UI框架

首先UI框架就是用来管理所有UI面板的一个框架，框架中会有一个所有面板类都会继承的基类，在这个基类中会有所有面板共同的方法，比如：初始化方法（Init）、面板显示时的方法（OnEnable），面板销毁时的方法（Destory）等等，当UI面板类去继承基类的时候能够重写这些方法，我们可以在面板类中去写各自面板的方法，在框架中有一个管理所有面板的类，在这个类中会有管理UI面板打开（OpenPanel）、关闭（ClosePanel）、隐藏（HidePanel）和释放（Destory）的方法，我们会通过这些方法去管理UI面板，当然还能有一些其他的方法，然后在这个类中会有三个数据结构去存储UI面板，一个存储所有的UI面板，一个是用来存储那些常驻内存的面板的，比如：背包，商城，宠物等等；一个是用来存储那些常驻内存的面板的，比如：排行榜等等，这些也不是绝对的，需要我们在测试游戏的时候去进行调试和测验。当我们运行游戏的时候我们会加载所有的UI面板，然后将所有面板隐藏掉，并根据面板的类型放到不同的数据结构里，常驻内存的放进常驻内存的数据结构，不常驻内存的放进不常驻内存的数据结构，我们不常驻内存的面板释放有两个方案，一是根据时间，如果这个面板有五分钟或者十分钟没有打开过，那么这个面板就会冲数据结构里进行删除，并且把这个面板对象销毁掉，需要显示的时候再重新加载这个面板进行显示，另一个方案是根据数据结构的容量（数量），当这个数据结构的容量达到规定容量时，会把这个数据结构里面所有的面板进行删除销毁，这就是不常驻内存面板的释放方案。

当然，我们也有MVC结构的UI框架，首先MVC一共分为三个层级，数据层（Model）、显示层（View）和业务逻辑层（Control），MVC形式的UI框架中会有三个基类，一个是数据方面的基类（Model），一个是UI面板显隐的基类（View），最后一个是写业务逻辑的基类（Control），当然这三个类也会有面板共有的方法，我们会通过编辑器去创建三个面板的类去分别继承这三个类，我们可以在这三个类中分别去写面板在这三个层级中会用到的方法，然后如果要调用其他类里面或者其他面板里面的方法的话，我们会通过消息中心或者事件中心去发消息通知其他类去执行对应的方法，实现数据与逻辑分离，这是MVC结构的UI框架。

我们在写UI框架的时候需要根据我们的项目去制定方案，UI框架不只有灵活性，还要有扩展性，能够让我们在使用UI框架的时候写一些额外的方法去扩展这个框架，这样才是一个好的UI框架。