

☰ Documentation

[visionOS](#) / Introductory visionOS samples

Introductory visionOS samples

Learn the fundamentals of building apps for visionOS with beginner-friendly sample code projects.

visionOS 2.0+ | Xcode 16.0+

Overview

The samples on this page are a starting point for developers new to visionOS. Each focuses on a specific feature, providing a solid foundation to build apps for the Apple Vision Pro.



Creating 2D shapes with SwiftUI

Draw two-dimensional shapes in your visionOS app with SwiftUI shapes or with your custom shapes.

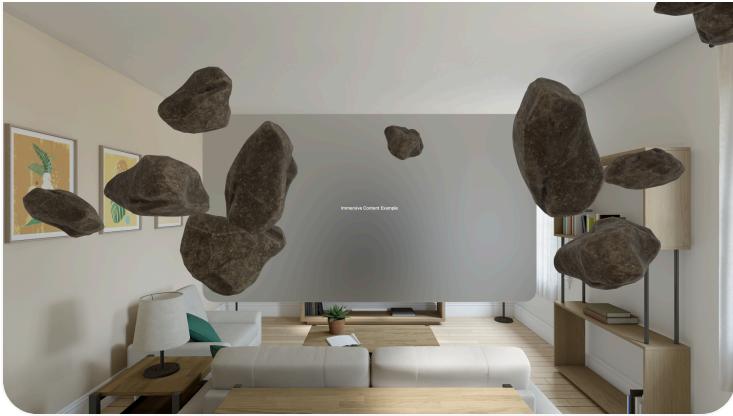
[View sample code >](#)



Creating 3D entities with RealityKit

Display a horizontal row of three-dimensional shapes in your visionOS app, using predefined mesh and white material.

[View sample code >](#)



Creating an immersive space in visionOS

Enhance your visionOS app by adding an immersive space using RealityKit.

[View sample code >](#)



Adding a depth effect to text in visionOS

Create text that expands out of a window using stacked SwiftUI text views.

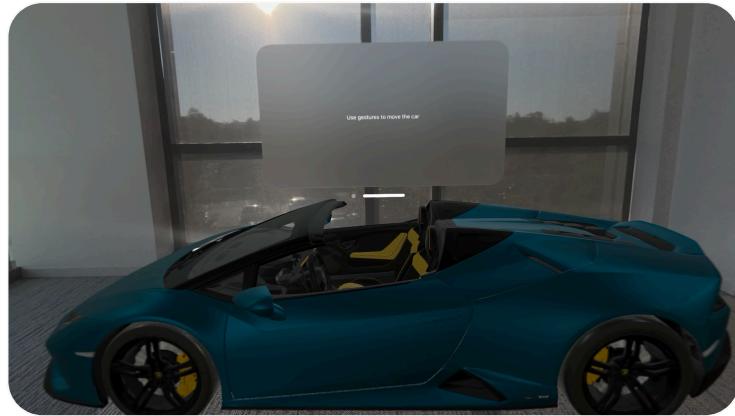
[View sample code >](#)



Tracking and visualizing hand movement

Use hand-tracking anchors to display a visual representation of hand transforms in visionOS.

[View sample code >](#)



Creating an interactive 3D model in visionOS

Display an interactive car model using gestures in a reality view.

[View sample code >](#)



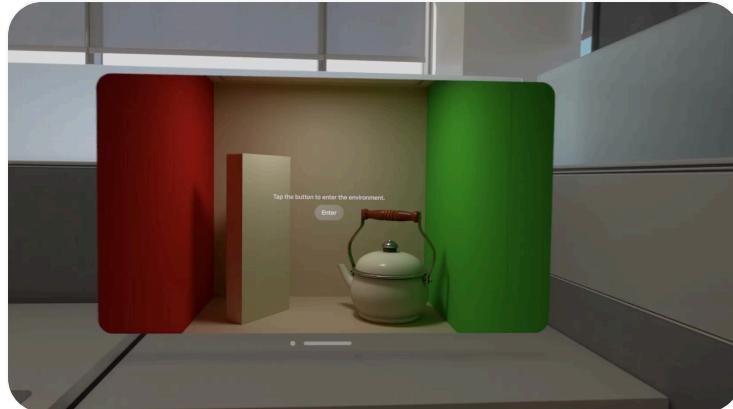
Creating a 3D painting space



Generating procedural textures

Implement a painting canvas entity, and update its mesh to represent a stroke.

[View sample code >](#)



Displaying a 3D environment through a portal

Implement a portal window that displays a 3D environment and simulates entering a portal by using RealityKit.

[View sample code >](#)

Display a 3D model that generates procedural textures in a reality view.

[View sample code >](#)



Obscuring virtual items in a scene behind real-world items

Increase the realism of an immersive experience by adding entities with invisible materials real-world objects.

[View sample code >](#)

Topics

Building shapes

{ } Creating 2D shapes with SwiftUI

Draw two-dimensional shapes in your visionOS app with SwiftUI shapes or with your custom shapes.

{ } Creating 3D entities with RealityKit

Display a horizontal row of three-dimensional shapes in your visionOS app, using predefined mesh and white material.

Working with windows

{ } Creating SwiftUI windows in visionOS

Display and manage multiple SwiftUI windows in your visionOS app.

{ } Creating 3D models as movable windows

Display 3D content with a volumetric window that people can move.

Drawing text

{ } Displaying text in visionOS

Create styled text in a window using SwiftUI.

{ } Adding a depth effect to text in visionOS

Create text that expands out of a window using stacked SwiftUI text views.

Implementing an immersive space

{ } Creating an interactive 3D model in visionOS

Display an interactive car model using gestures in a reality view.

{ } Creating an immersive space in visionOS

Enhance your visionOS app by adding an immersive space using RealityKit.

Integrating ARKit

{ } Creating a 3D painting space

Implement a painting canvas entity, and update its mesh to represent a stroke.

{ } Tracking and visualizing hand movement

Use hand-tracking anchors to display a visual representation of hand transforms in visionOS.

{ } Displaying an entity that follows a person's view

Create an entity that tracks and follows head movement in an immersive scene.

{ } Applying mesh to real-world surroundings

Add a layer of mesh to objects in the real world, using scene reconstruction in ARKit.

{ } Obscuring virtual items in a scene behind real-world items

Increase the realism of an immersive experience by adding entities with invisible materials over real-world objects.

Building materials

{ } Generating procedural textures

Display a 3D model that generates procedural textures in a reality view.

- { } Implementing adjustable material
Update the adjustable parameters of a 3D model in visionOS.
- { } Displaying a stereoscopic image
Build a stereoscopic image by applying textures to the left and right eye in a shader graph material.

Applying spatial audio

- { } Playing spatial audio
Create and adjust spatial audio in visionOS with RealityKit.

Creating portals

- { } Displaying a 3D environment through a portal
Implement a portal window that displays a 3D environment and simulates entering a portal by using RealityKit.
-

See Also

App construction

- 📄 Creating your first visionOS app
Build a new visionOS app using SwiftUI and add platform-specific features.
- 📄 Adding 3D content to your app
Add depth and dimension to your visionOS app and discover how to incorporate your app's content into a person's surroundings.
- 📄 Creating fully immersive experiences in your app
Build fully immersive experiences by combining spaces with content you create using RealityKit or Metal.
- 📄 Drawing sharp layer-based content in visionOS
Deliver text and vector images at multiple resolutions from custom Core Animation layers in visionOS.