# KGUI使用说明

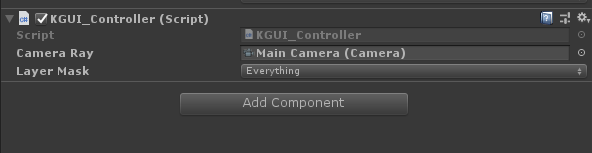
KGUI是基于Unity UGUI之后的在一次层次封装，意在解决UGUI一些不能满足的问题，KGUI集成了按钮（KGUI\_Button）,表格（KGUI\_Table）、背包（KGUI\_Backpack）、下拉框（KGUI\_Dropdown）、滚动条（KGUI\_sCRollBar）、滚动视图（KGUI\_ScrollView）、容器（KGUI\_Panel）等等。

KGUI的使用完全脱离与Unity自带的输入端（触摸、鼠标），只需要指定KGUI控制端的输入端口，KGUI将会完全运行。

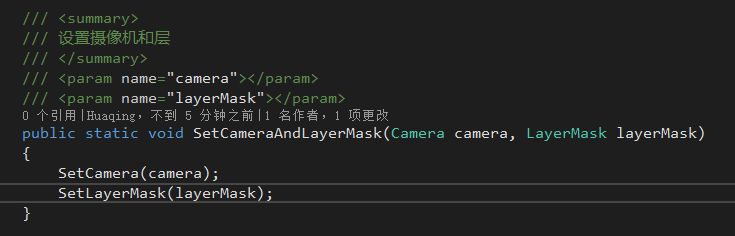
**最终目的：基于Unity GUI重新写一套属于自己的GUI。**

## 初始化

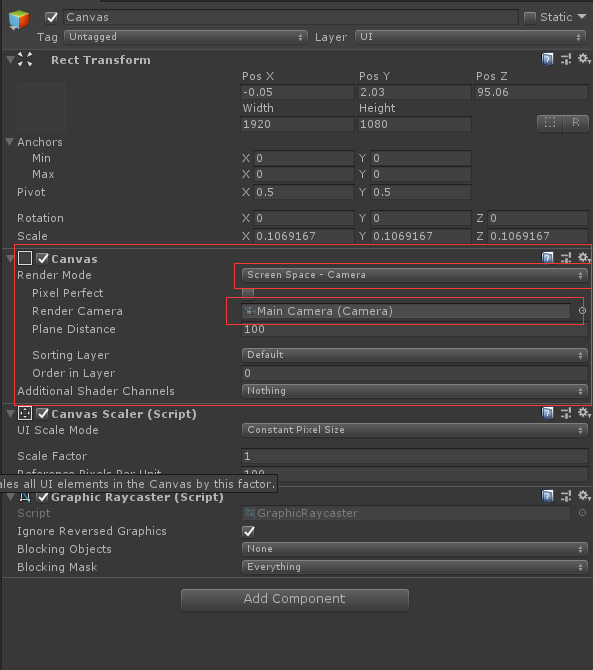
1. 将KGUI导入到项目中
2. 新建一个预制物体，添加KGUI\_Controller.cs脚本。选择UI发射线的摄像机以及UI射线层。



同时也可以代码设置CameraRay层以及LayerMask层。



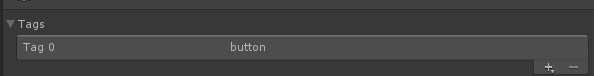
1. 如果此时你使用的是UGUI的Canvas，在Canvas下布局各种UI。



需要设置Canvas下的RenderMode属性、Render Camera属性。

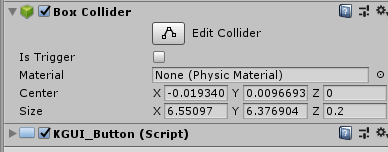
1. Button Tag添加

前三步设置完毕后，在编辑器中添加buttonTagID。

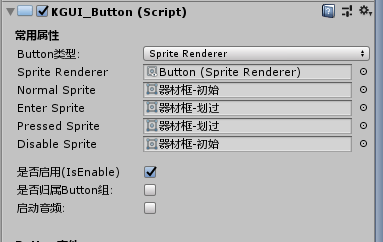


## Button(按钮)

1. 在物体中添加KGUI\_Button组件。
2. 设置该KGUI\_Button物体的碰撞体相关信息。



1. 可选择Button交互类型

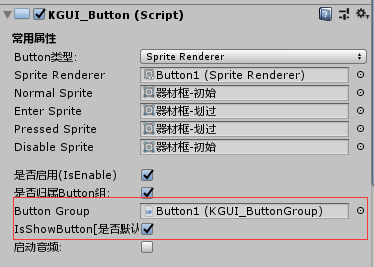


* None类型：当交互时没有任何处理
* Image类型：基于UGUI中的Image，当进行移入Enter、移除Exit、按下OnDown、禁用OnDisable时的Button表现。
* SpriteRenderer类型：应用与非UGUI的Button，以Sprite方式组成的Button。
* Object类型：应用与非Sprite，当交互时则是更换相关Button物体，以满足多样性。

1. IsEnable

当IsEnable为Ture时，该按钮激活，当为false时，按钮禁用。

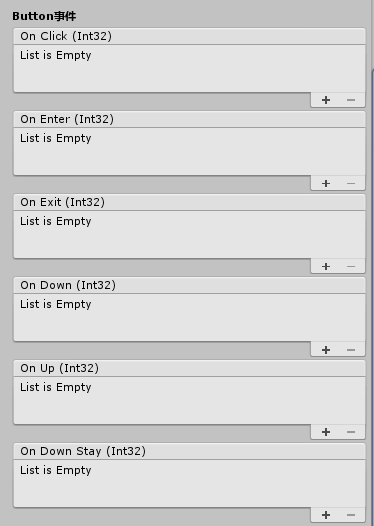
1. Button组
2. 选定一个物体，添加KGUI\_ButtonGroup组件。
3. 在需要设置为指定组的Button中，勾选“是否归属Button组”。然后将Button组对象赋予指定Button。



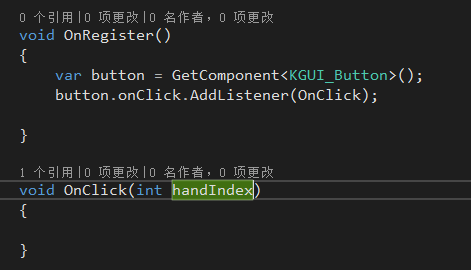
1. IsShowButton：勾选后，运行则该Button默认被选中
2. Button声音

勾选之后，当Button进行交互时，会播放声音。

1. 相关事件

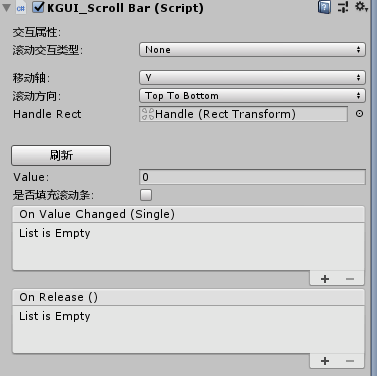


Button提供了相关事件，注册方式有两种，一种是直接在编辑器窗口中赋值，另一种是使用代码注册。如图：

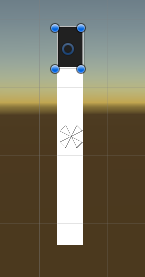


## ScrollBar(滚动条)

1. 在RectTransform的物体中添加KGUI\_ScrollBar组件



1. 需要注意的是HandleRect属性，该属性是滚动句柄。



根据自身需求，设置高度

1. 设置参数完毕后，点击刷新即可。
2. 设置Value值

Value值范围在0~1之间，通过设置Value值可移动滚动句柄。

1. 设置填充滚动条

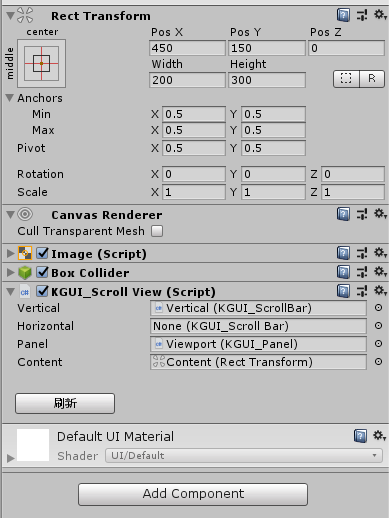
勾选“是否填充滚动条”，可设置滚动句柄的大小。

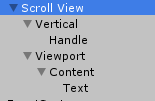
1. 提供相关事件

滚动条提供了OnValueChanged事件、OnRelease事件，通过注册这些事件，可获取相关滚动条信息。

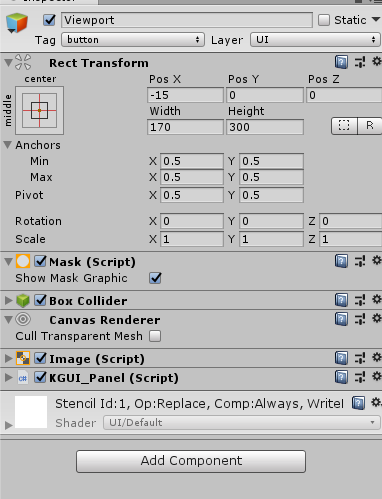
## ScrollView(视图)

在RectTransform物体中添加KGUI\_ScrollView组件，同时需要添加如下：





可在ScrollView视图下添加水平或垂直滚动条。在添加需要滚动的视图，也就是Viewport。



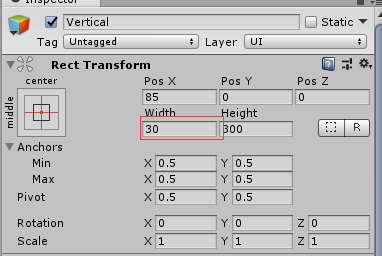
**注意：需要在Viewport物体中添加KGUI\_Panel容器，用于接收支持在视图中进行左右上下滚动事件**。

**Content**物体则是大小，也就是根据这个Content的高度值或宽度值来自动匹配滚动条的相关信息。

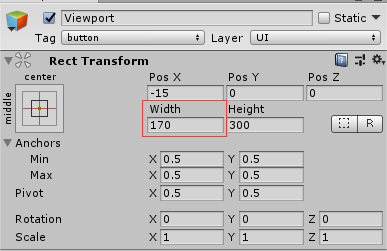
注意：如果滚动条是垂直滚动，那么此滚动条的宽度值+ViewPort的宽度值刚好要满足整个ScrollView的宽度值。从而来计算根据Content的高度值来是否显示滚动条。

什么意思呢？如下图所示：

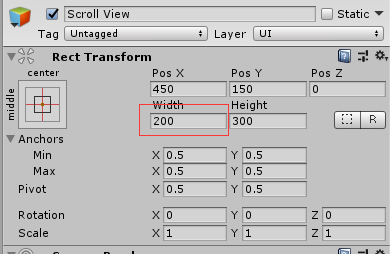
滚动条(Vertical)



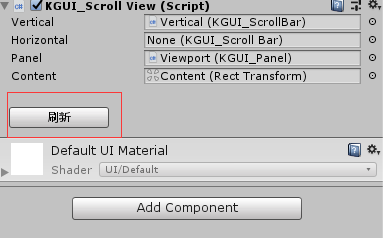
滚动视图(Viewport)



ScrollView组件所在的物体



配置好以上信息后，点击刷新即可。



## Panel(容器)

KGUI\_Panel容器是一个比较常用的功能，KGUI\_ScrollView组件和KGUI\_Dropdown组件都需要用到它，它提供了一些比较实用的事件。

* OnDirectionX事件

该事件用于接收当手势按下时，是往左移还是往右移。

往左移返回-1，往右移返回1；

* OnDirectionY事件

该事件用于接收手势按下时，是往上移还是往下移。

往上移返回1，往下移返回-1；

* OnEnter事件

当手势移入到Panle范围内时，执行此事件，该事件在移入时只会移入一次。

* OnExit事件

当手势移出到Panel范围时，执行此事件。

* OnDown事件

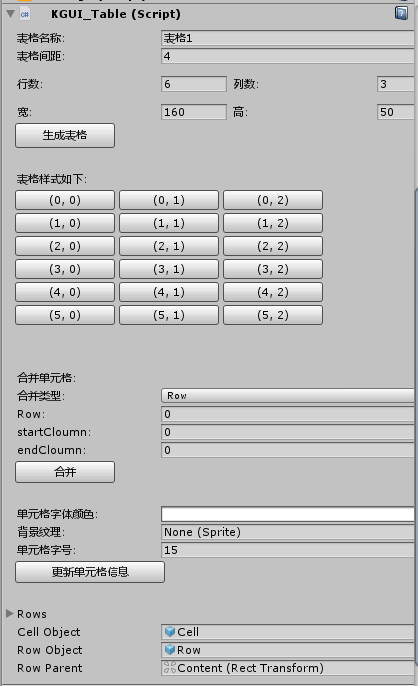
当手势按下时，在Panel范围内按下，返回OnDown(handIndex,True)。否则返回OnDown(handIndex,False)。

* OnUp事件

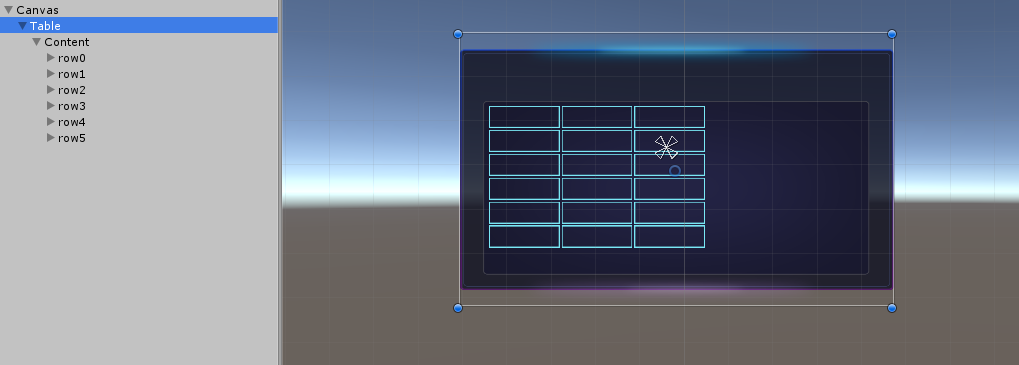
当手势释放时，在Panel范围内释放，返回OnUp(handIndex,True)，否则返回OnUp(handIndex,False)。

## Table(表格)

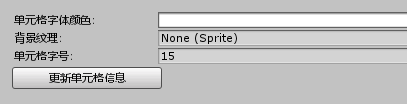
KGUI还提供了比较实用的功能，就是KGUI\_Table表格，可在编辑器中设置表格的样式。在RectTransform组件中添加KGUI\_Table组件。



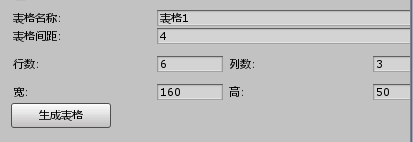
1. 首先需要配置好RowParent参数，也就是需要配置好表格行列的父对象，如图所示：



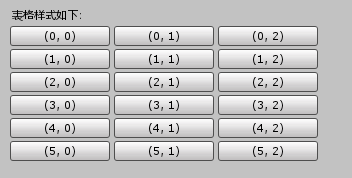
1. 可设置单元格的字体颜色、字号、以及背景，如果表格单元格背景为Null，则没有背景。



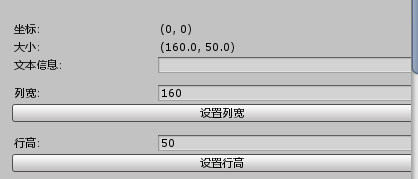
1. 设置表格的行列，以及宽高、表格间距、表格名称等信息。



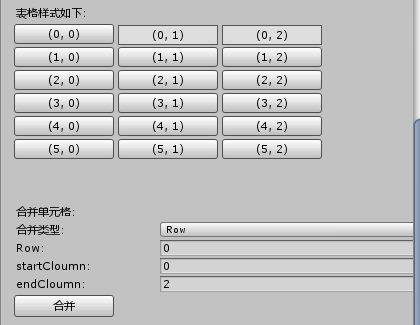
生存成功后，可在编辑器中看到此时表格的样式。

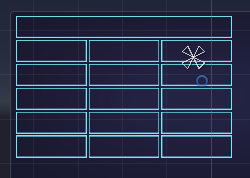


在单元格中点击某个单元格，还可以查看该单元格的信息，以及修改信息。

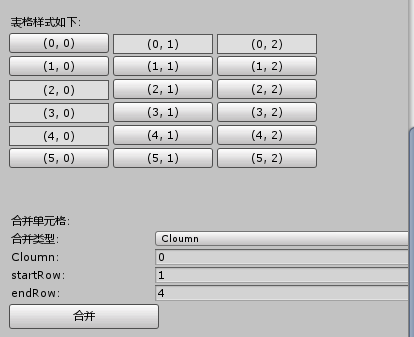


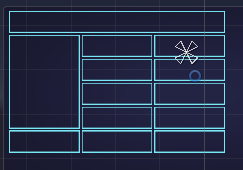
1. 合并单元格
2. 如果需要往对单元格进行列单元合并，只需要指定行，然后设置起始列、终止列。



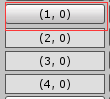


1. 同理，需要对行进行合并，只需要指定列，然后设置起始行、终止行。

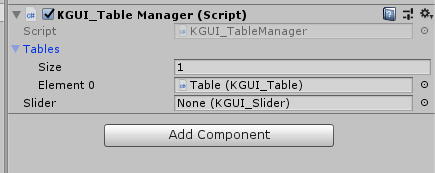




注意：合并后的最终大格，它的坐标值可在编辑器上查看到。



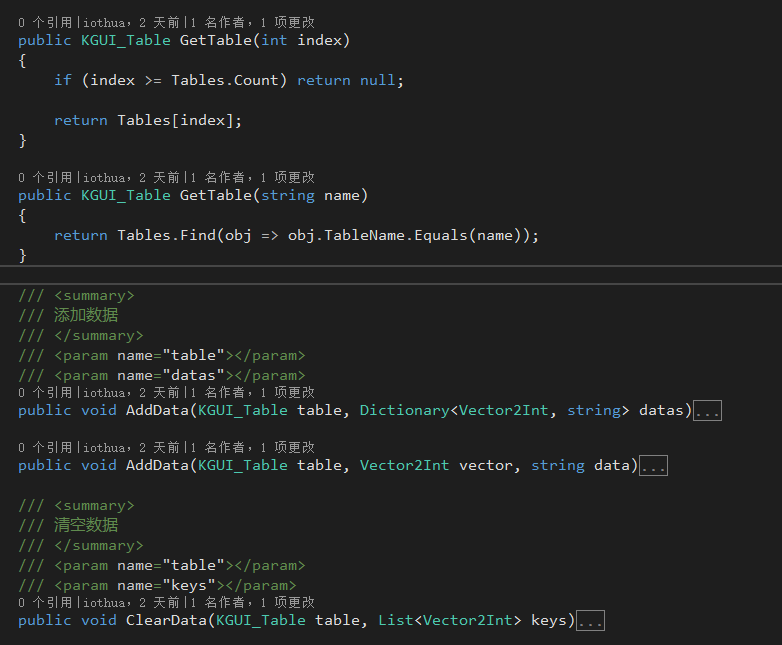
1. 代码设置单元格填充数据
2. 如果表格有多个需要管理，那么可在表格上方添加KGUI\_TableManager脚本，它可同时管理多张表。



Slider参数暂时无效。

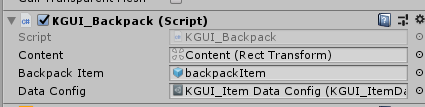
1. 获取到KGUI\_TableManager组件或者KGUI\_Table组件。

在这两个表格脚本中，都提供了如下常用的方法。用于对表格单元格的数据添加。



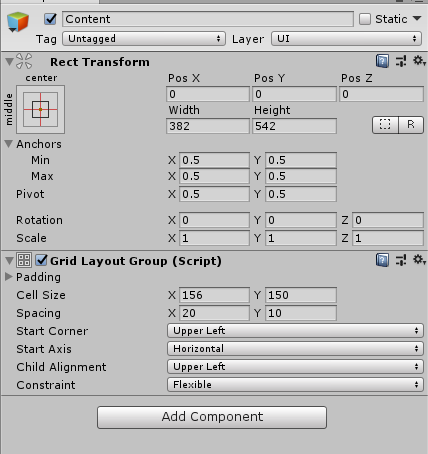
## Backpack(背包)

在RectTransform物体中，添加KGUI\_Backpack组件。



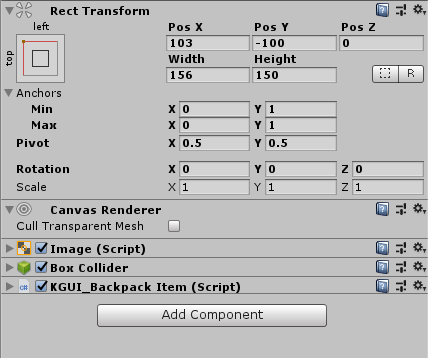
* Content属性

是生成子项的父对象，其中在Content中有GridLayoutGroup组件，用于进行排列的，可根据自身的背包信息，灵活设置。



* BackpackItem属性

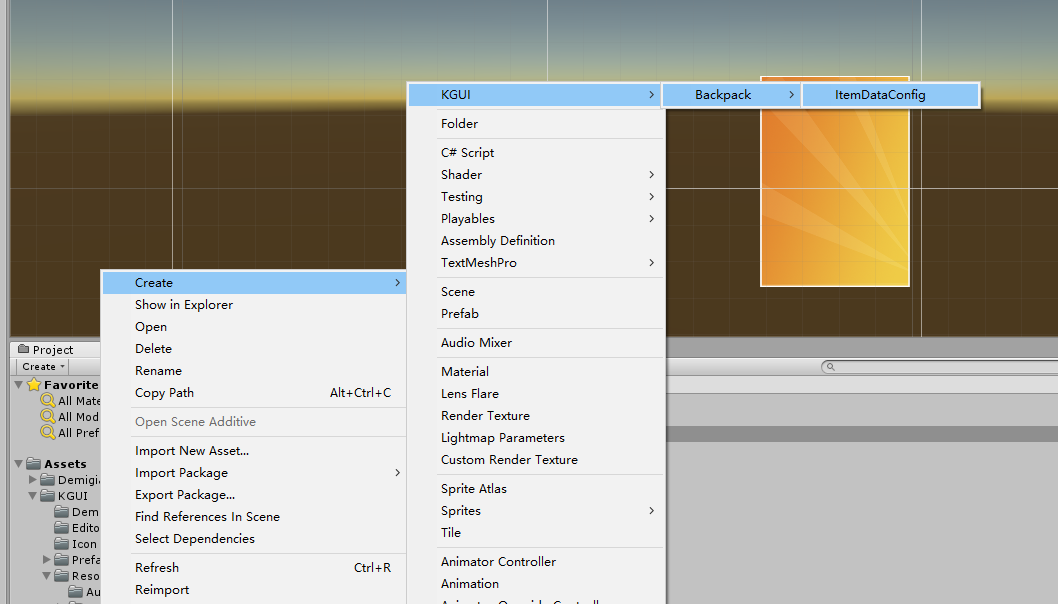
该属性是背包的子项，在背包子项中添加KGUI\_BackpackItem组件即可，并设置该子项的碰撞体大小。

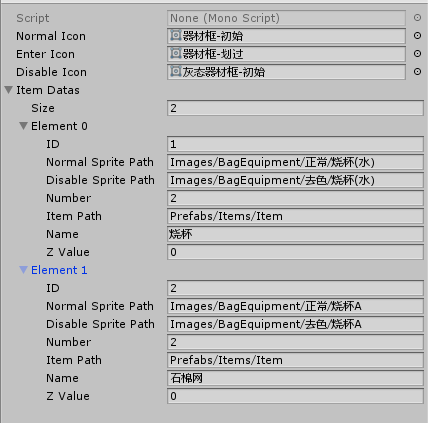


* DataConfig属性

该属性是一个资源文件，里面包含了背包的相关数据信息。

在Assets下右击创建KGUI\_Item Data Config资源文件。

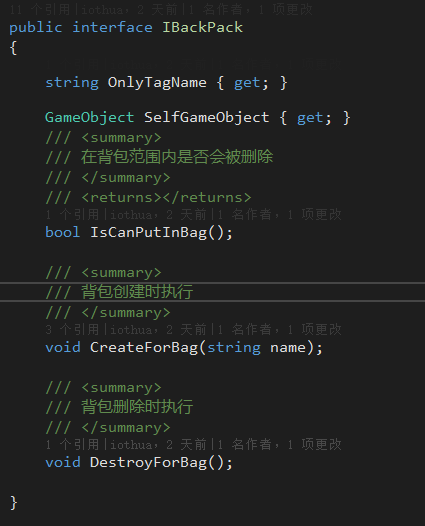




配置相关参数信息即可。

* NormalIcon属性：背包子项的默认样式
* EnterIcon属性：当手势移入到背包子项时的样式
* DisableIcon属性：当子项禁用时的子项样式
* ItemDatas属性则是背包格子数量。
  + ID属性：子项序号。
  + NormalSritePath属性：子项图标的默认样式路径，必须指定在Resources路径下。
  + DisableSpritePath属性：子项图标的禁用样式路径，必须指定在Resources路径下。
  + Number属性：仪器数量。
  + ItemPath：仪器预制物体。
  + Name：仪器名称。
  + ZValue：仪器Z轴初始值。

当运行时，Backpack组件会自动的初始背包数据，注意此时是没有办法将创建出来的仪器返回背包的，你需要在仪器预制物体中添加自己的脚本并且继承IBackpack。只有物体中的组件脚本继承了IBackpack接口，才会被放回背包，并且会提供相关的接口方法。

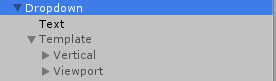


目前背包还是不够灵活，请期待后续背包会持续更新更加方便。

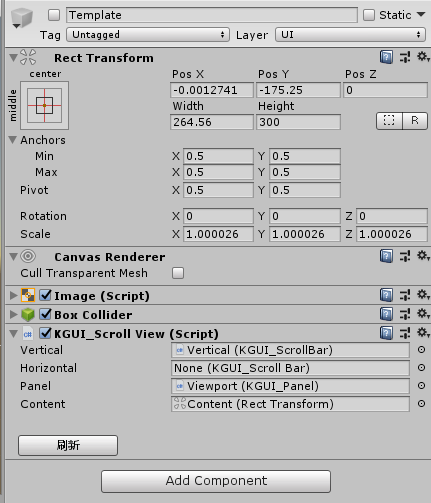
## Dropdown(下拉框)

在所需要的RectTransform对象中添加KGUI\_Dropdown组件，以及KGUI\_ButtonGroup组件。

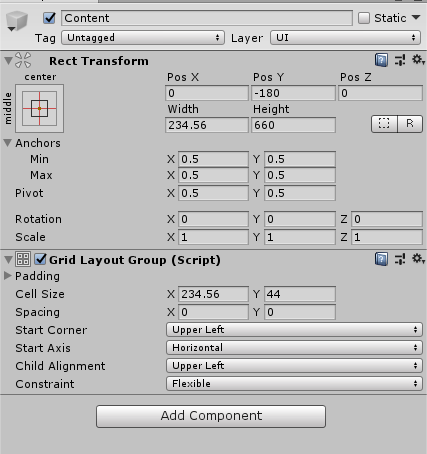
其中需要绑定下拉框的Template(模板)，也就是下拉框下的下拉子项集合对象，布局如图所示：

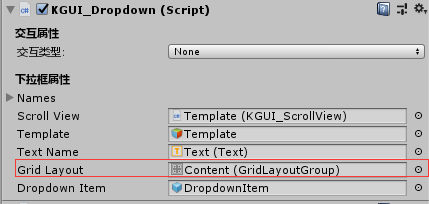


Template模板可直接采用ScrollView视图，该按钮已经提供了相应的滚动条以及相关功能，可满足模板功能需求。

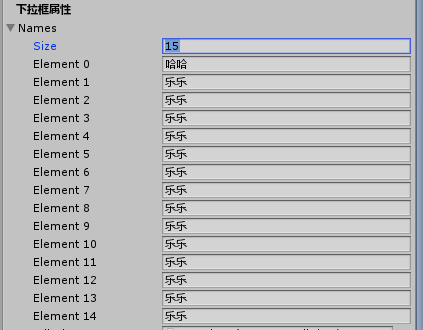


其中需要注意的是在Dropdown组件中，有一项需要注意的就是GridLayout属性，该属性是用于存储下拉子项的父对象，用于计算子项在有与滚动条的情况下来计算子项的大小。





配置好Dropdown相关参数后，可进行填充Names属性了，也就是下拉框子项的集合信息。



根据填充的信息，Dropdown组件会自动的计算滚动大小以及相关数据，同时也可以代码来设置该属性，该组件提供了相关的方法。