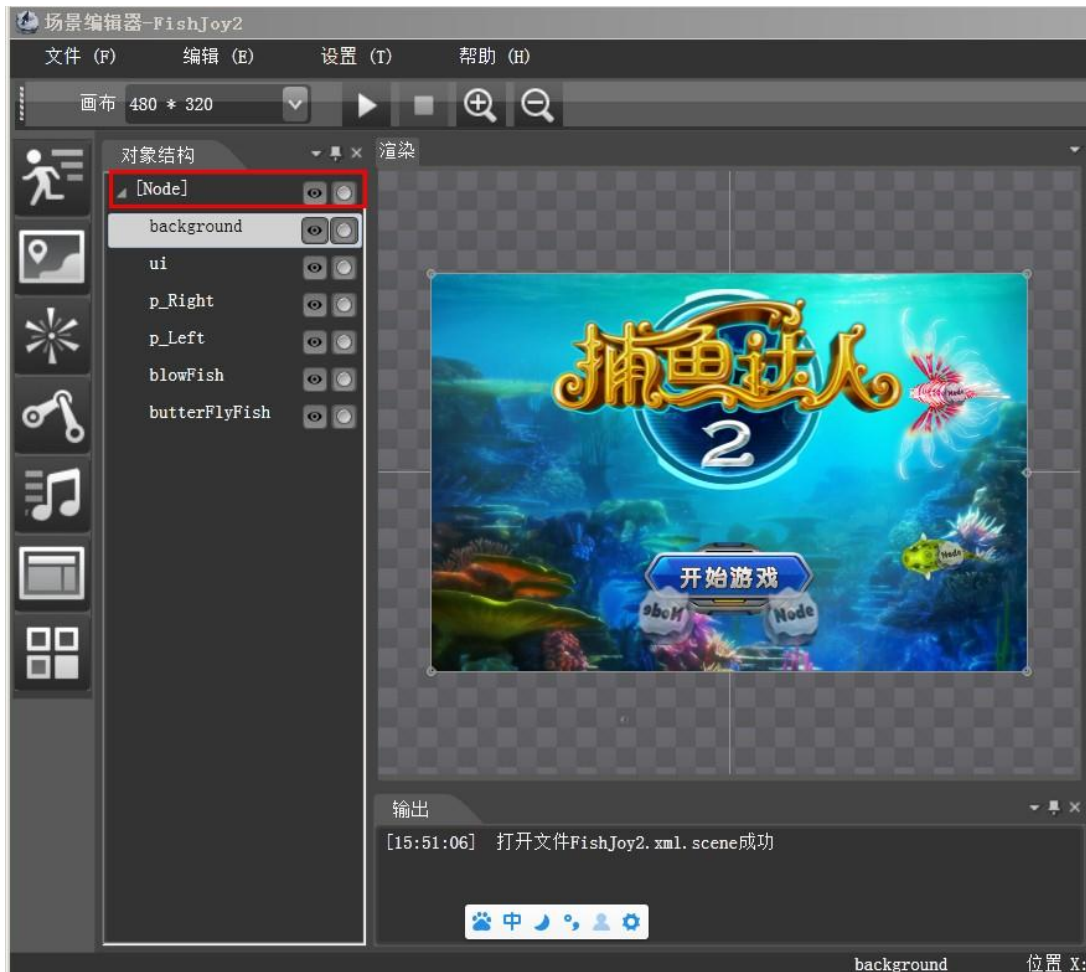


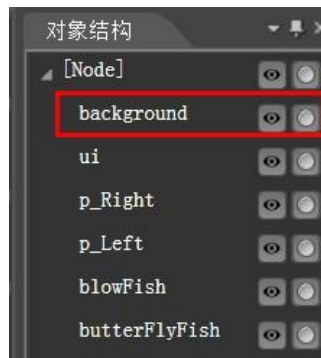
# 场景编辑器解析 API 使用手册

V0.1.0

场景编辑器是基于 cocos2dx 2.1.4 组件式编程来构建的，所以它的操作与场景编辑器的编辑方式有很大的关系，下面我们来分析 SceneJsonTest 的代码：



- 1、 上图红框代表根 Node，每个场景有且只有一个根 Node，而这个 Node 就是 cocos2dx 的 CCNode，在 SceneEditorTestScene.cpp 里面 SceneEditorTestLayer::createGameScene() 函数使用场景编辑器导出的 json 返回的就是这个根 CCNode，这一点请切记。



2、 Node 下面的每个对象，如图： background, ui, p\_Right, p\_Left, blowFish, butterFlyFish, 其实都是 CCNode，他们与跟 Node 的排列方式表明了他们都是根 Node 的子，并且挂上了不同的组件就具备了不同的功能，例如：background 就是挂载了 CCSprite 的 CCNode，看图：



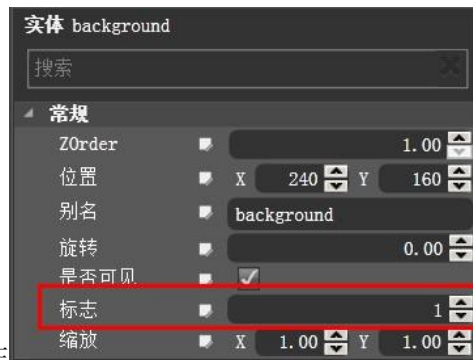
ui 是挂载了 CocoGUI 的 CCNode，p\_Right, p\_Left 是挂载了 CCParticleSystemQuad 的 CCNode，blowFish, butterFlyFish 是挂载了 CCArmature 的 CCNode，而所有的 CCSprite, CCParticleSystemQuad, CocoGUI, CCArmature 等，在场景编辑器里面叫做组件，他们是如何实现挂载的呢?请看 CCJsonReader.cpp 里面的 CCNode\*

CCJsonReader::createObject(cs::CSJsonDictionary \*inputFiles, CCNode \*parent) 函数，真相只有一个。

3, 既然我们知道了 background, ui, p\_Left, p\_Right, blowFish, butterFlyFish 这些都是 CCNode，而且是根 Node 的孩子，而 CCSprite, CCArmature, CCParticleSystemQuad, CocoGUI 都是这些 CCNode 的组件，那我们如何获取呢？

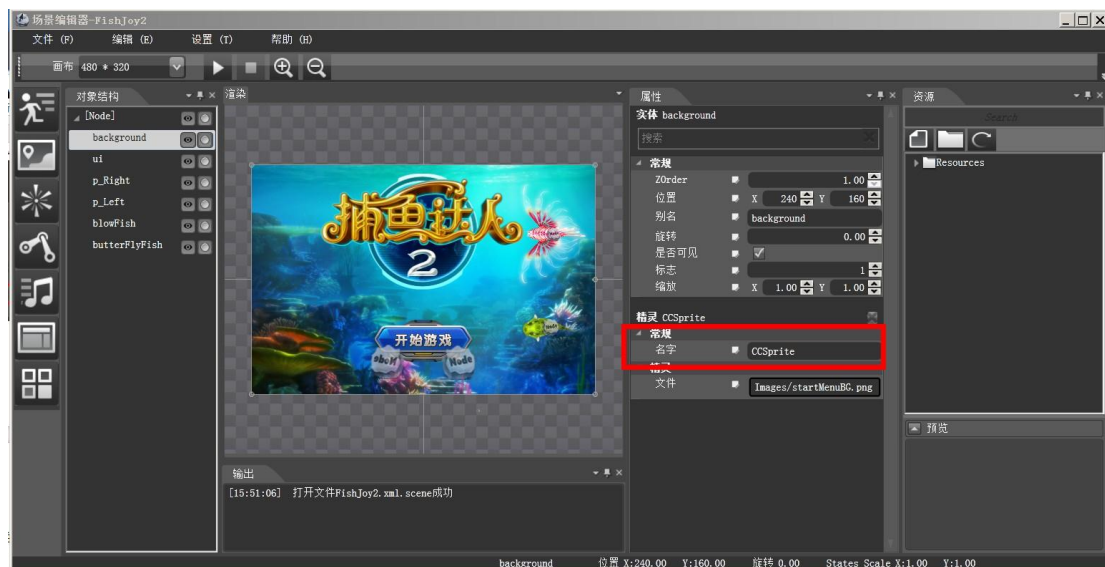
第一步：调用 CCJsonReader::sharedJsonReader()->createNodeWithJsonFile(“FishJoy2.json”) 返回根 CCNode，这个 CCNode 可以挂载到 CCLayer 上，从而可以绘制出来了。

第二步：我想操作 background 的背景，首先通过 getChildbyTag 获取到包含这个组件的



CCNode，这里注意一下 tag 是在 CCNode 指定的，目前默认是 -1，未来会修改成自增的 tag，就不用手动修改了。

第三步：获取到包含组件的 CCNode 之后，就可以调用 `getComponent("组件的名称")`，从而获取对应的组件，组件的名称在每个组件的属性栏设置，如：



特别注意一点：对于 CCSprite, CCParticleSystemQuad, CCArmature, CCTmxTiledMap, CocoGUI 组件，跟 `CCNode->getChildByTag(tag)->getComponent(name)` 所获取到的是

CCComRender 组件，具体的对象需要再调用一下 `getRender()`，这一点也请特别留意，而声音和属性组件可以直接获取。

以上便是组件式编程的实现方式，如果有兴趣可以看一下 `TestCpp/ExtensionTest/ComponentsTest`，在这个例子里面，我用组件的思想重写了 `coos2dx` 的 `SimpleGame`，大家可以对比一下这两种编程思路实现同一个游戏的差异，这也是一件很有意思的事情，感谢大家使用 `cocostudio`。