**Unity3D：AssetDatabase**

Posted on 2013年07月04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 44 次

AssetDatabase是一个能获取工程资源的API，它提供一些方法比如：查找、加载、创建、删除和修改。Unity需要了解工程文件夹里的所有改变，假如想要获取或修改资源文件，就使用 AssetDatabase的API而不是文件IO流。

**导入资源**

Unity导入资源通常是用鼠标手动拖动到工程面板里，但是也可能需要脚本控制资源的导入，为了使用脚本导入可以使用AssetDatabase.ImportAsset方法，比如：

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

|  |  |
| --- | --- |
| 1 | **using** UnityEngine; |
| 2 | **using** UnityEditor; |
| 3 | **public** **class** ImportAsset { |
| 4 | [MenuItem ("AssetDatabase/ImportExample")] |
| 5 | **static** **void** ImportExample () |
| 6 | { |
| 7 | AssetDatabase.ImportAsset("Assets/Textures/texture.jpg", ImportAssetOptions.**Default**); |
| 8 | } |
| 9 | } |

**加载资源**

加载资源可以用这些方法：AssetDatabase.LoadAssetAtPath, AssetDatabase.LoadMainAssetAtPath, AssetDatabase.LoadAllAssetRepresentationsAtPath 和AssetDatabase.LoadAllAssetsAtPath.

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

|  |  |
| --- | --- |
| 1 | **using** UnityEngine; |
| 2 | **using** UnityEditor; |
| 3 | **public** **class** ImportAsset { |
| 4 | [MenuItem ("AssetDatabase/LoadAssetExample")] |
| 5 | **static** **void** ImportExample () |
| 6 | { |
| 7 | Texture2D t = AssetDatabase.LoadAssetAtPath("Assets/Textures/texture.jpg", typeof(Texture2D)) **as** Texture2D; |
| 8 | } |
| 9 | } |

**用AssetDatabase做文件操作**

由于unity采用元数据的文件方式，你需要对文件做创建、移动或者删除操作，可以用以下这些方法替代操作： AssetDatabase.Contains,AssetDatabase.CreateAsset, AssetDatabase.CreateFolder, AssetDatabase.RenameAsset, AssetDatabase.CopyAss

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

|  |  |
| --- | --- |
| 01 | **public** **class** AssetDatabaseIOExample { |
| 02 |  |
| 03 | [MenuItem (“AssetDatabase/FileOperationsExample”)] |
| 04 |  |
| 05 | **static** **void** Example () |
| 06 |  |
| 07 | { |
| 08 |  |
| 09 | **string** ret; |
| 10 |  |
| 11 | *// Create* |
| 12 |  |
| 13 | Material material = new Material (Shader.Find(“Specular”)); |
| 14 |  |
| 15 | AssetDatabase.CreateAsset(material, “Assets/MyMaterial.mat”); |
| 16 |  |
| 17 | **if**(AssetDatabase.Contains(material)) |
| 18 |  |
| 19 | Debug.Log(“Material asset created”); |
| 20 |  |
| 21 | *// Rename* |
| 22 |  |
| 23 | ret = AssetDatabase.RenameAsset(“Assets/MyMaterial.mat”, “MyMaterialNew”); |
| 24 |  |
| 25 | **if**(ret == “”) |
| 26 |  |
| 27 | Debug.Log(“Material asset renamed to MyMaterialNew”); |
| 28 |  |
| 29 | **else** |
| 30 |  |
| 31 | Debug.Log(ret); |
| 32 |  |
| 33 | *// Create a Folder* |
| 34 |  |
| 35 | ret = AssetDatabase.CreateFolder(“Assets”, “NewFolder”); |
| 36 |  |
| 37 | **if**(AssetDatabase.GUIDToAssetPath(ret) != “”) |
| 38 |  |
| 39 | Debug.Log(“Folder asset created”); |
| 40 |  |
| 41 | **else** |
| 42 |  |
| 43 | Debug.Log(“Couldn't find the GUID for the path”); |
| 44 |  |
| 45 | // Move |
| 46 |  |
| 47 | ret = AssetDatabase.MoveAsset(AssetDatabase.GetAssetPath(material), “Assets/NewFolder/MyMaterialNew.mat”); |
| 48 |  |
| 49 | if(ret == “”) |
| 50 |  |
| 51 | Debug.Log(“Material asset moved to NewFolder/MyMaterialNew.mat”); |
| 52 |  |
| 53 | else |
| 54 |  |
| 55 | Debug.Log(ret); |
| 56 |  |
| 57 | // Copy |
| 58 |  |
| 59 | if(AssetDatabase.CopyAsset(AssetDatabase.GetAssetPath(material), “Assets/MyMaterialNew.mat”)) |
| 60 |  |
| 61 | Debug.Log(“Material asset copied as Assets/MyMaterialNew.mat”); |
| 62 |  |
| 63 | else |
| 64 |  |
| 65 | Debug.Log(“Couldn't copy the material”); |
| 66 |  |
| 67 | *// Manually refresh the Database to inform of a change* |
| 68 |  |
| 69 | AssetDatabase.Refresh(); |
| 70 |  |
| 71 | Material MaterialCopy = AssetDatabase.LoadAssetAtPath(“Assets/MyMaterialNew.mat”, typeof(Material)) **as** Material; |
| 72 |  |
| 73 | *// Move to Trash* |
| 74 |  |
| 75 | **if**(AssetDatabase.MoveAssetToTrash(AssetDatabase.GetAssetPath(MaterialCopy)))*//Unity3D教程手册：www.unitymanual.com* |
| 76 |  |
| 77 | Debug.Log(“MaterialCopy asset moved to trash”); |
| 78 |  |
| 79 | *// Delete* |
| 80 |  |
| 81 | **if**(AssetDatabase.DeleteAsset(AssetDatabase.GetAssetPath(material))) |
| 82 |  |
| 83 | Debug.Log(“Material asset deleted”); |
| 84 |  |
| 85 | **if**(AssetDatabase.DeleteAsset(“Assets/NewFolder”)) |
| 86 |  |
| 87 | Debug.Log(“NewFolder deleted”); |
| 88 |  |
| 89 | *// Refresh the AssetDatabase after all the changes* |
| 90 |  |
| 91 | AssetDatabase.Refresh(); |
| 92 |  |
| 93 | } |
| 94 |  |
| 95 | } |
| 96 |  |

当你完成资源的修改后，应该调用AssetDatabase.Refresh方法来确认你的改变。