该配置表工具，可以将策划使用的Excel配置表数据提取并转换成txt，然后自动生成表头代码，并使用配套的c#脚本，可以非常方便的加载\获取配置表的数据。

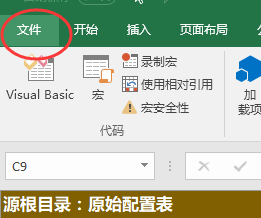
## 插件引入

插件是官方说法，其实就是一个外部链接（dll？？？）。

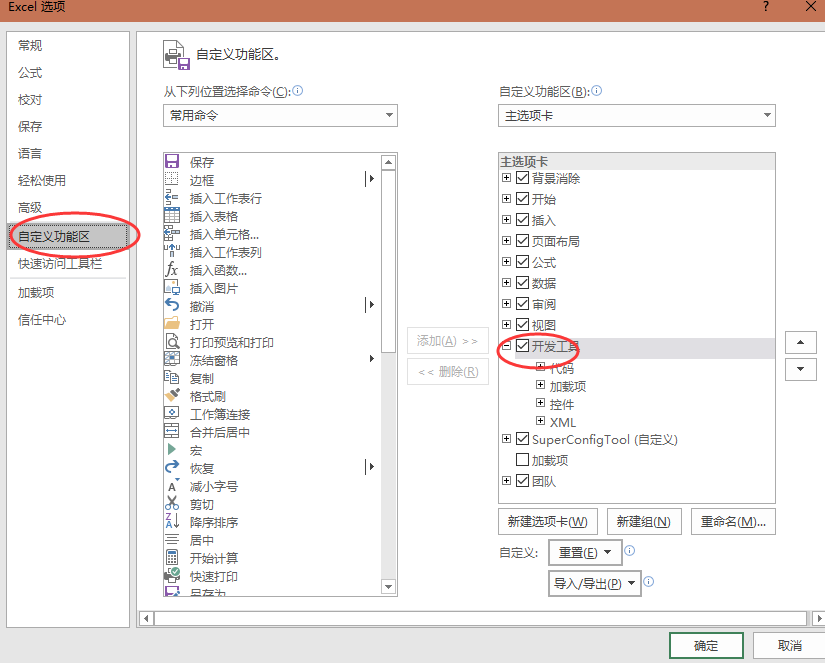
里面是我抽象出来的一些工具函数，所以其实我还是蛮喜欢把一些函数收集起来的？？？

Step1：

先把“开发工具”弄出来

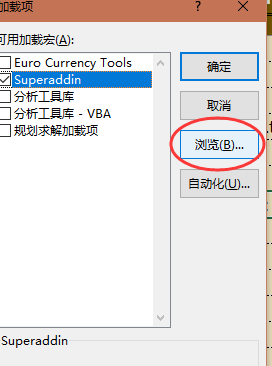


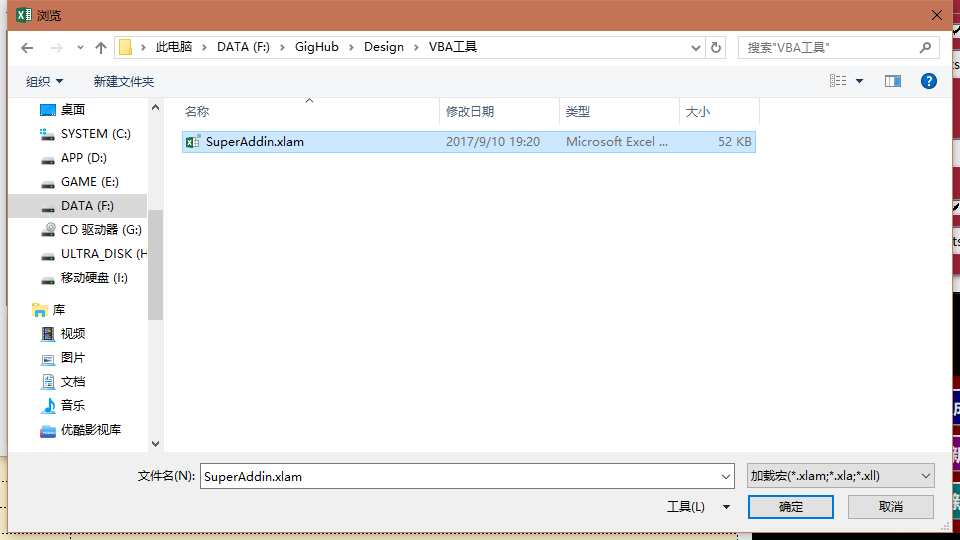




Stpe2：





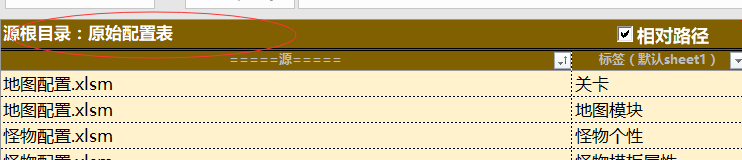


SuperAddin.xlam我已经上传到git上了

然后，原则上就应该可以使用下面的功能了。

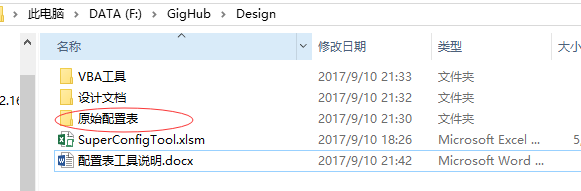
## 数据录入

在这里输入原始配置表（即策划使用的配置表，其中可以有公式，有很花哨的格式…）的根路径



其实这里也可以为空，然后每一个具体配置表路径为： 原始配置表\XXXX.xlsm，但多数时候配置表们都有一个共有的根目录，所以把这部分内容放这里会看起来顺眼。

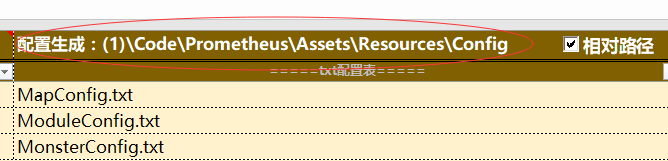
这个配置表已经是以git目录为目标结构进行的配置，那么这个“原始配置表”其实就代表的下级路径



因为这里使用的是相对路径（把后面的勾去掉就是使用绝对路径），所以以自己所在目录为基准，找到目录“原始配置表”，里面就是策划使用的原始配置表。

## 配置输出

生成路径就要长很多



前面那个（1）表示向后退1级。

注意，这个txt文本的名字，是生成类的名字，所以这里我就比较老实的用英文了…

OK，因为以上内容，你们应该不会怎么用，这部分是我要操作的，所以就简单的介绍到此。

## 脚本输出

这上面是告诉这个工具，要生成的代码应该放在哪里。

PS：最早我发现它生成的代码，在unity编辑器预览里不能显示（虽然在IDE里可以正常显示），原因是因为txt默认使用ANSI为编码格式，unity不认识。作为一个有强迫症的人，这明显是不能忍的…所以建议在新建cs文件的时候，就把它的编码改成Unicode（虽然我已经新建好了），我在这个工具里做了一些编码格式处理，以确保它能识别主流的txt编码格式。

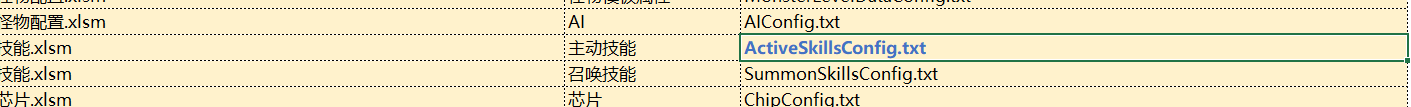
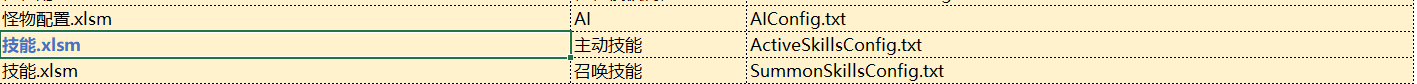
**类生成**

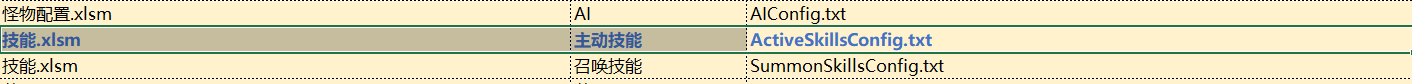
放大镜，文件查找，你可以新建好一个cs文件，然后点这个查找再找到这个cs，它会自动生成最优的路径方案（虽然你也可以手写…）。

文件夹，打开这个cs文件所在的文件夹。

预览，以notepad打开这个cs文件。

在已经生成好txt配置的前提下，选中你要生成的配置表所在行，下面几种方式是等价的



点击



它就会根据配置表的头3行信息生成c#代码。

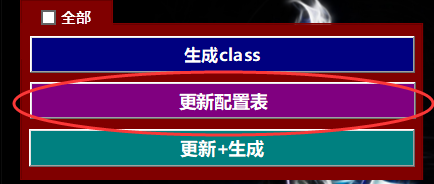
如果，勾选了全部，那么它会对全部的配置表执行一次对应的操作。

当然，你可以进行多选，比如这样



它会把你选中的3行数据进行你点击相应按钮的处理。

点这个，可以把策划用的配置表转换成txt（转换过程同样有很多处理，不过你们应该不关心）



简单说，如果表头没变点更新，如果表头变了点生成。

懒人按更新+生成就好… 不过会多花几秒钟。

PS：下面两个按钮是快捷操作，你选中某个excel或者txt，再点他俩就可以执行相应操作。



PS2：其实右键表格中的配置表，或者txt也可以打开他们。

**类备份**

考虑到本来git就有备份回滚功能，而且给程帆省钱…所以我就没有开启备份，不然每生成一次类，它就会先把老的表头文件保存起来，再生成一份新的。

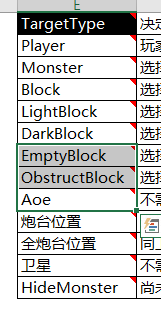
**枚举集合**

没这个功能之前，我每声明一个枚举，就要在excel里做一个格式筛选，还要添加c#代码，注释也要加两份，简直神烦…

所以偷偷的在上班的时间添加了这个功能（其实之前也给你们演示过，顺便我也把注释多余换行的BUG修复了），这里3个按钮作用和前面一样，就不重复描述了。

如何生成枚举：

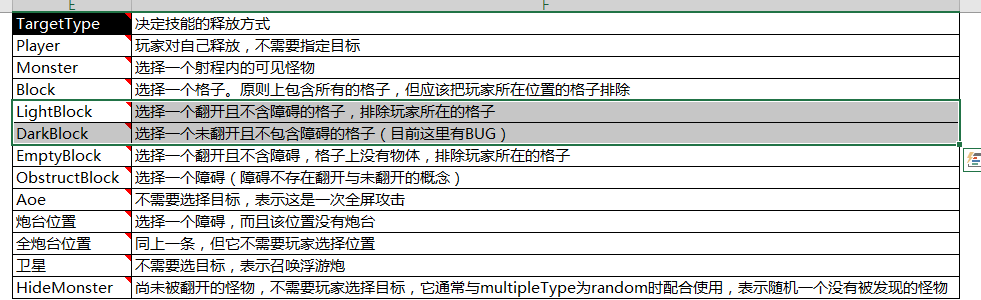
Step1：首先选中你要生成的枚举类型所在表格中的位置，比如这样



TargetType是枚举的名字，下面是它的值，这里虽然有个看起来好看的格式，但其实有没有格式都一样。

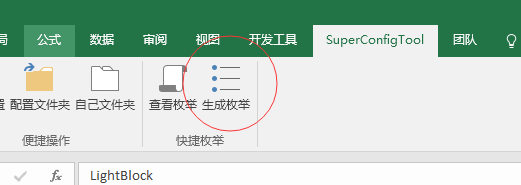
当然我手贱选了3格，其实无所谓，你只要是竖着放的，不间断，它都会认为这是一个枚举对象。

当然，如果我手贱横着选了呢？那么它会认为这是两个枚举对象，所以它会生成两个枚举。



当然，你可以按照ctrl点选多个位置，它会依次的把你每个选中的区域都转换成枚举（相同不会重复转换，就像我之前竖着选了3格也没重复生成一样）。

Step2：点这个

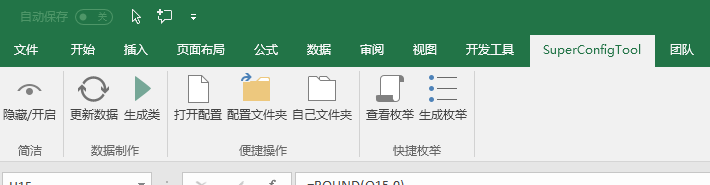


它就会按照你输入好的位置，生成c#代码。

当然，可以在excel里加一些备注，这些备注会一并作为注释内容放进c#代码里。

## 辅助标签页

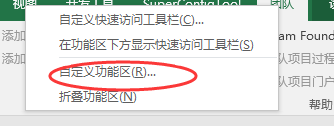
就是这个东西



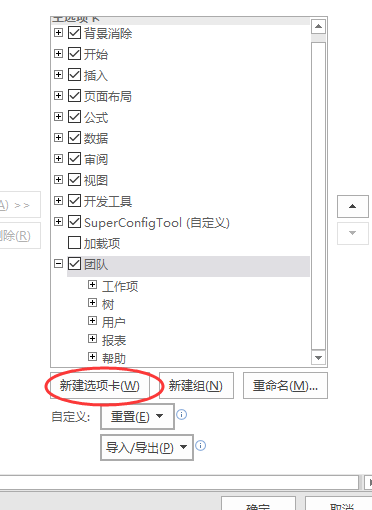
稍微有点麻烦。

Step1：

右键任意标签



Step2：



然后自己给这个选项卡起个霸气的名字吧，比如SuperConfigTool

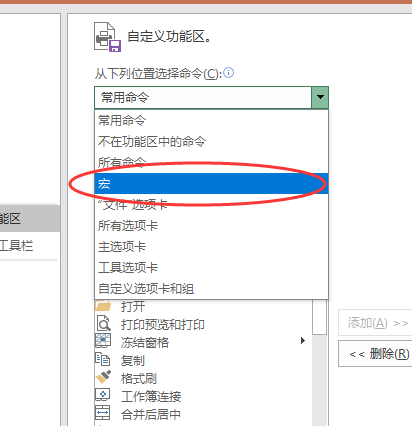
Step3：

你们可以按照这个结构，来新建一些组



Step4：

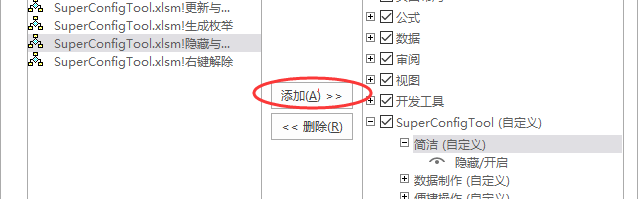
选中简洁，然后在左侧下拉列表选择“宏”



找到



选中，按添加

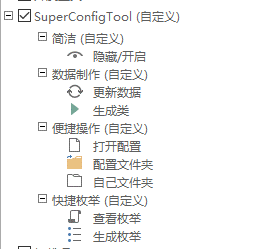


这时候应该多一个这个东西出来



然后，你自己将他重命名，然后选择一个好看的图标。

以下是我自己的组织方式，与对应的宏命令。





之后，你的excel就会多出一个选项页，只要当前的sheet被录入过配置表工具，你就可以直接在这个选项页完成配置表更新，甚至更新表头文件的操作。

## 关于SuperArray

凡是在配置表里，使用了“类型[]”形式，表示这个是一个数组类型的参数。比如string[|:\_]。

它的意思是把该数据依次使用中括号内字符进行分割。在代码里，这种数据类型会被封装成SuperArray<string>，它是一种只读的对象。

它主要有以下几种方法：

1. 可以通过下标直接获取值。

比如“1000002\_active:1\_0|1000003\_active:1\_1”，数据类型还是string[|:\_]。

superArray[0,0,0]就可以获取到1000002，superArray[1,0,1]获取到active。

1. Count

这个是一个方法，还以之前的数据为例

superArray.Count() 解释为，首次分割的长度，返回2（ | 分割后的元素个数）。

superArray.Count(0) 解释为，首次分割的长度，0号元素的个数（如果还可以进行分割）这里就是1000002\_active:1\_0的长度，因为第二个分割符是 ： ，所以还是返回的2，如果第二个分隔符是 \_ ，那么就应该返回3.

以此类推。

1. ToArra / ToList

将SuperArray内容返回为一个一维数组。SuperArray是n维的，所以需要指定返回那一部分的内容。

如果是一个superArray是一维的，那么直接superArray. ToArray()就可以了，如果是之前的例子，就必须指定一个纬度，比如superArray. ToList(1,0),得到的list将包含1000003与active两个元素。

综上，如果是一个一维的数组，其实用起来和list基本一样，Count和ToArra / ToList都不需要参数，在多维的时候，需要注意下纬度来决定参数。

另外，里面有缓冲机制，以确保二次访问的效率。

1. Foreach

可以用foreach进行遍历，它会无视纬度的，依次返回每个元素。