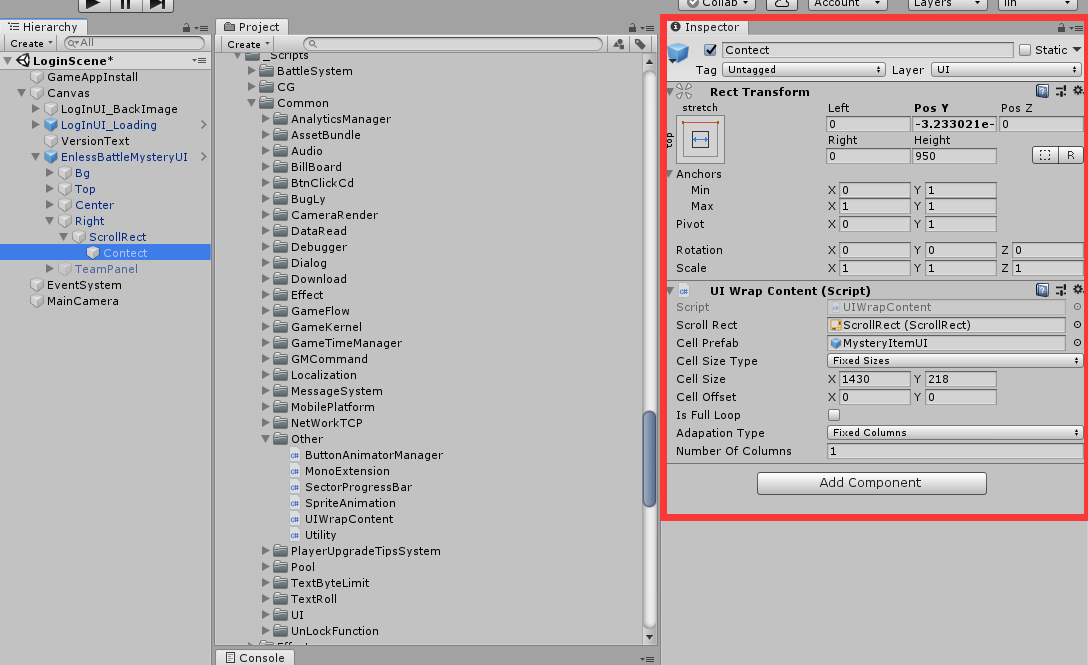
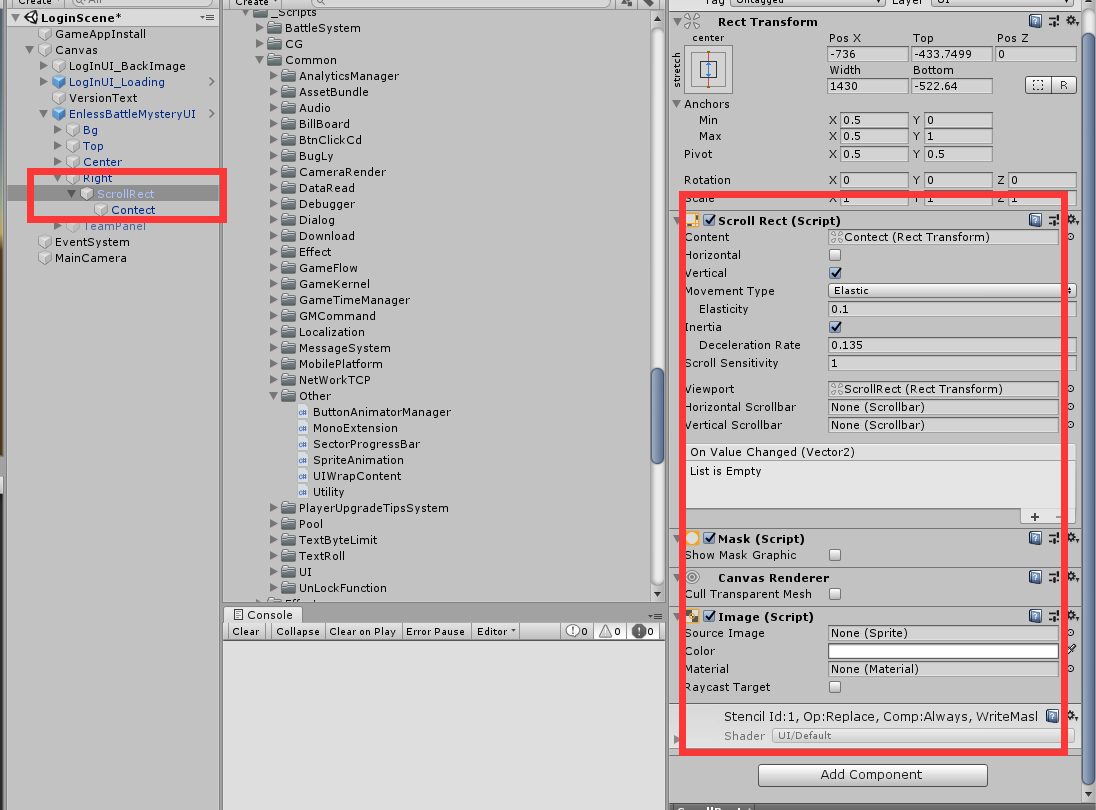
无限循环案例 (具体案例在EnlessBattleMysteryUI .prefab中) 后期可拓展。

1.使用到的脚本UIWrapContect .cs在Scripts/Common/UIwrapContect 中，

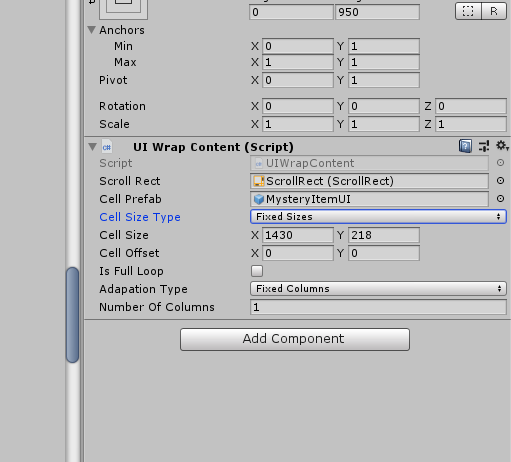
参照NGUI uiWrapContect 抄写的一个脚本，用于处理过量生成预制体的问题。

1. 层级结构,需要用到 SCrollRect 的组件和UIWrapContect 脚本.



注意插入UIWrapContent脚本的物体的锚点位置，以及不需要添加Consize 自适应大小的组件。

1. UIWrapContect 参数解析。



1. 是父物体的ScrollRect 组件，用于获取高宽 ， 滑动时候OnvalueChange 的函数改变。
2. 需要无限循环的子物体。
3. 是否固定物体大小，FixedSizes 是固定物体大小，ModifySize 是计算大小。

Ps;(因为当时思范的一个案例,有一些子物体的大小是不确定的,需要程序后面自适应大小后,告诉子物体,所以设置了一个这个选项.)

Ps;(如果在预制体中,选择了 ModifySize 则在程序初始化中进行手动赋值,下面分析代码的时候会说到.)

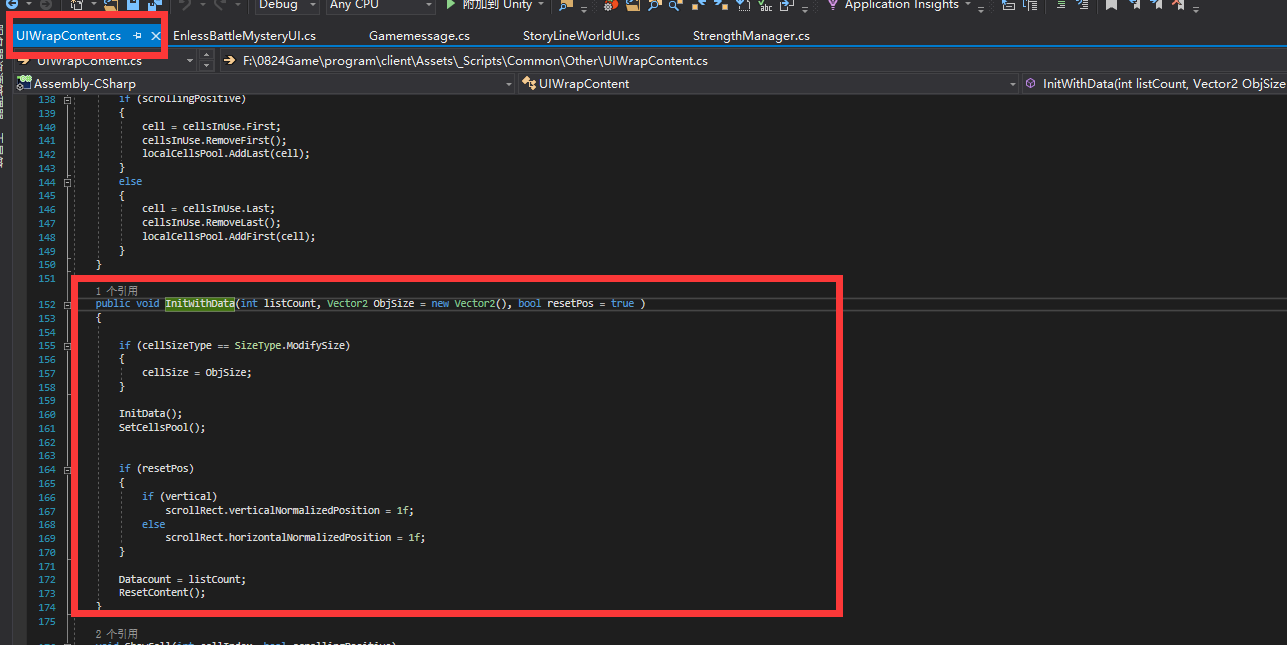
1. 在3中选择了FixedSizes 才会生效,是告诉UIWrapContent 子物体大小,初始化计算好物体的.
2. 告诉物体间隔
3. 是否循环播放.(意思就是让到底部了,不能再拉的情况下,会自动无缝衔接的回到初始第一个位置,循环播放.)
4. 自动计算行数类型,和3的意思差不多，如果FixedColums 会固定函数，下面的会生效，

如果Modify colums 会计算行数，按照父物体ScrollRect 组件的大小进行计算可以存放多少行。

1. 每一个列的行数

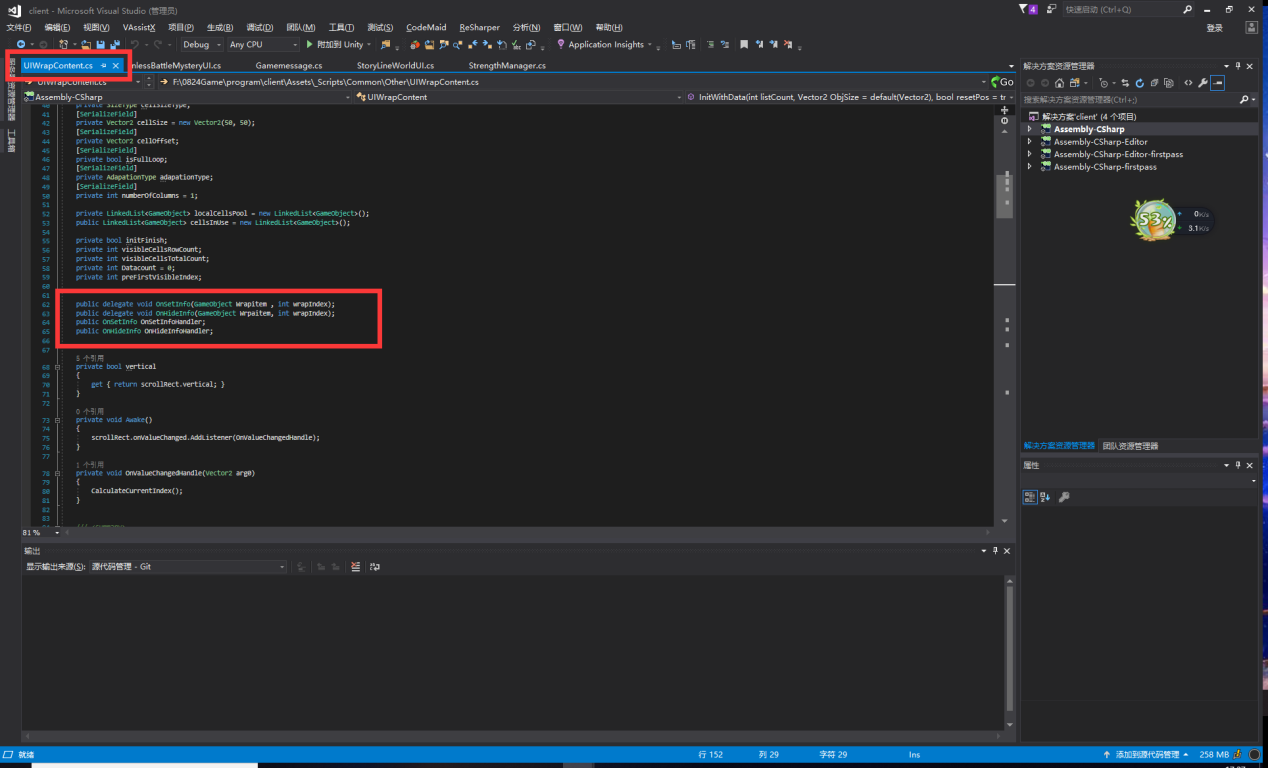
( 准确表示是如果选择了水平滑动,那么水平滑动中一行有多少个子物体，如果选择了垂直滑动，那么一列中有多少个子物体 )

3.重要代码结构分析。



InitWithData 是初始化子物体的设置。

1. 是代表子物体数据一共有多少，
2. 2.接上面所说，如果需要程序自适应大小，传入一个Vector2的值，让他进行计算大小,
3. 物体是否重置位置,就是刷新的时候ScrollRect 的父物体位置是否归零显示再最顶端，也可以自己再外面操控ScrollRect 函数进行设置，(考虑到奕佳之前写到聊天的时候，会默认设置位置在最下端。)

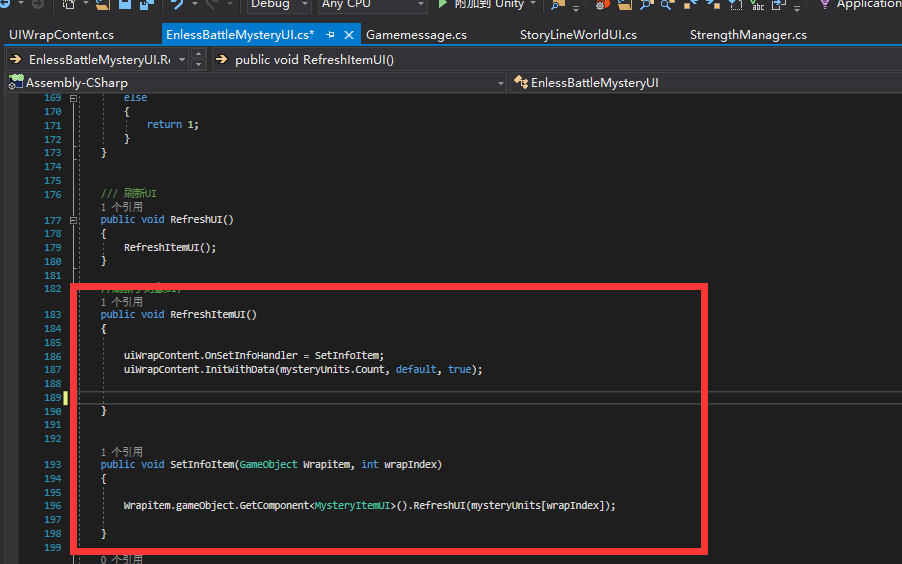


声明一个委托,在存储数据的父类脚本中调用

OnSetInfoHandler 是改变输入的委托。

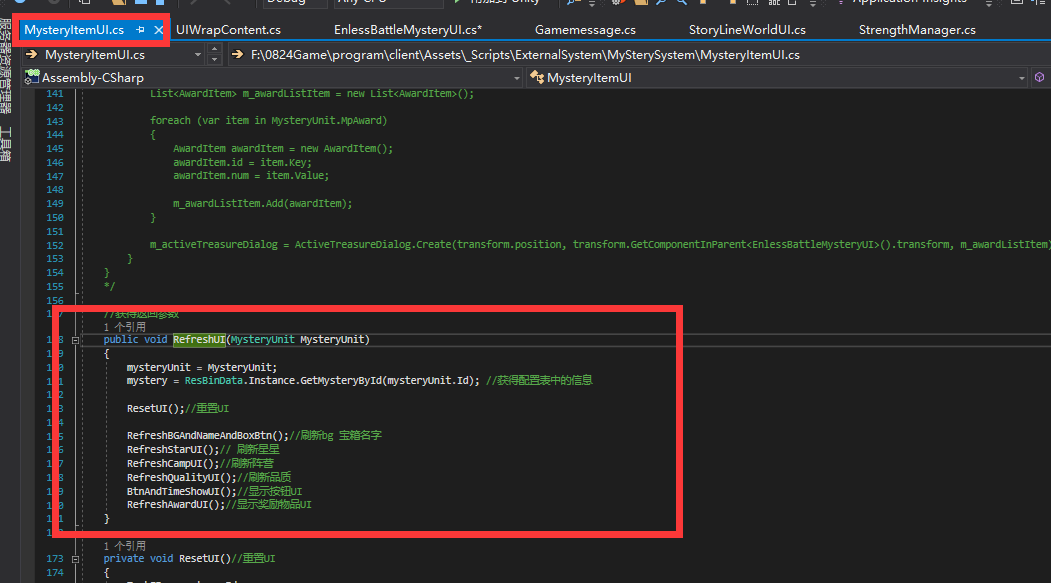
OnHideInfoHandler 是隐藏的委托. (设计到删除的委托，一般情况不用，可以忽略这个委托。)

4.实战代码结构环节。



定义一个委托，在父脚本中。 先告诉UIWrapContent 的显示设置委托,然后初始化子物体。

OnSetInfoHandler 会回调一个预制体 ,一个数据位置的ID ， 然后进行单独设置对应子物体的面板的数据，每当超出位置时，就是自动回调到对应的位置，然后调用SetInfoItem 函数，进行重置子物体。



(总结:获取到需要修改的预制体,然后有一个他所在对应位置数据的ID ，取得数据进行修改这个预制体)。

这个脚本可以拓展,例如快速移动到对应的脚本中，定向删除某一个数据，动态修改物体数据。有需要的老板们，可以告诉我一下，我拓展上去。也可以自己看下代码拓展，参照NGUI UIWrapContent 抄写的。。。