API Collection

Anchors

Lock virtual content to the real world.

Overview

RealityKit anchors all entities in a scene to an anchor target in the same scene, such as the ground, an image, or a position relative to its parent view. RealityKit provides several types you can use as the target for your anchors.

Topics

Anchoring components

`struct AnchoringComponent`

A component that anchors virtual content to a real world target.

`enum Target`

Defines the kinds of real world objects to which an anchor entity can be tethered.

`struct TrackingMode`

Options for how an entity tracks its target anchor.

`struct ARKitAnchorComponent`

A component that exposes the backing ARKit data of an anchored entity.

`class AnchorEntity`

An anchor that tethers entities to a scene.

`protocol HasAnchoring`

An interface that enables anchoring of virtual content to a real-world object in an AR scene.

Surface anchor characterization

`struct Alignment`

Defines the alignment of real-world surfaces to seek as targets.

`struct Classification`

Defines types of real-world surfaces to seek as targets.

Image and object tracking

`struct ImageAnchoringSource`

Defines the source of object anchoring target based on how it is created.

`struct ObjectAnchoringSource`

Defines the source of object anchoring target based on how it is created.

Hand tracking

{} Happy Beam

Leverage a Full Space to create a fun game using ARKit.

`struct HandLocation`

Defines the locations of tracked hands to look for.

`enum Chirality`

Defines the chirality of tracked hands to look for.

Spatial tracking

`class SpatialTrackingSession`

An object that incorporates spatial tracking capabilities into your RealityKit apps.

`struct Configuration`

A type for configuring the spatial tracking session.

`struct AnchorCapability`

A type that defines various anchor tracking capabilities.

`struct SceneUnderstandingCapability`

Defines how system behaviors use scene understanding.

`enum Camera`

Defines the camera feed the RealityView renders.

`struct UnavailableCapabilities`

A type that contains the unavailable capabilities of the current spatial tracking session.

Body and face tracking

{} Creating an App for Face-Painting in AR

Combine RealityKit's face detection with PencilKit to implement virtual face-painting.

{} Occluding virtual content with people

Cover your app's virtual content with people that ARKit perceives in the camera feed.

`struct BodyTrackingComponent`

A component for tracking people in an AR session.

`class BodyTrackedEntity`

An entity used to animate a virtual character in an AR scene by tracking a real person.

`protocol HasBodyTracking`

An interface that enables the animation of a virtual character by tracking a real person in AR.

Accessory tracking

{} Tracking a handheld accessory as a virtual sculpting tool

Use a tracked accessory with Apple Vision Pro to create a virtual sculpture.

Physics simulation space

`enum PhysicsSimulation`

Describes the physics simulation space of the entity and its descendants.

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.

☰ 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.

- Videos

  Present videos in your RealityKit experiences.

- Images

  Present images and spatial scenes in your RealityKit experiences.