

[RealityKit](#) / Collision detection

API Collection

Collision detection

Determine when entities collide with each other or the environment.

Overview

RealityKit can automatically detect when two objects participating in the physics system collide with each other if both entities have a [CollisionComponent](#) with at least one *collision shape*. Because doing collision detection with complex 3D models can be computationally expensive, collision shapes are simpler, invisible shapes RealityKit uses to detect collision, as well as doing hit testing, ray casts, and convex shape casts.

Entities can participate in the scene simulation in two different modes: as a *rigid body* or as a *trigger*. A rigid body fully participates in the collision simulation. It affects the velocity and direction of other rigid body entities with which it collides. An entity with a rigid body *mode* of [PhysicsBodyMode.dynamic](#), other rigid body entities can affect its velocity and direction. A trigger entity doesn't have any impact on the other rigid bodies in the scene, but can trigger code or Reality Composer Pro behaviors when another rigid body entity overlaps it.

Turn an entity into a trigger by adding a [CollisionComponent](#) to it and setting its *mode* to [CollisionComponent.Mode.trigger](#).

Turn an entity into a *rigid body* by adding a [PhysicsBodyComponent](#) to the entity in addition to a [CollisionComponent](#). The [PhysicsBodyComponent](#) defines the physical properties of the entity, such as its mass and collision shape.

Note

RealityKit ignores an entity's collision component mode if the entity also has a [PhysicsBodyComponent](#). An entity can be a rigid body, or a trigger, but not both at the same time.

Topics

Collision shapes and groups

- `{}` Simulating physics with collisions in your visionOS app
Create entities that behave and react like physical objects in a RealityKit view.
- `{}` Configuring Collision in RealityKit
Use collision groups and collision filters to control which objects collide.

`struct CollisionComponent`

A component that gives an entity the ability to collide with other entities that also have collision components.

`enum Mode`

A mode that dictates how much collision data is collected for a given entity.

`class ShapeResource`

A representation of a shape.

`enum ShapeResourceError`

`struct CollisionGroup`

A bitmask used to define the collision group to which an entity belongs.

`struct CollisionFilter`

A set of masks that determine whether entities can collide during simulations.

`class TriggerVolume`

An invisible 3D shape that detects when objects enter or exit a given region of space.

Collision-related notifications

`enum CollisionEvents`

`struct Contact`

Events associated with collisions.

Ray casting

`struct CollisionCastHit`

A hit result of a collision cast.

`struct TriangleHit`

Information returned when ray intersects a triangle mesh.

`enum CollisionCastQueryType`

The kinds of ray and convex shape cast queries that you can make.

`struct PixelCastHit`

Entity compliance

`protocol HasCollision`

An interface used for ray casting and collision detection.

See Also

Physics simulation

☰ Simulations and motion

Simulate physical interactions between entities or systems.

☰ Force effects

Control the movement of virtual objects with forces.

☰ Physics joints and pins

Simulate joint physics that connect virtual objects.