Class

ARSession

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

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

```
class ARSession
```

Mentioned in

- Displaying an AR Experience with Metal
- Managing Session Life Cycle and Tracking Quality

Overview

An `ARSession` object coordinates the major processes that ARKit performs on your behalf to create an augmented reality experience. These processes include reading data from the device's motion sensing hardware, controlling the device's built-in camera, and performing image analysis on captured camera images. The session synthesizes all of these results to establish a correspondence between the real-world space the device inhabits and a virtual space where you model AR content.

Create a Session

Every AR experience requires an `ARSession`. If you implement a custom renderer, you instantiate the session yourself.

```
let session = ARSession()
session.delegate = self
```

If you use one of the standard renderers (like ARView, ARSCNView, or ARSKView), the renderer creates a session object for you. When you want to interact with your app's session, you access it on your app's renderer.

```
let session = myView.session
```

Run a Session

Running a session requires a configuration. Subclasses of ARConfiguration determine how ARKit tracks a device's position and motion relative to the real world, and thus it determines the kinds of AR experiences you create. For example, ARWorldTrackingConfiguration enables you to augment the user's view of the world around them though the device's back camera.

Topics

Configuring and running a session

`func run(ARConfiguration, options: ARSession.RunOptions)`

Starts AR processing for the session with the specified configuration and options.

`var identifier: UUID`

A unique identifier of the running session.

`struct RunOptions`

Options for transitioning an AR session's current state when you change its configuration.

`var configuration: ARConfiguration?`

An object that defines motion and scene tracking behaviors for the session.

`func pause()`

Pauses processing in the session.

Responding to events

`var delegate: (any ARSessionDelegate)?`

An object you provide to receive captured video images and tracking information, or to respond to changes in session status.

`var delegateQueue: dispatch_queue_t?`

The dispatch queue through which the session calls your delegate methods.

`protocol ARSessionDelegate`

Methods you can implement to receive captured video frame images and tracking state from an AR session.

`protocol ARSessionObserver`

Methods you can implement to respond to changes in the state of an AR session.

Managing anchors

`func add(anchor: ARAnchor)`

Adds the specified anchor to be tracked by the session.

`func remove(anchor: ARAnchor)`

Removes the specified anchor from tracking by the session.

Saving or sharing state

`func getCurrentWorldMap(completionHandler: (ARWorldMap?, (any Error)?) -> Void)`

Returns an object encapsulating the world-tracking session's space-mapping state and set of anchors.

Recording and Replaying AR Session Data

Record an AR session in Reality Composer and replay it in your ARKit app.

Scanning 3D objects

`func createReferenceObject(transform: simd_float4x4, center: simd_float3, extent: simd_float3, completionHandler: (ARReferenceObject?, (any Error)?) -> Void)`

Creates a reference object (for 3D object detection) from the specified region of the session's world space.

Updating the world origin

```
func setWorldOrigin(relativeTransform: simd_float4x4)
```

Changes the basis for the AR world coordinate space using the specified transform.

Finding real-world surfaces

Cast a ray from a point on the screen to find intersections with real-world surfaces.

```
func raycast(ARRaycastQuery) -> [ARRaycastResult]
```

Checks once for intersections between a ray and real-world surfaces.

```
func trackedRaycast(ARRaycastQuery, updateHandler: ([ARRaycastResult])
-> Void) -> ARTrackedRaycast?
```

Repeats a ray-cast query over time to notify you of updated surfaces in the physical environment.

Converting local coordinates to geographic coordinates

```
func getGeoLocation(forPoint: simd_float3, completionHandler: (
CLLocationCoordinate2D, CLLocationDistance, (any Error)?) -> Void)
```

Converts a position in the framework's local coordinate system to latitude, longitude and altitude.

Accessing the camera frame

```
var currentFrame: ARFrame?
```

The most recent still frame captured by the active camera feed, including ARKit's interpretation of it.

```
class ARFrame
```

A video image captured as part of a session with position-tracking information.

```
func captureHighResolutionFrame(completion: (ARFrame?, (any Error)?) ->
Void)
```

Requests a frame outside of the normal frequency that contains a high-resolution captured image.

Managing collaboration

```
func update(with: ARSession.CollaborationData)
```

Updates your session with information about the physical environment that is collected by another user.

`class CollaborationData`

An object that holds information that a user has collected about the physical environment.

Providing a session

`protocol ARSessionProviding`

An object that provides a session.

Instance Methods

```
func captureHighResolutionFrame(using: AVCapturePhotoSettings?,
completion: (ARFrame?, (any Error)?) -> Void)
```

Requests a single, high resolution frame to be captured.

Relationships

Inherits From

`NSObject`

Conforms To

`CVarArg`
`CustomDebugStringConvertible`
`CustomStringConvertible`
`Equatable`
`Hashable`
`NSObjectProtocol`

See Also

iOS

Verifying Device Support and User Permission

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

`class` ARAnchor

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

ARKit in iOS

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