

StructureUnityAR Plugin

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

StructureUnityAR.unitypackage contains the Structure.Framework, the iOS plugin interface code, game assets and two sample scenes that will allow you to develop your own Structure Sensor augmented reality games. We also provide a simple scene for starting a game from scratch.

Section 0 - Overview

What's included

The StructureUnityAR.unitypackage for Unity 5.1.2 and up contains a few simple scenes and projects which allow for modification to serve as a starting point for your own project. With this plugin you'll be able to scan a small area of your room which can be used in your unity game to create a full 3d environment for your characters and objects to interact with. You'll also be able to use the tracked model to add camera tracking to your scene.

Section 0.1 - Requirements

In order to use the StructureUnityAR plugin, you will need:

- Unity version 5.1.2 or newer
 - The StructureUnityAR Plugin is compatible with both Unity Personal and Unity Pro
 - Older versions of Unity are not supported.
- iOS 8.4
 - iOS development requires apple certification
- Xcode version 6.x
 - Xcode 7 not yet supported
- OSX 10.10.x

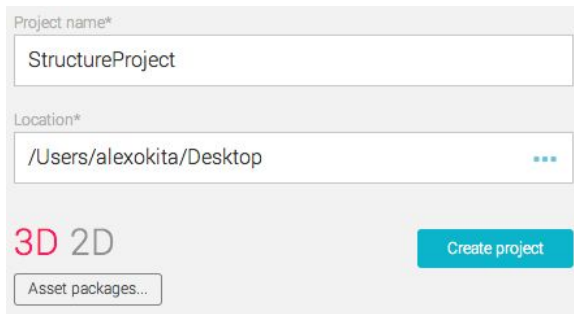
StructureUnityAR includes two sample game scenes: Fetch and Ball Physics. Both sample games consist of a single Unity scene, where in-game assets can interact with 3D-scanned real world geometry. We have also included a simple template scene for you to get started from scratch. In order to run these games, follow the procedure below. We describe the procedure for Fetch, but the same workflow applies with Ball Physics.

Currently tested with Unity **5.1.3f1**, **5.2.0f3** Xcode **6.4 (6E35b)** OSX **10.10.5** Yosemite iOS **8.4.1**

Section 1 - Starting from Scratch

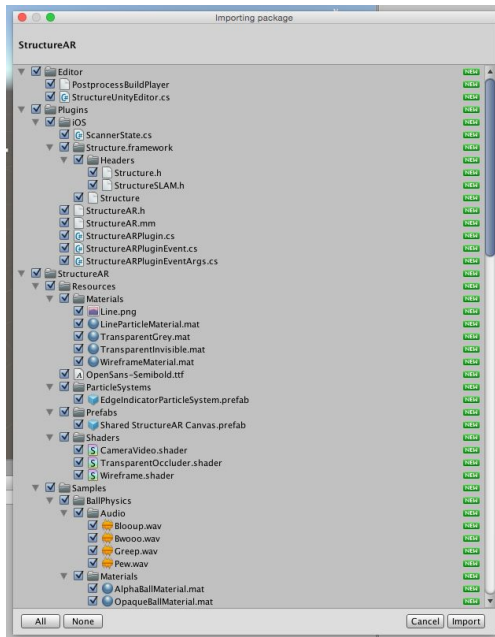
Section 1.1 - Unity Project Creation Package Import and Build Steps

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



c.

- d. Navigate to: **Assets** → **Import Package** → **Custom Package** and choose the StructureUnityAR.unitypackage, included with this guide.

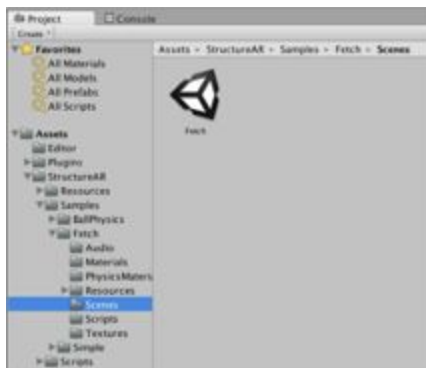


e.

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

2. Open the Fetch scene, located in: **Assets/StructureUnityAR/Samples/Fetch/Scenes/Fetch.unity**

- a. You should see the following project tree:



b.

3. Generating the Xcode Project

- a. To build and play Fetch, you'll want to change your **Build Settings**.

- b. **File** → **Build Settings...** or "Shift + Command + B" to open the "Build Settings" window.

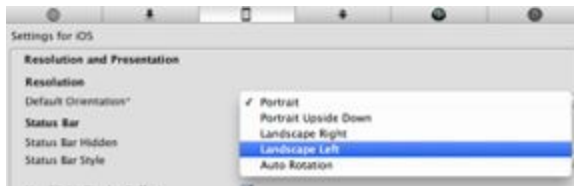
4. In the "Platform" list, select "iOS" and press the "Switch Platform" button.

5. Press the "Add Current" button, with the Fetch scene open. This will add the scene to the build, and the "Build Settings" window should now look as follows.



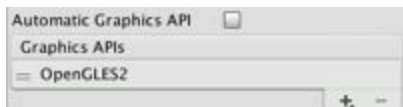
a.

6. Click the [**Player Settings...**] button under the “Platform” list.
 - a. The player settings inspector panel, in the Unity Editor.
7. In the “Resolution and Presentation” section, change the Default Orientation to **Landscape Left**, as depicted in the image below. (This is the optimal setting for the Structure Sensor attached to your iPad.)



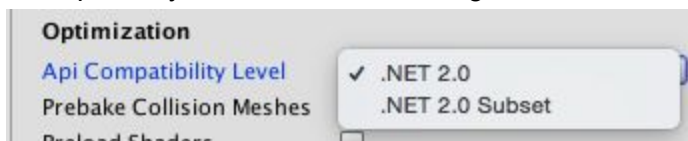
a.

8. In the “Other Settings” section, uncheck “Automatic Graphics API”. This will give you an option to choose your Graphics API. Only **OpenGL ES2** is supported right now; make sure it is added (if not, press the “+” button). Delete any other Graphics APIs listed by pressing the “-” button while they are selected. Your graphics API settings should look like the image below.



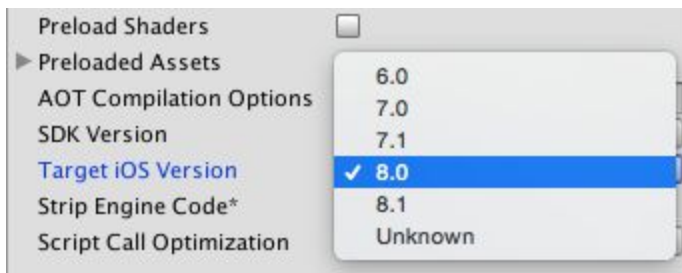
a.

9. The Api Compatibility Level should be changed to .NET 2.0 rather than .NET 2.0 Subset



a.

10. In the “Other Settings” section, set “Target iOS Version” to 8.0 or higher (Structure SDK does not support iOS7).

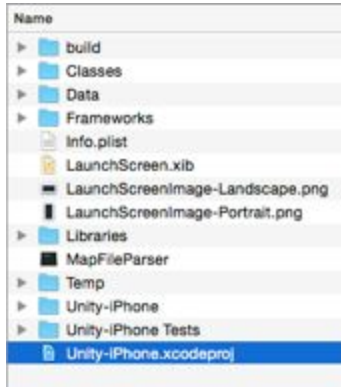


a.

11. 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.
12. Pick a directory, then press “Save”.

13. Deploying the iOS App

14. Once Unity is done generating an Xcode project, it automatically opens the directory that contains it:



a.

15. First, connect your iPad to your Mac using a Lightning cable.

16. Now, open “Unity-iPhone.xcodeproj”.

17. Press the “Run” button (or “Command+R”).



18.

19. If all goes well, then the Fetch game should automatically deploy on your iPad.

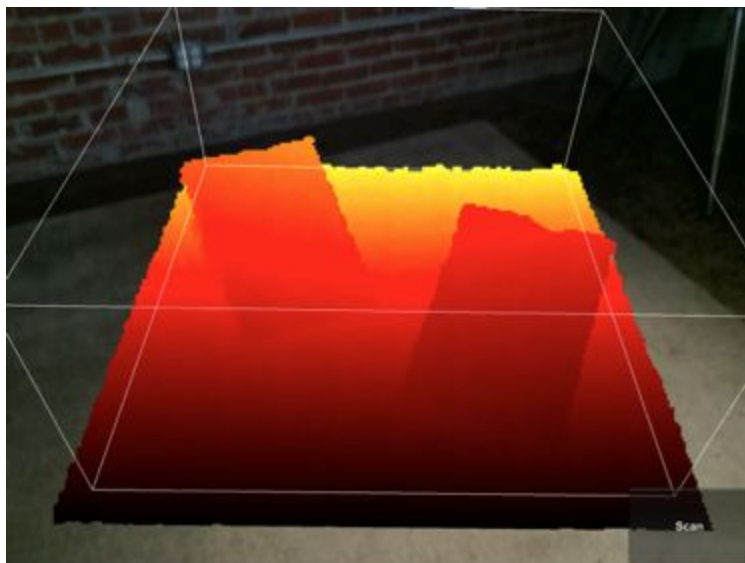
After the iOS app has been successfully deployed, you will need to unplug the iPad from your computer, then plug the Structure Sensor and re-launch the app. Playing StructureUnityAR Games

Section 2 - Scanning and Interacting

The gameplay of all StructureUnityAR 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, a transparent cube will be displayed on top of the iPad camera view.



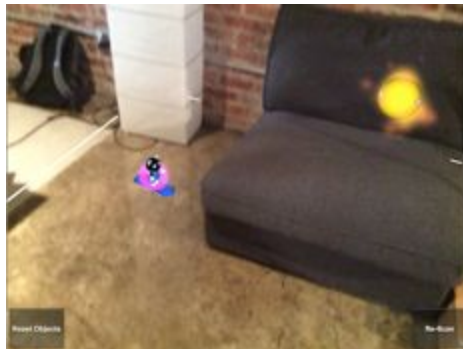
It depicts the volume of interest, used to initialize and 3D-scan the game world. You can perform a pinch gesture to zoom and change its size. When ready, press the “Scan” button to start scanning and build the game world mesh.

While keeping the cube in view, walk around slowly to add detail to the scene. 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 Fetch

When the Fetch game loop starts, a “Hover-cat” character will drop into the scene.



Tap and hold regions on the screen to move the Hover-cat. Tap with two fingers to make it jump. Collect the glowing ball for a jump height power-up! Press the “Reset Objects” button to start over and drop the Hover-cat and glowing ball the in scene, should they get stuck.

Section 2.3 - Playing Ball Physics

When the Ball Physics game loop starts two buttons appear on the lower left.



Press the “Launch Ball” button to launch balls into the scene from the camera’s point of view.

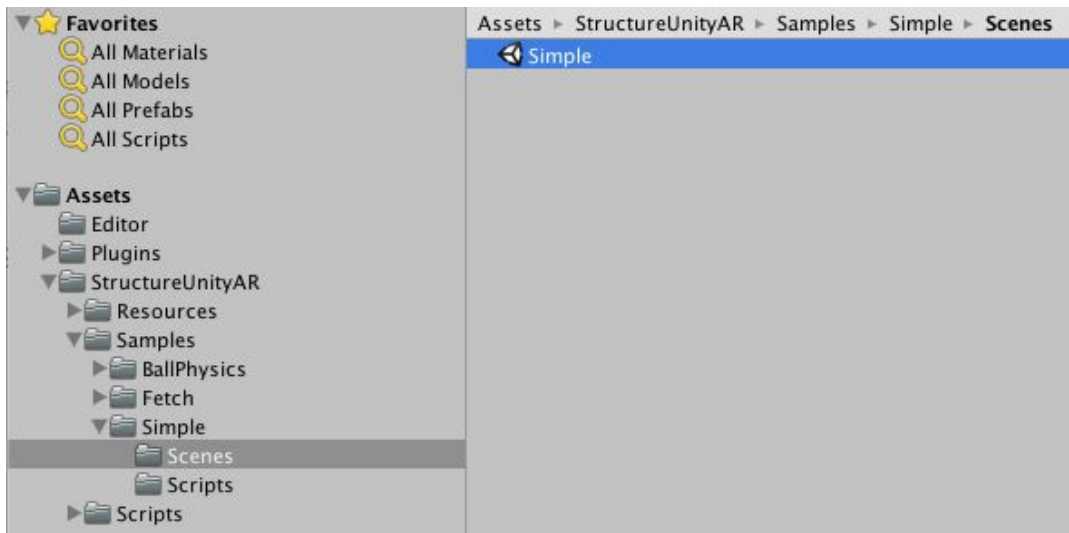
Press the “Drop Balls” button to cascade balls into the scene from above.

Section 3.0 - Creating a StructureUnityAR Game From Scratch

1. Import the StructureUnityAR.unitypackage into a new project.
2. Create a new Unity project
3. Navigate to: **Assets** → **Import Package** → **Custom Package** and choose the StructureUnityAR.unitypackage which was included with this guide. Make sure that all the assets are selected by clicking “All”. Then select “Import”.
4. Click anywhere on the Unity menu at the top of your screen.

Section 3.1 - From a Simple Scene

The simplest way to begin a new scene using the provided tools is to open the Simple.unity scene located in **StructureUnityAR/Samples/Simple/Scenes/Simple.unity** this has the necessary components and GameObjects in the scene required to get started. In the scene is a Cube with the SimpleObject.cs component added to it. The class can be used as a template that listens to events generated by the plugin.



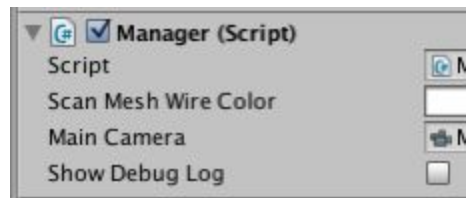
Alternatively, you can follow these instructions to create a new template scene to start from. You should now see a Structure AR menu item appear in the Unity menu bar. You can use this menu to create the following assets:

- **Shared StructureAR Canvas**
 - The 2D UI for StructureUnityAR Plugin, including a status bar at the top and, in the lower right, a button to advance through the stages of an AR game.
- **Buttons.cs**
 - This is a simple UI script used to send commands to the Manager. This can be replaced by any other UI system, so long as the correct methods are called in the Manager class.
- **Manager**
 - This class listens to events from the StructureUnityAR Plug-in. Please refer to the sample games for more information - you may inherit from this class, as we do with HoverCatManager in the Fetch sample.
- **PinchToScale**

- This script is also attached to the Main Camera automatically when the Manager is created. This listens for a pinch gesture to change the scale of the scan volume.
- **POV**
 - The POV script is automatically attached at runtime to the Main Camera in the scene. This script renders the color camera and the scan-volume of the scene. This also contains the methods to update the transform of the Main Camera while tracking.
- **ObjectLoader**
 - This is used during game play. This is created to store the mesh and render the wireframe once scanning is complete.
- **GroundPlane**
 - This asset represents the ground plane, as detected by the Structure Sensor.

Section 4.0 - Debugging Structure AR Games

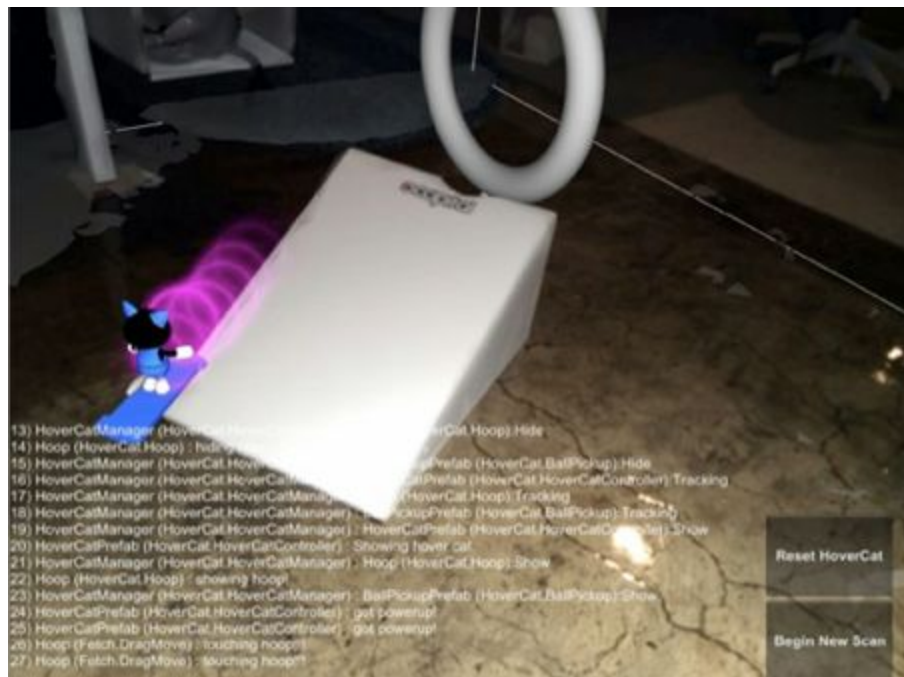
There is a special class, “GameLog”, which can be used to display debug messages to the iOS device during gameplay. To enable the Debug Log, select the ‘Show Debug Log’ checkbox on the Manager script.



To print a message to the display, use the following C# command in one of your Unity Project scripts:

```
GameLog.Log("Structure AR rocks!");
```

This will output the string argument in the translucent in-game log, shown below.



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 python 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.

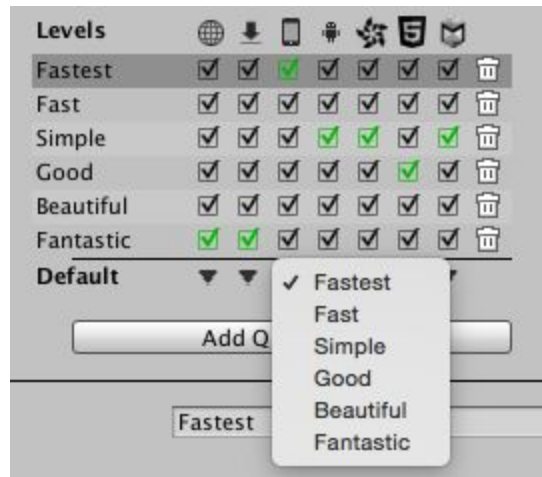
Debugging symbols (dSYM) are turned off by default in the generated Xcode project. To turn them on, modify the Debug Information Format from DWARF to DWARF with dSYM file in the Xcode Build Options.

We override Unity’s Exit on Suspend setting in Player Settings, as resuming from background can currently cause unexpected behavior.

Section 5 - 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.