RealityKit / Content synchronization

API Collection

Content synchronization

Synchronize the contents of entities locally or across the network.

Topics

Entity ownership synchronization

`protocol SynchronizationService`

An interface that enables entity synchronization among a group of local peers.

`typealias Identifier`

A type that represents a synchronization service identifier.

`protocol SynchronizationPeerID`

A type that represents a peer among a group of networked devices.

`struct SynchronizationComponent`

A component that synchronizes an entity between processes and networked applications.

`enum OwnershipTransferMode`

Modes of ownership transfer.

`enum OwnershipTransferCompletionResult`

The result of an ownership transfer request.

`enum SynchronizationEvents`

Events associated with network synchronization of scene information.

`protocol HasSynchronization`

An interface that enables an entity to be synchronized between processes and networked applications.

Multipeer synchronization

Loading remote assets in multiplayer apps

Ensure assets load on all connected peers before using them.

`class MultipeerConnectivityService`

A service that provides scene synchronization among all peers in a multipeer connectivity session.

`class NetworkCompatibilityToken`

An opaque token used to check the networking compatibility between two peers in a multipeer connection.

`enum Compatibility`

Indicates whether two devices running RealityKit are compatible and able to connect and sync scenes.

`protocol TransientComponent`

An interface for components that aren’t saved to file or cloned.

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.

☰ Audio

Create personalized and realistic spatial audio experiences.

☰ Videos

Present videos in your RealityKit experiences.

☰ Images

Present images and spatial scenes in your RealityKit experiences.