API Collection

Models and meshes

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

Overview

Meshes are the building blocks for every visible geometric shape and model in RealityKit, including those that come from a USDZ file. Create primitive shapes by calling a `MeshResource` factory method, such as `generateBox(size:cornerRadius:)`, or define your own mesh by creating and configuring a `MeshDescriptor` or a `LowLevelMesh` instance.

Display meshes in your scene by creating a `ModelComponent` for each mesh, and add that component to an entity. You can also add `Material` instances to your mesh by adding them to the same `ModelComponent` instance.

Topics

Model display

`struct` `ModelComponent`

A component that contains a mesh and materials for the visual appearance of an entity.

`class` `MeshResource`

A high-level representation of a collection of vertices and edges that define a shape.

`class` `ModelEntity`

A representation of a physical object that RealityKit renders and optionally simulates.

Render configuration

`struct ModelSortGroupComponent`

A component that configures the rendering order for an entity's model.

`struct ModelSortGroup`

A group that you assign to multiple entities to tell the renderer what order and how to render the entities in the group.

`struct OpacityComponent`

A component that controls the opacity of an entity and its descendants.

`struct AdaptiveResolutionComponent`

A component that provides the suggested pixels per meter necessary to render an object.

`struct ModelDebugOptionsComponent`

A component that changes how RealityKit renders its entity to help with debugging.

`struct MeshInstancesComponent`

A component that performs GPU instancing on the model of the same entity.

Static meshes

`struct MeshDescriptor`

Defines a 3D mesh's structure and data.

`enum Primitives`

Indicates which primitive shape type a mesh applies to its vertex indices.

`enum Materials`

Updatable meshes

{} Integrating virtual objects with your environment

Create an immersive game using native anchor support, environmental blending, model manipulation, and mesh instance duplication.

{} 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.

Creating a plane with low-level mesh

Create a low-level mesh and set its vertex positions and normals to form a plane.

`class` **LowLevelMesh**

A container for vertex data that you can use to create and update meshes using your own format.

`struct` **Descriptor**

An object that describes the data format and layout of the buffers in a low-level mesh.

`struct` **Part**

An object that describes a range of primitives to display, and their material index.

`struct` **Layout**

An object that describes a set of attributes that share a buffer index, offset, and stride.

`struct` **Attribute**

An object that determines how to store vertex attribute data in memory and map it to RealityKit shader attributes.

`enum` **VertexSemantic**

Designates the intended usage of a vertex attribute.

`struct` **PartsCollection**

An object that holds a mutable collection low-level mesh parts.

`class` **LowLevelBuffer**

`class` **LowLevelInstanceData**

Bounding box retrieval

`struct` **BoundingBox**

An axis-aligned bounding box (AABB).

`struct` **OrientedBoundingBox**

Representation for an oriented bounding box. Uses a combination of an axis-aligned bounding box and a rotation vector around the centroid of the said axis-aligned bounding box to represent an oriented bounding box.

Text generation options

`struct GenerateTextOptions`

A type that determines the configuration for rendering text in 2D, before it is extruded.

`typealias Font`

A platform-specific type that represents a font for use in generating a text mesh.

2D path extrusion for 3D mesh creation

`struct ShapeExtrusionOptions`

A type that determines the extrusion, chamfering, and material assignment of an extruded shape.

`struct MaterialAssignment`

A type that determines the material assignments for each part of an extruded shape.

`enum ChamferMode`

Determines which part of the extrusion to chamfer.

`enum CurveStrokeResolution`

Designates the resolution at which a smooth curve is discretized.

`enum ExtrusionMethod`

The options that determine the way in which to extrude a swept shape in 3D.

Mesh description

`struct MeshBuffer`

Mesh buffer containing elements of any type.

`protocol MeshBufferContainer`

Conforming objects contain a table of mesh buffers.

`protocol MeshBufferSemantic`

A protocol that holds an identifier value for mesh buffers.

`enum MeshBuffers`

An object that holds the data for an model entity's mesh.

`struct AnyMeshBuffer`

Mesh buffer stored in the container.

`struct` `MeshInstanceCollection`

An object that holds a collection of mesh resource instances.

`struct` `MeshModelCollection`

An object that holds a collection of mesh models.

`struct` `MeshPartCollection`

An object that holds a collection of mesh parts.

Mesh skeletons

`struct` `Skeleton`

A skeleton consists of a hierarchy of joints. Each joint defines a coordinate space. Portions of a model may be thought of as having a position in a joint's local space.

`struct` `Joint`

A named joint in a `MeshResource.Skeleton`.

`struct` `MeshSkeletonCollection`

An object that holds a collection of skeletons used by a mesh resource.

`struct` `MeshJointInfluence`

A binding to a joint, which consists of the joint's index and the weight of that joint's influence on a vertex.

`struct` `JointInfluences`

A buffer of vertex-joint influences which bind the mesh part's vertices to a skeleton via a skinning deformation.

Mesh resource data

`struct` `Contents`

Value of the contents of the resource.

`struct` `Instance`

An object that transforms a model to a location.

`struct` `Model`

A model consists of a list of parts.

`struct` `Part`

A part of a model consisting of a single material.

Blend shape management

`struct BlendShapeWeightsComponent`

A component that provides access to the current weights associated with all blend shape meshes on an entity.

`class BlendShapeWeightsMapping`

A mapping of blend weights to the target meshes that those weights affect.

`struct BlendShapeWeights`

A set of animatable weight values that collectively represent the blending amounts for all the blend shapes' blend targets.

`struct BlendShapeWeightsData`

A structure that encapsulates the blend shape name, blend shape weights and the names of those weights to be stored by the blend shape weights set.

`struct BlendShapeWeightsSet`

A custom collection of named blend shape weights.

Entity compliance

`protocol HasModel`

An interface that provides meshes and materials to define the visual appearance of an entity.

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.

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.

Videos

Present videos in your RealityKit experiences.

Images

Present images and spatial scenes in your RealityKit experiences.