# 客户端框架说明书

目录

[客户端框架说明书 1](#_Toc48293323)

[1、 UI管理器 1](#_Toc48293324)

[a) PageManager 1](#_Toc48293325)

[b) Page 1](#_Toc48293326)

[c) NodeManeger 2](#_Toc48293327)

[d) Node 2](#_Toc48293328)

[2、 资源管理 2](#_Toc48293329)

[a) BundleManager 2](#_Toc48293330)

## ****UI管理器****

采用Page+Node组合结构，Page只允许同时存在一个，加载新的Page时默认销毁旧的Page及相关资源；Node允许同时存在多个，生成时默认父节点为当前Page，并加载在最上层，切换Page时，默认销毁所有Node及相关资源。

### PageManager

Page管理器，负责所有Page的管理，存储UI关键数据

* + 1. 关键函数/接口说明：
       1. OpenPage<T>：根据Page类名打开指定Page
       2. OpenPage：根据Page名打开指定Page
       3. OpenLastPage：打开上一个Page

### Page

所有Page的基类，所有的Page必须继承该类

* + 1. 关键字段
       1. resourceDirectory：Page自己的资源目录
       2. spriteAB：创建Page时默认加载对应资源文件夹中的mainPic文件夹Bundle资源
    2. 关键函数/接口：
       1. Init()：默认初始化函数，默认只会执行一次，执行后Page隐藏
       2. Open()：每次执行对应OpenPage()后默认自动执行，执行后Page显示，等同于OnEnable
       3. Close()：加载新Page时旧Page的Close()自动执行，执行后隐藏旧Page，在新Page加载完成后销毁旧Page
       4. GetSpriteAB()：获取当前Page的默认图片资源AssetBundle
       5. GetSpriteByDefaultAB()：从当前Page的默认图片资源AssetBundle中获取一张图片
       6. GetJsonData()：获取当前Page的资源目录中获取指定名字的Json数据
       7. OpenFloatNode()：从当前Page的资源目录下加载一个FloatNode，该Node必须存放在该Page资源目录下的Prefab文件夹内
       8. OpenNode()：从当前Page的资源目录下加载一个Node，其余同上
       9. OpenPrefab()：从当前Page的资源目录下加载一个预制体，其余同上

### NodeManeger

Node管理器，负责Node部分UI相关内容的管理，存储最新Node的信息

* + 1. 关键属性/字段：
       1. currentNodePath：当前最新加载的Node的相对路径名
       2. DefaultNodeDirectoryPath：Node的默认存放根目录
    2. 关键函数/接口：
       1. OpenNode()：根据Node名打开对应Node，该Node影响Node间跳转
       2. OpenNode<T>()：根据Node类名打开对应Node，该Node影响Node间跳转
       3. OpenFloatNode<T>()：根据Node类名打开一个不影响跳转的Node
       4. DontDestroyAtChangePage()：设置某Node在加载新Page时不销毁

### Node

所有Node的基类，所有的Node必须继承该类

* + 1. Init()：默认初始化函数，默认只会执行一次，执行后Node隐藏
    2. Open()：每次执行对应OpenNode后默认执行，执行后Node显示，等同于OnEnable
    3. Close()：关闭Node时需要执行的函数，执行后销毁Node
    4. GetSpriteAB()：获取当前Node的默认图片资源AssetBundle
    5. GetSpriteByDefaultAB()：从当前Node的默认资源AssetBundle中获取一张图片
    6. GetJsonData()：获取当前Node的资源目录中获取指定名字的Json数据
    7. OpenPrefab()：从当前Node的资源目录下加载一个预制体

## 资源管理

### BundleManager

Bundle资源管理器，单例

* + 1. 关键函数/接口：
       1. GetCommonJson()：获取目标Json文本
       2. GetCommonJsonData()：获取目标JsonData对象
       3. GetAudioClip()：获取一个音频文件
       4. GetSpriteBundle()：获取一个图片Bundle，并且不执行Unload
       5. GetSprite()：从目标Bundle中获取目标图片
       6. GetSprites()：从目标Bundle中读取所有图片，可选Unload
       7. GetAnimationSprites()：从一组Sprite中获取一组对应名称的Sprite组
       8. GetBundle()：获取一个AssetBundle
       9. GetGameObject()：获取一个GameObject
       10. GetGameObjectByUI()：获取一个UI预制体（Node或Page）
  1. 动态资源存放路径：

根目录：Assets/Editor Default Resources/AssetBundle

注：只有在Assets/Editor Default Resources/AssetBundle文件夹下的资源才会自动设置BundleName并自动生成Bundle

* + 1. 公共资源目录:

根目录：../CommonResources

* + - 1. 音效：../Audio/Sound
      2. 音乐：../Audio/Music
      3. 配置文件(Json)：../Config
      4. 热更库文件：../HotFix
      5. 通用公共资源：../Sprite/Common/
    1. 预制体目录：
       1. 非UI预制体：../Prefab
       2. 杂项预制体：../Prefab/OtherSundry
       3. 通用Tip预制体：../Prefab/Tips
       4. 所有Node：../ UIResources/Node
       5. 所有Page：../UIResources/Page
    2. 公用字体：../Font
    3. 公用Shader：../Shader
    4. 所有Page自用资源，资源结构和公共资源结构相同，目录结构为

../ UIResources /Page/PageName（自身Page名）静态资源：所有Resources文件夹下资源均为静态资源，读取方式和文件结构同动态资源，在有同路径名动态资源的情况下，优先加载动态资源

1. **音频管理：AudioManager，单例**
   1. 关键属性说明：
      1. music：背景音乐声音大小
      2. sound：音效声音大小
   2. 关键函数/接口说明：
      1. PlayTempSound()：播放一个临时音效，该音效资源在相关界面销毁后会自动释放
      2. StopTargetTempSound()：停止播放临时音效
      3. PlaySound()：播放一个音效，该音效不会被自动释放
      4. PlayMusic()：循环播放一个音乐，直到关闭或切换另一个音乐
      5. SetMusic()：设置音乐声音大小
      6. SetSound()：设置音效声音大小
2. **下载管理器DownloadManager（单例）**
   1. DownloadFiles()：下载多个文件，使用协程
   2. DownloadFilesUserThread()：下载多个文件，使用线程
   3. DownloadFileUserThread()：下载单个文件，使用线程
   4. DownloadImage()：下载一张图片，使用协程
   5. DownloadAudio()：下载一个音频文件，使用协程
3. **工具管理**
   1. UGUI扩展组件工具
      1. UGUIEventListener：交互事件工具，不包含拖拽
      2. UGUIEventListenerContainDrag：交互事件工具，包含拖拽
      3. ClickAnimation：点击动画，按下缩放，抬起还原
      4. CommonAnimation：通用动画工具，包含位移、旋转、缩放、颜色变化、透明度变化、尺寸变化、填充动画(fillAmount)
      5. GradientColor：图片渐变色组件
      6. GraphicOutline：描边组件
      7. MutillgualText：多语言文本组件，继承自Text
      8. SequenceAnimation：帧动画组件
      9. TextSpacing：文本字间距组件
      10. TextTip：点击弹出描述Tip组件
      11. Timer：计时器组件
      12. BrokenLineImage：折线图组件
      13. CircleImage：圆形图组件
      14. CustomImage：自定义图片组件
      15. RoundRectangleImage：圆角图片组件
      16. RadioButton：单选按钮
      17. Tab、TabGroup：Tab签组件，继承自Toggle和ToggleGroup
      18. TextNonBreakingSpace：Text文本优化组件
      19. PanelScrollRectCenter：滑动面板组件
      20. CombinedPanelScrollRect：滑动面板组合组件（ScrollRect和PanelScrollRectCenter的组合）
      21. CombinedChildScrollRect：滑动面板组合组件（ScrollRect和ScrollRect的组合）
      22. ScrollRectChildCenter：ScrollRect元素居中组件
      23. ScrollRectCircle：ScrollRect元素首尾循环组件
   2. Util：扩展工具类
      1. ConstantUtils：常量定义
      2. GlobalVariableUtils：全局变量定义类
      3. InputUtils：获取用户输入，鼠标点击，触摸屏幕等
      4. MiscUtils：非UI相关扩展
      5. OSSUtils：OSS上传集成类
      6. TextUtils：文本工具扩展
      7. UIUtils：UI相关扩展
      8. LogUtils：日志类
      9. WaitForUtils：协程用延迟类
4. **网络通信相关：长链接，采用Socket/TCP传输ProtoBuffer方式实现**
   1. 发送通信请求：MessageProcessor.SendMessage()，两个重载
   2. 接受到返回后的处理：采用委托池方式
      1. 公共消息监听：MessageEventPoolManager.AddMessageListener()
      2. 在目标物体上绑定监听：gameObject. AddMessageListener()
      3. 在目标物体上删除对应监听：gameObject. RemoveTargetListener()
      4. 在目标物体上删除全部监听：gameObject. RemoveAllListener()
5. **提示弹窗：TipManager，单例**
   1. 核心函数：OpenTip()