

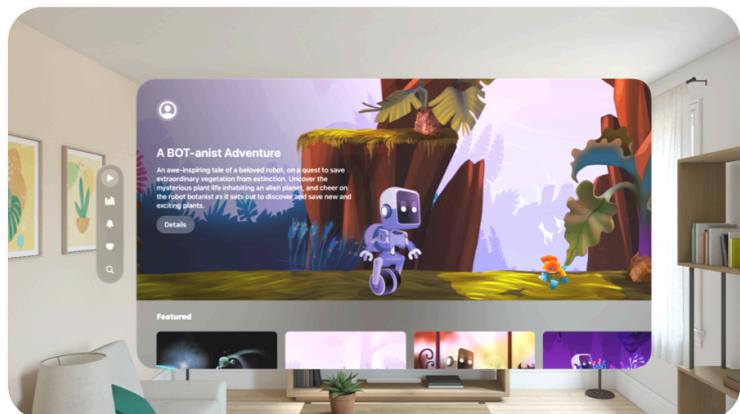
[RealityKit](#) / Videos

API Collection

Videos

Present videos in your RealityKit experiences.

Overview



Destination Video

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

[View 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.

[View sample code >](#)



Enabling video reflections in an immersive environment

Create a more immersive experience by adding video reflections in a custom environment.

[View sample code >](#)



Rendering stereoscopic video with RealityKit

Render stereoscopic video in visionOS with RealityKit.

[View sample code >](#)

Playing immersive media with RealityKit

Create an immersive video playback experience with RealityKit.

[View sample code >](#)

Topics

Video player configurations

`struct VideoPlayerComponent`

A component that supports general video-playback experience with an AV player.

`enum ImmersiveViewingMode`

Options for viewing the video during immersive-media playback.

`struct VideoMaterial`

A material that supports animated textures.

`class VideoPlaybackController`

An object that controls the playback of video for a video material.

`enum ViewingMode`

Options for viewing video playback.

Content placement

```
struct DockingRegionComponent
```

A component that docks a scene within a region of an immersive space.

Playback notifications

```
enum VideoPlayerEvents
```

Events associated with video playback for VideoPlayerComponent.

SwiftUI video content

{ } Destination Video

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

{ } Docking a video player in an immersive scene

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

See Also

Scene content

{ } Hello World

Use windows, volumes, and immersive spaces to teach people about the Earth.

{ } Enabling video reflections in an immersive environment

Create a more immersive experience by adding video reflections in a custom environment.

{ } Creating a spatial drawing app with RealityKit

Use low-level mesh and texture APIs to achieve fast updates to a person's brush strokes by integrating RealityKit with ARKit and SwiftUI.

{ } Generating interactive geometry with RealityKit

Create an interactive mesh with low-level mesh and low-level texture.

{ } Combining 2D and 3D views in an immersive app

Use attachments to place 2D content relative to 3D content in your visionOS app.

{ } Transforming RealityKit entities using gestures

Build a RealityKit component to support standard visionOS gestures on any entity.

{ } Responding to gestures on an entity

Respond to gestures performed on RealityKit entities using input target and collision components.

:≡ Models and meshes

Display virtual objects in your scene with mesh-based models.

:≡ Materials, textures, and shaders

Apply textures to the surface of your scene's 3D objects to give each object a unique appearance.

:≡ Anchors

Lock virtual content to the real world.

:≡ Lights and cameras

Control the lighting and point of view for a scene.

:≡ Content synchronization

Synchronize the contents of entities locally or across the network.

:≡ Audio

Create personalized and realistic spatial audio experiences.

:≡ Images

Present images and spatial scenes in your RealityKit experiences.