1.Panel开启&关闭（支持多开）

2.动态子界面(Page)支持

3.层级管理(支持子Page和多Canvas)

4.动态Panel间隐藏&交互关系支持

5.界面缓存支持(可设置最大缓存数量)

6.底层面板、顶层面板、自定义层级面板支持

7.依赖面板支持

8.Panel生命周期管理

9.依赖资源预先加载:(此处有一隐患:[LoadPanelResAsync 562][LoadPageResAsync 606] 目前的回调是接受主资源加载完成的，此时依赖资源是否确定也已经完成加载???)

a.关闭所有面板

b.隐藏所有面板

c.销毁所有面板(包括缓存)

d.获取当前顶层面板(参考导航面板)

///////////////

基本原理

1.面板隐藏原理:

通过修改GameObjectLayer为HideUI实现

2.层级调整原理

通过修改Canvas的SortOrder实现

1.从Page继承的类切记不要重写Awake和OnDestroy方法

2.从Panel 继承不要重写OnValidate，如有需求改成虚函数重载吧

3.Panel并不是通过GameObject的SetActivity来实现缓存功能的，所以如果面板中有Update方法，则面板被缓存的情况下会继续执行

4.GameObjectPool有缓存策略，UI的缓存策略不是通过GameObject的SetActivity来实现隐藏的，所以如果有实现Update方法，则在缓存中也会继续执行(可以通过继承或则添加PoolObjectComponment组件来接受进入缓存回调)

UI相关的生命周期函数:

Page:

OnUIInit():单个Panel&Page永远只有一次

OnOpen():和OnClose成对

OnClose():和OnOpen成对

OnViewEvent()

BeforeDestroy()

Panel:

OnBecomeHide()：

OnBecomeVisible()：

OnPanelOpen():每一次Panel被请求开启都会调用。依赖面板重入也会再次调用。

OnViewEvent() //子类继承需要调用base

BeforDestroy() //子类继承需要调用base