Article

Handling different-sized objects in physics simulations

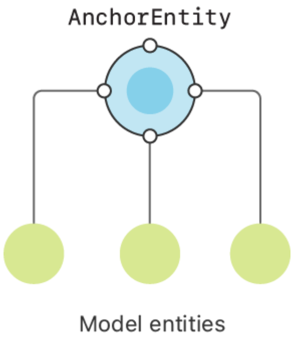
Set up a scene hierarchy for accurate physics simulations.

Overview

RealityKit provides a physics engine optimized to simulate a wide range of object sizes. Most RealityKit scenes contain objects within this range, so they simulate correctly without any extra work. However, objects smaller than one cubic centimeter fall outside the optimal size range. You need to set up scenes containing those tiny objects differently to ensure that they simulate realistically.

Use the default hierarchy for most scenes

Set up most RealityKit scenes — those without very tiny objects — so that you've parented all entities to a single anchor entity that acts as both the root of the visible scene and the origin of the physics simulation. With this setup, you can rotate and move the anchor in the scene without affecting the physics simulation because RealityKit applies anchor transforms after calculating the physics simulation.



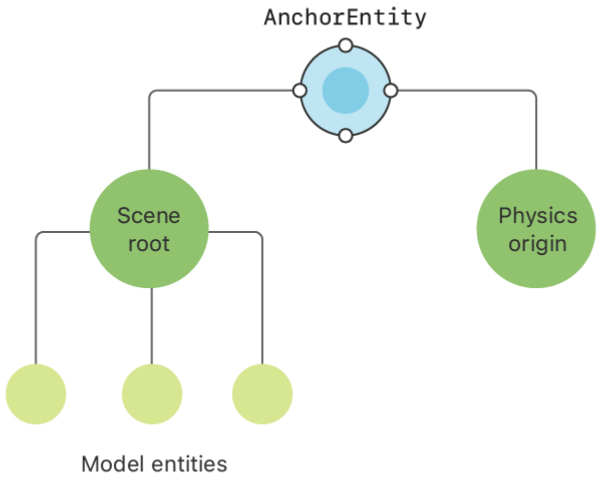
Reality Composer automatically sets up its scenes this way with no additional work needed. When building scenes programmatically or manually anchoring a Reality Composer scene into your ARView, set the physics origin to be the scene's AnchorEntity, like this:

```
let myAnchor = AnchorEntity()
arView.scene.addAnchor(myAnchor);
self.arView.physicsOrigin = myAnchor
```

When using this type of hierarchy, you can move, rotate, and scale objects after placing them in the ARView, and everything functions correctly, even if you make some objects smaller than a cubic centimeter after placing them in your scene.

Create a hierarchy for simulating tiny objects

Scenes that are authored with tiny objects, however, may not simulate optimally. To compensate, set up those scenes with a hierarchy that separates the scene root from the physics origin so you can change the scale of the physics origin independently of the scene.



To set up a scene like this, instead of using the AnchorEntity as both the scene root and the physics origin, add two empty entities as children of the anchor before adding any entities or a Reality Composer scene. One of the two entities functions as the scene root and the other acts as the physics origin. The following code shows how to set up a hierarchy like this:

```
// Create an anchor entity.
let myAnchor = AnchorEntity()

// Create the scene root.
let sceneRootEntity = Entity()
sceneRootEntity.name = "Scene Root"
myAnchor.addChild(sceneRootEntity)
```

```
// Create the physics origin.
let physicsRootEntity = Entity()
physicsOriginEntity.name = "Physics Origin"
myAnchor.addChild(physicsOriginEntity)

// Add the anchor and specify the physics origin.
arView.scene.addAnchor(myAnchor);
self.arView.physicsOrigin = physicsOriginEntity
```

When adding your scene to an ARView, instead of adding it directly as a child of the anchor entity, add it as a child of the scene root entity.

Since the scene root and physics origin are siblings in this hierarchy, you can apply separate transforms to each of them. Doing so allows you to adjust the scale of the physics calculations to increase the size of the tiny objects relative to the physics origin. To do that, scale the physics origin by the inverse of the desired amount of change. To calculate the inverse, divide `1.0` by the desired scale value. For example, if you want to scale up a physics simulation so that it behaves as if it's 10 times larger, apply a scale transform to the physics origin of `0.1`.

Here's an example that shows how to load a Reality Composer scene, add it to an ARView, and then adjust the physics origin so the scene's physics simulation behaves as if it were 10 times larger, allowing the tiny items in the scene to simulate in a more optimal manner:

```
if let myScene = try! Experience.loadMyScene() {
    sceneRootEntity.addChild(myScene)
}
physicsRootEntity.scale = SIMD3<Float>(0.1)
```

Adjusting the scale of the physics origin increases the size of the simulated entities *only* for the physics simulation. It has no effect on how RealityKit renders the entities.

See Also

Related Documentation

`var physicsOrigin: Entity?`

The entity that defines the origin of the scene's physics simulation.

Simulation setup

Designing scene hierarchies for efficient physics simulation

Configure your RealityKit scenes to avoid performance bottlenecks.

`struct PhysicsSimulationComponent`

A component that controls localized physics simulations.