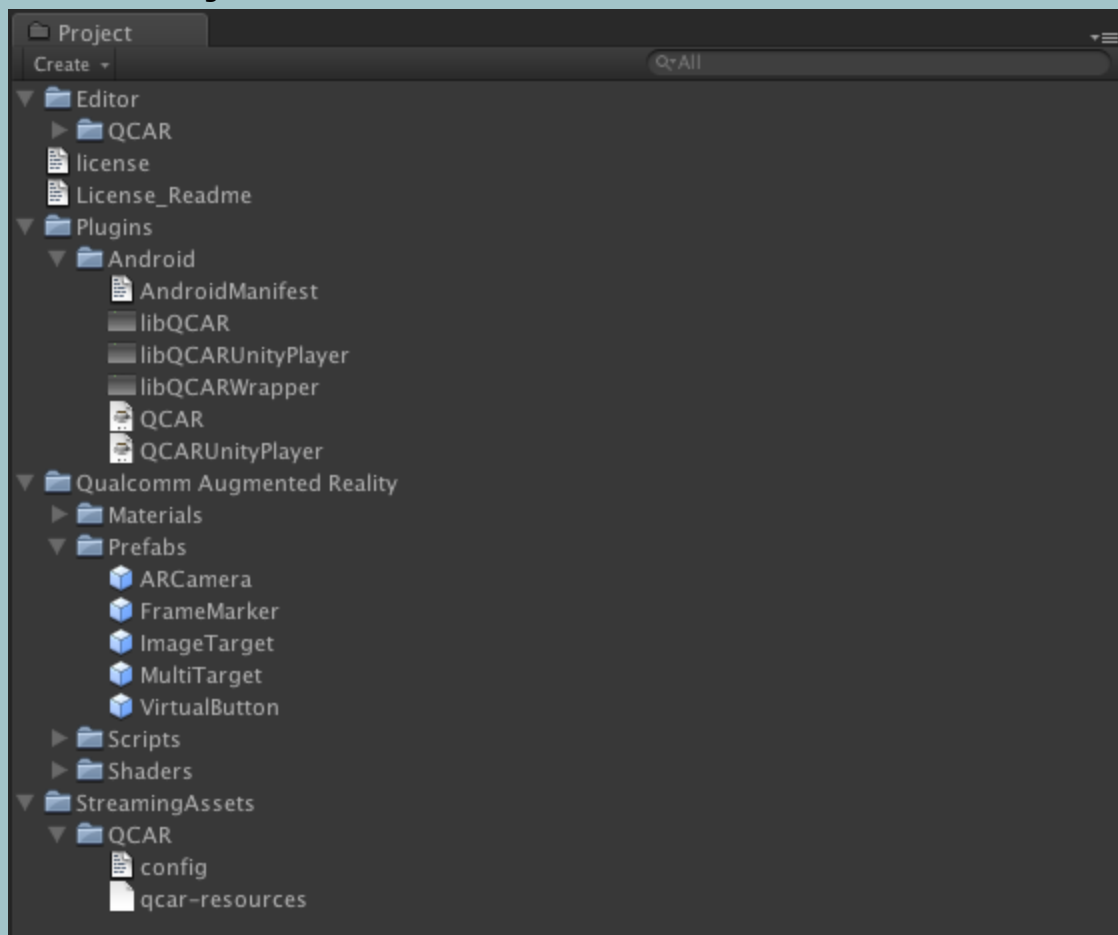


创建一个工程

创建一个新的**Unity**项目工程，选择**qcar-unity-android-xx-yy-zz.unitypackage** or **qcar-unity-ios-xx-yy-zz.unitypackage**导入到你的新工程当中，创建工程的时候不要使用空格来命名，这样会在之后创建**Xcode**程序的时候出现问题。

你需要添加一个**config.xml**文件和一个**qcar-resources.dat**文件到**StreamingAssets/QCAR**文件夹。你可以到<http://ar.qualcomm.at/qdevnet>创建这两个文件或从刚刚安装的**unitypackage**文件中得到它。你想获取的图形文件位于**Editor/QCAR/ImageTargetTextures**文件夹，这些纹理应用在**Unity editor**中的**plane**物体上。

你应该再**Unity**中看到以下这种目录结构：



以下是文件夹的内容：

- **Editor**

包含脚本所需的**Unity editor**插件

- **Plugins**

包含**QCAR SDK**所需的**Unity Android or Unity iOS**应用

- **Qualcomm Augmented Reality**

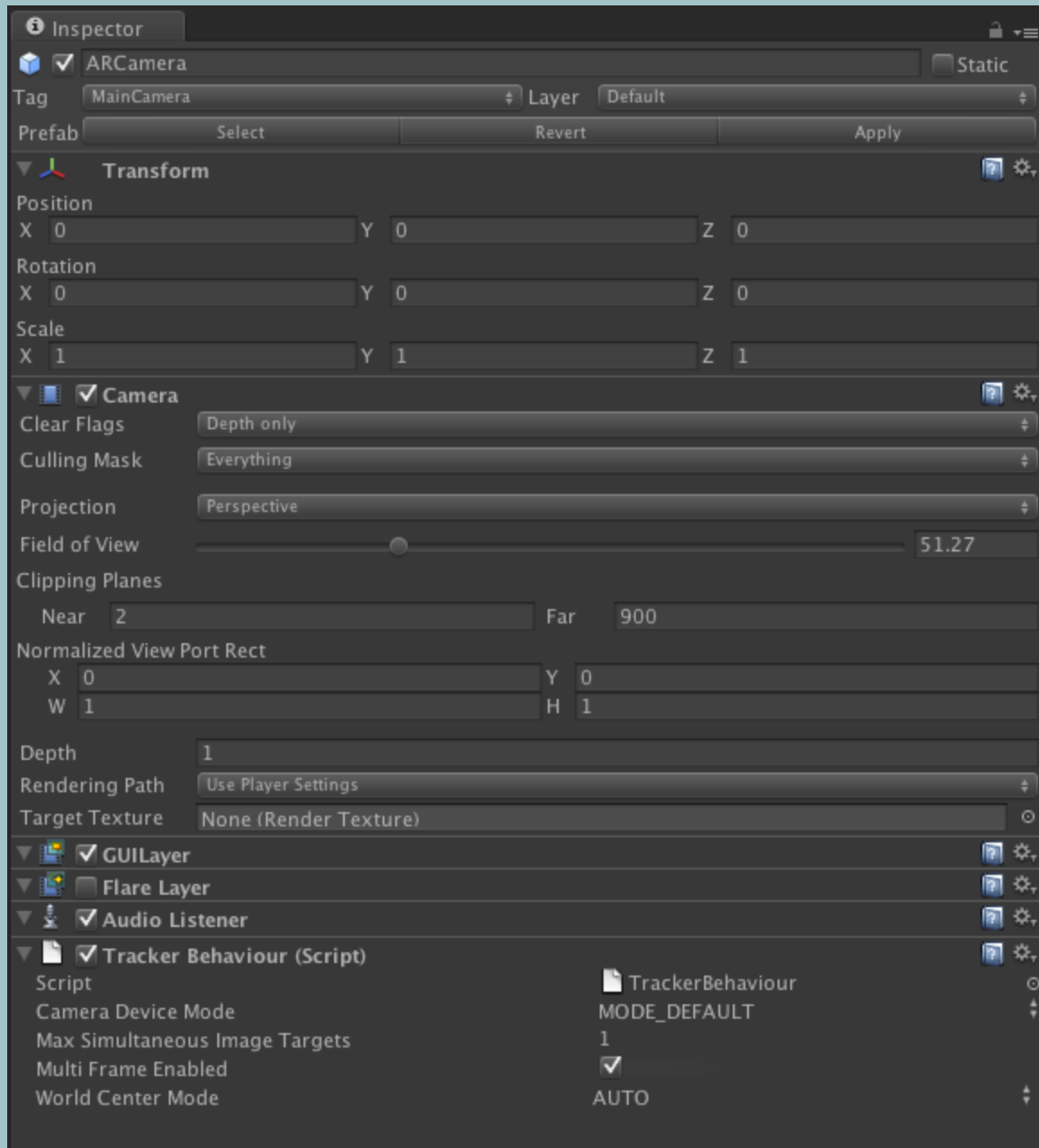
包含**prefabs**和**Unity**增强现实需要的脚本文件。

- **Streaming Assets**

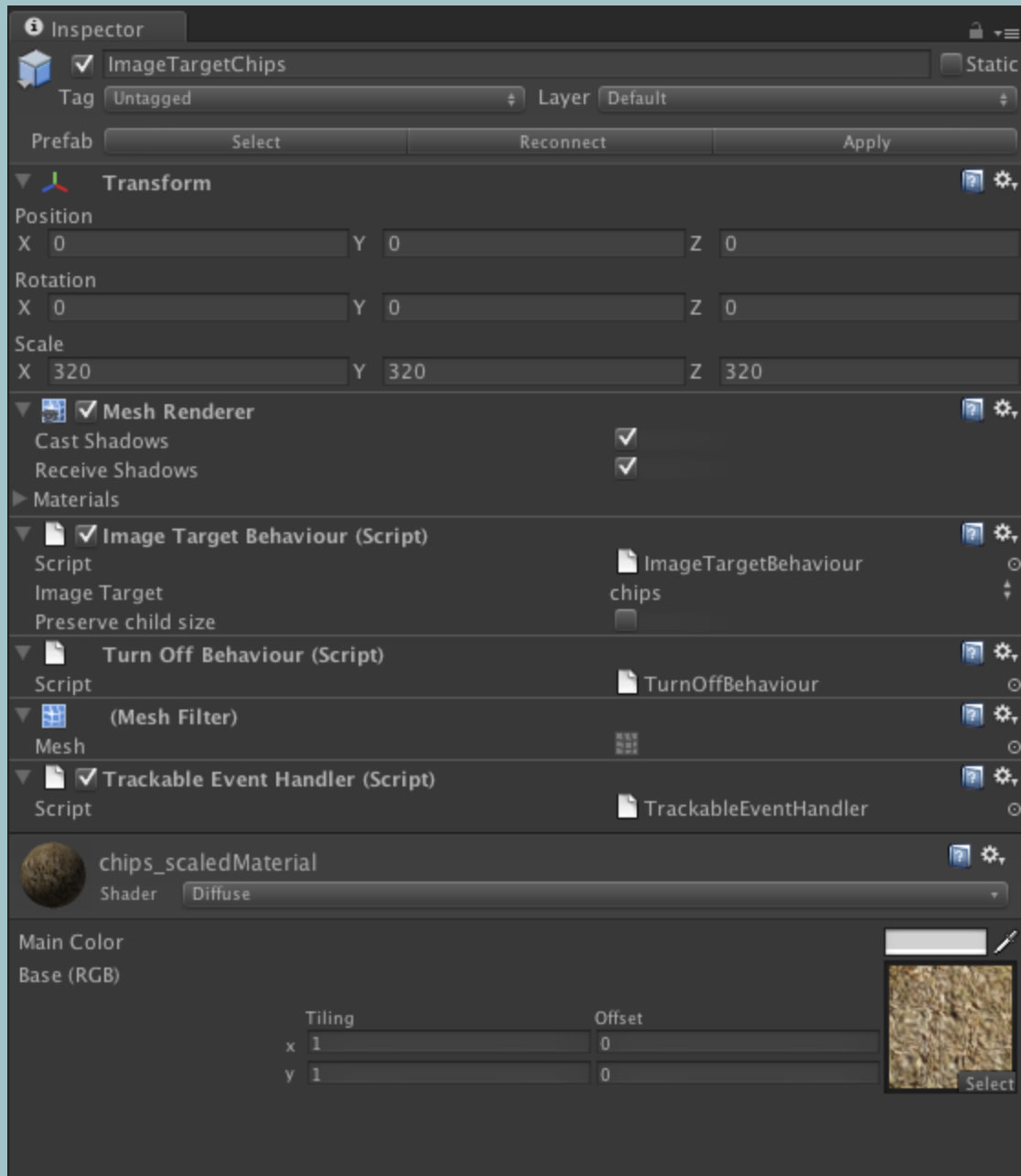
包含从网上下载的目标管理系统需要的配置和资源文件

添加**AR**资源和**Prefabs**到场景

现在你已经完全导入**QCAR**所有的**Unity**文件，你可以很容易的再你的项目当中创建增强现实。展开**Qualcomm Augmented Reality**和**Prefabs**文件夹。先拖拽一个**ARCamera prefab**到你的场景当中。用这个物体替代你原先场景中原有的**main camera**，你要删除场景中其他的所有**camera**物体。**ARCamera**可以负责渲染摄像头图像背景还有对场景中的数据做出反应。



接下来，拖拽**ImageTarget prefab**到你的场景中。这是一个图像目标预制物体。在场景中选择这个**ImageTarget**，看一下他的**Inspector**面板，上面应该有一个**TrackableBehaviour**，下面的**Image Target**写着属性名称。这个属性的下拉列表包含了所有可用的图像项目名称。名称会根据从网上下载回来的**config.xml**自动添加到这个列表当中。事例的程序包含了两个**Image Target**，一个“stones”和一个“chips”。使用它们或者到网站的“**My Trackables**”中自己创建。



添加三维物体并且可以被追踪

现在你可以添加**3D**物体给**Image Target**了。测试当中，创建一个简单的**cube**物体。把这个**cube**作为**Image Target**的子物体，你也许还需要在场景中再加一个灯光。

添加追踪时间处理功能

事先说一下，你虽然已经添加了这个**Cube**物体，并且作为**Image Target**的子物体，但并不意味着这个**cube**会自动根据摄像头做出变化，当**Image Target**消失的时候它仍然会被渲染出来。

这个时候我们就需要添加**TrackableEventHandler**类了。在这个事例中我们用了从**unitypackage**中导入的一个现成的脚本。

现在你只需要拖拽这个脚本到你的**Image Target**物体上。它会自动根据**Image Target**是否在视线内来判断**cube**是否可见，你也可以做一个淡出动画或者声音。相关的接口可以到[“Responding to Tracking Events”](#)查看。

配置应用程序

下一步就是配置你的应用程序到设备上。

Android

首先确定屏幕旋转方式为**Default Orientation**而不是**Auto Rotation**。**QCAR**不支持这项设置。接下来**Minimum API Level**设置成**Android 2.1 'Eclair' (API level 7)**或者更高版本。**Bundle Identifier**必须设置一个可用名。

你现在可以**build**程序了。首先你要保存一下场景，然后打开**build menu**。确保你的场景再**Scenes in Build**当中。都没问题后，点击**Build**。

ios

首先确定屏幕旋转方式为**Default Orientation**而不是**Auto Rotation**。**QCAR**不支持这项设置。接下来确保设置**Target Platform**不是**armv6 (OpenGL ES 1.1)**。**QCAR**目前支持**OpenGL-ES 2.0**。**Bundle Identifier**必须设置一个可用名。

你现在可以**build**程序了。首先你要保存一下场景，然后打开**build menu**。确保你的场景再**Scenes in Build**当中。都没问题后，点击**Build**。

当**Build**后**Unity**会产生**Xcode**项目。打开**Xcode**项目，应该会看到一个**PostProcessBuildPlayer**脚本，**library**当中也有**QCAR**的相关文件。

创建的**Xcode**项目当中有一个叫做**AppController.mm**的文件。可以通过它来定制**Unity**的程序。**PostProcessBuildPlayer**脚本默认使用

THREAD_BASED_LOOP设置，因为它会提供**QCAR**最佳性能。你可以更改这个选项来使自己的程序达到最佳的性能。

