**Unity3D Editor类(Inspector)经验总结**

Posted on 2013年03月18日 by U3d / [Unity3D 软件操作](http://www.unitymanual.com/category/manual/unity3d-%e8%bd%af%e4%bb%b6%e6%93%8d%e4%bd%9c)/被围观 235 次

[**Unity3D**](http://www.unitymanual.com) Editor类(Inspector)经验总结。

1. Editor类中的全部控件描画，都是通过GUI完成的，所以不一定非要使用EditorGUILayout，使用GUILayout有时候更方便。

2. 因为1， 所以我们可以使用GUI直接做一些完全自定义的描画。例如在一张图片上画一个区域选择框。

3. Inspector光标定位Bug，具体表现为，代码无错误，无任何提示。但Inspector的编辑光标定位却不准。我做了一些简单的测试：

3.1 该问题与 Component 脚本(就是你写的类)的编写和 Editor 类(也就是 Inspector 的自定义类)的编写无关。

3.2 该问题的出现，似乎与编译缓存有关，出现问题后，我尝试过删除Assets同级目录下的obj文件夹(这个文件夹可以随意删除)或Library文件夹(删除这个会导致全部的关系链接丢失，需要手动恢复脚本间的挂接关系)，之后重新打开Unity，Inspector光标正常了。

3.3 经过反复测试发现，该bug似乎和MeshFilter有关。当你的Component中依赖了MeshFilter，但Mesh却是空，就会出现该bug。例如：

[RequireComponent(typeof(MeshCollider), typeof(MeshRenderer))]  
public class Test : MonoBehaviour  
{  
……  
}

所以我将MeshFilter的添加，延迟到了Start中。

void Start()  
{  
// 没有在定义时直接添加 MeshFilter 的依赖关系，是因为，直接那么做会导致Editor出错。  
// 虽然很诡异，但不清楚原因。所以只有在Start中手动添加了。  
MeshFilter mf = GetComponent<MeshFilter>();

if (mf == null)  
mf = gameObject.AddComponent<MeshFilter>();

……

}

4. EditorGUIUtility.LookLikeControls(80f) 很有用，它可以让所有控件的前置标签的宽度统一。虽然不够灵活，但至少比没有强，不过也仅仅比没有强一点。这句代码可以写在 public override void OnInspectorGUI() 中。

5. 有些资料中说，Editor的派生类不能被继承，这个是错误的。你可以从Control派生出Button，然后让Button的ButtonEditor再继承ControlEditor，这样可以省去很多功夫。

6. Editor写成模板类的时候，需要小心。public class ControlEditor<T> : Editor这样的定义，将不能编译出正确的Inspector。

7. 编辑环境下，和运行环境下，是两套不同的体系，为Inspector带来的问题是，动态数据无法直接被应用到Runtime(运行时)中。例如，你根据一个xml解析出一些数据。在编辑中，你会描画一个文件选择框(TextAsset对象)，然后选择一个xml文件。之后文件被解析，在ExecuteInEditMode类中，编辑时这些解析出的数据也可以正常使用。一旦你点击了运行那个小三角，一切解析出的临时数据都将灰飞烟灭，记得在Start或Update中检查必要的数据，发现没有就及时处理。所见即所得有时候只是幻觉。

8. 你可以定义一些GUIStyle，用来控制描画样式。这些GUIStyle可以是系统默认的，例如：

GUIStyle Style = new GUIStyle(EditorStyles.toolbarButton);

也可以完全的自定义。但记得，直接修改EditorStyles.toolbarButton会让系统所有用到该GUIStyle的地方一起变化。所以建议不要直接修改系统默认样式。

9. GUI中没有ComboBox或List这样的控件，只能自己做。

10. 如果你想用一个字符串数组生成一个弹出菜单，只有enum类型可以这么做。  
11. GUI中的控件并不是以面向对象思想设计的，自然，Editor类中的描画也就变得一塌糊涂。其底层封装应该是类似GDI+这样的描画库。但GUI的封装做的并不好。

综上，如果不是因为编辑器类不能用别的办法描画，我早不用GUI了。在生产环境下，还是放弃GUI吧，有很多实用的UI扩展库，例如EZGUI、NGUI。