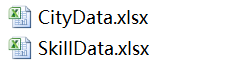
## 根据需求制作Excel表

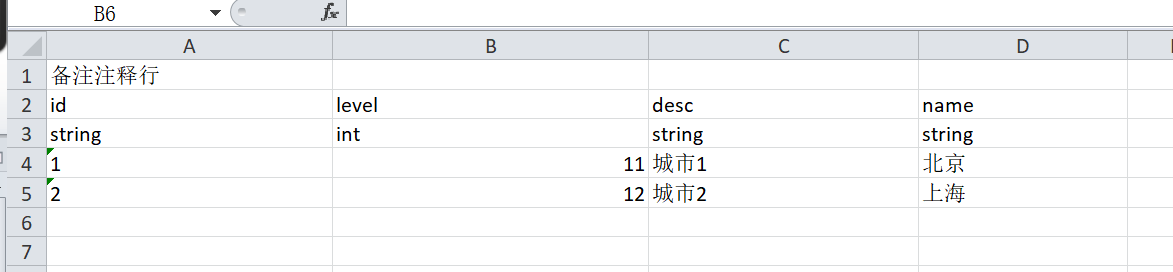
* 1. **Excel表目录：**

**C:\Users\July\AppData\Local\Temp\1544512176(1).png**

* 1. **Excel表命名：之后工具生成代码中数据结构的类名是表名**

****

* 1. **Excel表内容示例：**



格式要求：

第一行：备用行，可以用来写注释等，自由发挥，不会写入最后数据

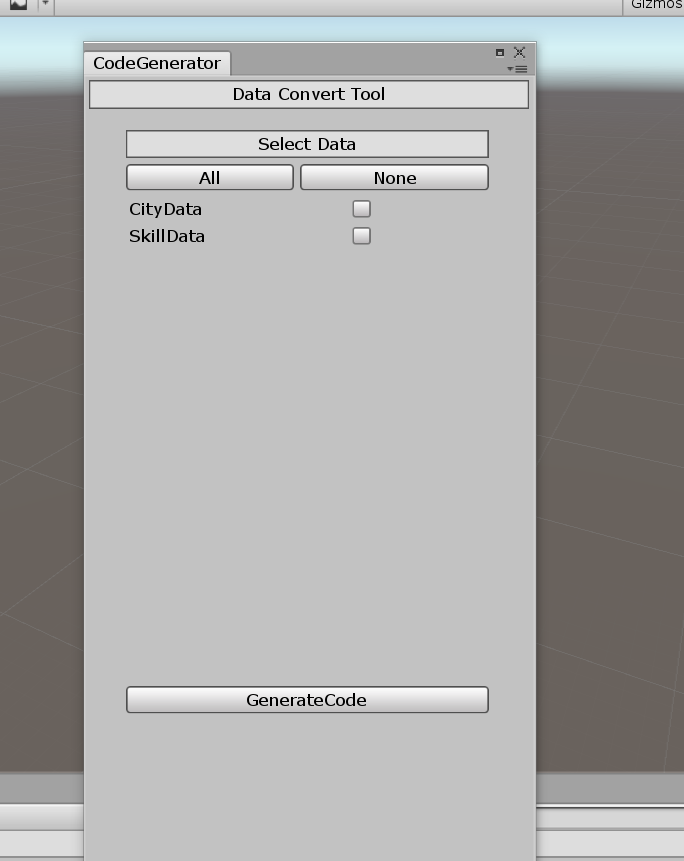
第二行：字段名称行，数据结构中各字段的名称

第三行：字段类型行，数据结构中各字段的类型，目前支持的类型：int, float, string, bool, List<int>.

第一列：id 数据，string 类型。

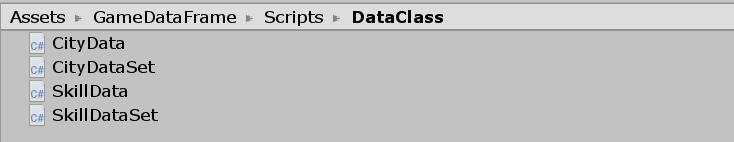
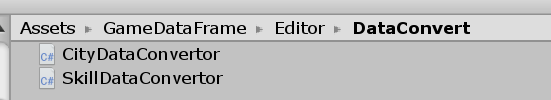
## 生成代码

* 1. DataTools -> CodeGenerator, 选择要生成数据机构类的Excel表，点击GenerateCode按钮生成。

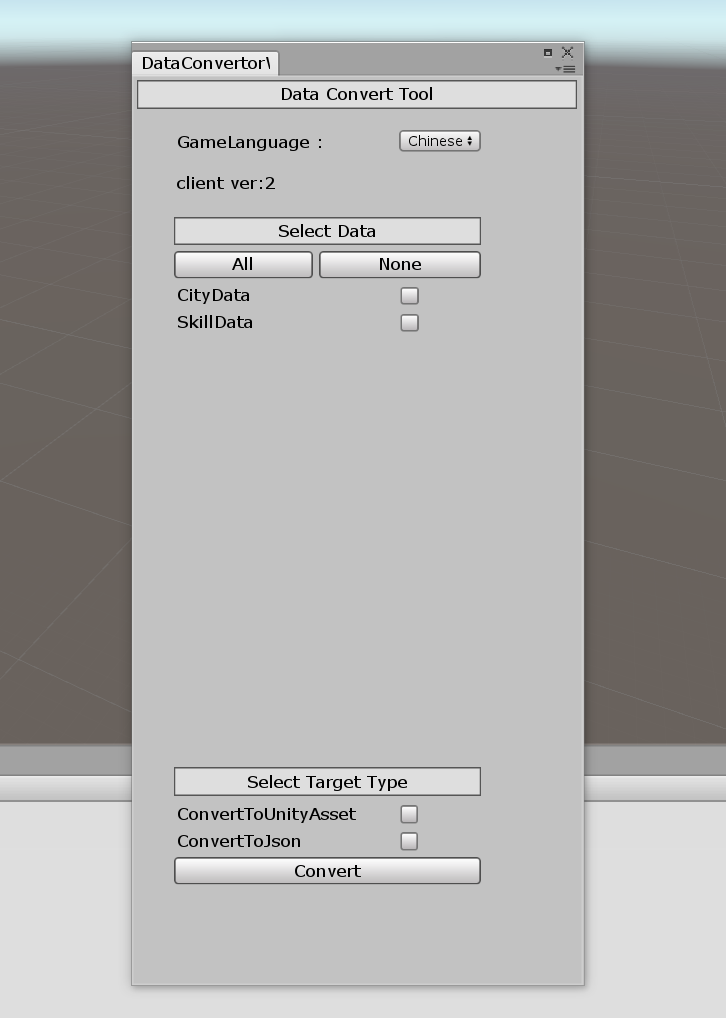
****

* 1. 每个Excel表将生成3个文件，数据结构类，数据结构集合，数据结构转换器。

例如：

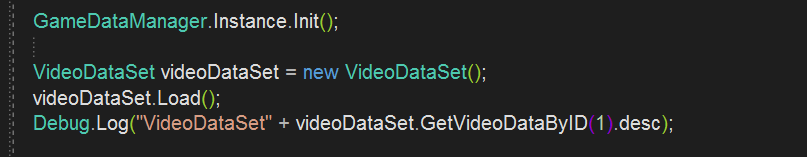
**** ****

## 转换数据

* 1. **DataTools -> GameDataConvertor, 勾选待转换的数据以及数据的目标类型，生成数据。生成的数据存放在 GameDataFrame -> Resources 对应文件夹内。**

## 加载数据

* 1. 在场景中创景一个gameobject，add GameDataManager脚本。然后初始化GameDataManager，之后加载所需数据**。**



## 配置数据版本号

DataVersion.json存储各数据转换的版本号，版本号目前规则为该数据转换一次则版本号加一。初始化工程时该json中"dataVersionDict"中的内容清空。

转换数据时会自动填充该json，如果某项数据不需要了，需手动将该数据对应字段从"dataVersionDict"中移除

## 待完成

AssetBundle 方式远程加载

文字多语言支持

## 数据封装成AssetBundle

**DataTools -> BuildAssetBundle。 AssetBundle的加载请自行解决。。。。。。**

