**导表工具使用说明文档**

**1.准备阶段**



这两个目录最好放置在同一层，方便程序使用相对路径查找文件。

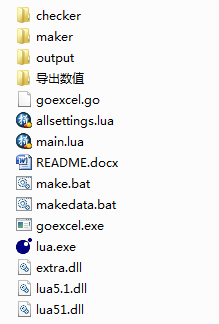
a). 策划配置目录: 配置表

存放所有策划配置表(\*.xlsx)，配置表有统一的表格格式，请严格按照格式填写。（下文将会介绍配置表格式）



b). 导表工具目录: tools

存放各种脚本和工具。



其中，makedata.bat 脚本将会调用goexcel.exe，在‘配置表’目录查找所有 “\*.xlsx”后缀的配置表，读取所有配置页数据，并为每个配置表生成对应的lua数据文件，生成的文件放在’tools/导出数值’。比如‘配置表/钓鱼配置表.xlsx’这个文件，将会生成lua数据文件’tools/导出数值/钓鱼配置表.xlsx .lua’。**注意，生成的数据文件(“\***.xlsx .lua’**”)不要提交到git仓库，只需要存放在本地即可.**

脚本 make.bat 会一键生成所有配置数据文件,并生成最终的前后端配置文件,make.bat里调用了makedata.bat,然后再根据maker脚本,输出所有配置文件.

**2.策划使用方法**

a).创建配置表(\*.xlsx),按配置规则填写好配置表

b).等待程序写好导表脚本(maker\_xxx.lua)

c).双击全导表脚本make.bat

d).有修改的配置,在提交git之前先比较一下前后版本更改的内容. **\*请保持这种习惯!\***

**注意,生成的配置只提交output/client和output/server两个目录下的配置文件.**

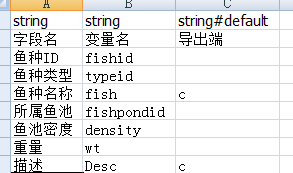
配置表填写规则:

i).每个配置表必须至少有”导出”和”常量配置”(统一格式不至于太乱).

**导出页**,是用于定义配置的中文字段对应生成lua配置表后的变量名字,变量名字一般都是程序定义,毕竟在代码里使用配置的是程序. 其中导出端由程序控制,大写或小写的字母c表示这个字段只导出到客户端; 同理,字母s表示只导出到服务端;而留空则是导出到双端. 例如有一些客户端描述字段,服务器并不需要,减少配置文件大小.

**常量配置页**,是用来配置这个玩法中,比较单独的毫无规则的数据,比如某个cd值,扣除金币,程序需要定义的字符串等等,让程序不至于在实现某个玩法的时候,把一些常量写死在代码行里.





ii).配置文件的其他页配置规则



**字段的类型**: 支持的类型请看 tools/checker/,每一种类型用一个lua文件实现.目前只支持基本类型,复杂的类型以后再根据需要添加.

**中文字段名字**:策划(或程序)自行定义,但不能为空.

类型特殊含义:

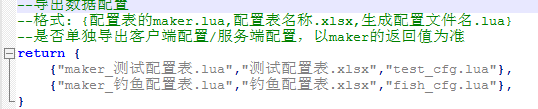
‘xxx#ukey’: 意思是主键,统一规定只在第一列生效,并且这一列的主键不能有重复值;

‘xxx#default’: 意思是这个类型的字段可以不填写任何数值,导出的lua配置表是不会出现这个字段值; 比如鲩鱼的描述没有填写,导出到客户端就没有201002这个主键对应的数据就没有Desc字段了.

**3.程序使用方法**

a).每个配置文件都需要一个导表脚本文件,也就是maker\_xxx.lua,名字可以自定义,但一般是跟配置表名称一致: ‘钓鱼配置表.xlsx’ -> “maker\_钓鱼配置表.lua”;

b).修改tools/allsettings.lua. 这个文件的格式是这样的:



第一个是导表脚本文件(maker),第二个是策划配置表名字,第三个是生成最终配置文件的名字.

c).checker. 是配置表的表头类型,策划需要的类型都需要写一个同名的checker文件,在脚本生成过程中会自动调用这个文件里的函数,并返回相应的值.

d).maker的编写. 参照maker\_测试配置表.lua, 新建的maker类都需要继承父类,并在文件末尾返回类对象. 需要注意的是,有时候不需要生成客户端或服务端配置表,这取决于重载函数 clsXXX:MakeTable()的返回值,第一个返回的table生成服务端配置,第二个是给客户端.如果不想生成某一端的文件,返回nil值即可.

e). 看代码.有优化或错误的地方及时修改.