Sample Code

Tracking specific points in world space

Retrieve the position and orientation of anchors your app stores in ARKit.

Download

visionOS 26.0+ | Xcode 26.0+

Overview

Use world anchors along with an ARKit session's `WorldTrackingProvider` to track points of interest in the world over time, as a person moves while wearing the device, and across device usage sessions. For example, someone might place a 3D object in a specific position on their desk and expect it to come back the next time they use the device.

Play

ARKit keeps track of a unique identifier for each world anchor your app creates and automatically places those anchors back in the space when the person returns to your app in the same location. A world tracking provider also provides the position of the device the person is wearing.

Start an ARKit session with world tracking

Use an `ARKitSession` configured for world tracking to start receiving updates on the world anchors your app places. The following shows updates to world anchors your app previously registered using the `addAnchor(_:)` method:

```
let session = ARKitSession()
let worldInfo = WorldTrackingProvider()

Task {
    try await session.run([worldInfo])

    for await update in worldInfo.anchorUpdates {
        switch update.event {
        case .added, .updated:
            // Update the app's understanding of this world anchor.
            print("Anchor position updated.")
        case .removed:
            // Remove content related to this anchor.
            print("Anchor position now unknown.")
    }
}
```

Important

If a person repositions the current space — for example, by holding down the Digital Crown — world anchor updates begin updating their position relative to the new world origin. For example, a world anchor placed on a table still reports information about the table's position, but those positions are relative to the updated world origin.

Create and add world anchors

You can create world anchors for any point of interest in your app's world coordinate system once you've started a world tracking ARKit session. For example, you might track that a person placed

an item at a particular offset from a desk in their space:

```
let anchor = WorldAnchor(originFromAnchorTransform: deskPlane.originFromAnchorTransf
try await worldInfo.addAnchor(anchor)
```

Once you add a world anchor to your app's tracking provider using the `addAnchor(_:)` method, the `anchorUpdates` sequence in the current session and future runs of your app provides updates to the current position of that new world anchor.

Persist world anchors across sessions

The only information ARKit persists about the world anchors in your app is their `UUID` — a `WorldAnchor` instance's `id` property — and pose in a particular space. It's your app's responsibility to persist additional information, such as the meaning of each anchor. For example, you might save local data about a custom 3D lamp model that a person placed on their desk.

As a person moves from town-to-town or room-to-room, your app won't receive all of the world anchor updates from each place someone used your app. Instead, the `anchorUpdates` sequence only provides world anchors for nearby objects.

Track the device position in the world

Use the Compositor Services framework and the `WorldTrackingProvider` class's `queryDeviceAnchor(atTimestamp:)` method to get low-latency information about the current and future-predicted pose of the person's device in world space. For more information, see Drawing fully immersive content using Metal.

See Also

ARKit

{} Happy Beam

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

Setting up access to ARKit data

Check whether your app can use ARKit and respect people's privacy.

{} Incorporating real-world surroundings in an immersive experience

Create an immersive experience by making your app's content respond to the local shape of the world.

{} Placing content on detected planes

Detect horizontal surfaces like tables and floors, as well as vertical planes like walls and doors.

Tracking preregistered images in 3D space

Place content based on the current position of a known image in a person's surroundings.

{} Exploring object tracking with ARKit

Find and track real-world objects in visionOS using reference objects trained with Create ML.

{} Object tracking with Reality Composer Pro experiences

Use object tracking in visionOS to attach digital content to real objects to create engaging experiences.

{} Building local experiences with room tracking

Use room tracking in visionOS to provide custom interactions with physical spaces.

{} Placing entities using head and device transform

Query and react to changes in the position and rotation of Apple Vision Pro.