Class

ARAnchor

An object that specifies the position and orientation of an item in the physical environment.

iOS 11.0+ | iPadOS 11.0+ | Mac Catalyst 13.1+

```
class ARAnchor
```

Mentioned in

- Displaying an AR Experience with Metal
- Providing 2D Virtual Content with SpriteKit
- Providing 3D Virtual Content with SceneKit

Overview

To track the static positions and orientations of real or virtual objects relative to the camera, create anchor objects and use the `add(anchor:)` method to add them to your AR session.

> **Tip**
>
> Adding an anchor to the session helps ARKit to optimize world-tracking accuracy in the area around that anchor, so that virtual objects appear to stay in place relative to the real world. If a virtual object moves, remove the corresponding anchor from the old position and add one at the new position.

Some ARKit features automatically add special anchors to a session. World-tracking sessions can add `ARPlaneAnchor`, `ARObjectAnchor`, and `ARImageAnchor` objects if you enable the

corresponding features; face-tracking sessions add `ARFaceAnchor` objects.

Subclassing Notes

In addition to creating your own `ARAnchor` instances to track the real-world positions of your virtual content, you can also subclass `ARAnchor` to associate custom data with anchors you create. Ensure that your anchor classes behave correctly when ARKit updates frames or saves and loads anchors in an `ARWorldMap`:

- Anchor subclasses must fullfill the requirements of the `ARAnchorCopying` protocol. ARKit calls `init(anchor:)` (on a background thread) to copy instances of your anchor class from each `ARFrame` to the next. Your implementation of this initializer should copy the values of any custom properties your subclass adds.
- Anchor subclasses must also adopt the `NSSecureCoding` protocol. Override `encode(with:)` and doc://com.apple.documentation/documentation/oslog/oslogentry/init(coder:) to save and restore the values your subclass' custom properties when ARKit saves and loads them in a world map.
- Anchors are considered equal based on their `identifier` property.
- Only anchors that do not adopt `ARTrackable` are included when you save a world map.

Topics

Creating Anchors

`init(transform: simd_float4x4)`

Creates a new anchor object with the specified transform.

`init(name: String, transform: simd_float4x4)`

Creates a new anchor object with the specified transform and a descriptive name.

`var name: String?`

A descriptive name for the anchor.

Tracking Anchors

`var identifier: UUID`

A unique identifier for the anchor.

`var sessionIdentifier: UUID?`

The unique identifier of the session that owns this anchor.

`var transform: simd_float4x4`

A matrix encoding the position, orientation, and scale of the anchor relative to the world coordinate space of the AR session the anchor is placed in.

Relationships

Inherits From

`NSObject`

Inherited By

`ARAppClipCodeAnchor`
`ARBodyAnchor`
`AREnvironmentProbeAnchor`
`ARFaceAnchor`
`ARGeoAnchor`
`ARImageAnchor`
`ARMeshAnchor`
`ARObjectAnchor`
`ARParticipantAnchor`
`ARPlaneAnchor`

Conforms To

`ARAnchorCopying`
`CVarArg`
`CustomDebugStringConvertible`
`CustomStringConvertible`
`Equatable`
`Hashable`
`NSCoding`
`NSCopying`
`NSObjectProtocol`
`NSSecureCoding`
`Sendable`

`SendableMetatype`

See Also

iOS

Verifying Device Support and User Permission

Check whether your app can use ARKit and respect user privacy at runtime.

`class` `ARSession`

The object that manages the major tasks associated with every AR experience, such as motion tracking, camera passthrough, and image analysis.

ARKit in iOS

Integrate iOS device camera and motion features to produce augmented reality experiences in your app or game.