**创建UI的步骤**

1，创建panel，确定好名字，后续不允许修改

2，添加UISystem，确定好ui的类型（full , pop , top）

3，在prefab资源路径表格中添加好对应的路径

4，在lua/ui文件夹下添加对应的lua文件，名称和prefab的uiName一致

5，在include.lua文件中添加require

**使用规范：**

1. UI分为三类：全屏界面（FULL）,弹出界面（POP），置顶界面（TOP）

1) 全屏界面：有一张全屏的非透明的贴图。UILayer为：FULL;

2) 弹出界面：可以透视看到后边的UI。UILayer为：POP;

3) 置顶界面：永远显示在最上层的界面，如跑马灯消息等。UILayer为：TOP

1. UI中，禁止将dlg的资源直接做在Prefab中（例如某个界面的帮助信息弹窗），凡是弹窗打开的界面，都需要按照一个单独dlg的样式制作。
2. 动态load的item类的prefab，事先在inspector面板上指定好，禁止在代码中通过ResourceManager进行load。界面专用的item，无需另建prefab。直接放在界面中就行。
3. 在UI中，动态设置图片的时候，需要通过UIManager.instance.GetSprite()获取新的资源，禁止直接使用ResourceManager.instance.LoadSprite()。
4. UIText中，除非特殊情况，都需要勾选上Best Fit，并且，其中Max Size应该是设计时使用的字体大小，并且在button上的UIText的Anchors必须min(0,0)，max(1,1)。
5. 脚本中为UIText动态设置中文时，禁止使用UIText.text=”内容”的形式，需要使用UIText.text=Localization.Get(id)；
6. 禁止Lua的Open()和Close()方法中添加任何逻辑，其中只是保存着界面Open和界面Close的逻辑。
7. 需要打图集的资源，统一在对应的文件夹下建立一个SpriteAtlas的文件，数据参考DlgMain\_Atlas。
8. 避免使用“不可见”的Image作为交互响应的控件，统一使用UIRaycast
9. 全屏背景，需要设置pivot为(0.5,0.5)，anchor为(0.5,0.5,0.5,0.5)，最后挂上FullScreenKeepAspect脚本。

**文件夹功能定义：**

|  |  |
| --- | --- |
| 文件夹 | 描述 |
| Res/Atlas/... | 每一个界面所用的图集文件夹，不在图集中的资源不放在此处 |
| Res/Atlas\_Common/Bottom | 界面背景层公用图集，例如弹窗背景，item背景 |
| Res/Atlas\_Common/Front | 界面非背景层公用图集，例如按钮，红点等 |
| Res/Sprites | 各个系统中公用的大图放在此文件夹中 |
| Res/**SpritesGroup**/... | 每个系统中独有的大图放在对应系统名称的文件夹中 |
| Res/Icon/Item | 整个游戏用的道具图标 |
| Res/Icon/Skill | 技能图标 |
| Res/Icon/Other | 无具体意义的图标，例如任务item中的图标 |
| Res/**Prefab/UI** | UI使用的prefab都直接放在此文件夹中，严禁建立子文件夹 |
| Res/Audio/BGM | 存放背景音乐 |
| Res/Audio/Sound | 存放音效 |
| Res/Audio/Voice(1...) | 角色对白，每个角色的一个文件夹 |
| Res/Font/... | 每个CustomFont的相关资源都放在对应的文件夹中 |
| Res/FX | 特效文件夹，特效相关的内容均放在此处 |
| Res/Spine/... | 存放spine的内容，一个文件夹存放一个spine文件，并且文件夹的名称和spine的名称一致 |

注：

* Res文件夹中的所有文件夹、子级文件夹的名称都不能重名
* 图集资源，界面做完之后，每个图集的宽高不能超过2048\*2048，程序设计完UI后，需要自行检查最终图集的大小

**资源命名规范：**

1. 文件夹名称：首字母大写，例：SG\_Main、SpritesGroup、DlgMain等
2. 资源名称：首字母大写，例：Cm\_Ft\_RedPoint、DlgMain\_HeadFrame等
3. 特效：Prefab以FX\_\*\*命名，其使用到的其他资源以FX\_Tex\_\*\*命名，shader文件除外。例如某特效，命名为FX\_Boom，其用到的贴图命名为FX\_Tex\_boom1,等
4. Spine：一个文件夹存放一个spine文件，并且文件夹的名称和spine的名称一致
5. 全身像立绘：存放在Sprites文件夹中，文件前缀SP\_Role\_\*\*，具体参考《通用命名规范》
6. **UI：**

* 图集资源：在Atlas文件夹下建立UI图集资源文件夹Dlg\*\*\*，例如DlgMain文件夹。文件夹中的图片资源需要都以DlgMain\_开头。例如DlgMain\_HeadFrame。
* Common图集：在Bottom文件夹中的资源以Cm\_Bt\_\*\*命名，例Cm\_Bt\_DialogBg；在Front文件夹中的资源以Cm\_Ft\_\*\*命名，例：Cm\_Ft\_RedPoint
* 非图集独用资源：此界面独用的资源，在SpritesGroup中建立对应的文件夹SG\_\*\*\*,例如SG\_Main,文件夹中的资源都以Sg\_Main\_开头，例如Sg\_Main\_Bg
* 非图集公用资源：多个界面公用的资源，直接放在Sprites文件夹中，此时，资源的名称以Sp\_\*\*\*命名，例如：Sp\_Bg

1. **Icon**

* 道具icon：以Icon\_Item\_\*\*命名；
* 技能icon：以Icon\_Skill\_\*\*；
* 其他icon：主要用于展示用，没有特别具体的意义

**切图规范：**

1. 所有的资源，尽量使用九宫格样式图，同时，九宫格资源尽量小。
2. 单张贴图的留白或透明区域不能超过整张图的1/4。
3. 对称型图，如果单张的高或者宽超过150像素，需要对半切或者1/4切，具体方式，问程序；宽高都小于150像素的，可以直接给图。除特殊情况，禁止违反此规则
4. 同类型的资源，例如Icon，卡牌框等。尺寸一致，并且尽量是2的幂，实在不行，保持4的倍数。
5. 样式相同而大小不同的资源，优先考虑使用其中最大尺寸的资源，其他的出图时标注好宽高，如果大小相差太大，可以考虑出一个小的，具体和程序商量。
6. 对于公用贴图的定义：

* 图宽高都小于300的，如果在3个以上界面使用的，需要提取到公用图集中
* 图宽高都小于100的，如果在5个以上界面使用的，需要提取到公用图集中

1. 贴图尺寸太大的，和UI讨论是否有优化方法。
2. 大图尺寸尽量保持2的幂。实在不行：1.考虑切图的可能。2.保持4的倍数。
3. 大图尽量保持非透明。

导入贴图时的格式。

1. Android：非透明：RGB ETC2-4bit。透明：RGBA ETC2-8bit。
2. IOS：非透明：RGB 16。透明：RGBA16。