UI框架的设计

我们项目中 UI框架主要是为了对游戏中的UI面板进行一个统一管理，基于MVC思想进行设计，然后就是需要一个基类，面板的v层继承基类

基类里面主要是封装一些共有的虚方法，比如初始化、更新、释放、打开、关闭等等，这些虚方法主要是根据项目需求然后添加，

然后框架是采用字典进行储存所有的面板，如果有需求要做成那种层级面板的话，就使用堆栈，框架也会有一些对外开放的接口，比如打开面板、关闭面板、获取当前面板、释放资源、加载资源等，也是可以根据需求进行添加的

打开面板的时候，他有个枚举类，列举了所有的面板，然后打开的时候往打开面板的方法里面传需要打开的那个面板的枚举，打开的时候是先判断面板有没有加载到字典里，如果没有的话就把这个面板加载到内存，存到字典里，然后通过调用基类初始化的方法，调到对应面板的具体实现，然后就是打开面板 给他显示出来。面板关闭的话也是使用CanvasGroup组件，调整透明度给面板消失掉。

然后 面板的管理规则，主要是区分一下面板的类型吧，就是常驻面板，和不常驻的面板，常驻面板比如说是 背包商城这些，打开频率比较高的，就加入内存后不卸载，一直存在内存中

不常驻的面板，就是那种打开频率不高的，可以就是直接卸载掉。，也可以走资源管理框架那个，超过一定时长后卸载。

区分这个频率的话，刚开始的时候可以是策划来定，后面可以根据测试的数据进行调整

然后面板功能的实现是 采用的一个MVC思想，v层会继承框架的基类然后挂载到面板上面，受框架的管理，然后面板的数据来源主要是服务器下发，一种是游戏初始化的时候下发，然后数据发生变化的时候再次下发，数据存到面板的m层，这种主要是适用于背包商城这种的面板，就是打开的时候可以直接用，还有一种是面板打开的时候，向服务器获取数据，比如拍卖行、邮件、好友列表这种，需要实时获取的。通过框架打开面板的时候，会调用v层初始化的方法，根据数据生成对应的显示，两种方案的数据发生变化后是通过消息系统通知v层进行一个面板的刷新。

框架的优化这一块是采用Canvas分层的一个方案，因为canvas是ui渲染的一个基本元素，分层也能避免反复渲染也就是overDraw，我记得当时项目分层是分了5层，背景层，面板层，tipes提示信息层，人物模型一个层，引导层，还有就是采用动态合批和静态合批，以及图集打包来进行一个ui的优化，需要注意的是，避免频繁删除或者增加ui对象，避免元素数目过多，这俩都会增加渲染方面的消耗，还要注意text文本文档的使用，他会打断合批，尽量使用艺术字或者换成TextMeshPro，打包图集的时候需要注意，加载图集的时候会把整个图集加载进入内存，哪怕有的图片不需要他也会加载进内存，还有图集元素差距过大也会导致空间浪费，所以打包的时候需要区分一下，不常用的图打一个，常用的那些图，公共的图打一个，面板自己的图集打一个

整个框架的设计主要是各个面板之间的耦合度尽可能降低，从而提高代码的可维护性和可扩展性，然后新同事上手快，也需要方便拓展一些其他功能。