XR Flood – Build instructions

[All platforms 1](#_Toc192610028)

[Requirements 1](#_Toc192610029)

[Getting started 1](#_Toc192610030)

[XR Flood client – Build instructions 2](#_Toc192610031)

[All platforms 2](#_Toc192610032)

[Building for iOS 2](#_Toc192610033)

[Requirements 2](#_Toc192610034)

[Xcode build instructions 3](#_Toc192610035)

[Building for visionOS 5](#_Toc192610036)

[Requirements 5](#_Toc192610037)

[Connecting with a Vision Pro device 6](#_Toc192610038)

[Problems pairing Vision Pro 6](#_Toc192610039)

[Building for Android 7](#_Toc192610040)

[XR Flood server – Build instructions 8](#_Toc192610041)

[All platforms 8](#_Toc192610042)

[Deprecated packages 8](#_Toc192610043)

[Building for Windows, Mac 8](#_Toc192610044)

## All platforms

### Requirements

* Unity 6000.0.26f1 or higher

### Getting started

Get [the project](https://github.com/IMD-Lab/XR-Flood-DigiTwin) from GitHub. You can do this by downloading a ZIP or by using git clone; for either, use the main branch with the tag V1.0.

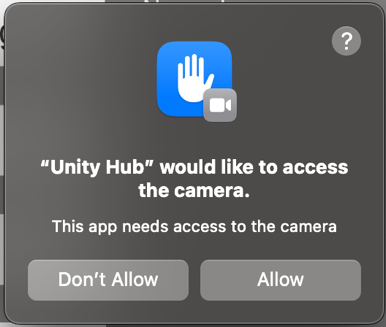
Please find all supporting documentation in the [/docs](https://github.com/IMD-Lab/XR-Flood-DigiTwin/tree/main/docs) folder:

* [Configuration schema](https://github.com/IMD-Lab/XR-Flood-DigiTwin/raw/refs/heads/main/docs/XR%20Flood%20Configuration%20Schema.docx).
* Open index.html in the [/docs/gen/html](https://github.com/IMD-Lab/XR-Flood-DigiTwin/tree/main/docs/gen/html) folder.

# XR Flood client ­– Build instructions

## All platforms

Open [Unity Hub](https://unity.com/download) and press Add > Add project from disk. Navigate to where you downloaded the project /client/MagicBook client and press Open. If the necessary Unity editor version and/or components are not installed yet, you can use Unity Hub to install them. It might be a good idea to install iOS, visionOS and macOS component for the selected Unity version at the same time, as well as development IDE such as Visual Studio. The first time you open the project might take a significant amount of time.

After opening the project, there will be some warnings and a Shader error in the Console panel. These are due to the standard build platform in Unity: Windows, Mac, Linux. You can safely ignore them. In the Project panel, browse to Assets/Scenes/ and open MainScene. Press the Play button. If you’re developing on MacOS with an available webcam, then the first time playing you will get a dialog asking for permission to use your webcam. Press Allow. By default, when the Unity active platform is not supported by XR Flood client, it will open the scene CardboardMain. Please proceed to the section specific to the platform you are building for.

## Building for iOS

### Requirements

* Unity iOS module > iOS Build Support
* iOS 11.0 or newer
* iOS device with A9 processor or newer
* [Xcode](https://developer.apple.com/xcode/) 14.0 or newer

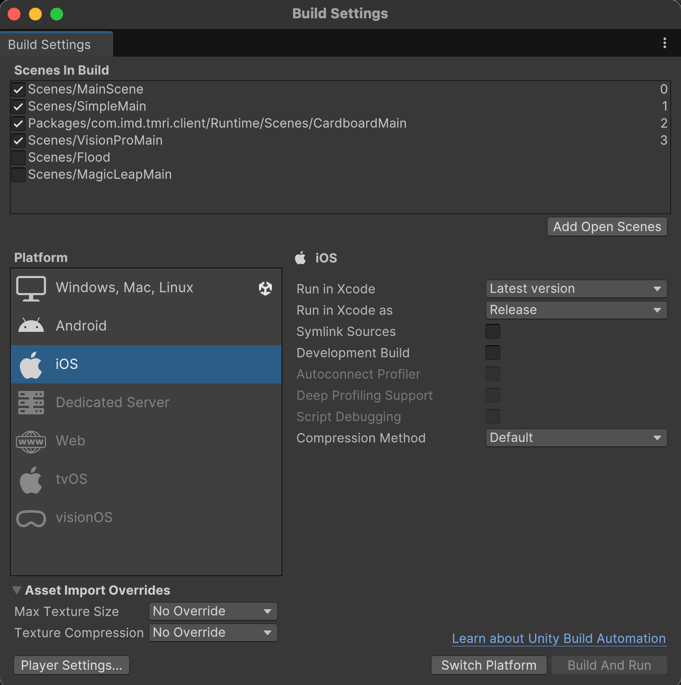
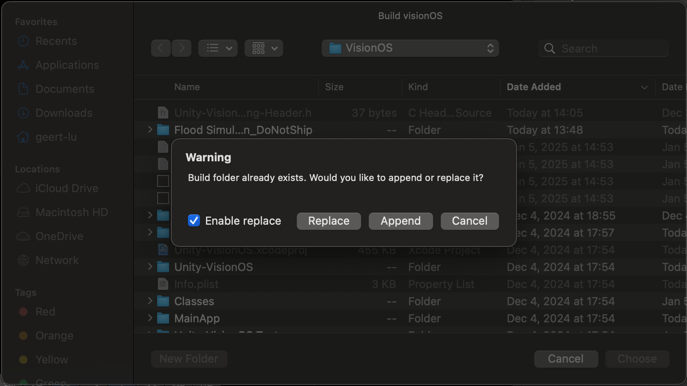
In Unity, go to Window > Panels > Build Settings. Under Platform, select iOS and press Switch Platform. Switching platforms might take some time.

Figure . iOS build settings.

In the Project panel, browse to Assets/Scenes/ and open MainScene. Press the Play button to try the client app operation in the Unity editor. Before building, make sure there are no errors in the console.

In the Build Settings panel, make sure the iOS configuration is set up like Figure 1.

If this is the first time building:

1. Press Build.
2. A save dialog appears. Create folders Build/iOS/ in the MagicBook client directory.
3. Press Choose. The build process will take some time.
4. Browse to the build directory and open Unity-iPhone.xcodeproj with Xcode.

If this is not the first time:

1. Press Build and Run.
2. A save dialog appears. Browse to your Build/iOS directory.
3. You have the option to Replace or Append. Press Append for faster building times.
4. Xcode automatically opens the project and starts building.

When Unity is finished, verify the following message in the console: Build completed with a result of 'Succeeded' in x seconds.

### Xcode build instructions

In Xcode, to be able to sign the application while building (called: provisioning), you need to have a developer account with Apple. Go to Xcode > Preferences > Accounts and make sure you have a valid development account. Visit [developer.apple.com](https://developer.apple.com/documentation/xcode/building-and-running-an-app) for more information.

On the left-hand side of Xcode, select Unity-iPhone then under TARGETS select Unity-iPhone. Press on the Signing & Capabilities tab to see a similar configuration as in Figure 2.

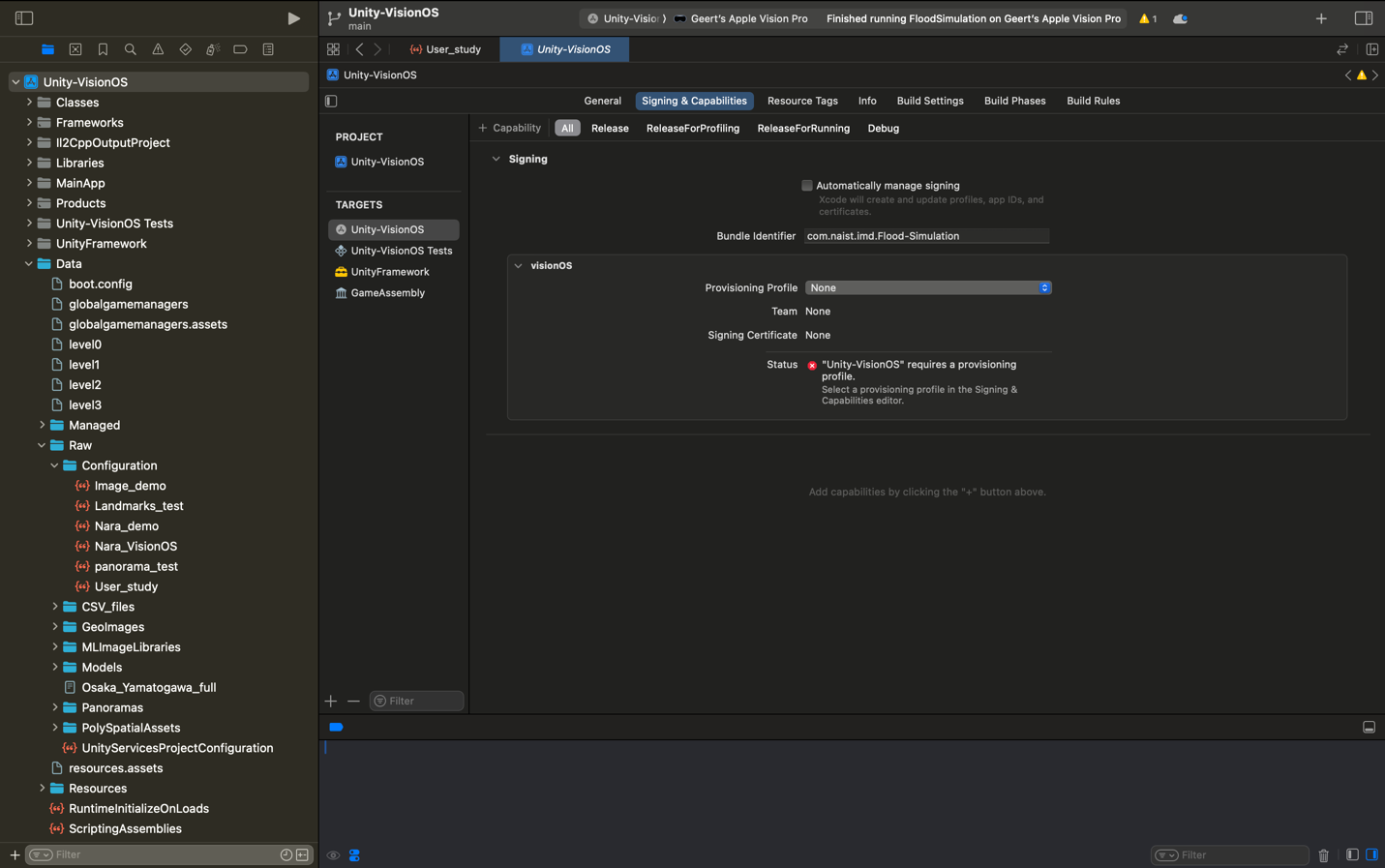


Figure . Xcode signing settings without a provisioning profile.

1. Verify that Bundle Identifier points to a valid in app your Apple Development team, and if not, change the bundle identifier.
2. Select your Provisioning Profile, if available.
3. If your Apple Development team handles signing automatically, or if you want to use a Personal Team, check Automatically manage signing and then press Enable Automatic.
4. Select your Team again.

If all configuration was done correctly, Signing Certificate should contain a valid ID without errors, see Figure 3 bottom.

Connect your iOS device to Xcode via a USB cable or by connecting both to the same (Wi-Fi) network. Connected and saved devices appear in Window > Devices and Simulators. Select your target device (Figure 3 top) and press the Play button. After building, the app should automatically open on your device.

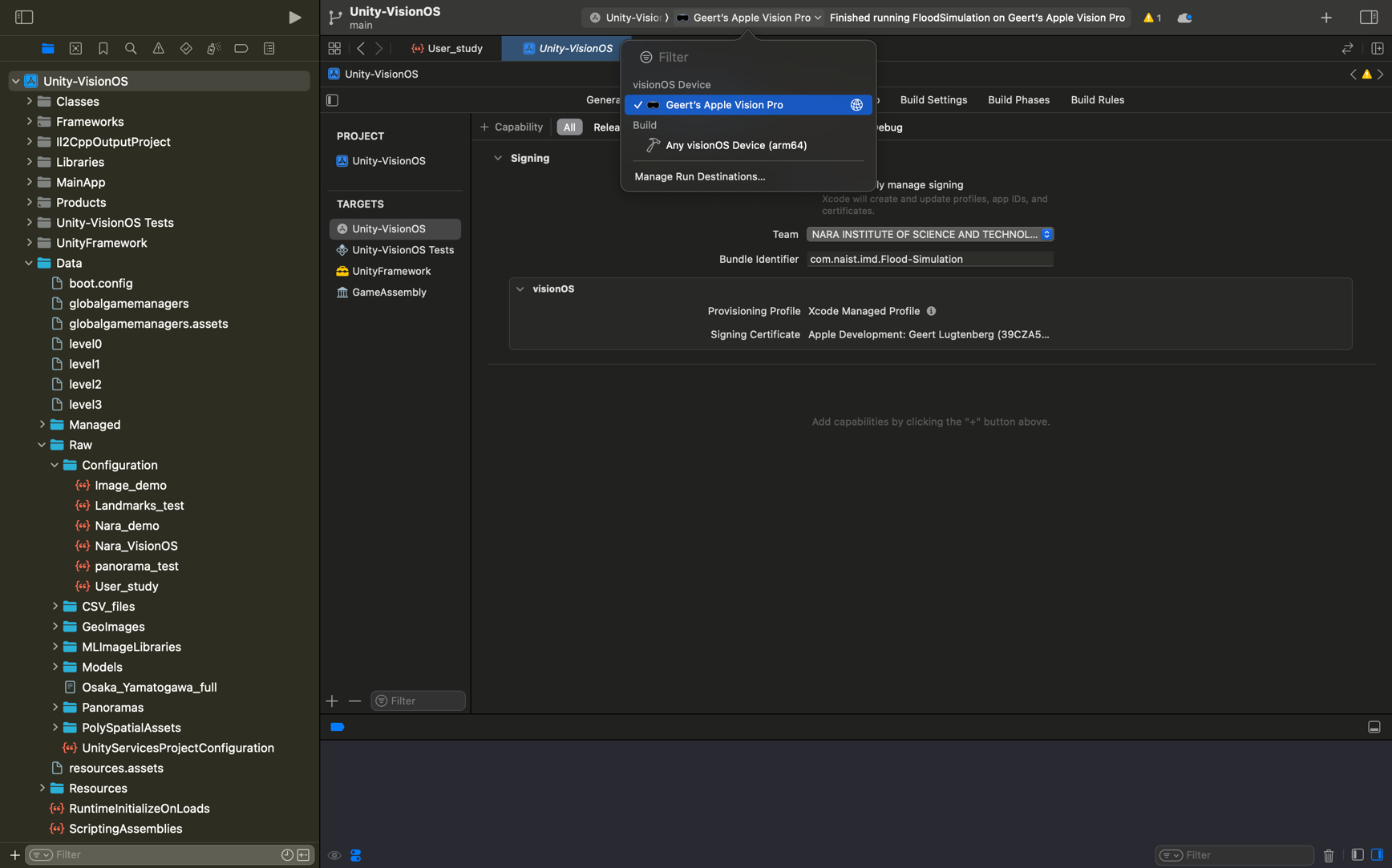


Figure . Selecting build target device in Xcode.

## Building for visionOS

### Requirements

* Apple Silicon Mac device (M-series chip)
* Unity Pro or Enterprise or Education license
* Unity visionOS module > visionOS Build Support
* Xcode 15.2 or newer
* Target device with visionOS 2.0 or newer

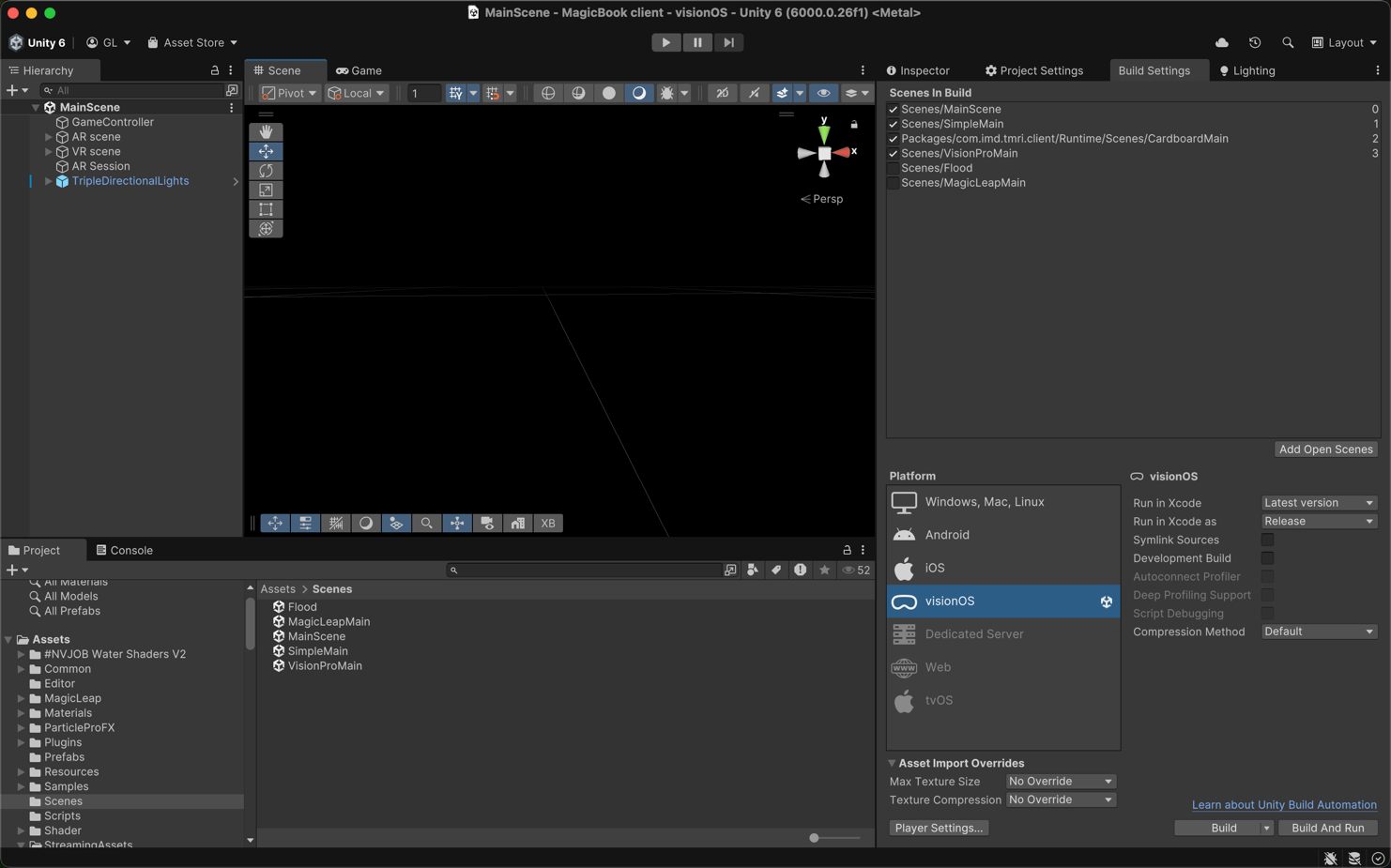
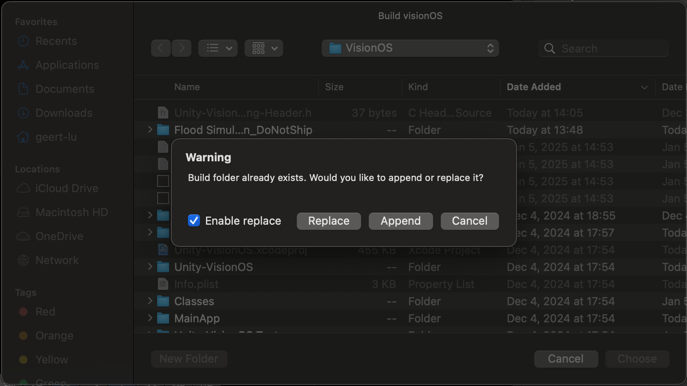
In Unity, go to Window > Panels > Build Settings. Under Platform, select visionOS and press Switch Platform. Switching platforms might take some time.

Figure . VisionOS build settings in Unity.

In the Project panel, browse to Assets/Scenes/ and open MainScene. Press the Play button to try the client app operation in the Unity editor. Before building, make sure there are no errors in the console.

In the Build Settings panel, make sure the visionOS configuration is set up like Figure 4.

If this is the first time building:

1. Press Build.
2. A save dialog appears. Create folders Build/visionOS/ in the MagicBook client directory.
3. Press Choose. The build process will take some time.

If this is not the first time:

1. Press Build.
2. A save dialog appears. Browse to your Build/visionOS directory.
3. Check Enable replace and press Replace.

When Unity is finished, verify the following message in the console: Build completed with a result of 'Succeeded' in x seconds. Browse to the build directory and open Unity-VisionOS.xcodeproj with Xcode.

The build steps for visionOS are identical to iOS Xcode build instructions apart from connecting a target device. At time of writing, the only supported visionOS device is Apple Vision Pro.

### Connecting with a Vision Pro device

Make sure your Vision Pro is connected to the same (Wi-Fi) network as the Mac running Xcode. For general information on VisionOS development, read [Apple’s documentation](https://developer.apple.com/visionos/pathway/). Alternatively, you may read [instructions for preparing your Vision Pro device for development](https://michael-flores.medium.com/how-to-set-up-apple-vision-pro-for-development-cdd7e19ed4c0).

If your Vision Pro is not (yet) available as a Connected build target in Xcode:

1. In Xcode, open Window > Devices and Simulators (Figure 5)
2. On Apple Vision Pro, open Settings and navigate to General > Remote Devices.
3. Back in Xcode, your Vision Pro should appear under Discovered.   
   Press Pair (Figure 5 right).
4. After succesful pairing, select Vision Pro as the target device (Figure 3) and press Play.

After building and installing on Vision Pro, the app should start automatically. Note that the first time building might take some time, and installation over Wi-Fi always takes more time.

If this is the first time building to the Vision Pro device, Xcode will stop prematurely and request to enable Developer mode. Follow the indicated instructions:

1. On Apple Vision Pro, open Settings and navigate to Privacy & Security > Enable Developer Mode.
2. Restart Vision Pro.
3. Repeat the steps in the previous paragraph.

### Problems pairing Vision Pro

If your Apple Vision Pro does not appear in step 3 of Connecting with a Vision Pro device, try the following instructions:

* Make sure your Vision Pro and Xcode are connected to the same (Wi-Fi) network.
* Make sure your Vision Pro has Developer Mode enabled (see above).
* Make sure your Vision Pro has the latest Apple software updates.
* Make sure your Mac running Xcode has the latest Apple software updates.
* Update Xcode to the latest version (using App Store).
* Restart Apple Vision Pro by disconnecting/connecting the power cable.
* Restart your Mac running Xcode.
* If all of the above fails: disable Developer Mode on Vision Pro, then enable it again.

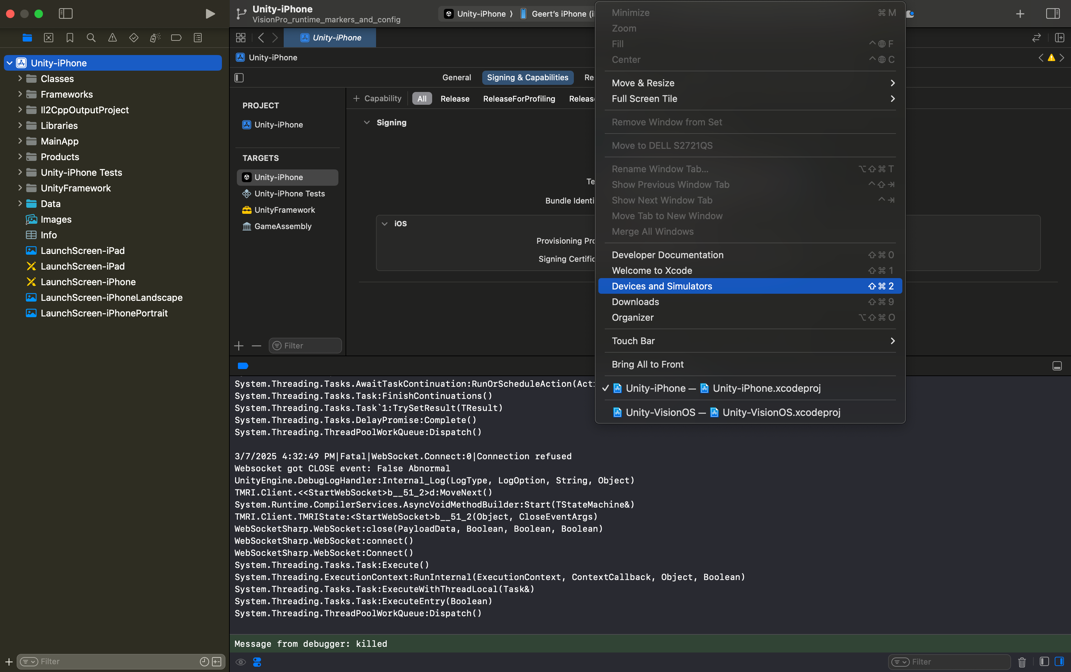
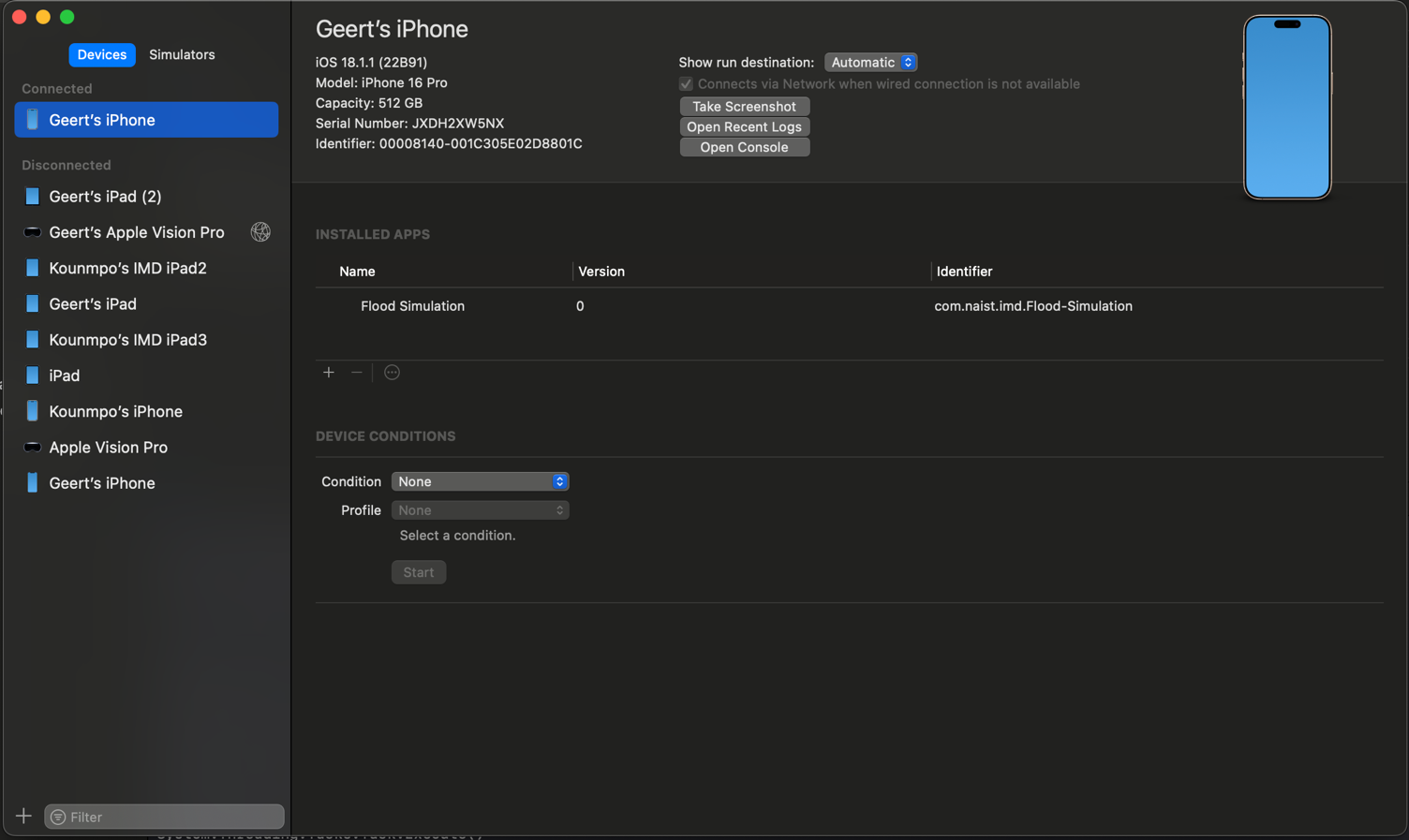
 

Figure . Devices and Simulators in Xcode.

## Building for Android

TBD

# XR Flood server ­– Build instructions

## All platforms

Open [Unity Hub](https://unity.com/download) and press Add > Add project from disk. Navigate to where you downloaded the project /server/MagicBook server and press Open. If the necessary Unity editor version and/or components are not installed yet, you can use Unity Hub to install them.

In Unity, go to the Project panel and navigate to Assets/Scenes/ folder. Open NewServerGUI scene. Press the Play button and verify that there are no errors in the Console panel.

### Deprecated packages

As external packages and dependencies continue to update, some packages might become deprecated. Upon opening the project, Unity might display a dialog as seen to the right. You are free to update or remove packages during future development, but for V1.0 of the XR Flood server app, please press Dismiss.

## Building for Windows, Mac

In Unity, go to Window > Panels > Build Settings. Under Platform, select Windows, Mac, Linux and press Switch Platform. Leave the build settings on their default and press Build And Run. Using the Save Dialog that appears, create a Build folder and save the server application under any name. When posed with the question Do you want TriLib to configure the StandaloneFileBrowser plugin to be included? please press No.