**Unity3D的资源管理**

Posted on 2013年03月29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 373 次

[**Unity3D**](http://www.unitymanual.com)的资源管理。

**Asset Bundles**

制作：BuildPipeline.BuildAssetBundle()

加载：AssetBundle.Load()

卸载：AssetBundle.Unload(bool unloadAllLoadedObjects)

unloadAllLoadedObjects = true：释放解压数据，所有已实例化的资源亦被释放。

unloadAllLoadedObjects = false：释放解压数据，已实例化的资源不释放。

**Resource Folders**

制作：Project视图中任意名为“Resources”文件夹下的资源，无论场景中是否用到。

加载：Resources.Load()

卸载：Resources.UnloadUnusedAssets()，实例通过Object.Destroy()释放。

**工作方式：**

所有Resources资源及其相关资源都存储于resources.assets中，但如果有资源被某场景使用，该资源则被存储于该场景对应的sharedassets<0~n>.assets中。

如果编译目标平台为Streaming Web Player，则情况略有不同。你可以在设置中通过First Streamed Level With Resources一项指定所有Resources资源跟随哪一场景（Level）加载。在该模式下，所有先于“First streamed Level”加载的场景，用到的Resource资源同上，都存储于其对应的sharedassets中；后于“First streamed Level”加载的场景，其用到的Resource资源，却存储于resources.assets中。

**Streaming Web Players**

该模式中的Streaming是以场景（Level）为单位的，你可以在编译设置中指定各个场景的载入顺序，Unity Web Player会按该顺序自发异步加载各场景。同时，Unity也提供了通过WWW加载.unity3d场景文件的API，使开发者对场景加载有更灵活的控制权。

实际上，WWW可以用于网络下载任意资源（纹理、文本、二进制文件），视频和音频也可以通过WWW下载，而且可以做到边下载边播放。

**AssetBundle**

想打包进AssetBundle中的二进制文件，其文件名后缀必须为“.bytes”，Unity会将其视为TextAssets对待。

AssetBundle bundle = www.assetBundle;

获取WWW.assetBundle的一刻，由WWW下载的数据压缩包被解压，AssetBundle对象被创建，此时你可以用AssetBundle.Load()实例化资源了。

除非使用AssetBundle.LoadAll()获取所有资源，否则无法得到一个AssetBundle中的资源列表。一般做法是在该AssetBundle中放置一个定好名称的TextAsset，在其中维护一份所有资源的名称列表。

一个AssetBundle包可以引用其他AssetBundle包中的资源。制作时，需要调用BuildPipeline.PushAssetDependencies()和BuildPipeline.PopAssetDependencies()函数对；加载时，被引用的包需要先于引用包加载。

另外，重编这种包需要开启BuildAssetBundleOptions.DeterministicAssetBundle，以保证每次重编时所使用的资源ID都是一样的。

你可以在一个游戏里使用另一个游戏的AssetBundle，前提是该AssetBundle中所引用的资源，要么存在于该AssetBundle中，要么已经被载入了游戏中。为确保AssetBundle引用的资源被编进同包中，制作时可以使用BuildAssetBundleOptions.CollectDependencies选项。

**AssetDatabase**

Unity Editor使用AssetDatabase维护项目中的所有资源文件，Unity使用这套API取代传统的filesystem。这是一个Editor类，所以只有在名为“Editor”的文件夹下的Script中才有用。

BuildPipeline.BuildStreamedSceneAssetBundle()或BuildPipeline.BuildPlayer(..., BuildOptions.BuildAdditionalStreamedScenes)生成的场景包，使用WWW下载后，只需调用WWW.assetBundle，而不需AssetBundle.LoadAll()，即可激活包中的场景，即调用WWW.assetBundle后，你就可以通过Application.LoadLevel()来载入场景了。Unity运行时刻资源管理