Sample Code

Docking a video player in an immersive scene

Secure a video player in an immersive scene with a docking region you can specify.

Download

visionOS 2.0+ | Xcode 16.0+

Overview

In visionOS, all full-screen videos automatically dock when an immersive space is open. This behavior places the video in a fixed position within your immersive space. You can use `DockingRegionComponent` to customize the width and location of the docking region.

This sample code project demonstrates how to use the RealityKit API for the docking-region component. You define and specify the boundary of the docking region and docking position using a docking-region component, and attach it to an entity within the immersive scene. Then you position the entity that has the docking-region component to customize the docking position.

The docking-region component allows developers to customize the docking region. Alternatively, you can add and define a docking-region component using Reality Composer Pro. See Building an immersive experience with RealityKit for more details.

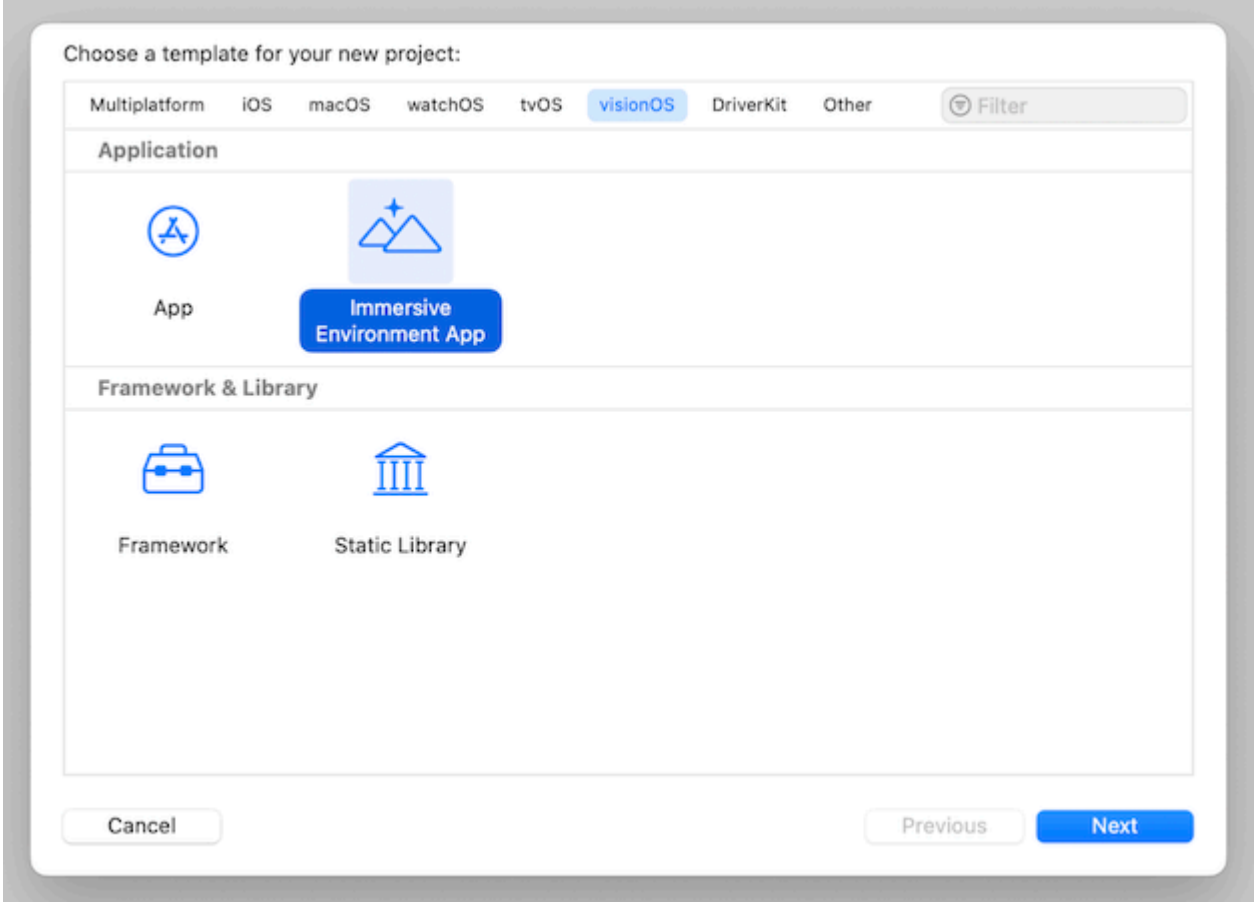
The sample app presents an AV player view controller with default playback controls, and a scene picker for selecting immersive scenes. You can start video playback with the default controls, and enter an immersive scene with the immersive view picker.

Play ⦿

You can dock a video player by following the steps below.

Create an immersive visionOS app

In Xcode, choose File > New > Project, select visionOS and Immersive Environment App in the New Project dialog, and click Next.



Enter your product name and organization identifier, and choose between a progressive and full immersive space. You can change the immersive setting later.

To continue, include a video in the bundle or provide a URL to an external video in `AVPlayerViewModel.swift`. The Immersive Environment App template launches with a window that includes a single button to show the immersive space. Showing the immersive space starts playing the video and displays the `AVPlayerView` full screen in the window. The system automatically docks this full-screen video in the default docking position.

Play the video in the shared space

To play the video in the shared space and immersive space, remove the calls to the `avPlayerViewModel` in the `onAppear(perform:)` and `onDisappear(perform:)` methods of `ImmersiveView`. Remove the `ToggleImmersiveSpaceButton` from the `ContentView` and replace it with a button that plays the video. At this point, the app plays the video full screen after someone taps the Play Video button, but the custom immersive space doesn't appear.

Add an immersive scene-viewing option to your video player

Create an immersive picker view that toggles the immersive scene through existing immersive space APIs.

```
struct ImmersivePickerView: View {
    let appModel: AppModel

    /// An asynchronous call returns after dismissing the immersive space.
    @Environment(\.dismissImmersiveSpace) private var dismissImmersiveSpace
    /// An asynchronous call returns after opening the immersive space.
    @Environment(\.openImmersiveSpace) private var openImmersiveSpace

    var body: some View {
        // Add a button to toggle the immersive environment.
        Button("Sky Dome", systemImage: "photo") {
            Task {
                if appModel.immersiveSpaceIsShown {
                    await dismissImmersiveSpace()
                } else {
                    await openImmersiveSpace(id: appModel.immersiveSpaceID)
                }
            }
```

```
        }
    }
}
```

Use `immersiveEnvironmentPicker(content:)` to include this picker as an option in the list of immersive scene selections.

```
AVPlayerView(viewModel: avPlayerViewModel)
    .immersiveEnvironmentPicker {
        ImmersivePickerView(appModel: appModel)
    }
```

Create a docking-region component with a docking entity

The Immersive Environment App template comes with a Reality Composer Pro project that includes a `Player` entity with the docking region. To customize the docking-region settings in Xcode, open the Reality Composer Pro package and remove the `Video_Dock` entity that contains the `Player` entity.

Within your `ImmersiveView`, create a docking-region component with a width of your choice. The docking region uses a cinematic `2.4:1` width-height ratio to determine the height.

```
// Create a docking-region component to customize your docking region.
var dockingRegionComponent = DockingRegionComponent()
// Set the docking region width to 9.6 meters.
dockingRegionComponent.width = 9.6
```

Create a docking entity and position the entity where you want to dock the AV player. Then attach the docking-region component to the docking entity.

```
// Create a docking entity as a docking anchor.
let dockingEntity = Entity()

// Set the position of your dock, in meters.
dockingEntity.position = [0, 2, -10]

// Attach the docking-region component to the docking entity.
dockingEntity.components.set(dockingRegionComponent)
```

And, finally, add it to your `RealityView`.

```
content.add(dockingEntity)
```

See Also

SwiftUI video content

{} Destination Video

Leverage SwiftUI to build an immersive media experience in a multiplatform app.