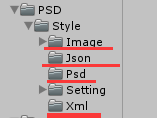
1. **Psd解析工具**

A：流程：提取原始信息（Extractor.jsxinc）-> 解析原始信息（Parser.jsxinc）-> 输出解析结果Json文件（JsonFileWriter.jsxinc） -> 输出资源XML列表（XmlFileWriter.jsxinc）-> 输出图层图片（ImageExporter.jsxinc）

Psd文件解析生成图层图片（Image）文件，Json文件和Xml文件。



B:重要技术元素：

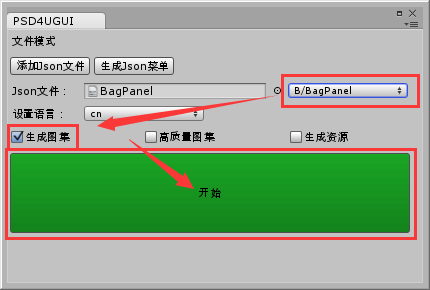
**Extractor.jsxinc**:提取PSD文件的原始信息。

**Parser**.**jsxinc**:根据组件定义规则对第一步生成的Psd原始信息进行加工、处理、验证。在这个文件中定义了每个组件类型的**解析器（parser）**，**验证器（validator）**，**子元素命名正则表达式（childrenRegExp）**以及各种参数解析、功能设置。例如：允许使用的字体名称列表，是否输出镜像图片等开关。

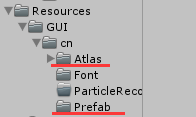
**ImageExporter.jsxinc：**将Psd文件中的图像图层输出为图片，当开启公共资源后，Psd中的公共资源图片不会输出，镜像图片的非保留图片不会输出，公共资源定义文件Shared.psd文件中的图像图层会全部输出。

1. **Unity中生成Prefab工具：**

Unity菜单PSD4UGU –> 文件模式



选择Psd解析结果生成的Json文件，勾选生成图集，点击开始按钮，生成图集及Prefab。

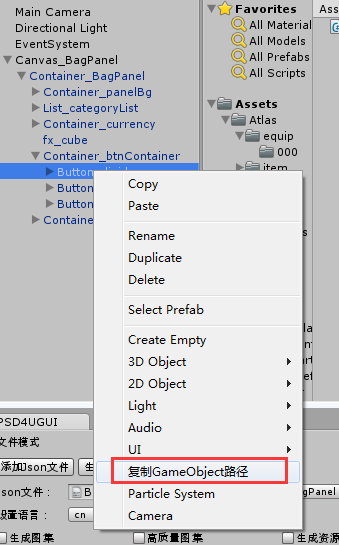


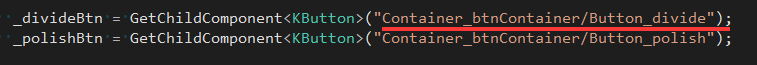
技术要素：

1. **生成图集（AtlasGenerator）**
2. 生成最小面积九宫图片（Scale9GridTextureProcessor.cs）：根据图片的九宫参数生成最小面积图片。
3. 图片描边（TextureClamper.cs）：给所有图片边缘添加2像素，内容和边缘相同，避免界面缩放时产生的黑线和裂缝问题。
4. 生成图集：将多张图片生成一张大的图集，可以在BatchSetting.json中定义将多个界面的图集合并。
5. ETC通道分离：将上步生成的图集分离rgb和alpha通道两张图片并使用etc1压缩。可以在QualitySetting.json中定义不使用etc而保留turecolor32的图集名城。
6. **生成Prefab（PrefabGenerator）**
7. 各个Creator根据Json信息生成GameObject及添加图片（ImageWrapper）和文本（TextWrapper）组件。
8. Creator分成**类型（Type）**和**特定（Specific）**，类型创建器根据类型名，特定创建器根据名字选择相关Json节点创建内容。
9. 创建Prefab过程中基本还原Psd中的图层结构，少数组件会有结构调整，如ScorllPage，ScrollView。
10. 应用Json中组件带的参数，创建阶段的参数hide，hideChildren等参数在创建阶段即应用，Deferred等运行时参数添加BuildHelper组件并记录参数留待Build阶段处理。
11. **创建阶段参数说明：**

**容器组件参数**：Hide：创建后通过SetActive(false)隐藏GameObject，HideChildren：隐藏容器子元素，**图片组件参数**：九宫参数：设置图片类型为Slice，UV参数：设置图片类型为Filled和填充方向。**文本组件参数**：对齐方式参数：设置文本的对齐方式，默认为UpperLeft。行间距参数LineSpaceing。

1. **快速获取GameObject路径：**



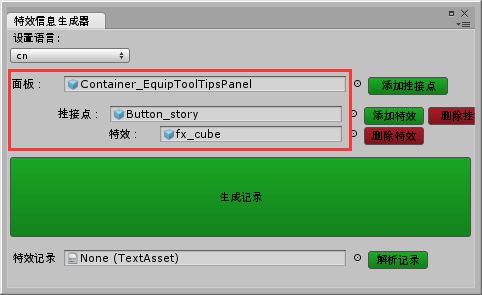


使用该工具可以快速、正确的获取组件的路径，并用于业务代码中。

1. **UI组件库：**
2. Build阶段：Prefab资源加载后根据各个节点的GameObject的类型或名称添加相应的功能组件。Build阶段参数：Deferred：添加DeferredComponent，其功能为将对应Go的Build工作延迟到业务阶段的第一次显示时。
3. 扩展组件库：根据游戏业务特点和Ps制作便利性，扩展UnityEngine.UI组件库。主要有以下几点：
4. 组件的坐标基准点为左上角。
5. 不使用Unity中自带的布局控制设置Anchors，Anchors设置为（0,1）, （0,1）。
6. Selectable组件默认的Transition设置为None，使用控制子状态Go的显示\隐藏来表示类似SpriteWrap效果。
7. 通过new定义覆盖底层事件，事件增加一个参数：事件触发对象，例如常用的按钮点击事件：public new KComponentEvent<KButton> onClick。
8. List组件是功能最复杂，带有灵活布局功能的一个组件，附带Unity工程示例。名为item的子元素是根据业务创建多条item的模板，名为template的子元素是根据业务创建item的内容时可以通过Instantiate克隆其资源再添加至item中，使用该方法可以减少计算资源消耗，提高游戏性能。
9. **UnityEngine.UI库修改：**
10. Image：**增加**：Filp属性用于表示镜像图片的翻转方式，Alpha属性用于改变图片的透明度值。**修改**：由于游戏业务中极少出现需要动态调整图片颜色值的情形，调整Color属性的原来用途，配合Shader的修改，用于表示图片的叠加特效，以减少图片数量。
11. BaseMeshEffect：当在文本上同时应用描边和投影特效时，Unity中对应添加Outline和Shadow组件，当这两个组件同时使用的时候存在一个可能的缺陷，会导致需要绘制的顶点数增多，视觉效果和Ps中也存在偏差。
12. **特效配置**

添加特效方式有如下几种：

1. 通过代码动态添加特效资源。
2. 通过扩展组件或占位符及相关代码统一添加特效。
3. 使用特效配置工具。



使用该工具可以配置在面板上的挂接点添加相关的特效，点击生成记录按钮相关的配置信息会以Json文件保存在GUI\cn\ParticleRecord文件夹下，生成Prefab过程中会读取特效的配置信息将特效包含在Prefab中。注意特效会作为兄弟节点添加在挂接点处，不是作为其子元素。