**Unity3D教程：AssetBundles**

Posted on 2013年07月05日 by U3d / [Unity3D 基础教程](http://www.unitymanual.com/category/manual/unity3d-%e5%9f%ba%e7%a1%80%e6%95%99%e7%a8%8b)/被围观 64 次

AssetBundles是你能导出的从Unity到由资源文件组成的一些文件。这些文件在程序中有被专有的压缩格式。这能允许你通过流的方式（比如：模型、纹理、音频甚至整个场景文件）区分你将要使用的是哪个场景，他们被设计成一个很容易下载的文件到程序中。

AssetBundles能包含任意一种被unity认证的资源文件类型，这些资源类型视文件的扩展名而定。如果你想要包含自定义的二进制文件，你必须用“.bytes”的扩展名来重命名这些文件。在导入的时候unity把它们当作TextAsset文件。

**创建AssetBundles**

有三种创建AssetBundles的类方法： BuildPipeline.BuildAssetBundle, BuildPipeline.BuildStreamedSceneAssetBundle andBuildPipeline.BuildAssetBundleExplicitAssetNames.

BuildPipeline.BuildAssetBundle：允许你任意类型的AssetBundles

BuildPipeline.BuildStreamedSceneAssetBundle：仅仅包含场景文件的时候用到。

BuildPipeline.BuildAssetBundleExplicitAssetNames：创建的这些AssetBundles为每一个对象附有额外的参数来制定一个自定义的id或者name。

**下载AssetBundles**

推荐下载AssetBundles的方法是用WWW.LoadFromCacheOrDownload，一旦下载完成就可以获取assetBundle 的属性值。例如：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 1 | **string** url = "http://www.mywebsite.com/mygame/assetbundles/assetbundle1.unity3d"; |
| 2 | IEnumerator Start () { |
| 3 | *// Start a download of the given URL* |
| 4 | WWW www = WWW.LoadFromCacheOrDownload (url, 1); |
| 5 | *// Wait for download to complete* |
| 6 | **yield** **return** www; |
| 7 | *// Load and retrieve the AssetBundle* |
| 8 | AssetBundle bundle = www.assetBundle; |
| 9 | } |

当你使用下载后的.assetBundle属性值时，AssetBundle 对象已经被创建好了。使用LoadFromCacheOrDownload方法需要制定要下载的AssetBundle的版本号，如果AssetBundle 在cache中不存在或者说假如存在但是版本号低 ，那么就从直接下载AssetBundle 。否则AssetBundle 将从cache中加载。

**从AssetBundle 加载和卸载对象**

从下载好的数据中创建一个AssetBundle 对象，可以用三种不同的方法导入这个对象： AssetBundle.Load,AssetBundle.LoadAsync and AssetBundle.LoadAll.

AssetBundle.Load：用对象的名字标识符做参数导入对象。这个名字在工程视图里是可见的。你可以完全地通过一个对象类型导入方法来确定这个导入的特殊类型的对象。

AssetBundle.LoadAsync：和load方法差不多，但是加载资源时这个方法不限制主线程的运行。当导入较大的资源或者很多资源时，这个方法很好的避免了程序的暂停。

AssetBundle.LoadAll：能导入包含在AssetBundle的所有的对象。

卸载资源可以用AssetBundle.Unload方法。这个方法有一个布尔型的参数，这个参数表示是否卸载所有数据或者仅仅卸载被压缩的数据。假如程序从AssetBundle中用一些对象，并且想要释放一些内存，那么你可以用false参数。如果想要完全的卸载所有数据，你可以用true销毁从AssetBundle导入的所有资源。

**AssetBundle中的对象列表**

用AssetBundle.LoadAll从AssetBundle中获取一个包含所有对象的数组。

实例化AssetBundle

一旦从AssetBundle中加载了一个对象，就可以在场景中实例化了。例如：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | **string** url = "http://www.mywebsite.com/mygame/assetbundles/assetbundle1.unity3d"; |
| 02 | IEnumerator Start () { |
| 03 | *// Start a download of the given URL* |
| 04 | WWW www = WWW.LoadFromCacheOrDownload (url, 1); |
| 05 | *// Wait for download to complete* |
| 06 | **yield** **return** www; |
| 07 | *// Load and retrieve the AssetBundle* |
| 08 | AssetBundle bundle = www.assetBundle; |
| 09 | *// Load the TextAsset object* |
| 10 | GameObject go = bundle.Load("myGameObject", typeof(GameObject)) **as** GameObject; |
| 11 | *// Instantiate the GameObject* |
| 12 | Instantiate(go); |
| 13 | } |