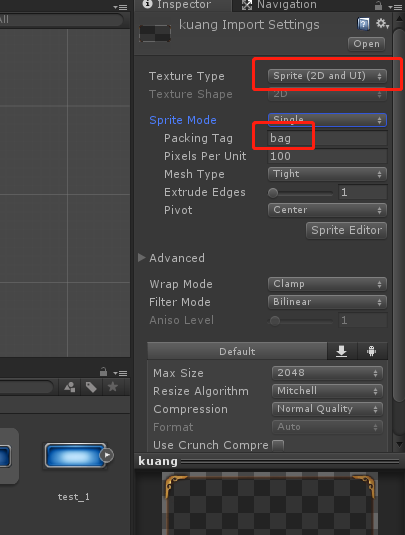
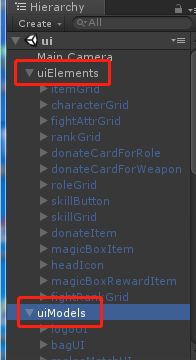
UI制作流程

1. 启动Unity选择Assets/source/unity/ui.unity场景
2. 需要新增图片资源的，把所需图片放置到Assets/source/uiTexture下的对应界面的目录下,如Assets/source/uiTexture/bag为背包所用图片资源,然后选中你新增的资源，设置纹理格式和打包tag



Tag设置为你的ui目录名即可

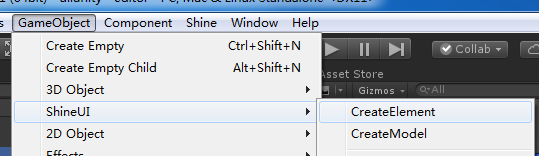
1. ui设计结构

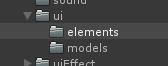


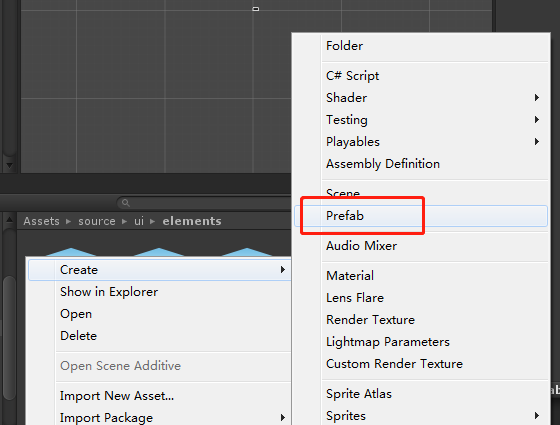
1.其中uiElements是存放ui元素(预制)的节点，uiModels是存放 ui 界面的节点。

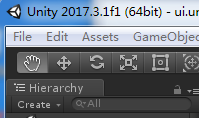
例如我们要做一个背包，里面有物品格子，那么itemGrid可做成 一个element预制，然后创建一个bagUI的界面，里面拖放若干 个itemGrid来实现。

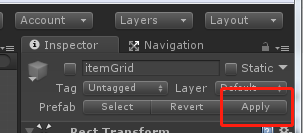
2.创建uiElement预制，必须使用shine的创建指令,不能自己直接 创建空GameObject。



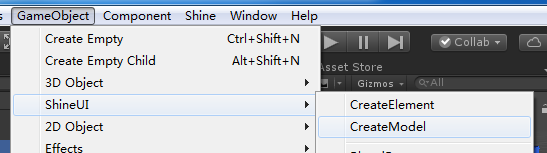
创建完element后,在目录下创建prefab,就是普通的右键标 准创建



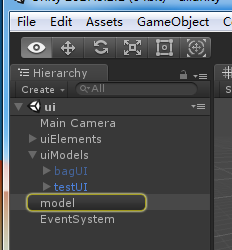
然后将你创建好的Hierarchy视图的element对象拖到 你刚创建的prefab上，完成更新预制操作，并且以后每次做完element/model都要更新 一下预制可点击右侧的apply按钮(未来工具会做导出时自动apply)



1. 创建model,执行GameObject/ShineUI/createModel创建

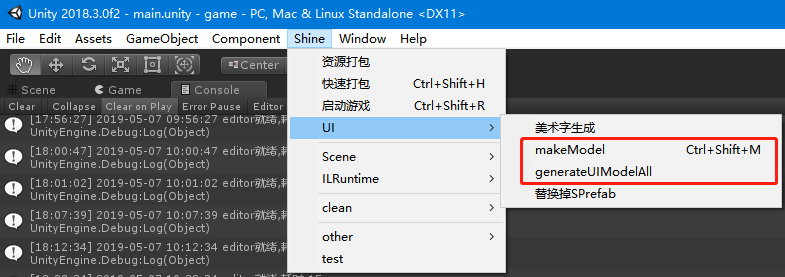


会出现一个名为model的GameObject

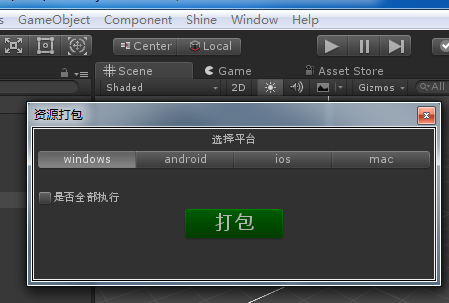


将其重命名为XXXUI(注意,UI大写，而且必须加)

之后开始编辑UI界面，编辑完成后执行makeModel来生成UIModel类

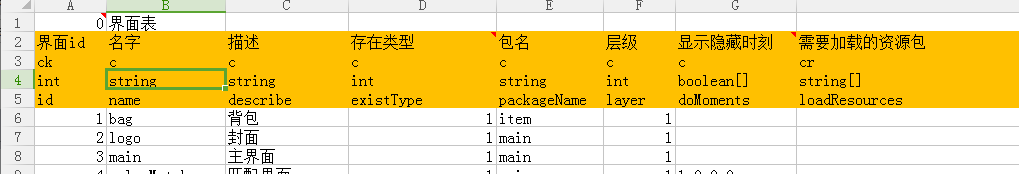


然后再执行资源打包

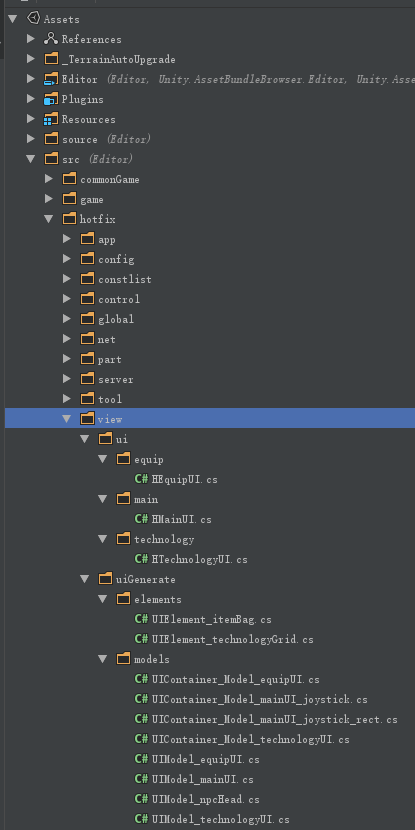


来将编辑好的界面打出bundle

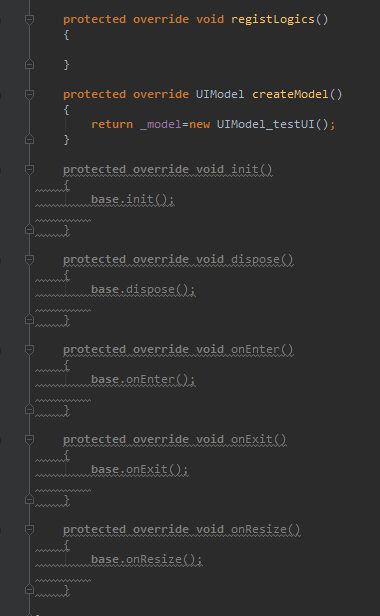
1. 在develop\common\config\hotfix目录下找到下图的表写好name，packageName，layer这些关键字段后执configExport.py会生成 对应的UI类代码

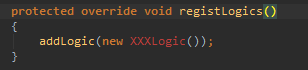


1. 代码使用

下图为客户端ui相关类存放位置事宜图 放在hotfix目录下因为ui支持热更

UI类内部的接口



registLogics:用来补充界面逻辑体,如

Init:,与dispose成对，每个界面只执行一次（除非开了界面析构，并且界面被dispose 过了，下次再开会再调用一次init）（只会在界面资源齐备，并且创建好Model后调用）

dispose:与init成对，界面析构时调用

onEnter:与onExit成对,界面打开时调用(如未init,则不会调用)

onExit:与onEnter成对,界面关闭时调用(如未init,则不会调用)

onResize:屏幕尺寸改变时调用（适配用）(如未init,则不会调用)

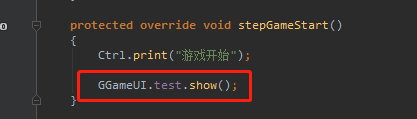
基本思路，在init中书写注册性质代码，例如给Button加click，一些控件的初始化，

以及registPlayerEventListener(注册Player的GameEvent部分)

在onEnter中做显示更新（除了registPlayerEventListener之外的，因为注册过的会自动 调用）

在enExit中做显示效果中断逻辑

界面显示调用如下



如需传参显示，则自行增加具象化传参接口，或使用默认传参接口

