* 标签：

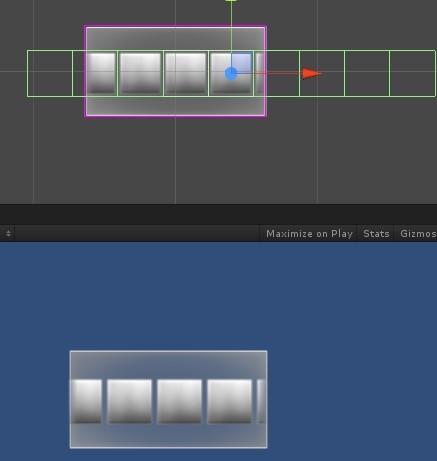
* [unity3d](http://www.qiujiawei.cn/tags#unity3d),
* [NGUI](http://www.qiujiawei.cn/tags#ngui)

本文主题主要是跟unity3D的一个扩展插件NGUI相关，在我这个版本的NGUI中的一个example： >Example 7 - Scroll View (Panel)

演示了如何实现一个可以拖动的panel。大概步骤就是:

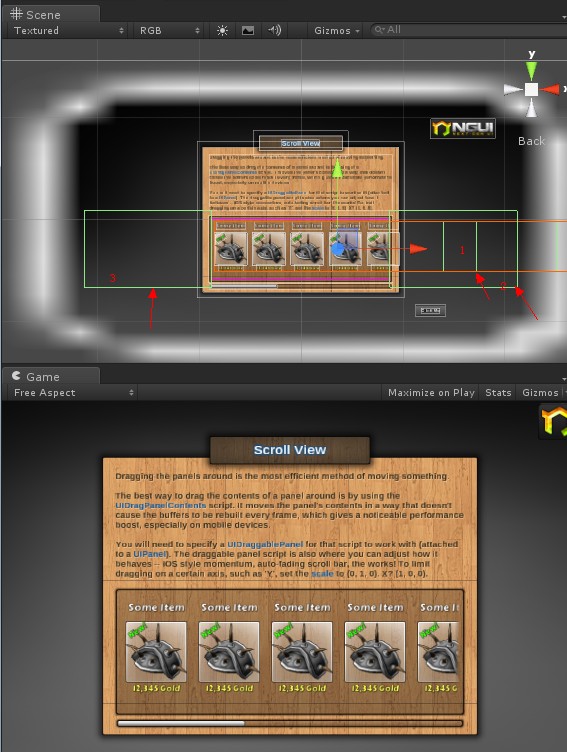
* 在一个普通panel对象添加一个IDraggablePanel组件，进行一些设置
* 再在该panel节点下添加一些content，如UISprite,UILabel（可以加多一层UIGrid来自动对齐）
* 这些content对象必须赋予一个组件UIDragPanelContents，和box Collider， 前者是NGUI可拖动面板相关的必要组件，添加即可，不需要设置；后者是用来触发事件的，如果不添加collider，按住这个content（可能是一张图片）会无法进行整个Panel的移动，而我们需要的是，再Panel的裁剪范围里的可见对象，按住它后可以移动整个面板（非常常见的一种功能吧）。

以上功能的实现还是比较容易的，稍微熟悉NGUI的人都可以按这个步骤做出来。但是会出现一个问题，当一个content移出panel裁剪边界后，它仍然处于可响应状态，尽管它已经被裁减、已经隐形了。原因就是，这个content的box collider仍然是active的。虽然看不到该对象，但组件是激活状态的。



*（绿色框就是box collider，那些出界了的、隐形了的方块仍然是可以被点到的）*

NGUI的这个example对此的解决方案是，在这个panel的轴向上的两个端点处，加了两个空的gameobject，并添加box collider，来遮挡本来出界了的content。



*（2把1给挡住了= =）*

这真是尼玛的坑爹啊！！！！

难道要每实现一个draggable panel都要在两端加这么一个玩意？而且这两个box collider可能会挡到其他控件。实在是不可取。

不考虑NGUI这个坑爹的方法，考虑其他方案：

第一种解决方案

panel里的content出界后，disable掉它的box collider。

这个方案也有问题，因为有可能一个content面积巨大，尽管它的一大片面积已经移出边界了，但是还有相当一部分面积还在panel里面。这时候我们需要的效果是，按住剩余的可见的那部分，还是可以拖动整个面板的，同时那部分出界的透明的，不可以触发拖动效果。

进一步考虑是，让box collider可以自适应，当content它的一部分出界后，box collider变形，只跟content的可见部分匹配。

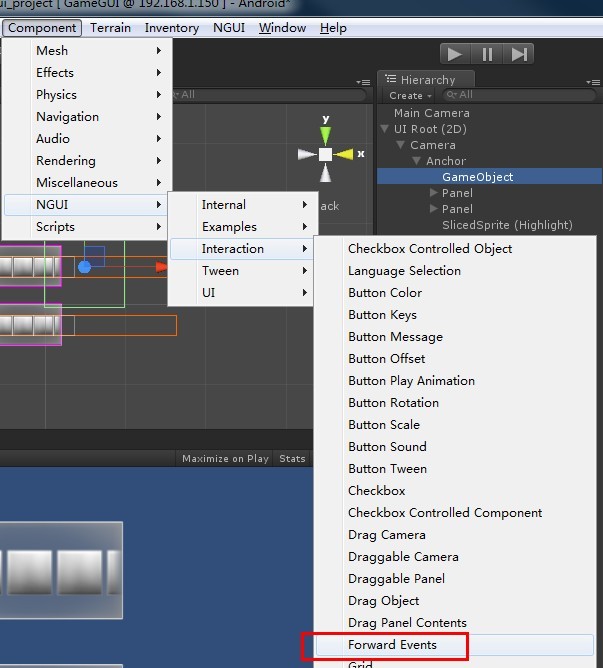
这个也许可以实现，但要做很多编码工作，而且可能会影响性能。

博主稍微研究了下draggable panel的相关源代码后，还是觉得这个自适应的扩展脚本很不好编写。

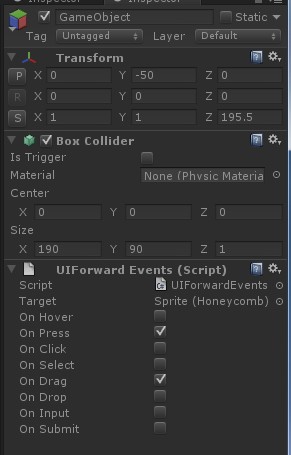
第二种方案

苦逼了一段时间后，发现其实可以不需要这种所谓自适应的box collider，可以换一种方法实现这种拖动panel功能：

* 保留panel里的各个子对象的UIDrag Panel Contents组件，删除它的box collider组件。
* 在draggable panel同层次创建一个空的gameobject，给它增加一个box collider，大小和位置，和draggable panel 的大小和位置对应（就是说，这个game object就是该panel的触发框了）
* 关键！在该gameobject添加一个组件：NGUI里的UIForward Events



设置target为目标draggable panel里的任意一个content对象，事件为onPress onDrag



这样，这个新的外部box collider会接收到点击事件，并调用target的回调函数去处理该事件。

出来的效果就是，只要在这个新的box collider内的拖曳事件都会正常地触发。

but，这样还是有问题，就是说当这个panel的各个content对象是可以被点击，触发某类事件的时候（比如是一堆Button)，就点不到啦。

所以这个解决方案只能解决content是普通静态对象的时候。比如content是一个或多个UILabel，用来展示一些游戏信息。

第三种方案

这个方案是应付上文说的,当content是可点选对象的情况的。

为了保留各个content的box collider组件，可以采取分页的方式，即这个draggable panel是分页的，当你拖曳结束的时候，panel会自动适配到某一页，而不会说停留在页与页的中间。

这样，只要当触屏事件结束的时候，判断出当前所属的是哪一页，然后把除了该页面外的所有content对象的box collider控件都disable掉，而当前页的就enable， 这样就行了。

另外，博主我也扩展了NGUI，实现了一个分页脚本，只需要拖到panel对象就可以自动应用上滑页效果了。不过等把这三种方案实现了，再开源出来。

目前博主就做到这个程度，这第二个方案确实解决了一部分问题，目前还是够用的。 等以后发现完美解决方案的时候再更新。

<http://www.qiujiawei.cn/blog/2012/09/06/unity3d-1/>