

界面功能制作

一、界面预制体

UI界面的调整制作任务中有很很大一部分的修改是针对UI界面预制体的，灌篮（竞技）的界面相关的重要路径如下：

UI界面预制体目录为：**SlamDunk\Assets\data[prefab]ui_prefab**

UI图集（主要存放的为使用的小图）路径：**SlamDunk\Assets\data\ArtControl[UIAtlas]**

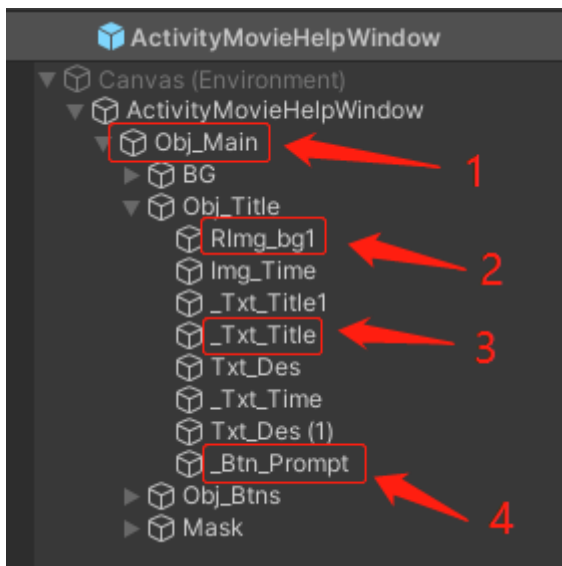
UI大图路径：**SlamDunk\Assets\data\ArtControl[texture]ui_texture**

界面预制体均挂载了处理控件的脚本 UIBase 该脚本挂载了逻辑需要控制的所有控件，控件包括：Button，Image，RawImage，Slider等常用组件

灌篮竞技的界面预制体在制作时需要遵循以下基本规范：

（一）命名规范

以下面的界面预制体为例：首先，预制体中的固定控件（即不需要逻辑进行控制的控件，**死控件**），命名需要设置为XXX_Name的方式命名，例如：Obj_Main, ObjTitle, Txt_Title，即 类型_名称；需要逻辑来进行控制的控件（**活控件**）则在命名前需要添加下划线，例如：_Txt_Title1 _Txt_Time等

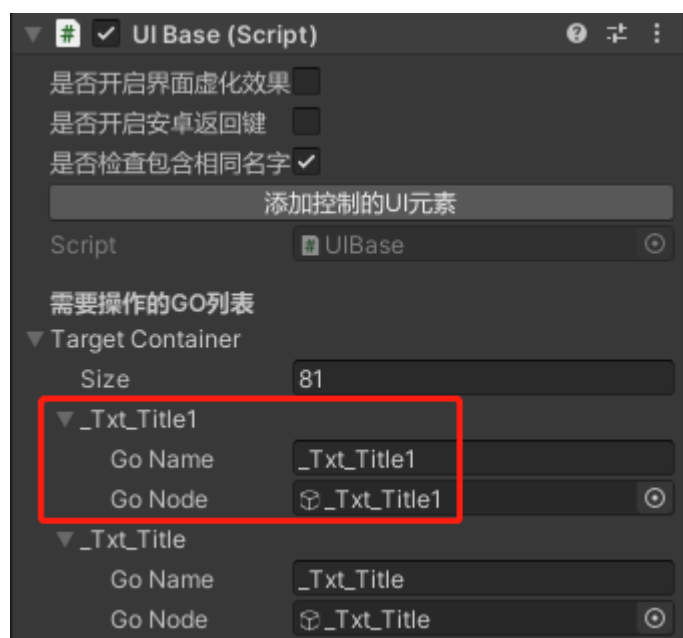


设立这样的规则是为了两个作用：

- 区分静态控件和动态控件，以便于在调整时快速定位组件，便于搜索区分
- 在UIBase脚本中内置了自动绑定控件的功能，点击 添加控制的UI元素 按钮，即可将前缀为下划线的组件添加到UIBase中

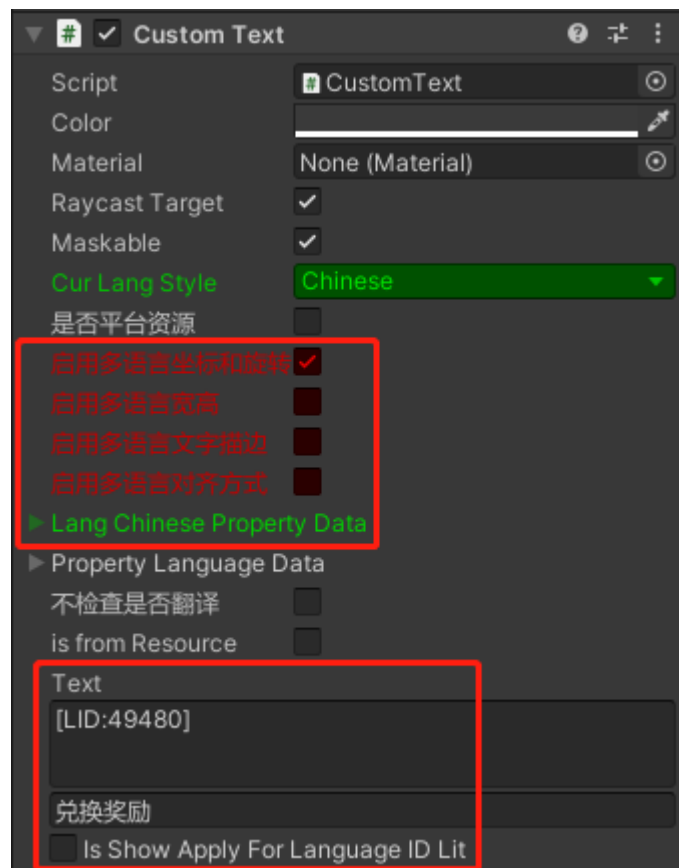


在点击按钮后，自动添加控件时会设置同名的名称，以便逻辑通过名称直接控制控件；若手动添加的控件，则需要检查GoName 是否与控件名相同



(二) 组件使用规范

Text 文本控件，目前灌篮竞技项目中，UI界面预制体中使用的基本是：**CustomText** 组件，该控件支持了一些多语言的便捷功能，部分文本配置了多语言，在 Text 中输入文本，即可判断到是否为多语言文本，是则会在下面显示 LID，显示了LID即可将该 LID设置到 Text 中，即可将该组件内容变为多语言可控制的内容

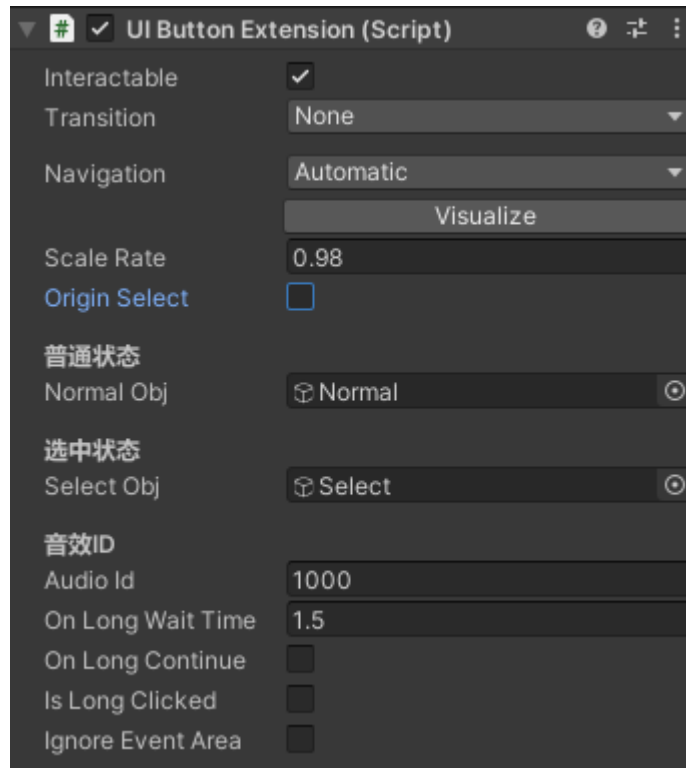


RawImage 与 **Image** 图片组件，灌篮竞技中常使用两类图片组件用于处理图像的显示，RawImage使用的资源为 Texture 格式，直接使用贴图进行显示，非精灵图片，仅为贴图资源，命名时前缀为：RIImg；Image组件使用的资源为Sprite，若图片是需要进行切图的，则需要使用Image控件，来正确设置图片

使用时需要注意：尺寸规格较大的图片，一般是直接作为 Texture 放到 RawImage 中的，此类图片不会打到图集中，所以存放目录为：**SlamDunk\Assets\data\ArtControl[texture]ui_texture**

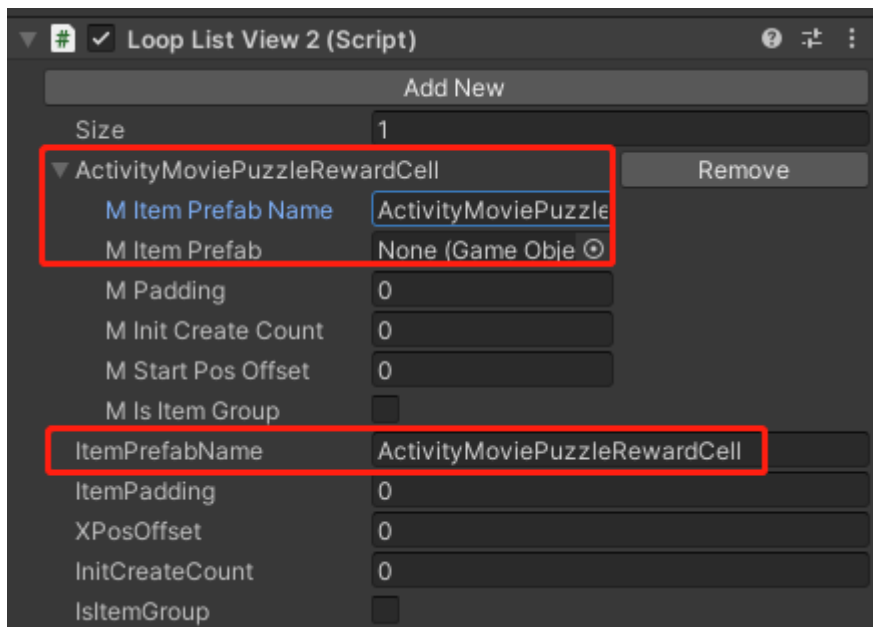
图标类的Icon小图，界面中使用的通用按钮背景，文本背景图则是会打到图集中，且其中很大一部分图片需要进行切图设置，这类图片一般存放目录为：**SlamDunk\Assets\data\ArtControl[UIAtlas]**

Button按钮组件，项目中有时会使用 UI Button 控件，使用方式即常规方式，不作赘述。主要说明下自定义Button扩展的使用方式，**UIButtonExtension**，按钮扩展组件，继承 **UnityEngine.UI.Selectable**，实现了按钮常用的接口；如下图，支持了：三态状态，按下缩放倍率变化，音效播放等功能

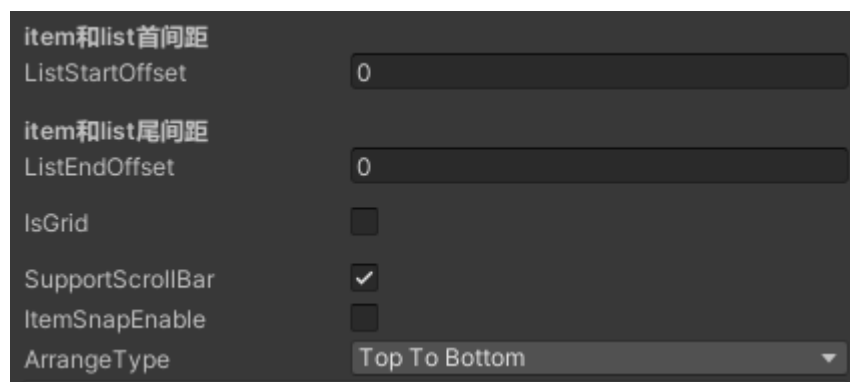


LoopListView 以及 LoopListView2，自定义循环列表组件，用于控制循环列表，以LoopListView2为例，需要做以下设置：

- 添加要控制的循环子界面，此处添加了 ActivityMoviePuzzleRewardCell 作为控制的子界面



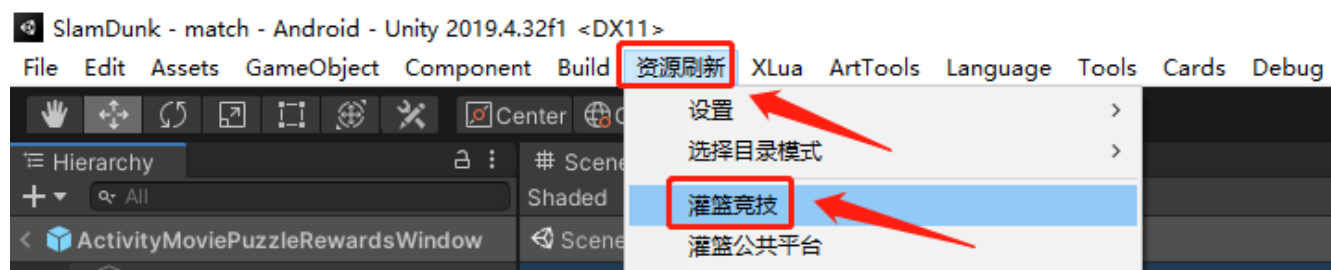
- 可设置Item List 的首尾间距，单个子界面为格子，是否支持滑动条，吸附是否开启以及排列方式



代码逻辑中的处理可跳转到：

(三) 预制体编辑规范

在界面预制体编辑时，若 UIBase 中设置的控件列表有增删改的变动，需要在保存后，进行资源刷新，即修改结束后 > 保存预制体 > 点击Unity顶部工具栏：资源刷新 > 选择 灌篮竞技 按钮，对竞技的资源进行刷新



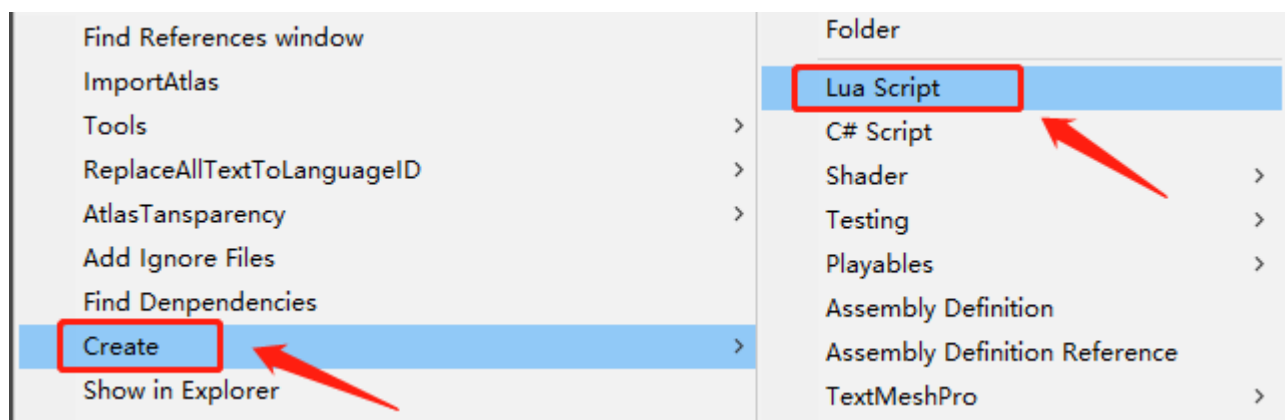
刷新后的界面才能重新进入游戏后显示变更的内容，否则可能会加载修改之前的内容

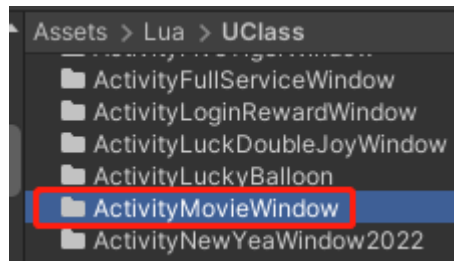
- 存在一个偶现的问题：在预制体编辑模式下，对控件位置，内容进行调整，调整后保存退出预制体模式，重新进入，预制体丢失了刚刚的修改调整，若出现这种状况，需要将控件设置的位置取整，再关闭，重开一下控件

二、逻辑处理

(一)脚本创建与初始化设置

灌篮竞技的UI功能逻辑使用Lua脚本来进行编辑，Lua脚本的创建路径为：**SlamDunk\Assets\Lua\UClass** 在Unity编辑器文件夹下，右键选择创建Lua脚本，注意根据不同的功能划分，将脚本放在不同的文件夹中，如下图：将脚本创建放在ActivityMovieWindow文件夹中



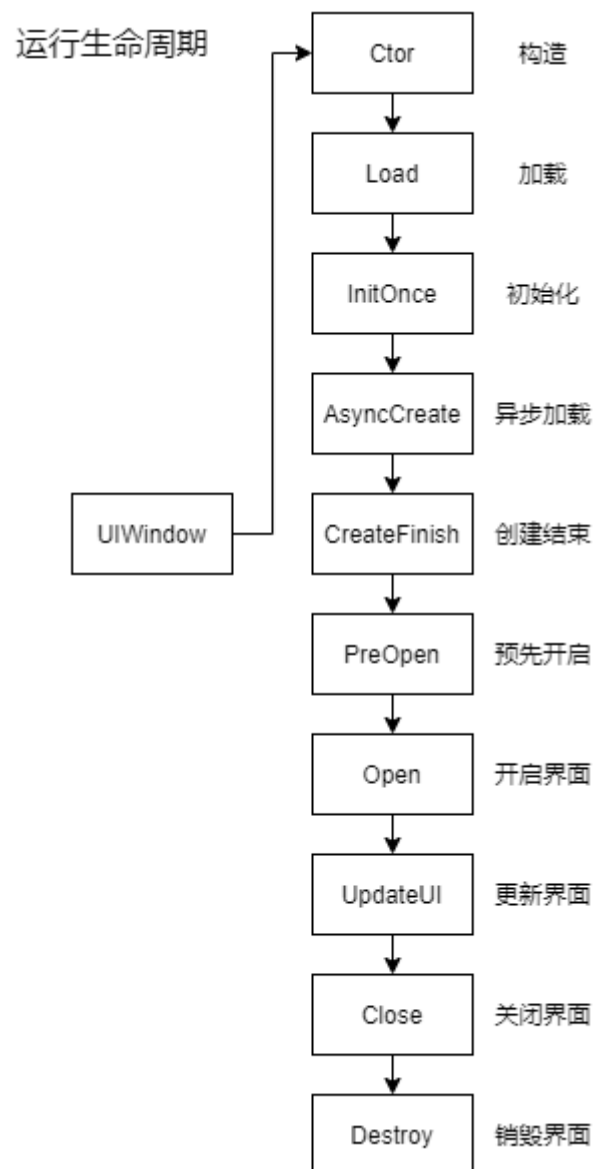


与界面相关的脚本都需要统一放在文件夹下，且 **脚本命名** 与 **界面预制体** 命名相同，便于脚本和预制体对应

初始化脚本时需要将脚本放在 UIGroup.lua 中进行注册，这样在初始化时才会根据名称将界面预制体进行预先加载

```
--UIGroup
local UIGroup = {}
UIGroup.UI = {
    --某个界面组
    xxxwindowGroup = {
        --属于该组的多个界面
        { name = "xxxwindow1", active = true, isFastLoad = true },
        .....
        { name = "xxxwindow2", active = false },
    }
}
```

创建Lua脚本时，会根据模板自动生成一些默认方法，主要为该界面的逻辑生命周期函数，如下图所示：



界面的功能实现就基于这几个生命周期函数进行设计制作，在不同的生命周期阶段进行逻辑处理，例如：在Open时初始化参数，进行事件绑定，事件监听