Article

Implementing object tracking in your visionOS app

Create engaging interactions by training models to recognize and track real-world objects in your app.

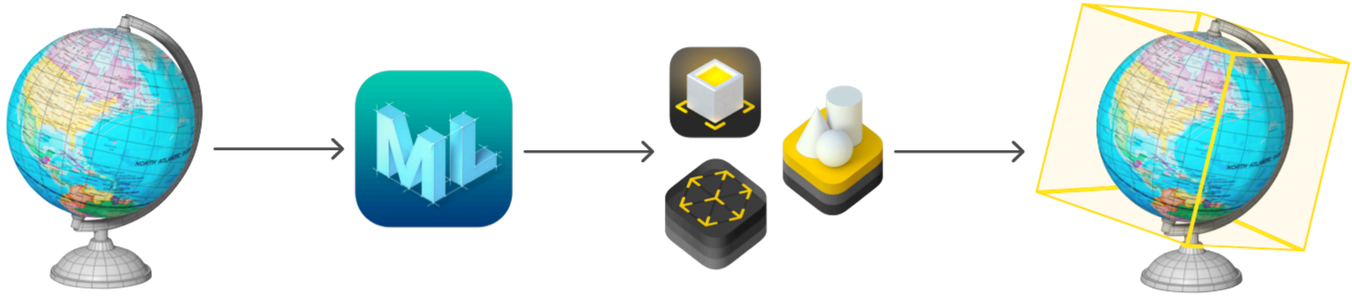
Overview

When you implement object tracking in your visionOS app, you can seamlessly integrate real-world objects in people's surroundings to enhance their immersive experiences. By tracking the 3D position and orientation of an object, or several objects, your app can augment them with virtual content.

You can use object tracking to provide virtual interactions with objects in a person's surroundings, such as:

- Guiding someone through using an item's features, reading about its history, or learning about its behaviors when they look at it in their surroundings.
- Helping people troubleshoot issues with household items and appliances with a virtual manual.
- Creating an immersive storytelling experience to make collectables and toys come to life.

To integrate object tracking into your app, you start with a 3D model of a physical object, train a machine learning model in Create ML with that 3D model asset to obtain a reference object file, and then use the resulting reference object file to track the physical object in your app. The reference object file is a file format with a `.referenceobject` extension, specifically for object tracking in visionOS.



Implementing object tracking requires an Apple Vision Pro with visionOS 2 or later, and a Mac with Apple silicon and macOS 15 or later for the machine learning training in Create ML.

Ensure your objects are suitable for object tracking

Object tracking performs optimally for a specific set of object characteristics. For object tracking to work best in your app, make sure your object is rigid, nonsymmetrical, and stationary.

Rigid
: Select an object that maintains its shape and appearance during tracking. For example, a pair of scissors is challenging to track because it changes shape while a person uses it.

Nonsymmetrical
: Select an object with a nonsymmetrical shape or texture, so that when you rotate the object, it doesn't have the same appearance from different angles. For instance, a globe has a symmetrical shape, but has a nonsymmetrical texture on all sides, making it a suitable object. In contrast, a styrofoam cup has the same appearance on all sides when you rotate it, making it challenging to track.

Stationary
: Select an object that's mostly stationary in a person's surroundings. If you're tracking a moving object, there can be a delay in following its position. For example, a pickleball racket constantly moves in different directions while a person plays with it, making it challenging to track.

Obtain a 3D model of your object

You use Create ML to begin the machine learning training to obtain your reference object file. Create ML requires a 3D model asset in the USDZ file format that represents your real-world object. You can obtain your 3D model using computer-aided design (CAD) software to accurately model an object's geometry and apply physically based rendering (PBR) materials to it, and save it in the USDZ file format. Using this method, the 3D model can realistically represent objects that consist of multiple parts made from different materials, like glass, metal, plastic, wood, and other common materials. This method is helpful for capturing objects that are entirely or partly transparent, shiny, or reflective. The better the 3D model represents the appearance of the physical object, the better the quality of tracking is in visionOS.

Another way to create your 3D model is by using the Object Capture feature in the Reality Composer app in iOS or iPadOS. You can use your iPhone or iPad to capture images of an object, and then save the USDZ file to import into your app. For more information about using the Object Capture feature to create a 3D model, see Meet Object Capture for iOS and Scanning objects using Object Capture.

Before beginning the training process in Create ML with the 3D model asset, keep the following guidelines in mind to ensure it works well for object tracking in visionOS:

- Ensure the 3D model is as photorealistic as possible — essentially a digital twin of your real-world object.
- Ensure the scale of the 3D model is as precise as possible and matches its specified units. If the scale doesn’t match the real-world object, the augmentation appears offset in the viewing direction, and may appear either in front of or behind the object.

> Note
>
> While training the machine learning model with the 3D model asset, Create ML ignores any animations, virtual cameras, and lights within the asset, treating them as static.

Train a machine learning model with the 3D model asset in Create ML

Object tracking requires a reference object file to track the spatial location and orientation of the corresponding real-world object. You use Create ML to train a machine learning model to create a reference object file unique to your object. The training of machine learning models with your 3D asset and the creation of the reference object file both run locally on your Mac. You can either train a model with the Create ML developer tool that comes with Xcode, or with the Create ML command-line tool.

The following are the steps to train a model in the Create ML app:

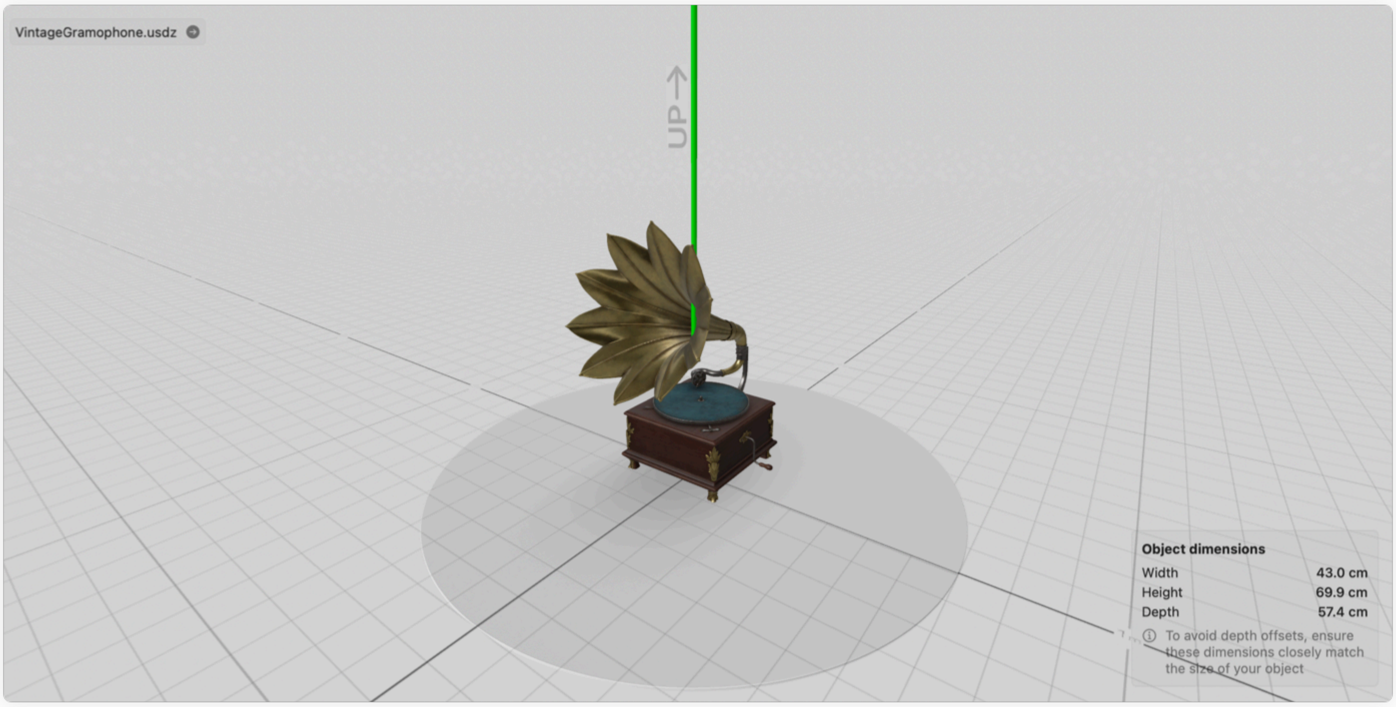
1. Open Xcode and choose Xcode > Open Developer Tool > Create ML.
2. Click the New Document button in the Open Project dialog that Create ML presents at launch.
3. In the Choose a Template dialog, select Object Tracking template, which is in the Spatial category, and click Next.

4. Give your project a name and, optionally, enter additional information about the model, and click Next.

5. Select a location for your project and click Create.

6. Create ML opens a training configuration view with an empty 3D viewport. Drag the USDZ file of your 3D model asset into the 3D viewport.

The 3D viewport is an interactive space where you can view your 3D model asset from different angles. After it appears in the viewport, check the appearance of the 3D model asset and confirm that it matches the absolute dimensions of your real-world object. Also make sure that the dimensions of the 3D model asset at the bottom right of the viewport match the actual dimensions of your object. If the scale doesn’t match, one option is to use Reality Composer Pro to rescale the 3D model and then add the adjusted USDZ file to Create ML.

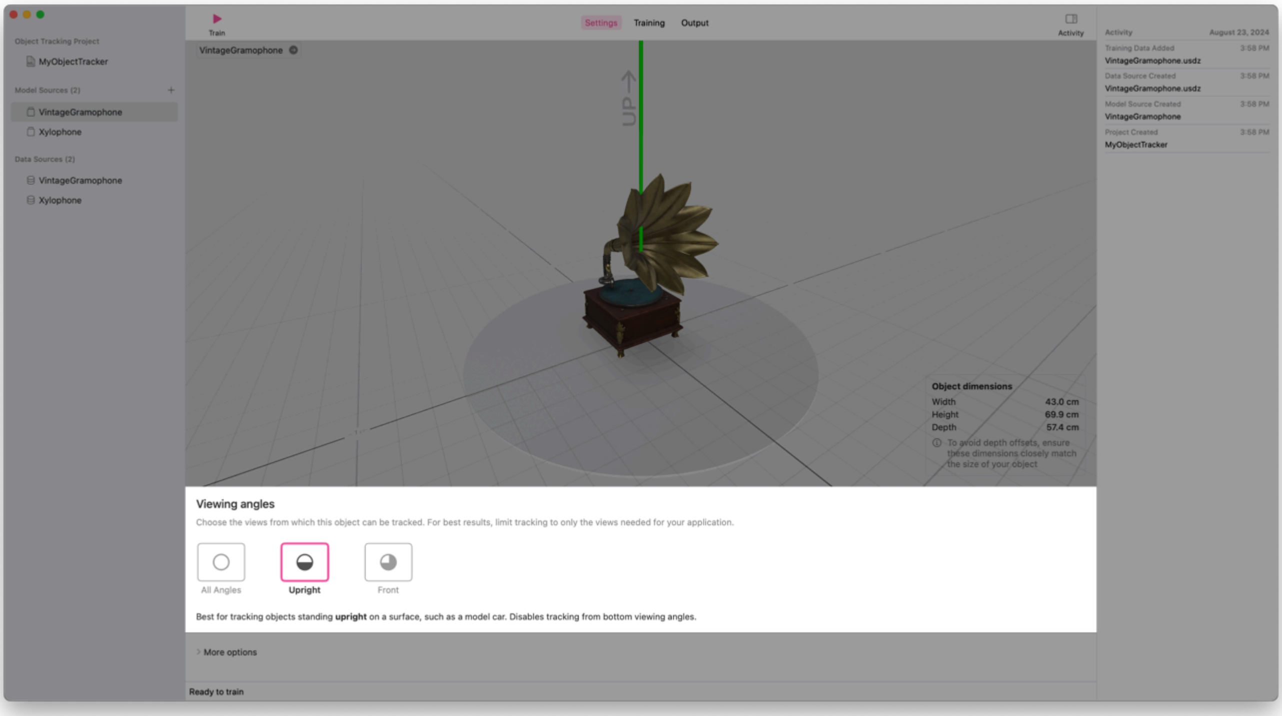


Object dimensions

Width	43.0 cm
Height	69.9 cm
Depth	57.4 cm

ⓘ To avoid depth offsets, ensure these dimensions closely match the size of your object

The next step is to select the best viewing angle for your real-world object. Consider how people view and interact with the object in your app, and decide which angle you need for tracking it. The “Viewing angles” setting appears below the 3D viewport, and has three viewing angles you can use: All Angles, Upright, or Front. It’s important to choose the best option for your object.



Activity	August 23, 2024
Training Data Added	3:58 PM
VintageGramophone.usdz	
Data Source Created	3:58 PM
VintageGramophone.usdz	
Model Source Created	3:58 PM
VintageGramophone	
Project Created	3:58 PM
MyObjectTracker	

Object dimensions

Width	43.0 cm
Height	69.9 cm
Depth	57.4 cm

ⓘ To avoid depth offsets, ensure these dimensions closely match the size of your object

Viewing angles

Choose the views from which this object can be tracked. For best results, limit tracking to only the views needed for your application.

○ All Angles [✓ ◒ **Upright**] ◔ Front

Best for tracking objects standing **upright** on a surface, such as a model car. Disables tracking from bottom viewing angles.

› More options

Ready to train

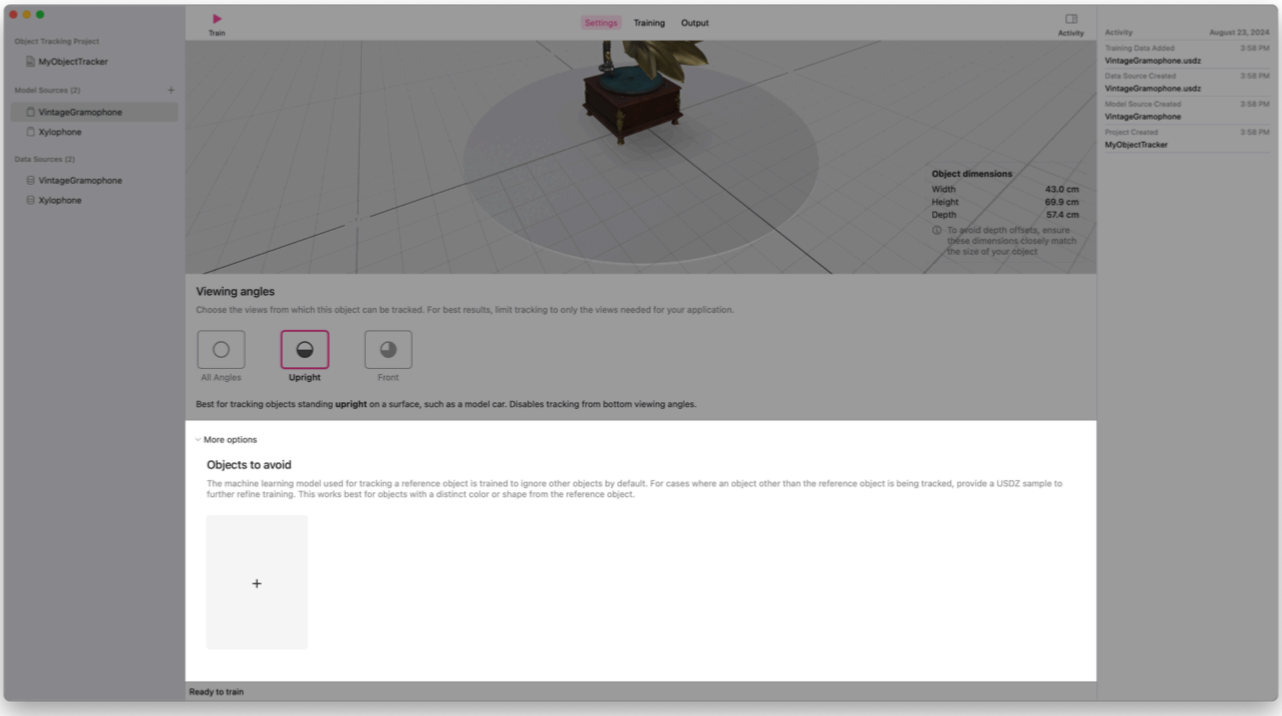
The All Angles option includes views from every angle. It works best for tracking objects that have a distinct and unique appearance from all sides, such as a patterned Christmas ornament that people see from all sides as it hangs on a tree.

The Upright option works only for tracking objects that stand upright on a surface, such as a microscope that sits on a counter and stays in the same position as people interact with it. This option disables tracking from the bottom viewing angle.

The Front option works only for tracking objects that stand upright on a surface where the back of the object isn’t visible, such as a coffee machine that sits on a counter while people operate it from the front. This option disables tracking from both the bottom and rear viewing angles.

> Note
>
> Only choose the All Angles option if you want to track your object from all sides. The more restricted the viewing angle is, the more accurate the object tracking is in visionOS.

If there’s an object in a person’s surroundings that’s similar to the object you want to track, the object-tracking feature might recognize it and track it instead of your object. To prevent this from happening, add the similar object as a negative example when training the machine learning model with your reference object. Below the 3D viewport, choose More Options > Objects to avoid. Use this section to add USDZ samples of similar items to ensure the machine learning model doesn’t identify them as the object you want to track.



Activity August 23, 2024
Training Data Added 3:58 PM
VintageGramophone.usdz
Data Source Created 3:58 PM
VintageGramophone.usdz
Model Source Created 3:58 PM
VintageGramophone
Project Created 3:58 PM
MyObjectTracker

Object dimensions

Width	43.0 cm
Height	69.9 cm
Depth	57.4 cm

ⓘ To avoid depth offsets, ensure these dimensions closely match the size of your object

Viewing angles

Choose the views from which this object can be tracked. For best results, limit tracking to only the views needed for your application.

○ All Angles [✓ Upright] Front

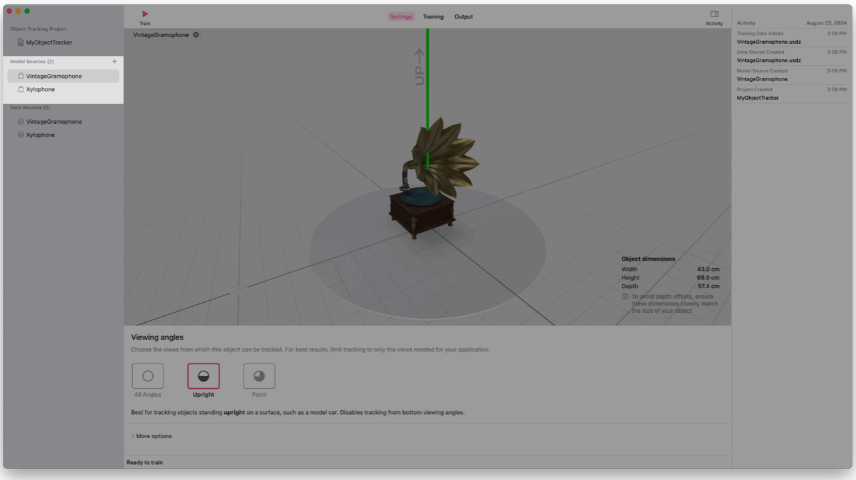
Best for tracking objects standing **upright** on a surface, such as a model car. Disables tracking from bottom viewing angles.

Objects to avoid

The machine learning model used for tracking a reference object is trained to ignore other objects by default. For cases where an object other than the reference object is being tracked, provide a USDZ sample to further refine training. This works best for objects with a distinct color or shape from the reference object.

Ready to train

Create ML supports training multiple machine learning models in the same object-tracking project. In the Model Sources section in the left pane, you can click the Add button (+) to add more 3D model assets to your Create ML project. Use this feature to track multiple objects in your app at the same time.



Activity August 23, 2024
Training Data Added 3:58 PM
VintageGramophone.usdz
Data Source Created 3:58 PM
VintageGramophone.usdz
Model Source Created 3:58 PM
VintageGramophone
Project Created 3:58 PM
MyObjectTracker

Object dimensions

Width	43.0 cm
Height	69.9 cm
Depth	57.4 cm

ⓘ To avoid depth offsets, ensure these dimensions closely match the size of your object

Viewing angles

Choose the views from which this object can be tracked. For best results, limit tracking to only the views needed for your application.

○ All Angles [✓ Upright] Front

Best for tracking objects standing **upright** on a surface, such as a model car. Disables tracking from bottom viewing angles.

Ready to train

Note

You can track up to 10 different reference objects simultaneously without an impact to performance.

After inspecting your 3D model asset and configuring the training settings, click Train to begin the training process. A progress bar indicates the amount of time until the machine learning training is complete. The machine learning training can take a few hours, depending on the configuration of your Mac. A more advanced processor and additional RAM significantly improve the training time.

Train your assets with the Create ML command-line tool

Starting with Xcode 26, which requires macOS 15.4 or later, you can train a machine learning model with your 3D asset by running the Create ML developer tool from a command line prompt. With an asset in the USDZ file format, you can use the tool to train the asset and get a reference object file to use for object tracking.

The Create ML command-line tool automates object tracking tasks in your workflow, like using your scripts and cloud-based parallel setups to run the training process. You can also use the tool when you need to automate training a large number of objects while you continue to work on other tasks.

You need to have Xcode command-line tools installed before using the Create ML tool, which you can check by running the following command:

```
% xcode-select -p
```

Note

You can use the Create ML command line tool if the command's output refers to a `Contents/Developer` directory in Xcode or to the `/Library/Developer/CommandLineTools` directory. If the command returns an error that indicates that Xcode or the command-line tools can't find the directory, install the command-line tools package by running the `xcode-select --install` command.

Begin the training process by invoking the Create ML command-line tool with the `xcrun` command. You need to modify the example below to provide the locations on your system for the commands inputs and outputs.

```
% xcrun createml objecttracker -s source.usdz -o tracker.referenceobject
```

The system uses the `xcrun` prefix to locate the path of the training tool in the Xcode command-line tools. The `-s` flag points to the source path for the 3D asset of the physical object you want to train, and the -o flag points to the output path to store the final trained reference object file. Before running this command, update it to include the name of the source and output of your object.

After you run the tool, it starts training your object. Use the —`help` option for more information on training and topics like viewing angles, objects to avoid, and redirection to alternative pipes:

```
% xcrun createml objecttracker --help
```

Export the reference object file

When training is complete, Create ML provides the reference object file for you to use in your app. Click the Output tab and save the resulting reference object file.

The reference object file contains the machine learning model you trained, packaged with the 3D model asset, in the USDZ file format. You can use the USDZ file for visualizing the tracking quality by rendering it as an overlay on the real-world object, and as a guide for adding immersive effects. The USDZ file may take up a lot of space in your app if your 3D model asset is large, so you can remove it from the reference object file if you need to optimize space.

You use the Reference Object Compiler in Xcode to remove the USDZ data from the reference object file during the build process. Select your project in Xcode, click the Build Settings tab, and enable the Strip USDZ Files from Reference Object option. This setting contains the `REFERENCEOBJECT_STRIP_USDZ` build flag. The default setting of the flag is No, so Xcode copies any reference object files you add to the project as-is unless you change the setting.

Integrate the reference object file into your app

After you generate the reference object file, you can set up object tracking in your app using Reality Composer Pro, RealityKit, or ARKit. For more information about each of these methods, see Using a reference object with Reality Composer Pro, Using a reference object with RealityKit, and Using a reference object with ARKit.

> Note
>
> Object tracking works only in an `ImmersiveSpace` within Xcode. Attempting to use object tracking in a window or volume results in a silent failure.

For more information about object tracking, see Explore object tracking for visionOS. For an example of using ARKit for object tracking, see Exploring object tracking with ARKit.

Topics

Object tracking within an app

Using a reference object with Reality Composer Pro

Import a reference object file to track a real-world object in your visionOS app.

Using a reference object with RealityKit

Import a reference object file to track a real-world object in your visionOS app.

Using a reference object with ARKit

Import a reference object file and track a real-world object in your visionOS app.

See Also

RealityKit and Reality Composer Pro

Reality Composer Pro

Build, create, and design 3D content for your RealityKit apps.

Petite Asteroids: Building a volumetric visionOS game

Use the latest RealityKit APIs to create a beautiful video game for visionOS.

BOT-anist

Build a multiplatform app that uses windows, volumes, and animations to create a robot botanist’s greenhouse.

Swift Splash

Use RealityKit to create an interactive ride in visionOS.

Diorama

Design scenes for your visionOS app using Reality Composer Pro.

Building an immersive media viewing experience

Add a deeper level of immersion to media playback in your app with RealityKit and Reality Composer Pro.

Enabling video reflections in an immersive environment

Create a more immersive experience by adding video reflections in a custom environment.

Combining 2D and 3D views in an immersive app

Use attachments to place 2D content relative to 3D content in your visionOS app.

Understanding the modular architecture of RealityKit

Learn how everything fits together in RealityKit.

Using transforms to move, scale, and rotate entities

Learn how to use Transforms to move, scale, and rotate entities in RealityKit.

Capturing screenshots and video from Apple Vision Pro for 2D viewing

Create screenshots and record high-quality video of your visionOS app and its surroundings for app previews.

{} Placing entities using head and device transform

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