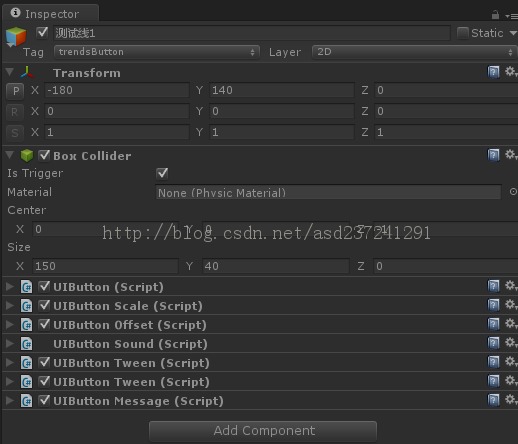
**原创文章如需转载请注明：转载自**[**脱莫柔Unity3D学习之旅**](http://blog.csdn.net/asd237241291)**QQ群：【119706192】 本文链接地址:**[**Unity3D NGUI动态创建按钮**](http://blog.csdn.net/asd237241291/article/details/9329631)

以动态创建服务器列表为例。

1. **public** UIAtlas mAtlas;
2. **public** UIFont mFont;
3. **public** **string** mSriteName;
4. **public** GameObject Playerlist;
5. /// <summary>
6. /// 动态加载一个NGUI按钮
7. /// </summary>
8. /// <param name="name"></param>
9. **private** **void** CreateServerButton(**string** key,**string** name, **int** index)
10. {
11. //获得深度（要创建button的Panle的深度）
12. **int** depth = NGUITools.CalculateNextDepth(PanleMe);
13. //创建button物体,命名、设tag、摆所在panle的相对位置。
14. GameObject go = NGUITools.AddChild(PanleMe);
15. go.name = name;
16. go.tag = "trendsButton";
17. go.transform.localPosition = **new** Vector3(-180 + (index % 3) \* 180, 140, 0);
19. //添加button的背景图片UISprite
20. UISprite bg = NGUITools.AddWidget<UISprite>(go);
21. bg.type = UISprite.Type.Sliced;
22. bg.name = "Background";
23. bg.depth = depth;
24. //背景图片使用的图集
25. bg.atlas = mAtlas;
26. //图集中使用的精灵名字
27. bg.spriteName = mSriteName;
28. bg.transform.localScale = **new** Vector3(150f, 40f, 1f);
29. //跳转位置（必要）
30. bg.MakePixelPerfect();
32. //如果有字体，添加UILabel
33. **if** (mFont != **null**)
34. {
35. UILabel lbl = NGUITools.AddWidget<UILabel>(go);
36. lbl.font = mFont;
37. lbl.text = name;
38. lbl.transform.localScale = **new** Vector3(22f, 22f, 1f);
39. lbl.transform.localPosition = **new** Vector3(0, 0, -1f);
40. lbl.color = Color.black;
41. lbl.MakePixelPerfect();
42. }
44. //添加碰撞（有碰撞才能接收鼠标/触摸），大小与Button背景一致
45. BoxCollider box = NGUITools.AddWidgetCollider(go);
46. box.center = **new** Vector3(0,0,-1);
47. box.size = **new** Vector3(bg.transform.localScale.x, bg.transform.localScale.y, 0);
49. //添加UIButton触发事件的必要组件，并关联之前生成的UISprite
50. go.AddComponent<UIButton>().tweenTarget = bg.gameObject;
51. //添加动态效果组件（大小、位移、音效）。（可选）
52. go.AddComponent<UIButtonScale>();
53. go.AddComponent<UIButtonOffset>();
54. go.AddComponent<UIButtonSound>();
56. /\*------ 一下是自己要用到的UIButton功能，可以绑自己需要的组件 ------\*/
57. //添加一个开启Player list Panel 的按钮事件
58. UIButtonTween serverListPanel = go.AddComponent<UIButtonTween>();
59. serverListPanel.tweenTarget = PanleNext;
60. serverListPanel.playDirection = AnimationOrTween.Direction.Forward;
61. serverListPanel.ifDisabledOnPlay = AnimationOrTween.EnableCondition.EnableThenPlay;
63. //隐藏自己panel
64. UIButtonTween hidemePanel = go.AddComponent<UIButtonTween>();
65. hidemePanel.tweenTarget = PanleMe;
66. hidemePanel.playDirection = AnimationOrTween.Direction.Reverse;
67. hidemePanel.disableWhenFinished = AnimationOrTween.DisableCondition.DisableAfterReverse;
69. //添加点击事件，参数是服务器ID（点击获取本服务器ID的下所有角色）
70. UIButtonMessage ubm = go.AddComponent<UIButtonMessage>();
71. ubm.target = Playerlist;
72. //给UIButtonMessage子增加传递‘参数’功能。
73. ubm.functionName = "Playerlist";
74. ubm.parameterStr = key;
75. }
77. /// <summary>
78. /// 清空动态生成的按钮
79. /// </summary>
80. **void** cleantrendsButton()
81. {
82. GameObject[] tbs = GameObject.FindGameObjectsWithTag("trendsButton");
83. **for** (**int** i = 0; i < tbs.Length; i++)
84. {
85. Destroy(tbs[i].gameObject);
86. }
87. }



http://blog.csdn.net/asd237241291/article/details/9329631