DOSP_DEMOv2

现状：

文件夹命名分类

- AutoGenCSV — 各种配置表 — 问题：本地不该有配置表，配置表导入ue4里更是不行，这样不安全。所有配置应只从java拿取
- DospDoc — 一些文档 — 问题：文档归属文档文件夹，不应该放在工程目录下
- Source下的proto文件夹 — proto的.h和.cc文件 — 问题：这算第三方库，应放在一个source外的单独的文件夹，在buil.cs中引用该目录

Gameinstance

更改：为保证功能单一性，新建一个事件类，广播init和shutdown事件，其余的都移植到各各自的模块管理

问题：初始化所有system耦合性太大，Gameinstance功能太多，不符合Gameintace设计。命名混乱。注释混乱。

init游戏初始化，加载配置表，监听地图变换。对UI进行操作，对输入设备进行操作，对音频系统进行操作，shutdown对所有system进行 销毁操作。关闭ws链接

Gamemode

命名太过分！一点都不认真

System

问题：逻辑混乱。数据层和逻辑层混在一起，命名不规范，注释不规范。子类当模块使用，模块有的是业务逻辑有的是网络请求，甚至有的只是定义数据，非常混乱没有做到模块的作用。并且没有做到依赖倒置。模块里的某个类发生改变模块system就会编译，接着使用模块的类就会编译，就会造成该一个函数编译半个小时的情况。耦合性非常大。

更改：去掉system系统，模块管理manager。多加一个事件层类。该类声明委托。调用接口绑定或者执行相关委托。委托是不添加任何头文件，调用者和被调用者两者没有直接关联。更改一方另一方不受编译影响，大大缩减编译时间。增加自定义数据层类。该类只自定义数据，如结构体枚举等。如果使用自定义数据只需要引入该头文件即可，做到了最小使用。增加数据缓存层，缓存公用数据，但不缓存来自网络的数据。网络数据由java服务器缓存。增加网络层来收发网络请求。该类。除事件层类以外其他的都不给模块以外的类或对象使用

Base类：发送接收数据，InitSys初始化改模块管理类，绑定HTTP请求回调函数。绑定ws请求回调。获得gameinstance，system类：生成子类单例。子类：当模块如果使用。

UI

UIManager类：管理UI,自定义枚举管理，依靠枚举切换。System中有UI某些功能模块类，比如邮箱系统有邮箱的system

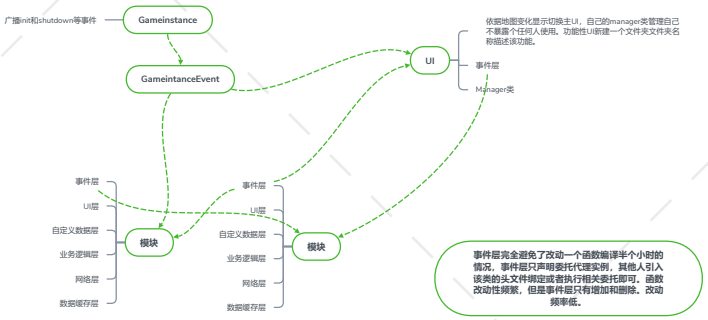
问题：在gameinstance里初始化UI，依靠枚举切换。首先UI和gameinstance毫无关系，不应有gameinstance管理。依靠枚举切换就会不停第一枚举。而且这些枚举是完全不必要的，UI某个系统功能是独立的，公有的。应新建一个文件夹存放，UI只是显示功能，向邮箱系统它是功能性模块，不应和UI混在一起

更改：UI模块通过gameinstance的事件类监听地图变化。依据地图变化显示UI，自己的manager类管理自己不暴露个任何人使用。功能性UI新建一个文件夹文件夹名称描述该功能。

单例处理

现在单例在editor里不会被析构，会造成很多问题比如构造函数里初始化数据

稍微好点的情况：



单例处理

使用ue4自带的editor宏判断是否在eidtor情况。在editor情况手动析构单例