Bridge Engine Unity Plugin

This plugin uses the Structure Sensor's depth sensor to generate a 3D scan of your real world, creating a digital physical space. The real world can occlude virtual objects, and virtual objects cast shadows and bounce off of your real world.

BridgeEngine.unitypackage contains the BridgeEngine.Framework, and the iOS plugin interface code, that will allow you to develop your own Bridge Engine games. We also provide a few example scenes and assets for starting a game from scratch. Bridge Engine will automatically retrieve a compatible version of GoogleVRForUnity.unitypackage **v1.70** that is needed to run with this release of Bridge Engine.

Section 0 - Overview

What's Included

The BridgeEngine.unitypackage contains the libraries and code needed to scan a large area, and begin tracking in VR mode. BridgeEngine for Unity is designed to naturally extend the capabilities of Google VR (formerly Cardboard). With this plugin you'll be able to scan a large area, which is used to precisely track the Bridge Headset movement and render a collision avoidance mesh when you approach real world objects.

Section 0.1 - Requirements

The plugin was built and tested on:
Xcode **8.2** and **9.0**
iOS **9.3, 10.3, 11**
Unity **5.6.3f1, 2017.1.1f1**
GoogleVR **1.70**

Using other configurations may be possible, though we've not tested outside of the prescribed environment and cannot guarantee compatibility.

Known Issues:

GoogleVR 1.60 and older
BridgeEngineForUnity is no longer compatible with Unity 5.5.x series of releases due to the switch over of GoogleVR 1.60 optics and pointer. And we recommend using a minimum of GoogleVR 1.70. The BridgeEngineForUnity plugin will auto-install GoogleVR, so go ahead and delete the older GoogleVR folder to allow BridgeEngine to pull the known compatible version of GoogleVR.

GoogleVR auto-installation does not run
Auto install runs if the **Assets/GoogleVR** is not present in your project, and there are no .check files located at the root of your project path. Look for **BridgeEngineIgnoreGVR.check** or

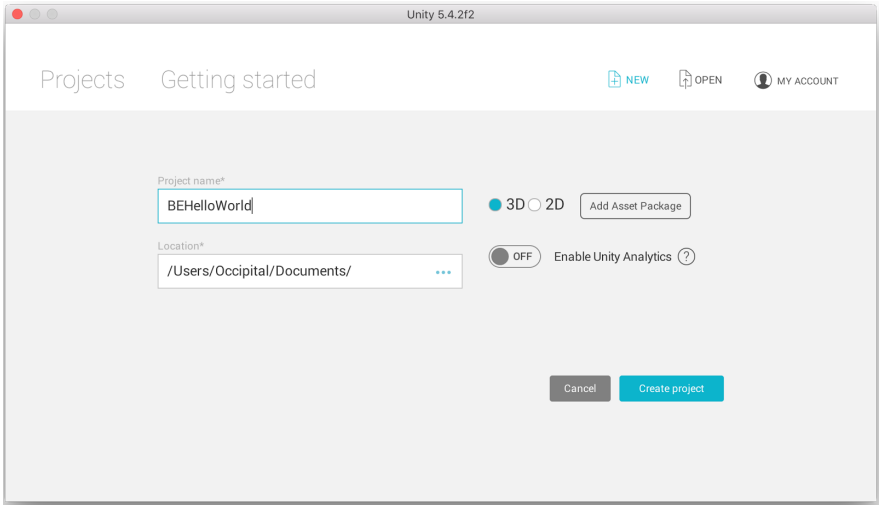
BridgeEngineGVRInstallInProgress.check and delete them. Return to Unity and it should automatically ask to install GoogleVR.

Section 1 - First Steps

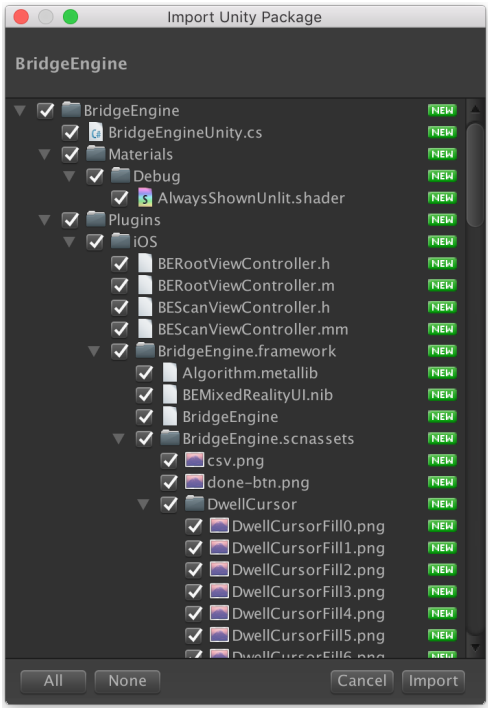
Section 1.1 - Starting with the MR Example

Unity Project Creation, Package Import and Build Steps

1. Create a new Unity project
    a. Unity Menu: select **File** → **New Project…** Save as: “BEHelloWorld”
    b. The project name is arbitrary, and can be located anywhere.

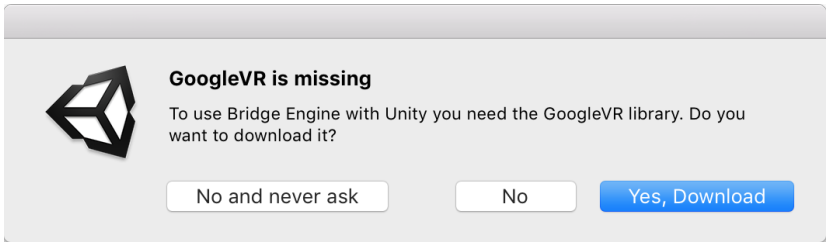


    c. Navigate to: **Assets** → **Import Package** → **Custom Package** and choose the `BridgeEngine.unitypackage`, included with this guide.

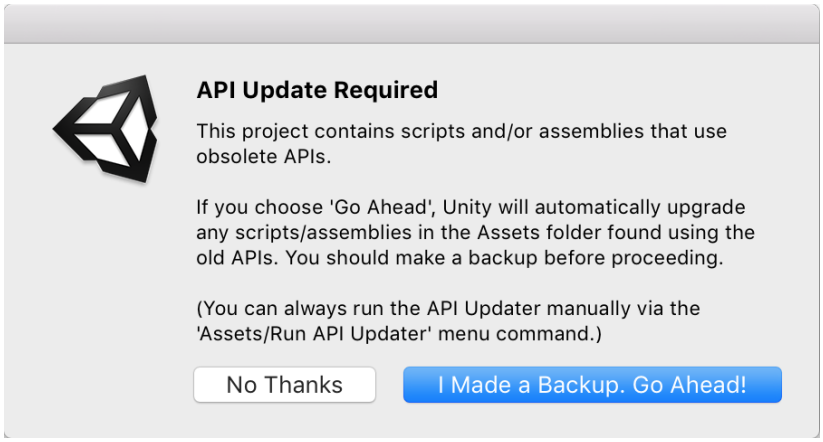


    d. Make sure that all the assets are selected by clicking **All** Then select **Import**.

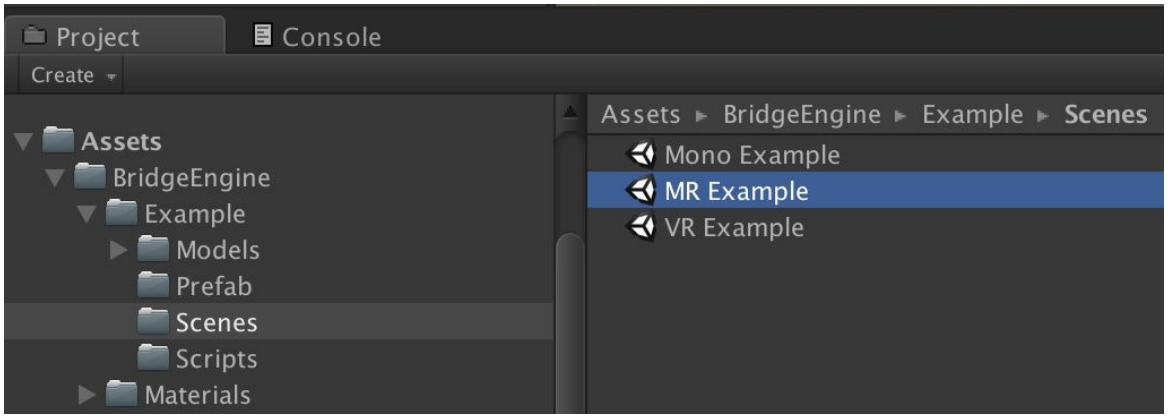
e. Upon installation GoogleVR is needed, click **YES, Download** to begin downloading **GoogleVRForUnity.unitypackage** from the official github repository.



f. Sometimes an API Update Required will appear, select **I Made a Backup, Go Ahead!**

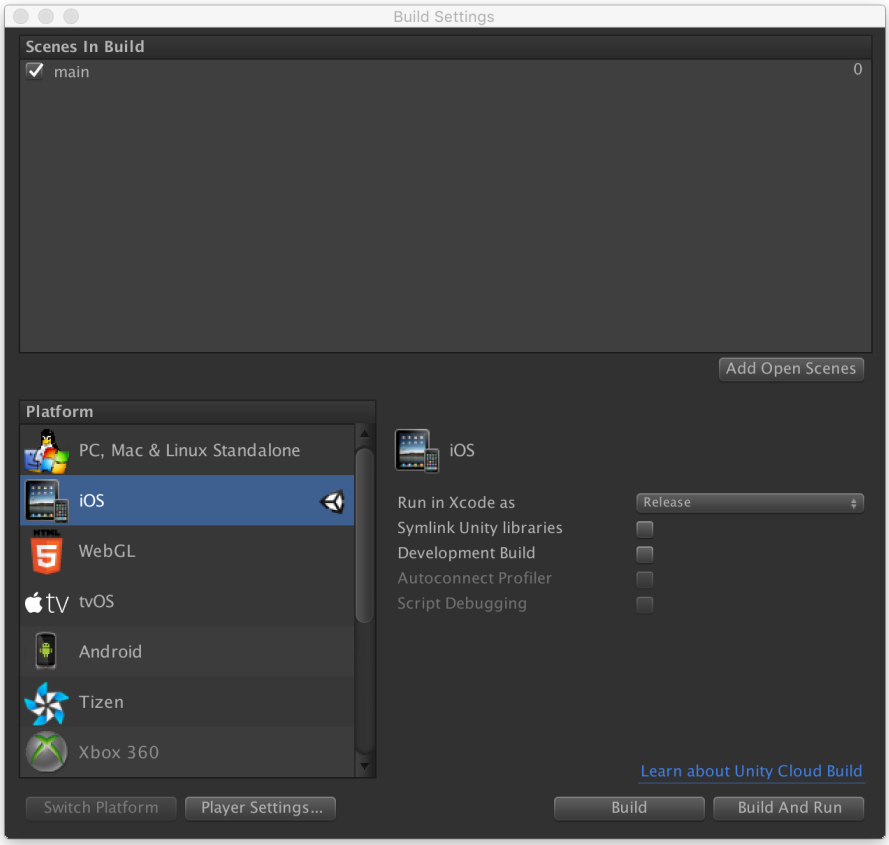


g. Go to the Project, **Assets** → **BridgeEngine** → **Example Scenes** folder and open the MR Example.

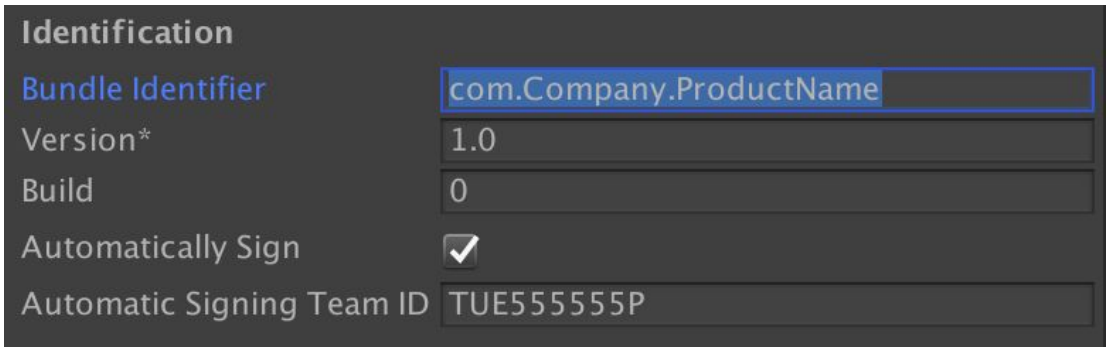


2. Generating the Xcode Project
   a. To build and play BEHelloWorld, you'll want to change your **Build Settings**.
   b. **File** → **Build Settings…** or “Shift + Command + B” to open the “Build Settings” window.
   c. In the “Platform” list, select “**iOS**” platform and press the **Switch Platform** button. This can take a few minutes to re-import the assets.

d. Press the **Add Open Scenes** button, with the Main scene open. This will add the scene to the build, and the “Build Settings” window should now look as follows.

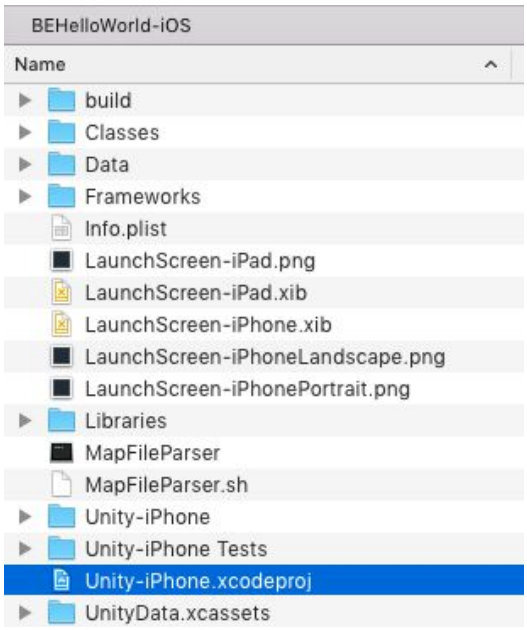


e. Click the [**Player Settings…**] button under the “Platform” list.
f. In the “Other Settings” section, add in your Bundle Identifier for the project.
Example: “com.Company.ProductName”



g. In Automatic Signing Team ID, add in your iOS Team ID
You can find this on your Apple Developer Page here:
https://developer.apple.com/account/#/membership
Look under “Membership” and you’ll see your Team ID
h. Go back to the “Build Settings” window, and press the “Build” button. Unity will then ask you for a location to save the Xcode project it will generate.
i. Pick a directory, then “Save”.

3. Deploying the iOS App
    a. Once Unity is done generating an Xcode project, it automatically opens the directory that contains it:



    b. Connect your iOS device to your Mac using a Lightning cable.
    c. Now, open “Unity-iPhone.xcodeproj”.
    d. Press the “Run” button (or “Command+R”).

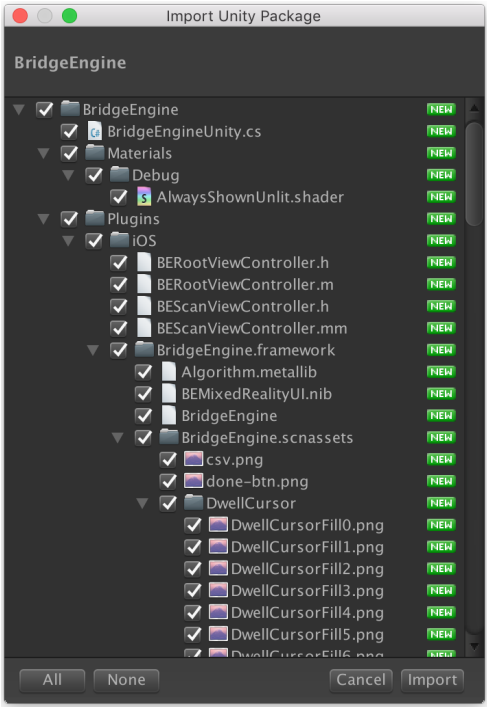
    e. If all goes well, then Bridge Engine should automatically deploy on your iPhone.

After the iOS app has been successfully deployed, insert your iPhone into the Bridge Headset and plug the Structure Sensor.

Section 1.2 - Adding to an Existing GoogleVR Project

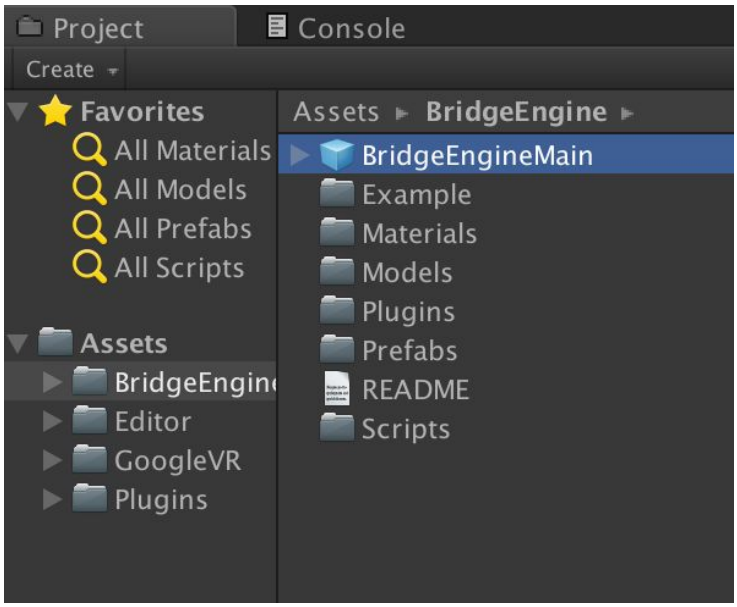
GoogleVR must be Version 1.70, if it isn't, please first delete the **/Assets/GoogleVR** folder.

1. Navigate to: **Assets** → **Import Package** → **Custom Package** and choose the **BridgeEngine.unitypackage**, included with this guide.

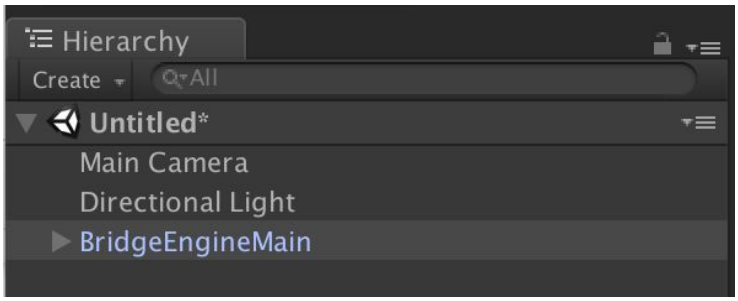


Make sure that all the assets are selected by clicking **All** then select **Import**.

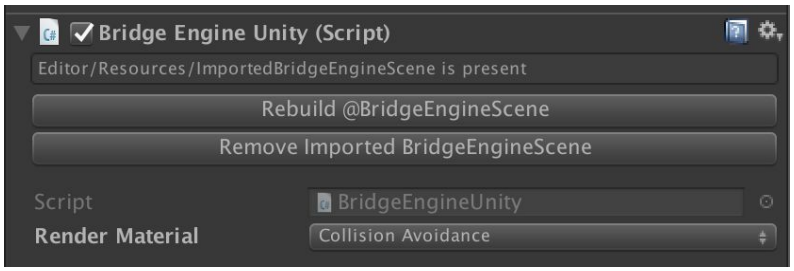
2. Open **/Assets/BridgeEngine** and drag-drop **BridgeEngineMain.prefab** to your scene.



3. Select the **BridgeEngineMain** GameObject



4. Look at the **Inspector** and **Rebuild @BridgeEngineScene**

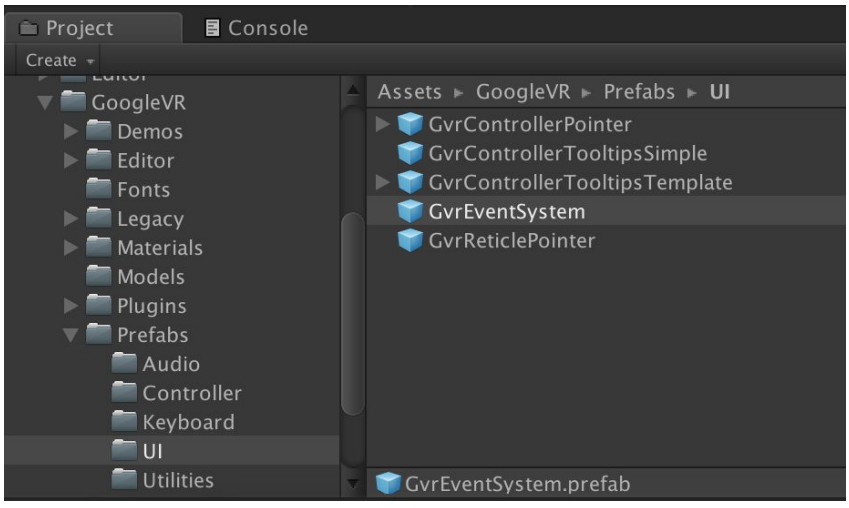


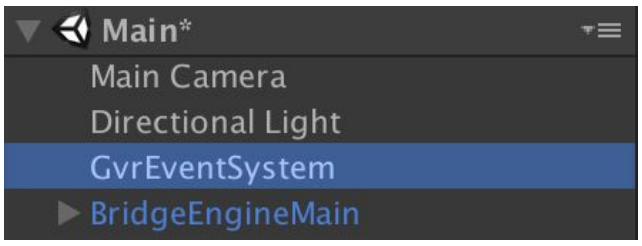
This will recreate the mixed reality **@BridgeEngineScene** world geometry GameObject for experimenting with pre-scanned scene. You can also remove the **ImportedBridgeEngineScene** folder, or if you like import your own **BridgeEngineScene** folder from a prior scan.

Section 1.3 - Enabling Controller Interaction

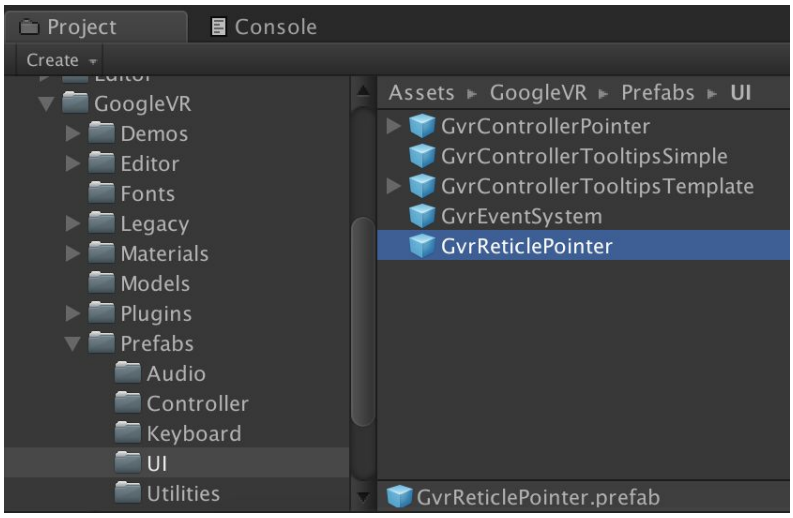
If you're setting up a fresh project, GoogleVR requires a couple components for the controller to work.

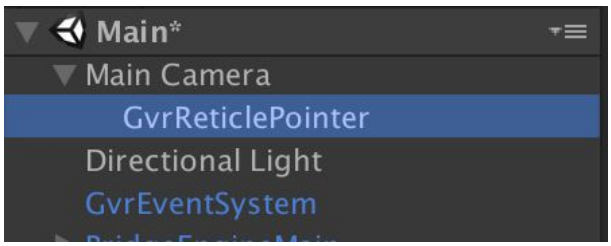
1. Navigate in the Project to: **Assets**/**GoogleVR**/**Prefabs/UI** and drag the **GvrEventSystem** into the Hierarchy.



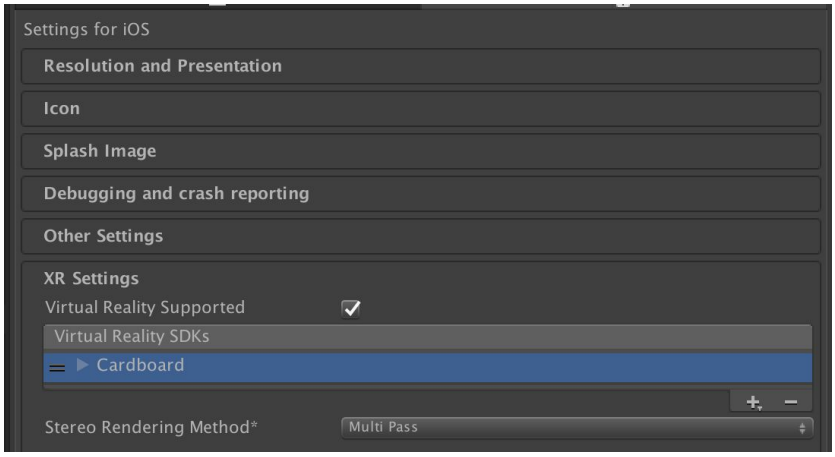


2. Select your “Main Camera” and drag the **GvrReticlePointer** onto it.





3. Make sure to add the Cardboard to the supported Virtual Reality SDKs.
   Navigate to the **Player Settings** → **XR Settings** (or Other settings) → **Virtual Reality SDKs**, click the “+” and add **Cardboard**.



Section 2 - Scanning and Interacting

The gameplay of all Bridge Engine games is made of two distinct steps. First, the real-world geometry is captured, and then the actual game loop starts.

Section 2.1 - Capturing Real-World Geometry

After the game has started, you scan the full room that you wish to track on. When ready, press the “Scan” button to start scanning and build the game world mesh. When you’ve scanned enough of the scene, press the “Done” button. Press “Re-Scan” to scan a new mesh and start over. Now you’re ready to play in the real world!

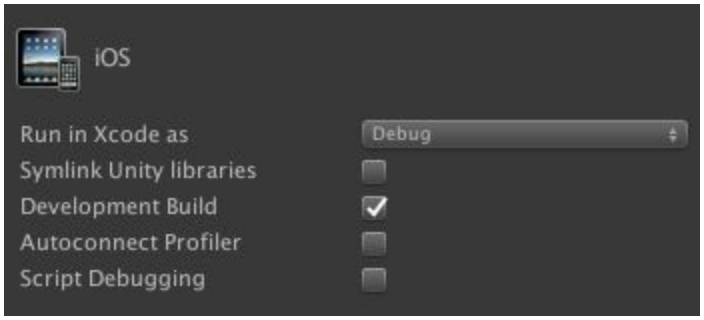
Section 2.2 - Playing

When the Bridge Engine game loop starts you can freely move around the scene and look in the direction of wherever there’s a scanned surface.

Section 3 - Debugging Bridge Engine Games

Section 3.1 - Troubleshooting Bridge Engine

Unity must generate **IL2CPP** code to compile natively on iOS. You can create readable CPP generated output by **File → Build Settings…** or “Shift + Command + B” to open the “Build Settings” window, and turning on the **Development Build** setting. Then do a Build.

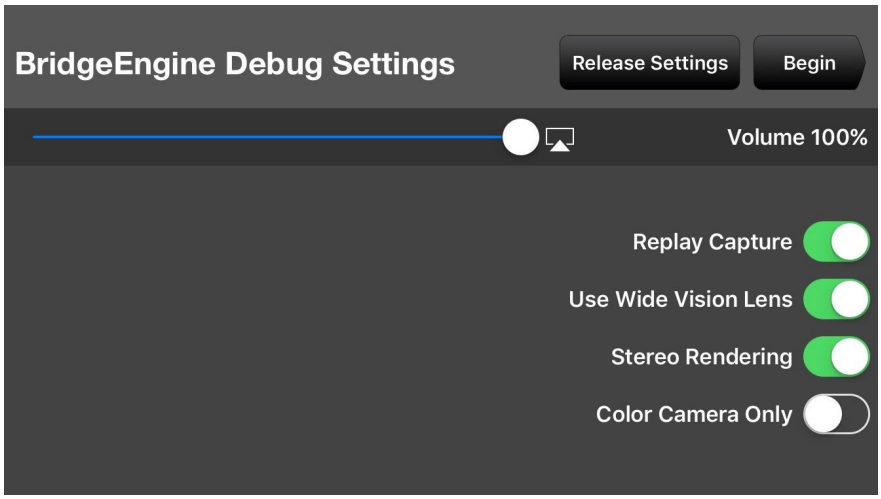


You can place breakpoints on the generated output. For a great tutorial on debugging IL2CPP check out this document on Unity’s website “IL2CPP INTERNALS – DEBUGGING TIPS FOR GENERATED CODE”: https://blogs.unity3d.com/2015/05/20/il2cpp-internals-debugging-tips-for-generated-code/

Whenever scanning a scene, an OCC recording captures all the sensor information into a single stream. It’s helpful to enable OCC capture replay so you can effectively debug your code in Xcode while connected by a wired connection.

- When scanning, hold still and look at center of the floor for about 10 seconds, before moving around. This gives you some time to navigate through the Scanning screen and move into the app experience.

- On next run, you can enable “Replay Capture” which Enables OCC capture playback:

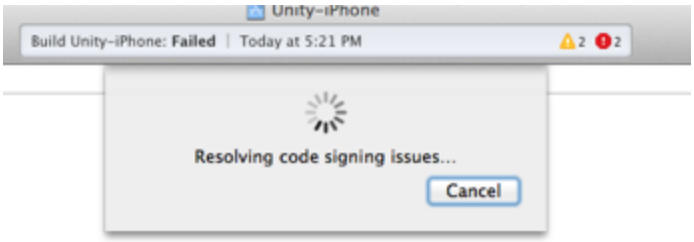


- Another option is to turn on “Color Camera Only” which allows tracking by only using the color camera, allowing you stay plugged into the Xcode debugger.

Section 3.2 - Code Signing Issues

When building the Xcode project, you may get an error message about missing provisioning profile.

If everything is setup properly then you should be able to resolve this with the “Fix Issue” button.

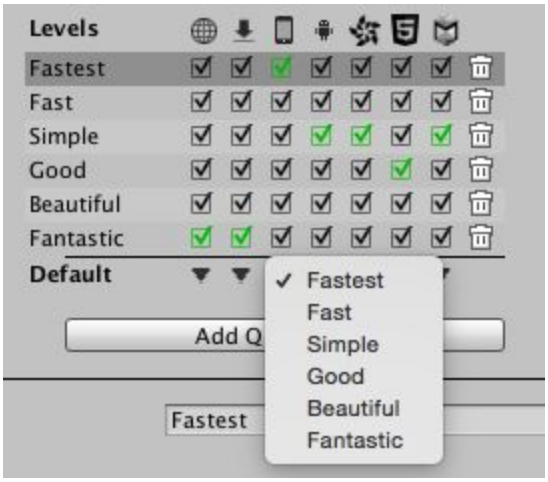


If this issue persists, please refer to the “Troubleshooting” section of Apple’s iOS app Distribution Guide.
Additional Notes:
For this release we’ve decided to set Exit on Suspend to true by default. This was added in the PostprocessBuildPlayer script found in Assets/Editor. This is a script which sets the Xcode project up. If your project requires this to be false, then you may edit the PostprocessBuildPlayer script by commenting out the last two lines.
Debugging symbols (dSYM) are turned off by default in the generated Xcode project. To turn them on, modify the Debug Information Format from DWARF to DWARF with dSYM file in the Xcode Build Options.
We override Unity’s Exit on Suspend setting in Player Settings, as resuming from background can currently cause unexpected behavior.

Section 4 - Tips

Section 4.1 - Use “Fast” Quality for Better Performance

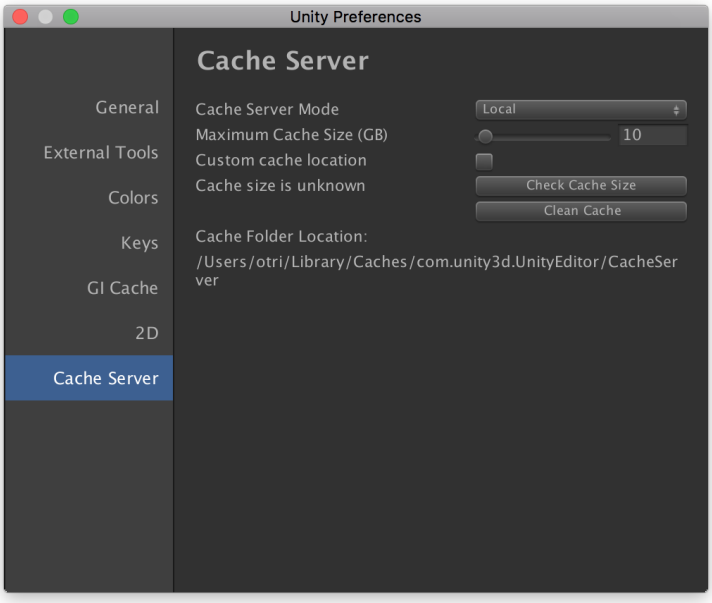
Depending on your device you have the option to change the quality settings within Unity to improve the rendering speed. This will have a direct effect on the frame rate and the perceived quality of the unbounded tracking.
To change the quality settings select Edit → Project Settings → Quality



In the inspector Simple is used as the quality used for the iOS device. Select either Fast or Fastest for better performance.

Section 4.2 - Turn on Local Cache Server

Whenever switching platforms to iOS, it can can take a lot of time reimporting all the Bridge Engine and GoogleVR image assets. It’s a good idea to turn on the Local Cache server built into Unity. This will make take the chore out of processing these images. Go to **Preferences** → **Cache Server**, and change the “**Cache Server Mode**” to “**Local**”.



Section 4.3 - Plus/Pro Tip : Turn Off “Show Splash Screen”

Once Bridge Engine has completed scanning, there’s a moment where Unity will start up and present a splash screen that is not presented in stereo mode. If you have the Plus or Pro license for Unity, go to **Player Settings** → **Splash Image**, and turn off **Show Splash Screen** to avoid this inconvenience and delay to startup.