**Unity3D AssetBundle的无缝地图实现**

Posted on 2013年02月26日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 462 次

[**Unity3D**](http://www.unitymanual.com) AssetBundle的无缝地图的实现，通过下面的制作过程和代码来逐步解释。

脚本如下：

Script\_Cube\_Trigger\_Terrain1.cs

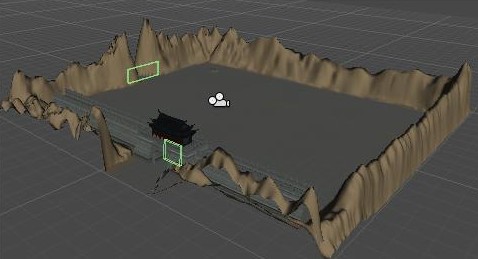
Script\_Cube\_Trigger\_Terrain3.cs

**Objects：**

1、人物资源，需要有CharactorController，不然无法做object的Trigger碰撞响应。

2、两个Cube，在其Inspector面板中，将BoxCollider下的IsTrigger属性勾选上，调整该Cube的形状，作为加载地形的触发点，我把它们放在地形交界点处，玩家碰触后加载对面的地形。

如下图：

[](http://www.unitymanual.com/wp-content/uploads/2013/02/2222.jpg)

Unity3D AssetBundle的无缝地图实现

3、将脚本绑定在两个cube上。

4、把各个地形块都制作好后，将地形上的Object作为地形对象的子节点。在Project视图里，创建几个prefab文件，将地形文件分别拖入其中，然后制作成AssetBundle。

5、制作过程中发现一个重大的问题，客户端模式Build出来的文件或者在编辑器中直观看到的结果，和Build成Web格式后的结果有些不一样。可能是WebPlayer的数据要依据固定的stream来源---WWW对象。而在CS脚本里所定义的成员或静态成员Object对象均无法保存当前WWW对象刚下载后的资源，造成下载部分和加载部分代码不能分开写，否则会因为丢失对象而加载失败，更糟糕的是在Web模式下，是不会报告什么错误的。所以这里解释下，两个脚本分别对应两个Cube的原因，就是各自保存了一份WWW对象，分别记录着自己所下载过的资源。注意不要用AssetBundleRequest对象来保存WWW对象中的asset。

脚本代码：

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

|  |  |
| --- | --- |
| 001 | **using** UnityEngine; |
| 002 |  |
| 003 | **using** System.Collections; |
| 004 |  |
| 005 | **public** **class** Script\_Cube\_Trigger\_Terrain1 : MonoBehaviour { |
| 006 |  |
| 007 | **int** Flag = 0; *//记录是否已经加载过该资源，避免多次加载* |
| 008 |  |
| 009 | *//AssetBundleRequest abr; string debugstr = ""; //调试信息* |
| 010 |  |
| 011 | **static** WWW wwwObj; *//每个场景中一个单独的对象* |
| 012 |  |
| 013 | **void** Start () |
| 014 |  |
| 015 | { |
| 016 |  |
| 017 | **if**(Flag == 0) |
| 018 |  |
| 019 | { |
| 020 |  |
| 021 | *//判断平台信息，如果是webPlayer就从网络上加载，如果是本地，就从本地资源目录中加载。* |
| 022 |  |
| 023 | **if**( Application.platform == RuntimePlatform.OSXWebPlayer || Application.platform == RuntimePlatform.WindowsWebPlayer ) |
| 024 |  |
| 025 | { |
| 026 |  |
| 027 | *//实例WWW对象，相当于从参数代表的网络地址上下载资源过程。* |
| 028 |  |
| 029 | *//GlobalConfig.GetConnectIP()方法可以取得当前机器的全局IP地址（这个例子中，服务器端和客户端在同一台机器上）* |
| 030 |  |
| 031 | wwwObj = new WWW("http://"+GlobalConfig.GetConnectIP()+"/AB/Terrain/Terrain1.unity3d"); |
| 032 |  |
| 033 | } |
| 034 |  |
| 035 | **else** |
| 036 |  |
| 037 | { |
| 038 |  |
| 039 | *// 因为玩家是玩Web形式发布的游戏，所以本地没有资源，所以这里可以省略了，本地调试的时候可以打开用。* |
| 040 |  |
| 041 | *// wwwObj = new WWW("file:///E:/123/Terrain1.unity3d");* |
| 042 |  |
| 043 | } |
| 044 |  |
| 045 | } |
| 046 |  |
| 047 | } |
| 048 |  |
| 049 | **void** Update () { |
| 050 |  |
| 051 | } |
| 052 |  |
| 053 | *// 当前对象（此例中就是那俩Cube）如果Collier的IsTrigger属性被勾选上了，那么当另一个Collier碰撞到此对象上，就会触发这个函数。* |
| 054 |  |
| 055 | **void** OnTriggerEnter(Collider other) |
| 056 |  |
| 057 | { |
| 058 |  |
| 059 | *// this if-type hierarchy is just for simple debug, we can put debug message in each curly braces* |
| 060 |  |
| 061 | **if**(Flag == 0) |
| 062 |  |
| 063 | { |
| 064 |  |
| 065 | **if**(wwwObj!=**null**) |
| 066 |  |
| 067 | { |
| 068 |  |
| 069 | print(wwwObj.isDone.ToString()); *//判断是否WWW对象已经下载完了* |
| 070 |  |
| 071 | **if**(wwwObj.assetBundle!=**null**) |
| 072 |  |
| 073 | { |
| 074 |  |
| 075 | **if**(wwwObj.assetBundle.mainAsset !=**null**) |
| 076 |  |
| 077 | { |
| 078 |  |
| 079 | *//load resouce into the assetbundle of wwwObject* |
| 080 |  |
| 081 | wwwObj.assetBundle.LoadAsync("Terrain1", typeof(GameObject)); |
| 082 |  |
| 083 | *//创建地形实例* |
| 084 |  |
| 085 | **if**(Instantiate(wwwObj.assetBundle.mainAsset ) != **null**) |
| 086 |  |
| 087 | Flag = 1; |
| 088 |  |
| 089 | } |
| 090 |  |
| 091 | } |
| 092 |  |
| 093 | } |
| 094 |  |
| 095 | } |
| 096 |  |
| 097 | } |
| 098 |  |
| 099 | *// 调试用的，通过GUI来调试，可视化，蛮方便的* |
| 100 |  |
| 101 | **void** OnGUI() |
| 102 |  |
| 103 | { |
| 104 |  |
| 105 | GUI.Label(new Rect(0,0,200,200),debugstr); |
| 106 |  |
| 107 | } |
| 108 |  |
| 109 | } |