Documentation

visionOS / Tracking preregistered images in 3D space

Article

Tracking preregistered images in 3D space

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

Overview

Use ARKit’s support for tracking 2D images to place 3D content in a space. ARKit provides updates to the image’s location as it moves relative to the person. If you supply one or more reference images in your app’s asset catalog, people can use a real-world copy of that image to place virtual 3D content in your app. For example, if you design a set of movie posters and provide those assets to people in the form of real-world environments, they can view the trailer for the movie in a fully immersive experience.

The following example tracks a set of images loaded from an app’s asset catalog:

```
let session = ARKitSession()
let imageInfo = ImageTrackingProvider(
    referenceImages: ReferenceImage.loadReferenceImages(inGroupNamed: "playingcard-
)

if ImageTrackingProvider.isSupported {
    Task {
        try await session.run([imageInfo])
        for await update in imageInfo.anchorUpdates {
            updateImage(update.anchor)
        }
    }
}
```

```
func updateImage(_ anchor: ImageAnchor) {
    if imageAnchors[anchor.id] == nil {
        // Add a new entity to represent this image.
        let entity = ModelEntity(mesh: .generateSphere(radius: 0.05))
        entityMap[anchor.id] = entity
        rootEntity.addChild(entity)
    }

    if anchor.isTracked {
        entityMap[anchor.id]?.transform = Transform(matrix: anchor.originFromAnchorT
    }
}
```

If you know the real-world dimensions of the images you're tracking, use the `physicalSize` property to improve tracking accuracy. The `estimatedScaleFactor` property provides information about how the scale of the tracked image differs from the expected physical size you provide.

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 specific points in world space

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

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.