Bridge Engine Unity Plugin

This plugin uses the Structure Sensor's depth sensor to generate and track real world elements and introduce them to your game experience.

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 **v1.40** 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 iOS **9.3** & **10.3**.
It is known to be compatible with Unity versions **5.5.0f3** & **5.5.3f1**.

Using other configurations may be possible, though we've not tested outside of the prescribed environment and cannot guarantee compatibility. There are multiple known issues with GoogleVRForUnity and Unity v5.6.0 beta, please refrain from using this release until GoogleVR native integration is stabilized.

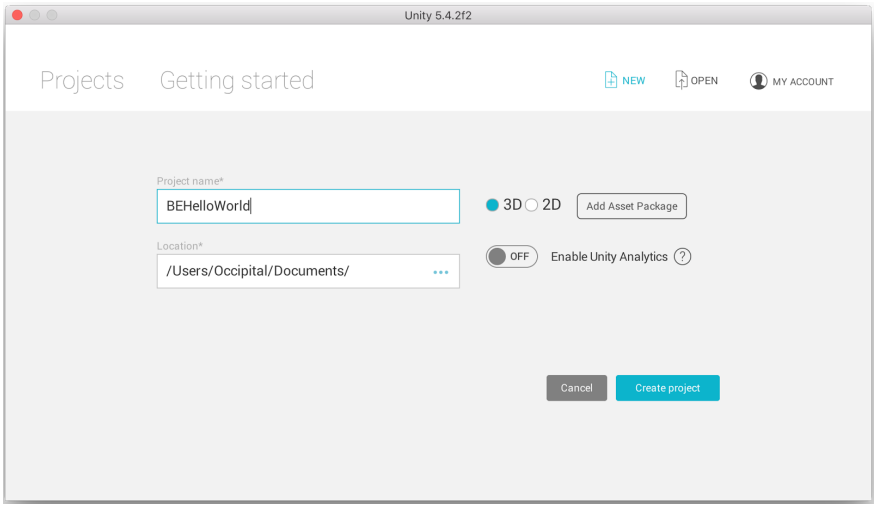
Known Issues:

BridgeEngineForUnity is not currently compatible with Unity 5.6.x and 2017.x series of releases, there's a lot of changes going on that relate to GoogleVR native integration which currently prevent an application from setting the optical profile of a headset. Therefore Bridge Engine Headset will not look correct. We are working on a solution with Unity and Google. Please download the latest Unity 5.5.x release from here:
https://unity3d.com/get-unity/download/archive

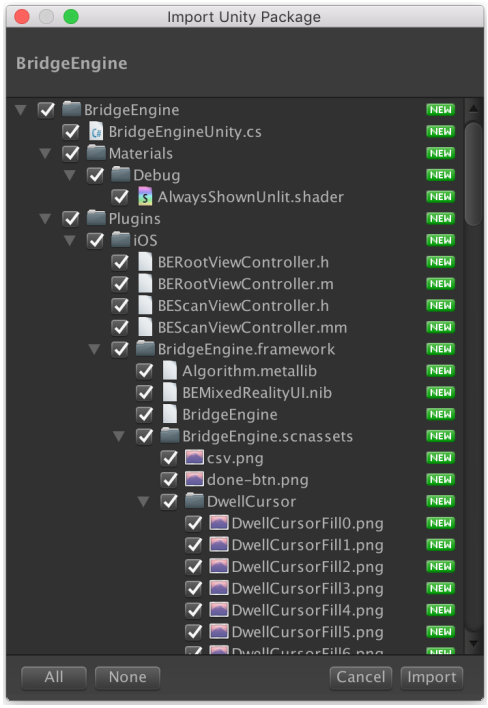
Section 1 - Starting from Scratch

Section 1.1 - 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.

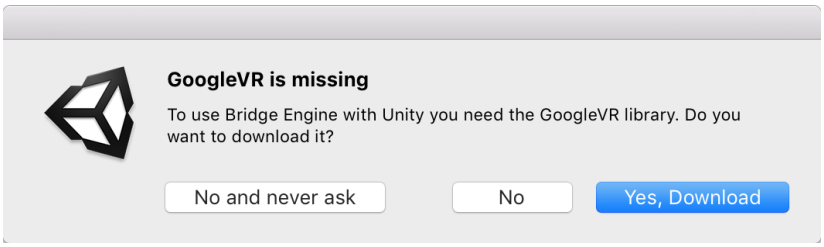


    c. Navigate to: **Assets** → **Import Package** → **Custom Package** and choose the `BridgeEngine.unitypackage`, included with this guide.

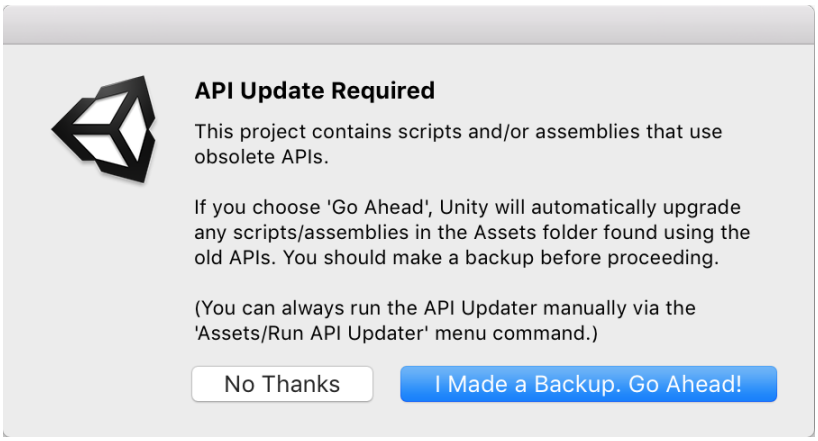


    d. Make sure that all the assets are selected by clicking **All** Then select **Import**.

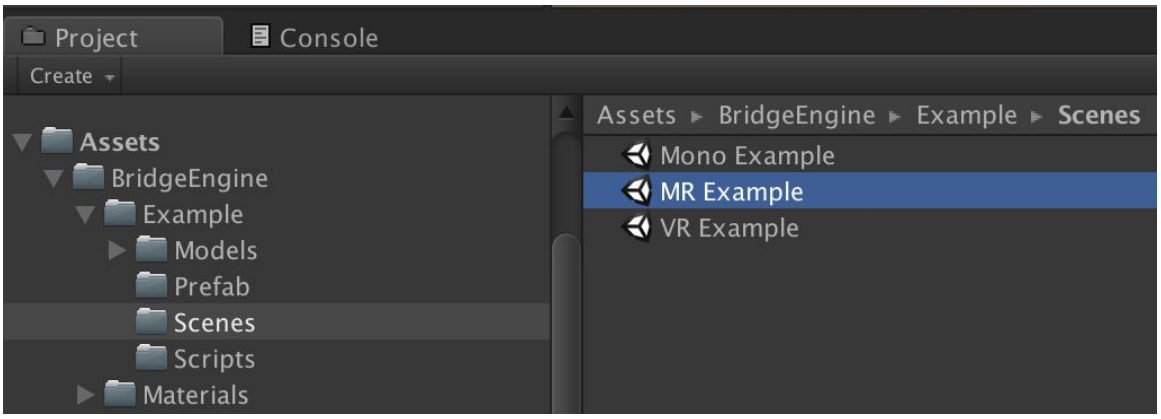
e. Upon installation GoogleVR is needed, click **YES, Download** to begin downloading **GoogleVRForUnity.unitypackage** from the official github repository.



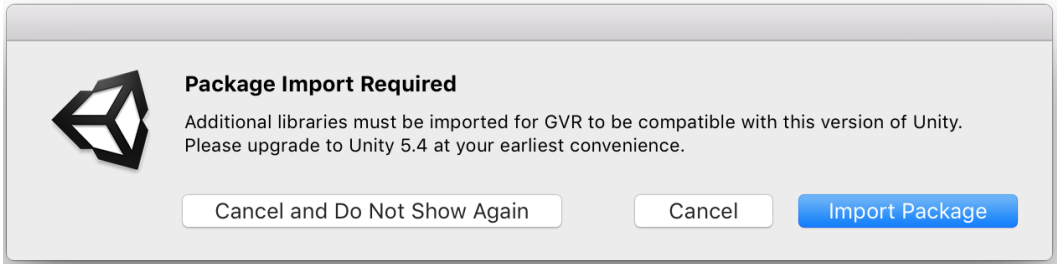
f. Accept API Update Required, select **I Made a Backup, Go Ahead!**



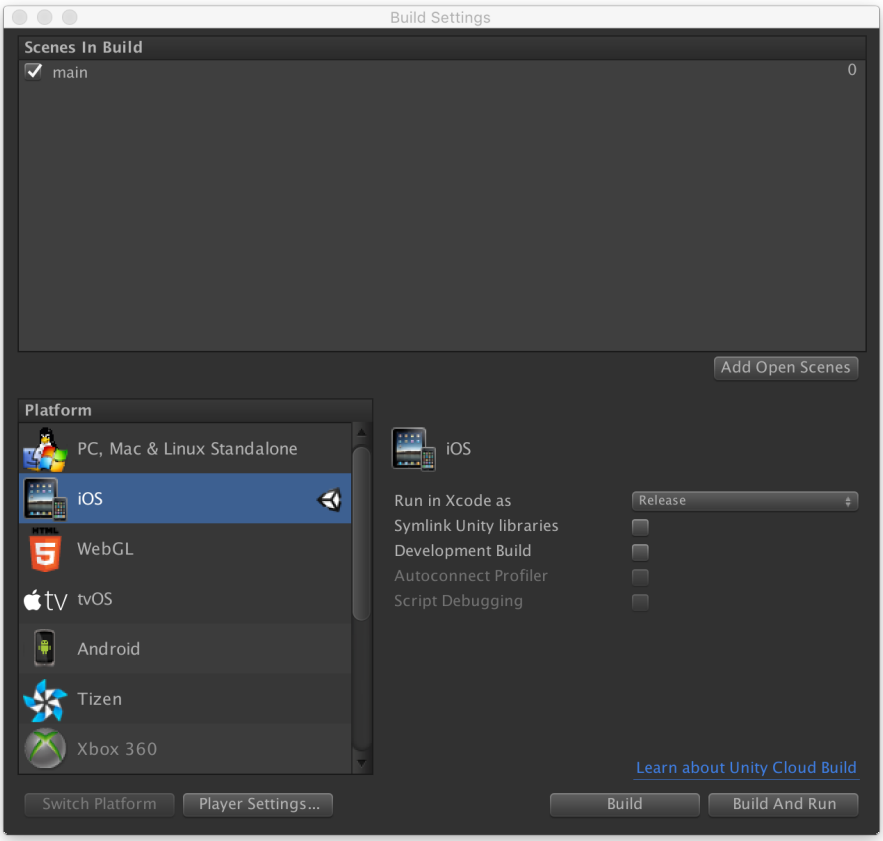
g. Go to the Project, **Assets** → **BridgeEngine** → **Example Scenes** folder and open the MR Example.



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. GoogleVR will detect it’s not ready to run yet, follow the prompt to install “Additional libraries” for compatibility with GVR. Select **Import Package**



e. 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.



f. Click the [**Player Settings…**] button under the “Platform” list.

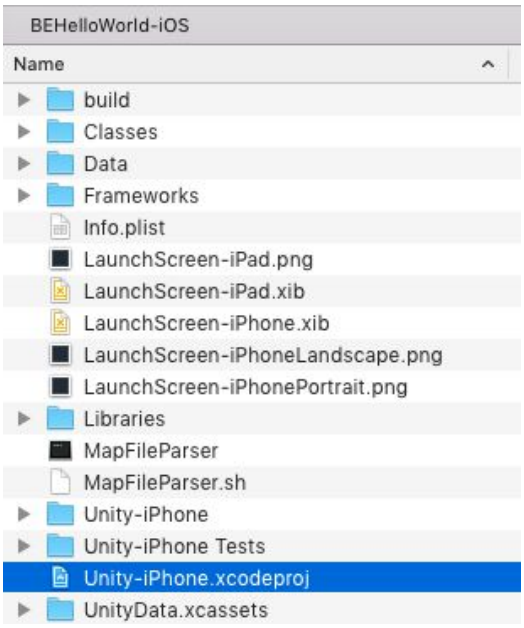
g. In the “Other Settings” section, add in your Bundle Identifier for the project.
Example: “com.Company.ProductName”

h. In iOS Developer 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

i. 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.

j. Pick a directory, then press “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, you will need to unplug the iPhone from your computer, then insert your iPhone into the Bridge Headset and plug the Structure Sensor.

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!

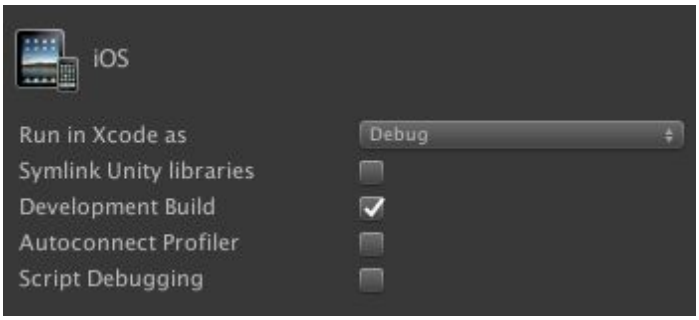
Section 2.2 - Playing

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

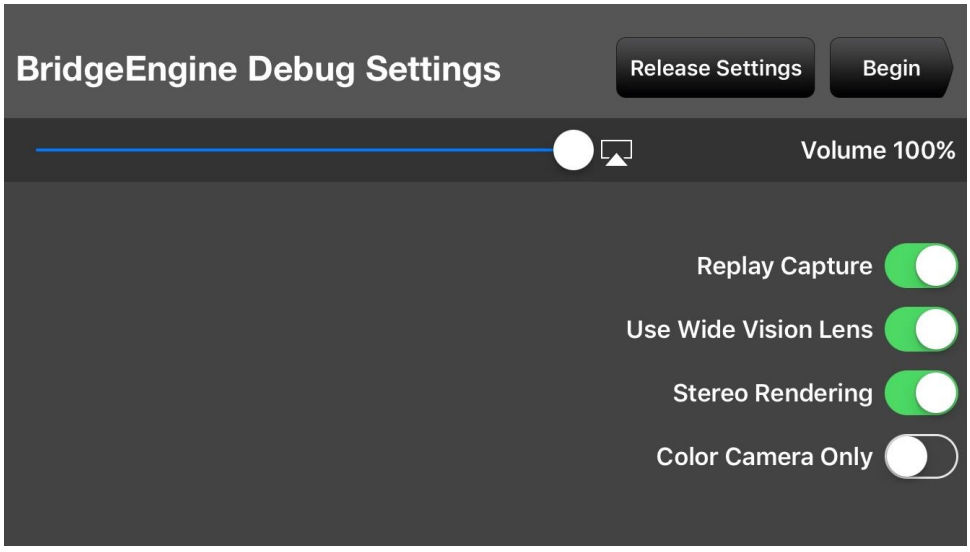
Unity must generate **IL2CPP** code to compile natively on iOS. You can create readable CPP generated output by **File → Build Settings…** or “Shift + Command + B” to open the “Build Settings” window, and turning on the **Development Build** setting. Then do a Build.



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:

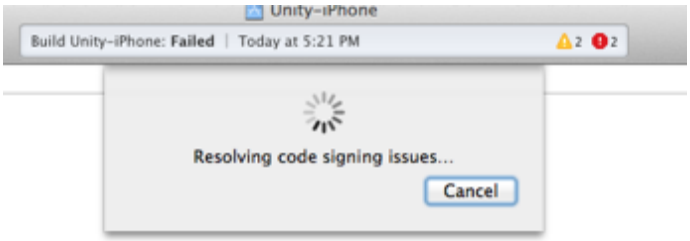


- Another option is to turn on “Color Camera Only” which allows tracking by only using the color camera, allowing you stay plugged into the Xcode debugger.

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 - Performance improvements

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.