UI框架所具备的功能:

1. 问题:----缓存常用UI窗体

游戏中多个场景会反复出现相同的”UI窗体”,造成多个场景中反复加载相同的UI窗体。

1. 问题:----UI窗体间的传值

各个UI窗体(UI脚本)之间的传值，容易出现”紧耦合”的情况，”可复用性”降低。

解决思路:

1. P2p

GameObject wnd = WindowsManage.GetWindows(“TestWnd”);

If (wnd != null)

Wnd.GetComponet<TestWnd>().TestMethod();

2.观察者模式

UI窗体之前不直接传值，用“消息传递中心”（观察者模式）的机制负责UI窗体间的通信。

1. 问题: -----管理UI窗体层级，UI导航

游戏中很多情况下会出现多个“窗体”叠加的现象，需要“手动”维护窗体间的层级关系。

解决思路：

3.1层级管理:

网络转菊花层

Loading加载层

引导层

Tip层

通用层(1.功能层 2.弹窗层)

主界面层

3.2 UI导航

使用自定义“**栈”**的数据结构，保存与控制当前所有需要显示的“U I窗体”的层级关系。

1. 问题:----模态窗口

多个“UI窗体”叠加出现时，窗体具有屏蔽上一层窗体机制,遮蔽点击游戏中的3D游戏对象等等。

解决思路:

1. 在每个窗体创建的时候，自动添加一个遮蔽层，就像cocos那样的做法
2. 用“栈”来缓存UI窗体，每创建一个新的UI窗体，就会遍历上一层UI窗体的所有BoxCollider，将其关闭。----我认为这个方法不太好，如果UI窗体的BoxCollider很多呢，去遍历一篇，本身就挺消耗的。
3. 在制作预制体的时候就创建好一个MaskLayer遮蔽层(不够智能,可以写个工具自动排序好depth，并且添加一个MaskLayer)-自定义的扩展性比较强
4. 专门创建一个全局MaskLayer图层,一旦有窗口创建层级就放在他下面(要写好管理，要不然容易穿帮)

5.问题:---窗体生命周期

1.Show()

2.Freeze() 冻结-不接收消息处理

4. Hide() 接收消息处理

5.ReShow()

6.End() 销毁窗体

6.问题:---窗口动画

7. 问题:---国际化

我这里也有一个问题： 1.在切换场景的时候是否要清除所有的窗体(包括缓存的),那样在每次切换场景的时候每个需要缓存的窗口都是第一次打开的状态，界面复杂点的，第一次打开的时候会卡顿， 2.需不需要在loading界面的预加载需要缓存的窗体

UI框架设计图:



