--------------------------------------------------CM实现----------------------------------------------

1. 文字、图片做多语言表，其他模块填Key，通过LocationMgr读取
2. 翻译表导出根据语言生成多个文件，加载的时候会节省
3. 给Text和Image添加静态赋值组件
4. 动态替换Text和Tmp的字体，Tmp字体使用Fallback实现
5. 语言包（字体、图片）下载

--------------------------------------------------多语言全部内容----------------------------------------------

##本地化的内容：文字、图片、音效

文字：

1.数据表进行翻译，通过LocationMgr.GetText(key)读取文字

2.Text组件需要AutoSetText(key)设置文字

3.服务器下发和代码里不能直接写文字，都要用key

4.需要根据实际情况下载对应语言的字体

注：图片、音效与文字处理原则一致，即不能直接和物体进行关联，多需要用key到表里查找不同语言的资源。当然也可以用规则比如文件夹名字

##如果需要动态切换语言，所有需要替换文字的界面或者别的类都要实现Translate的接口

##Prefab上的适配情况

Text组件考虑变长的情况

Texture组件考虑图片尺寸变化的适配

音效组件需要考虑时间长短不一样的情况

注：如果Prefab不能兼容那么就要考虑多套Prefab

注：有的设计上避免图片不一样，如所有版本标题都是大写英文字母缩写

注：UI意外的内容，还需要考虑模型、特效上的字，动画跟文字相关

#关于下载：

所有内容放在一个包里不太现实

每种语言一个语言包，切换的时候下载

如果做的洗点，对于每种语言包需要拆分为：基础包+下载系统包

#关于聊天

比如我是中文版本，需要在聊天界面显示各种语言，但是又不能把全部语言包的字体都下载了，那么就用系统默认语言

使用TMP解决字体不同源的问题

------------------------框架中多语言的实现-----------------------

##翻译表使用CSV，多种语言在一张配置表中，通过Luban的导出功能把数据导出为每一种语言一个Json(byte)。

问题：luban是一次性加载了所有的翻译表，是浪费内存的。

解决方案：1修改luban生成Table代码；2通过多个Table.xlsx控制；3自己做语言表。

##通过Tmp 字体 Fallback功能支持Text显示多语言

##UI对应的Texture和Sprite和音效资源还没做

##如果要支持游戏内切换语言，需要

1. Table数据增加Translate方法
2. Bean中增加Translate方法，参考Luban老版本的本地化
3. 所有文字显示的地方（包括但不限于UIText）都要注册切换语言的处理函数