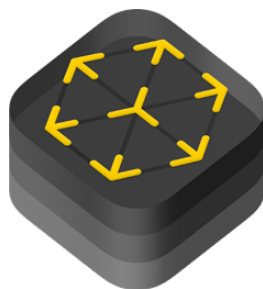
基于RealityKit和SwiftUI的AR开发

AR development based on RealityKit and SwiftUI



确认设备支持与获取用户权限

因为AR功能需要较新的设备支持



基于RealityKit和SwiftUI的AR开发

AR development based on RealityKit and SwiftUI

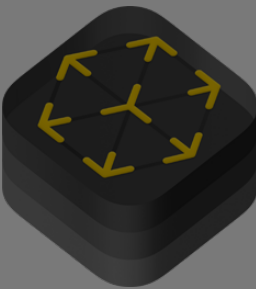
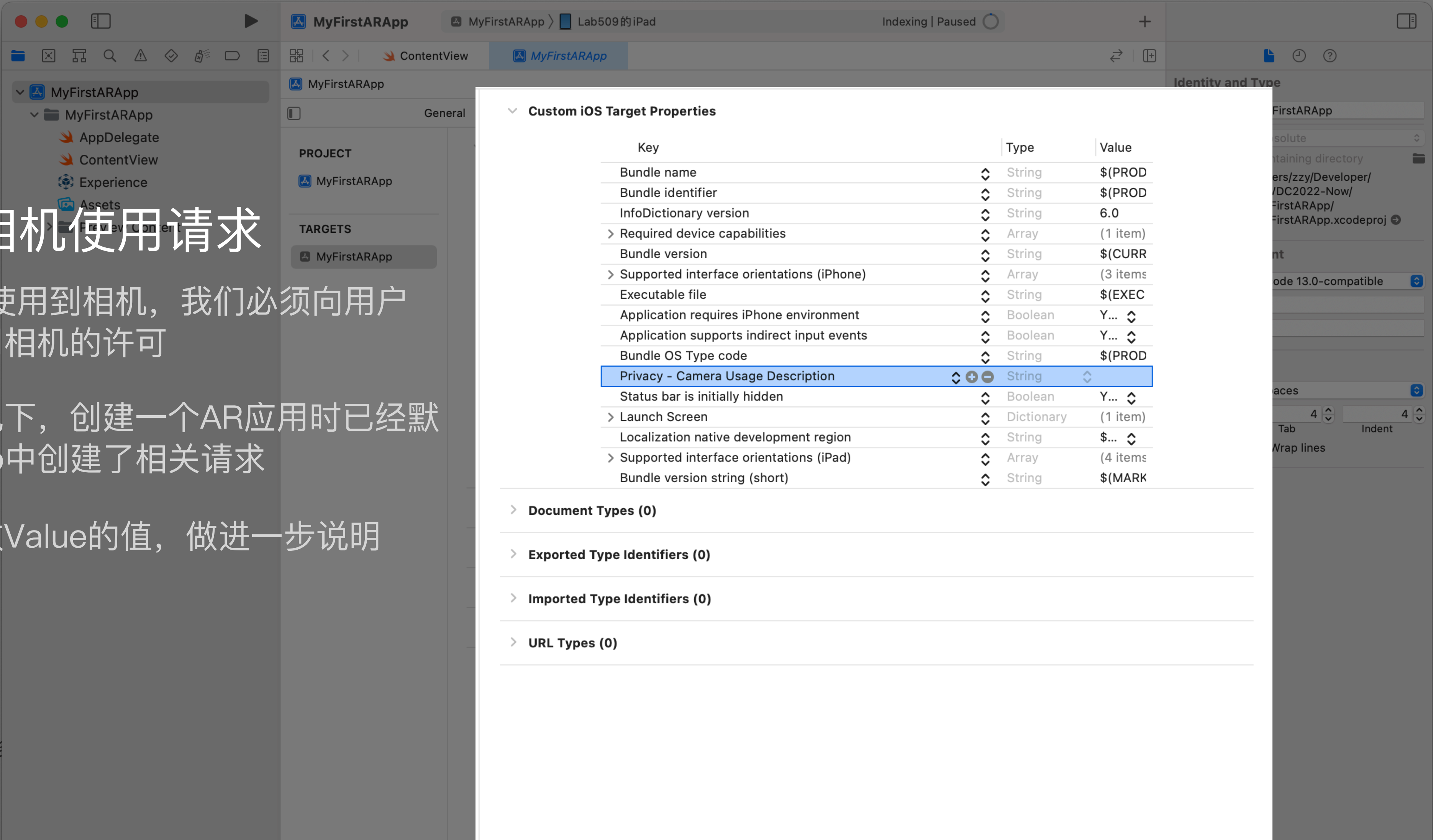


处理相机使用请求

AR需要使用到相机，我们必须向用户
征求使用相机的许可

默认情况下，创建一个AR应用时已经默
认在 Info中创建了相关请求

可以更改Value的值，做进一步说明



基于RealityKit和SwiftUI的AR开发

AR development based on RealityKit and SwiftUI

```
import SwiftUI
import RealityKit

struct ContentView : View {
    var body: some View {
        return ARViewContainer().edgesIgnoringSafeArea(.all)
    }
}
```

```
struct ARViewContainer: UIViewRepresentable {

    func makeUIView(context: Context) -> ARView {

        let arView = ARView(frame: .zero)

        // Load the "Box" scene from the "Experience" Reality File
        let boxAnchor = try! Experience.loadBox()

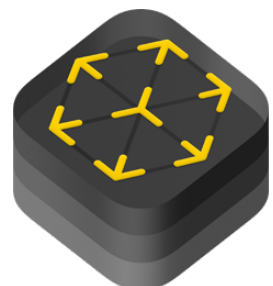
        // Add the box anchor to the scene
        arView.scene.anchors.append(boxAnchor)

        return arView
    }

    func updateUIView(_ uiView: ARView, context: Context) {}
}
```

```
#if DEBUG
struct ContentView_Previews : PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
#endif
```

创建之后，ContentView.swift 中的代码结构



基于RealityKit和SwiftUI的AR开发

AR development based on RealityKit and SwiftUI

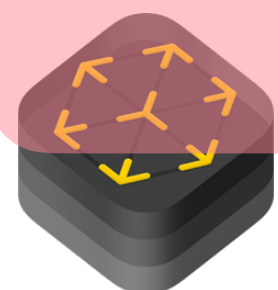


浙江大学
ZHEJIANG UNIVERSITY

```
import SwiftUI
import RealityKit

struct ContentView : View {
    var body: some View {
        ARViewContainer()
            .edgesIgnoringSafeArea()
    }
}
```

```
#if DEBUG
struct ContentView_Previews : PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
#endif
```



iOS移动平台AR基础

```
struct ARViewContainer: UIViewRepresentable {

    func makeUIView(context: Context) -> ARView {

        let arView = ARView(frame: .zero)

        // Load the "Box" scene from the "Experience" Reality File
        let boxAnchor = try! Experience.loadBox()

        // Add the box anchor to the scene
        arView.scene.anchors.append(boxAnchor)

        return arView
    }

    func updateUIView(_ uiView: ARView, context: Context) {}
}
```


基于RealityKit和SwiftUI的AR开发

AR development based on RealityKit and SwiftUI



浙江大学
ZHEJIANG UNIVERSITY

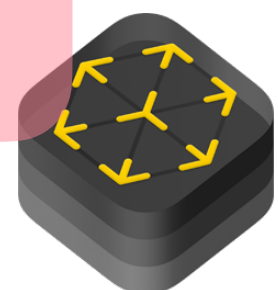
UIView —→ **ARView**

frame: **.zero** 表示

这个 View 渲染的深度为 0

area()

der {



iOS移动平台AR基础

```
struct ARViewContainer: UIViewRepresentable {  
    func makeUIView(context: Context) -> ARView {  
        let arView = ARView(frame: .zero)  
  
        // Load the "Box" scene from the "Experience" Reality File  
        let boxAnchor = try! Experience.loadBox()  
  
        // Add the box anchor to the scene  
        arView.scene.anchors.append(boxAnchor)  
  
        return arView  
    }  
  
    func updateUIView(_ uiView: ARView, context: Context) {}  
}
```

基于RealityKit和SwiftUI的AR开发

AR development based on RealityKit and SwiftUI



浙江大学
ZHEJIANG UNIVERSITY

UIView —→ **ARView**

frame: **.zero** 表示

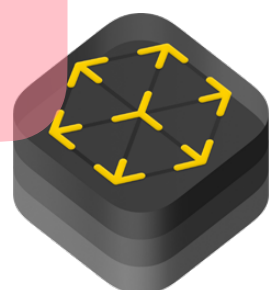
这个 View 渲染的深度为 0

我们不使用默认的Box场景

删除这两行

```
struct ARViewContainer: UIViewRepresentable {  
    func makeUIView(context: Context) -> ARView {  
        let arView = ARView(frame: .zero)  
  
        // Load the "Box" scene from the "Experience" Reality File  
        let boxAnchor = try! Experience.loadBox()  
  
        // Add the box anchor to the scene  
        arView.scene.anchors.append(boxAnchor)  
  
        return arView
```

```
: Context) {}
```



在iPad上进行AR开发



AR development on iPad

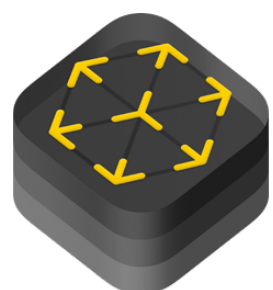


Swift Playgrounds是一款适用于iPad和Mac的应用程序，它使Swift学习变得互动而有趣！

自Playground 4.0发布以后

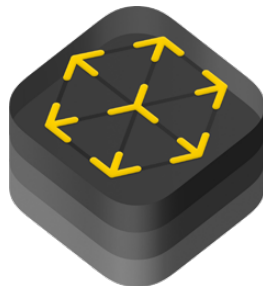
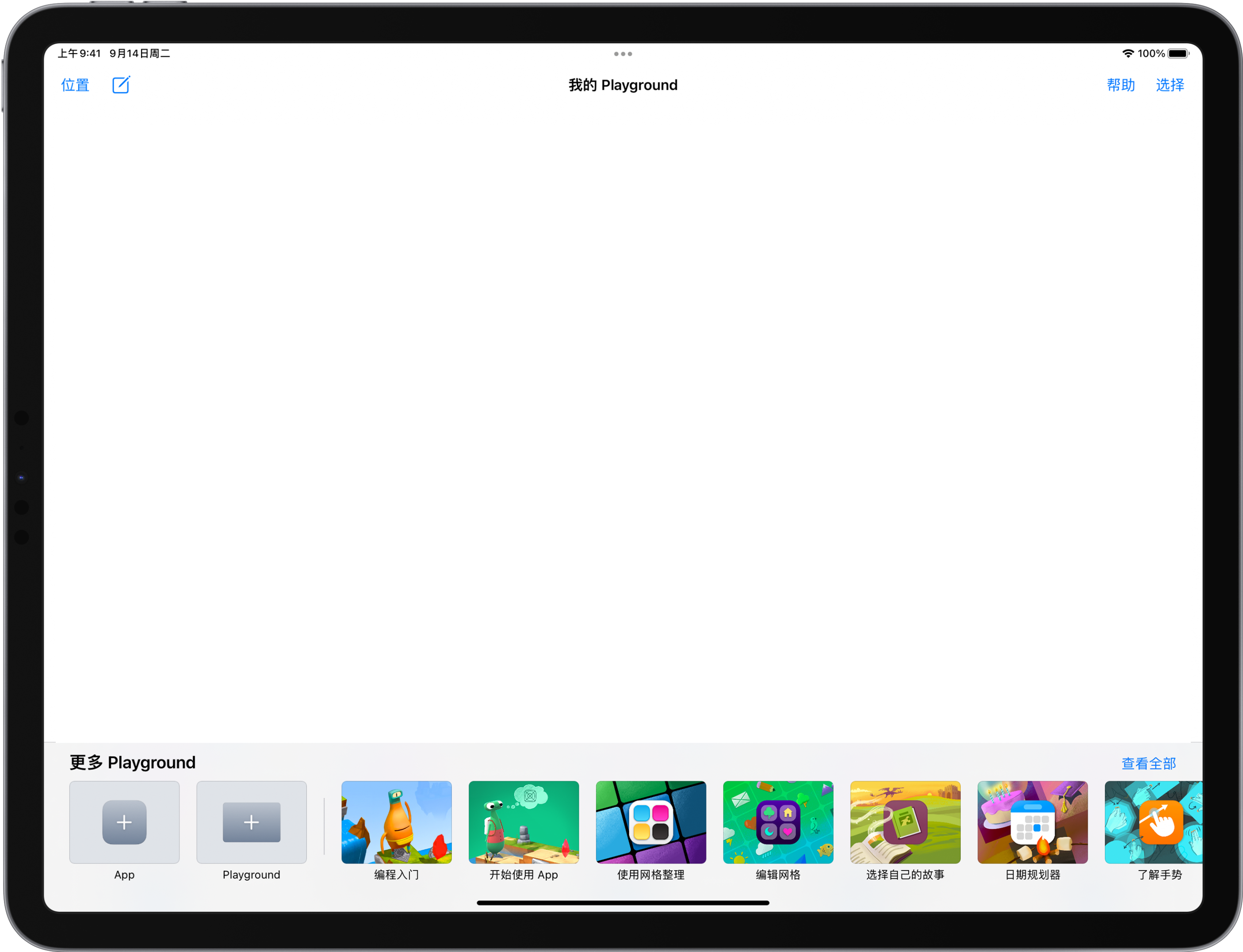
我们可以在 iPad 上编译Swift语言就像在 XCode 上！

接下来的我们会演示如何在 iPad 的 Playground 上开发！



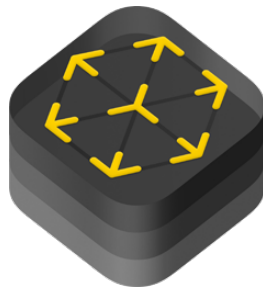
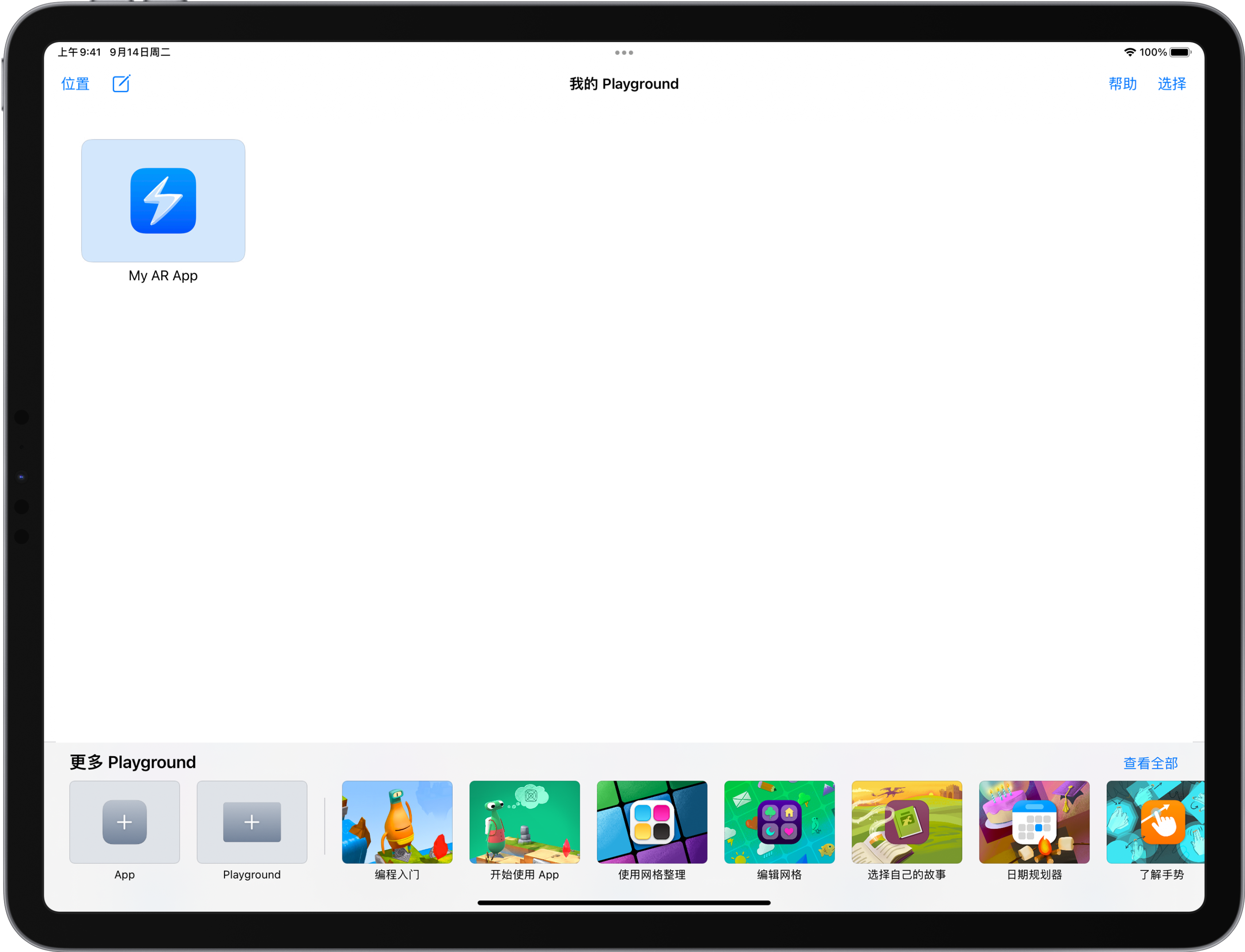
在iPad上进行AR开发

AR development on iPad



在iPad上进行AR开发

AR development on iPad



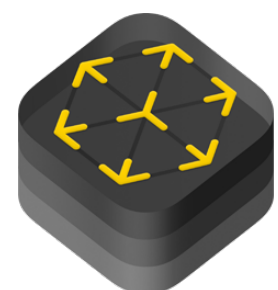
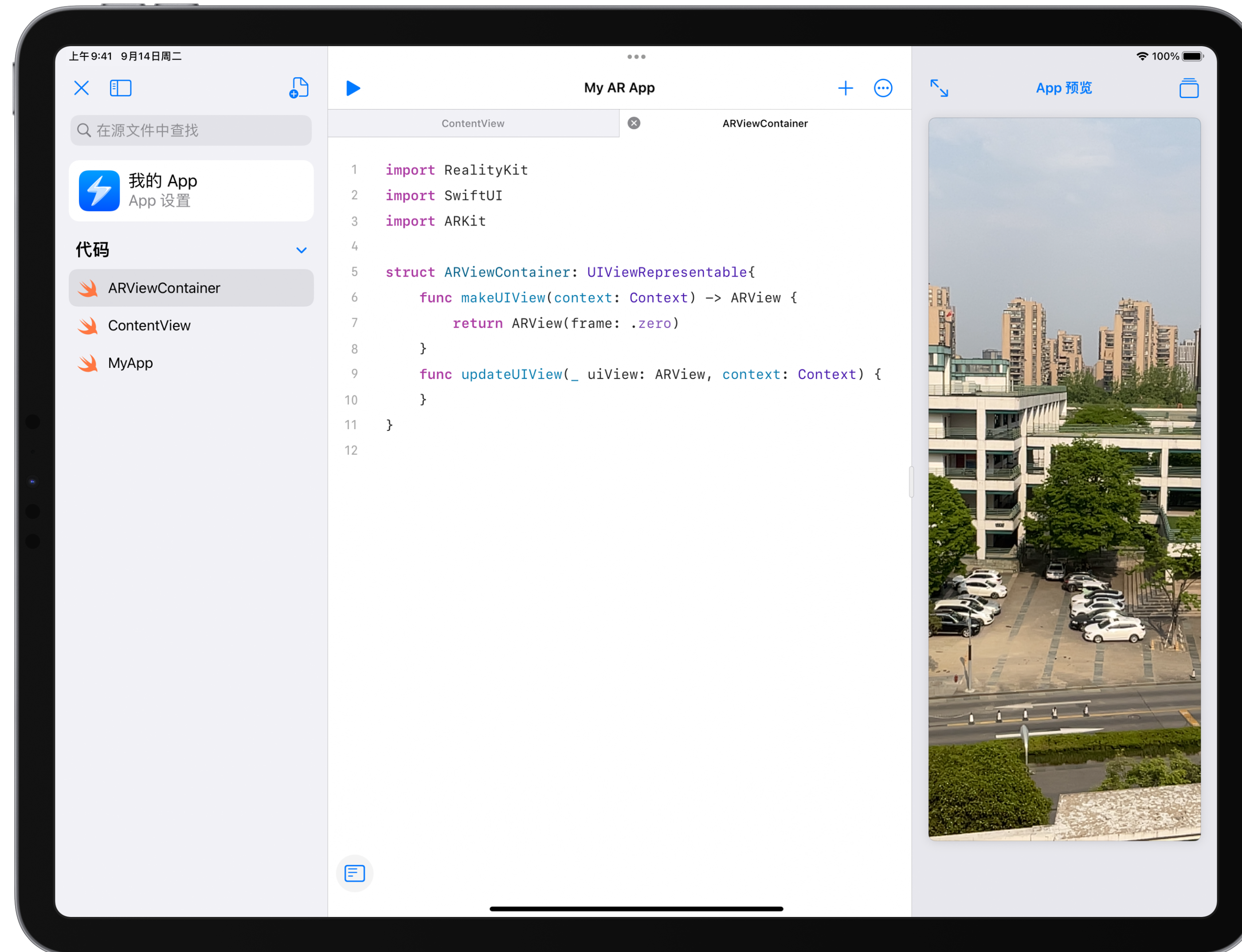
在iPad上进行AR开发

AR development on iPad



浙江大学
ZHEJIANG UNIVERSITY

与Xcode模版一致



iOS移动平台AR基础

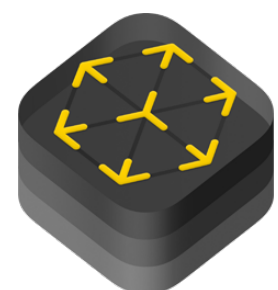
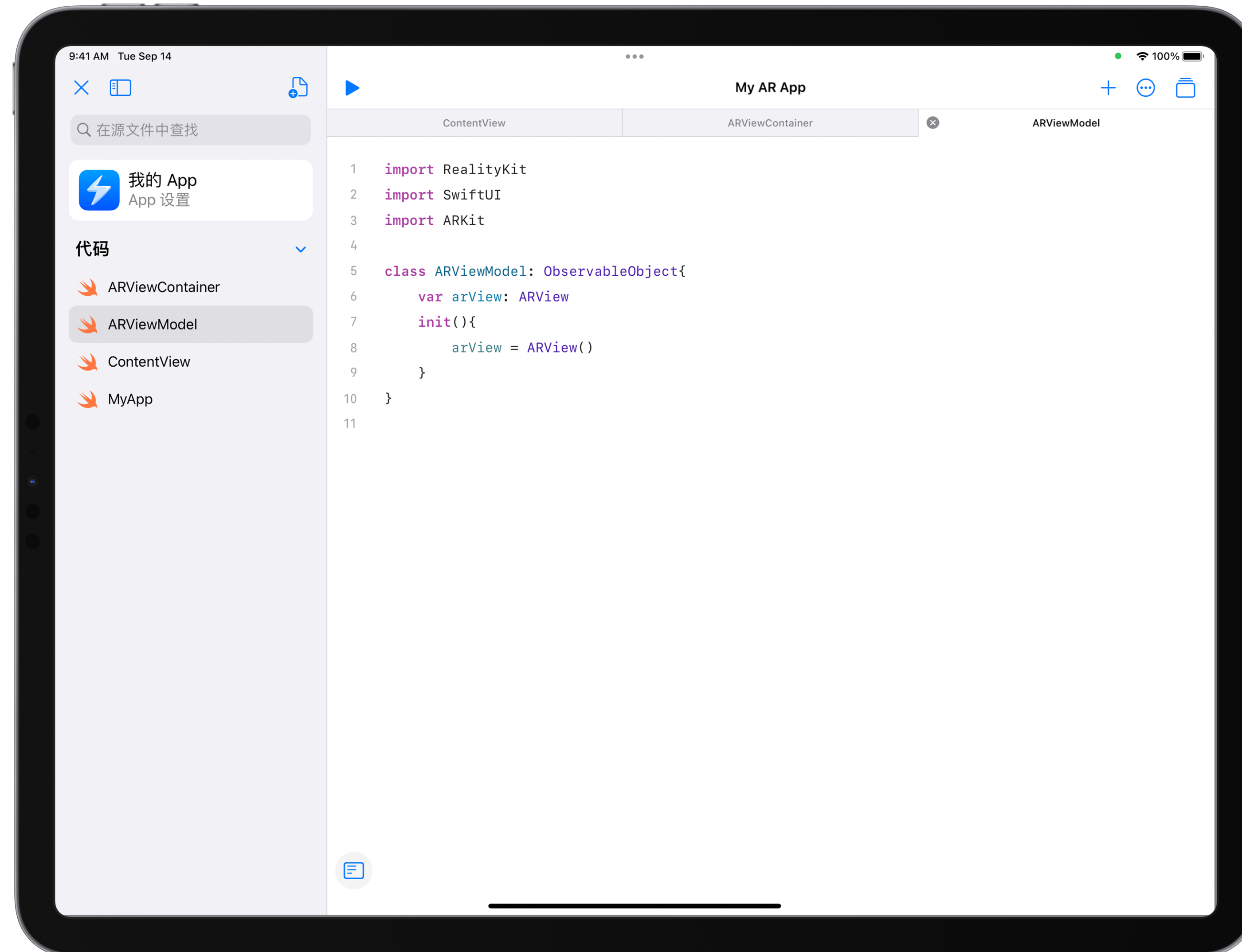
在iPad上进行AR开发

AR development on iPad



浙江大学
ZHEJIANG UNIVERSITY

采用MVVM模式



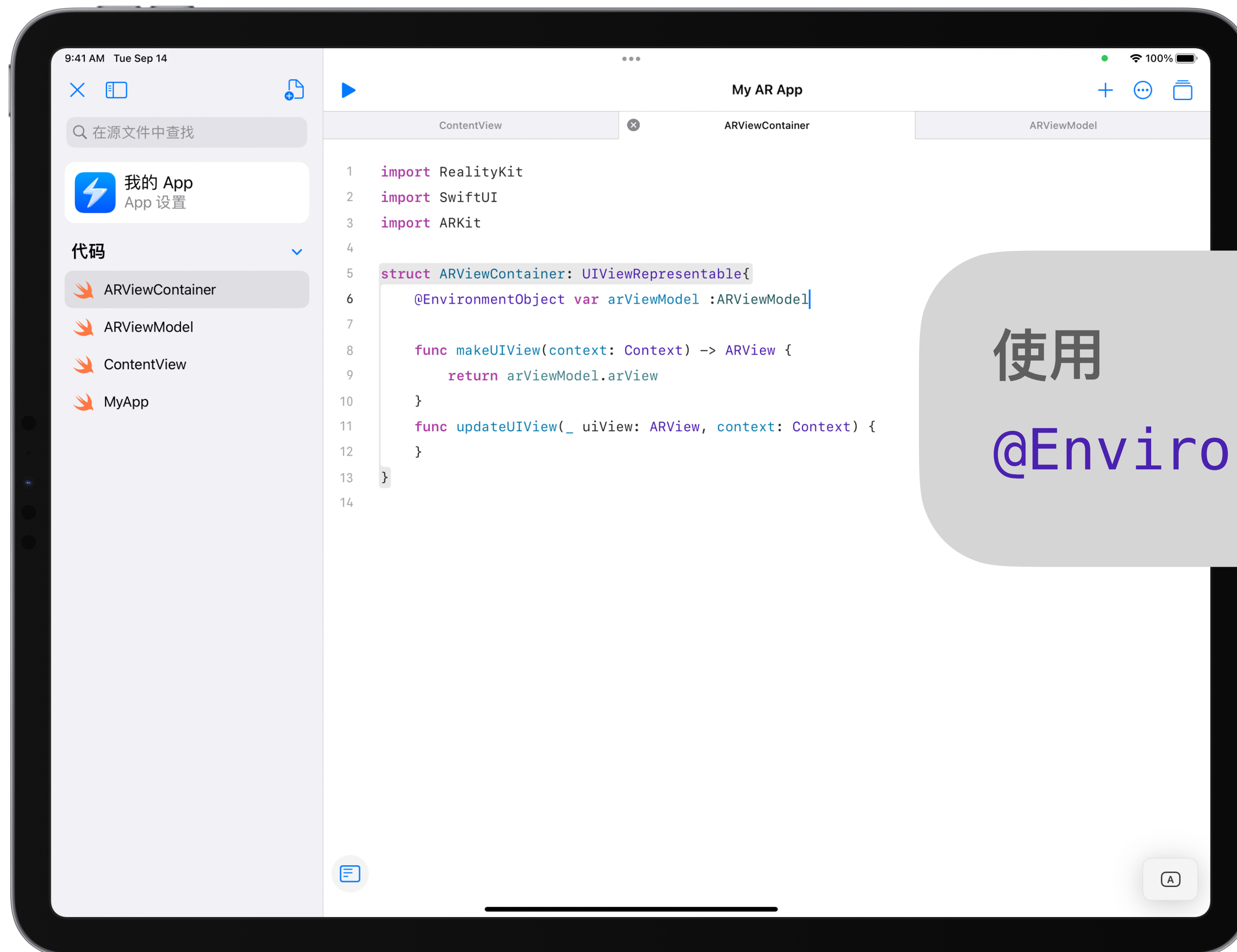
iOS移动平台AR基础

在iPad上进行AR开发

AR development on iPad

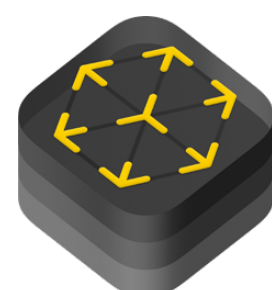


浙江大学
ZHEJIANG UNIVERSITY



使用

@EnvironmentObject



iOS移动平台AR基础

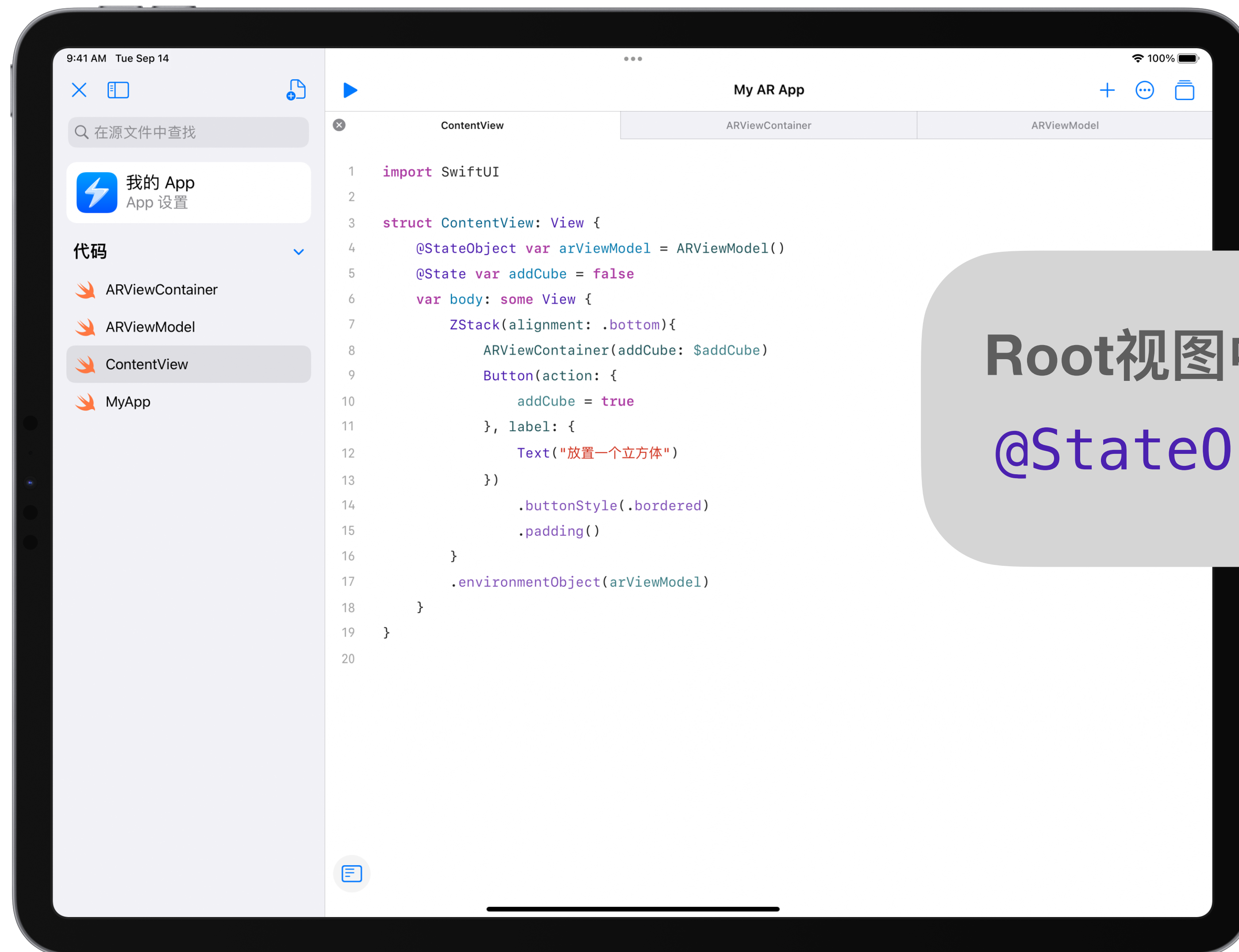
在iPad上进行AR开发

AR development on iPad

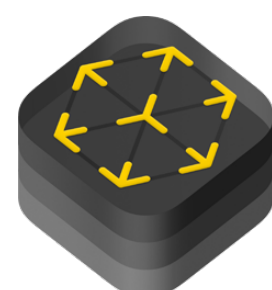


浙江大学
ZHEJIANG UNIVERSITY

添加一个按钮



Root视图中使用
`@StateObject`



iOS移动平台AR基础

AR development on iPad



The image shows an iPad screen with the Xcode IDE open. The top status bar displays the time as 9:41 AM on Tuesday, September 14, and the battery level at 100%.

Left Sidebar (Project Navigator):

- Search bar: 在源文件中查找
- Section: 我的 App (My App)
- Sub-section: App 设置 (App Settings)
- Section: 代码 (Code)
- Files listed: ARViewController, ARViewModel, ContentView, MyApp

Center Editor (ARViewController.swift):

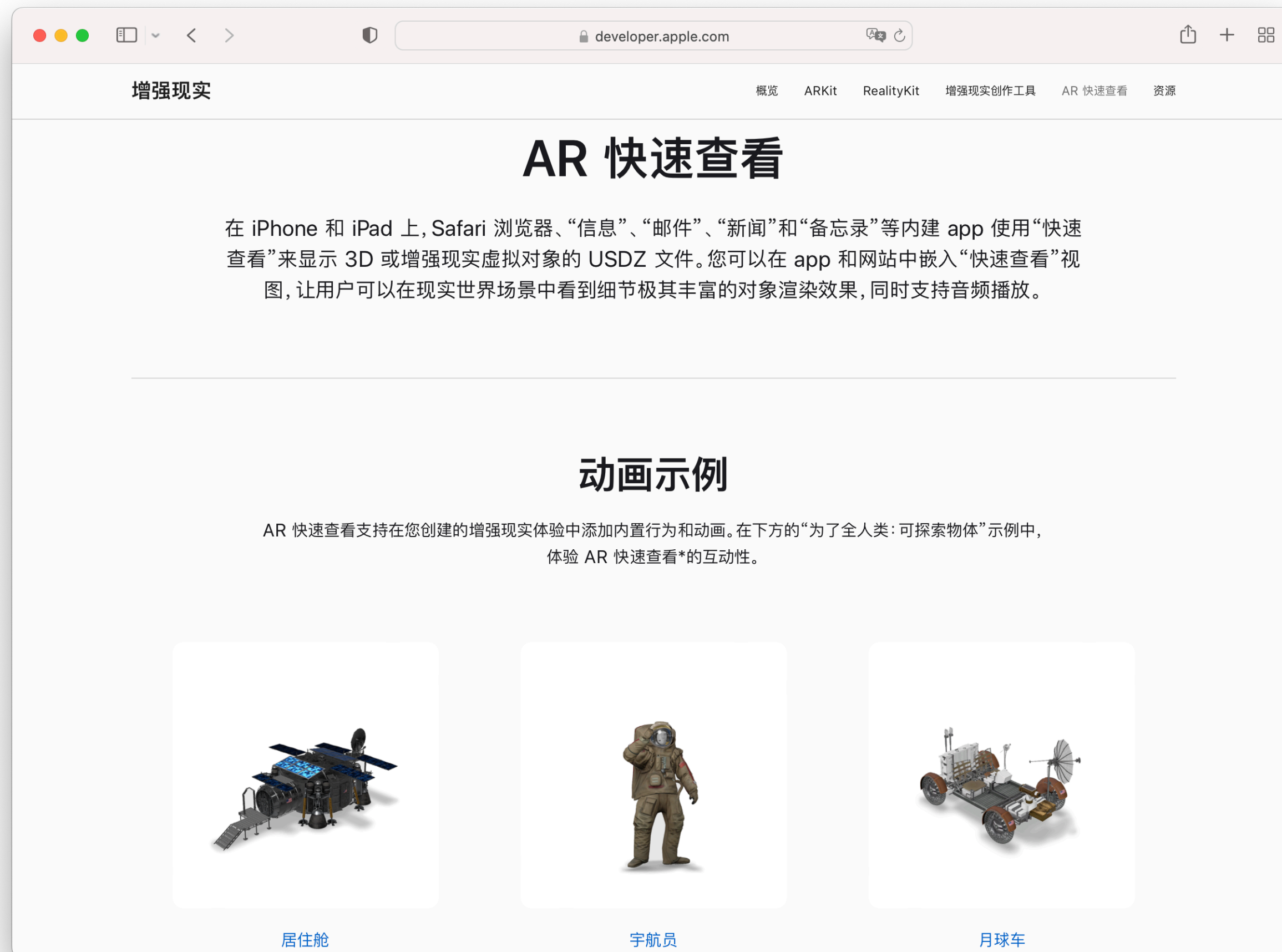
```
1  import RealityKit
2  import SwiftUI
3  import ARKit
4
5  struct ARViewController: UIViewRepresentable{
6      @EnvironmentObject var arViewModel :ARViewModel
7      @Binding var addCube: Bool
8
9      func makeUIView(context: Context) -> ARView {
10         return arViewModel.arView
11     }
12     func updateUIView(_ uiView: ARView, context: Context) {
13         if addCube{
14             let boxMesh = MeshResource.generateBox(size: 0.1)
15             let material = SimpleMaterial(color: .blue,
16                                           isMetallic: false)
17             let modelEntity = ModelEntity(mesh: boxMesh,
18                                           materials: [material])
19             let anchorEntity = AnchorEntity(plane: .any)
20
21             anchorEntity.addChild(modelEntity)
22             arViewModel.arView.scene.addAnchor(anchorEntity)
23             DispatchQueue.main.async {
24                 addCube = false
25             }
26         }
27     }
28 }
```

Right Preview (App 预览):

The preview area shows a 3D rendering of a blue cube placed on a wooden surface. A text label at the bottom of the preview area reads "放置一个立方体" (Place a cube).

更多模型

Models



想显示更复杂的模型?

下载或创建USDZ模型

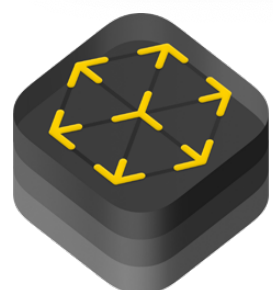
Apple和皮克斯公司为AR特别创建的
基于皮克斯的“通用场景描述”标准
向Apple设备提供了AR和3D内容的新模式

3D模型可以锚定在现实世界中

3D模型会对真实场景做出反应

可以融合物理效果模拟

可以在空间某点添加音频



AR开发需要的工具

Tools we need



Xcode 13



Reality Composer

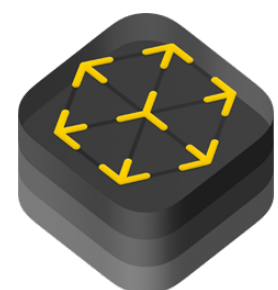


Reality Converter



USDZ Tools

创建USDZ模型



iOS移动平台AR基础

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