

# Demo3-加载UI

2018年3月6日 17:41

对于Demo2，可能写的有点乱，因为我没想到切换流程和场景是要涉及到多个知识点的，我的错。

这次我们简单一点，来演示一下如何用Game Framework框架加载UI，不涉及其他操作。

## 1.创建基础项目

请确保你已经阅读过Demo1和Demo2，或者你已经知道如何创建Game Framework空项目，以及切换流程的方式。

我们需要新建一个Demo3项目，并且把初始场景和菜单场景给创建好，文件列表如下：

Demo3：初始场景

Demo3\_Menu：菜单场景

Demo3\_ProcedureLaunch.cs：初始流程脚本

Demo3\_ProcedureMenu.cs：菜单流程脚本

至于初始场景要把Game Framework的基础组件预制体拖到场景里这种事情我就不啰嗦了。

## 2.初始流程脚本

Demo3\_ProcedureLaunch脚本的内容如下：

```
using System.Collections;
using System.Collections.Generic;
using GameFramework;
using GameFramework.Procedure;
using UnityEngine;
using UnityGameFramework.Runtime;
using ProcedureOwner =
GameFramework.Fsm.IFsm<GameFramework.Procedure.IProcedureManager>;
public class Demo3_ProcedureLaunch : ProcedureBase {
    protected override void OnEnter(ProcedureOwner procedureOwner)
    {
        base.OnEnter(procedureOwner);
        SceneComponent scene
            =
UnityGameFramework.Runtime.GameEntry.GetComponent<SceneComponent>
();
        // 切换场景
```

```

        scene.LoadScene("Demo3_Menu", this);
        // 切换流程
ChangeState<Demo3_ProcedureMenu>(procedureOwner);
    }
}

```

即，在一开始的时候就切换到菜单场景和菜单流程。

### 3.菜单流程脚本

Demo3\_ProcedureMenu脚本内容如下：

```

using System.Collections;
using System.Collections.Generic;
using GameFramework;
using GameFramework.Procedure;
using UnityEngine;
using UnityGameFramework.Runtime;
using ProcedureOwner =
GameFramework.Fsm.IFsm<GameFramework.Procedure.IProcedureManager>;
public class Demo3_ProcedureMenu : ProcedureBase {
    protected override void OnEnter(ProcedureOwner procedureOwner)
    {
        base.OnEnter(procedureOwner);
        // 加载框架UI组件
        UIComponent UI
            =
UnityGameFramework.Runtime.GameEntry.GetComponent<UIComponent>();

        // 加载UI
        UI.OpenUIForm("Assets/Demo3/UI_Menu.prefab",
"DefaultGroup");
    }
}

```

这里就要稍微啰嗦一下了，UIComponent是Game Framework框架的基础组件，专门用于处理界面相关的逻辑。

比如我们现在要加载UI，就需要用到它的OpenUIForm的函数。

函数的第一个参数需要的是UI预制体所在的路径，需要完整路径，从Assets目录开始。

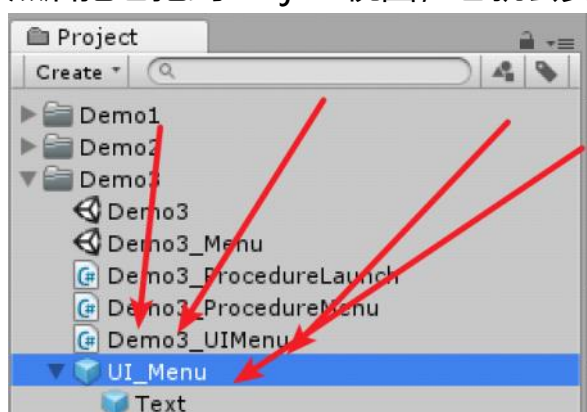
函数的第二个参数需要的是UI所属的分组，比如不同的UI分组显示的层次会不一样，这里我们随便填一个“DefaultGroup”就好，应该没关系的吧，呵呵。（旁白：总感觉好随意啊！不是很靠谱的感觉）

#### 4.UI\_Menu.prefab预制体

UI\_Menu是我们自己创建的预制体，怎么创建的，我就不说了，这个是Unity3D基础中的基础，如果连这个都不知道的话...（旁白：你大声说出来）

那我就稍微说两句吧，直接在Hierarchy视图里创建一个TextUI，然后调调文字，改改颜色，大家自己喜欢就好。

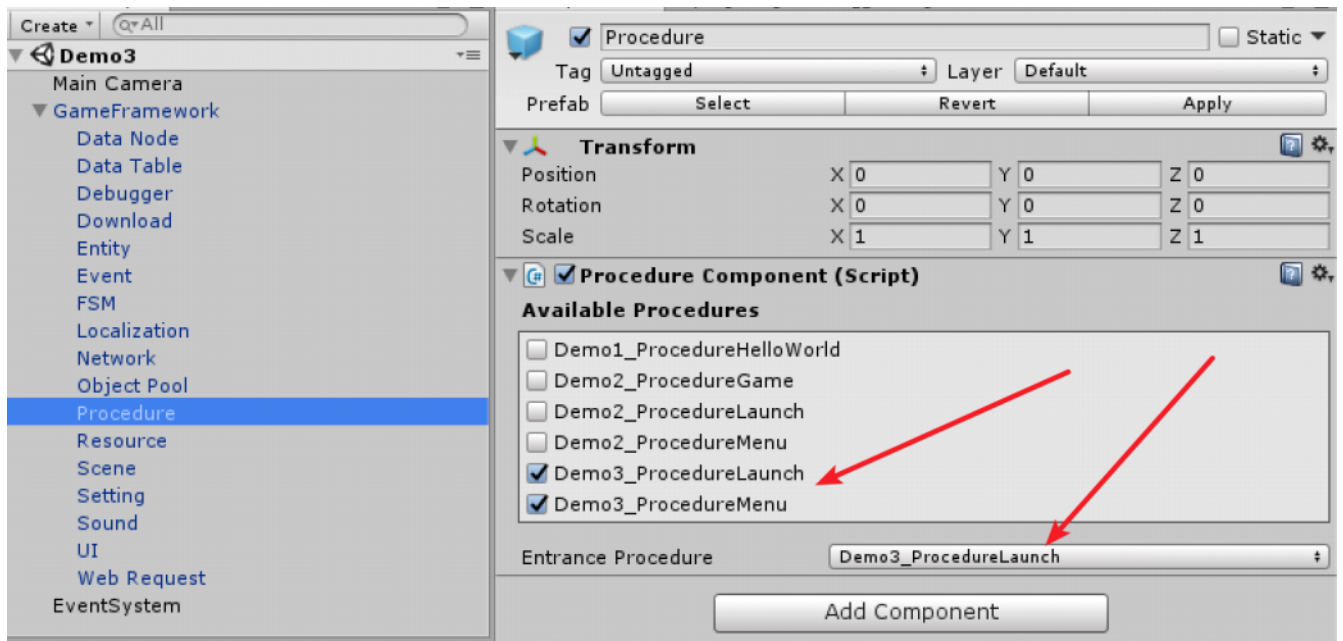
然后把它拖到Project视图，它就会变成一个蓝色的东西：



这个就是预制体了，Hierarchy视图里的那个UI就可以删掉了。

#### 5.运行测试

在运行之前，记得把我们的流程给勾上，初始流程也要选上：

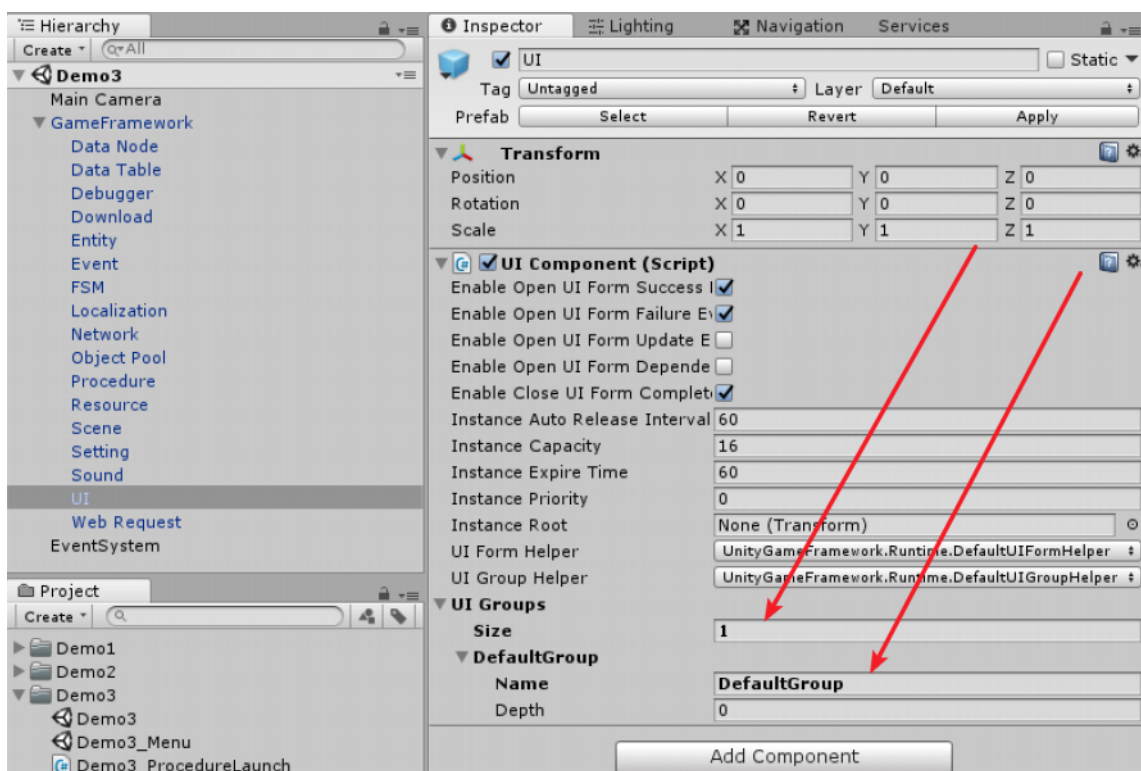


然后运行Demo3场景，你们会发现...（旁白：报错了！）

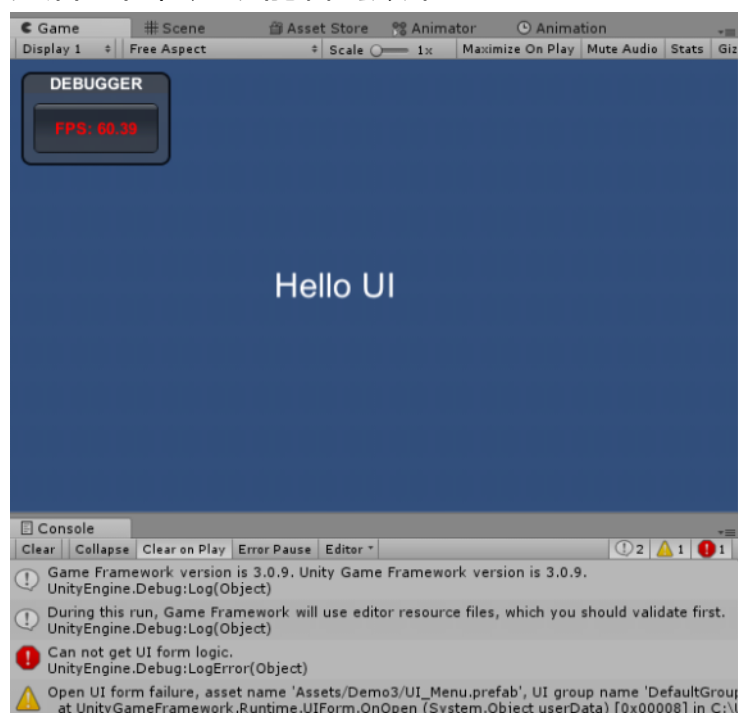
对，它报错了，说找不到UI什么的...

果然，UI分组的名称不能乱填呢，呵呵的（旁白：刀我已经买好了，大家说句话就行）

选中Demo3场景的Hierarchy视图里的UI组件，然后给它添加一个分组即可：



然后运行，应该能看到效果：



OK，已经看到我们的UI加载出来了，本次Demo完美结束。（旁白：等等！又报错了！）

那么，下一篇我们再演示点别的什么吧...（旁白：别走！报错了！）

## 6.UI逻辑处理

好吧，果然逃不掉呢。

是的，又报了新的错误：Can not get UI form logic，意思是，无法获得能量，运行失败...才怪呢。是因为缺少UI的逻辑处理脚本了。

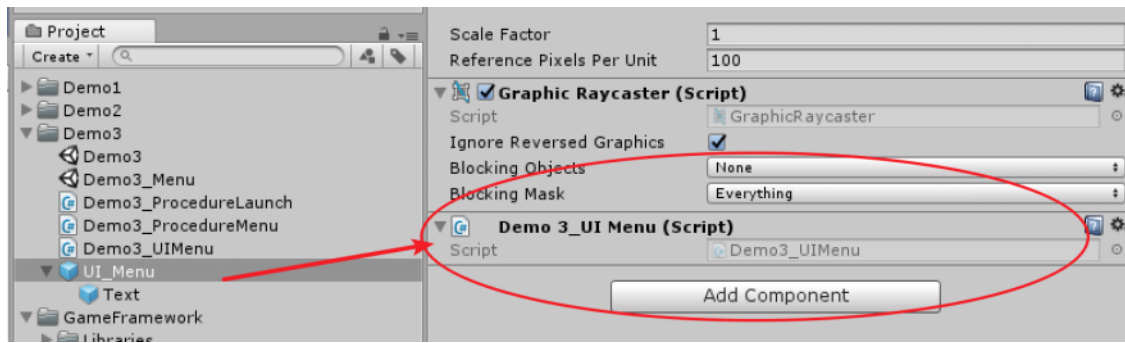
我们新建一个脚本，命名为Demo3\_UIMenu.cs：

```
using GameFramework;
using UnityEngine;
using UnityGameFramework.Runtime;
public class Demo3_UIMenu : UIFormLogic {

}
```

对，就是这么简单，但是必须继承UIFormLogic，这个脚本是用于处理UI逻辑的。

然后把这个脚本拖到UI\_Menu预制体上：



OK, 真的没问题了。