1. Cocos 2D-JS开发环境配置
2. 安装WebStorm;
3. 安装JDK，安装与系统对应的32位或者64位，记录下当前的安装路径:

D:\Program Files\Java\jdk1.7.0\_60

1. 下载Ant,并解压，解压路径为:

D:\Android\ANT

1. 安装Cocos Studio V2.0以上版本,安装此版本之后，系统会默认添加Cocos所需要的环境变量:
2. 配置环境变量:

ANT\_HOME D:\Android\ANT

ANT\_ROOT D:\Android\ANT\bin

JAVA\_HOME D:\Program Files\Java\jdk1.7.0\_60

以上环境变量，后面均没有分号。

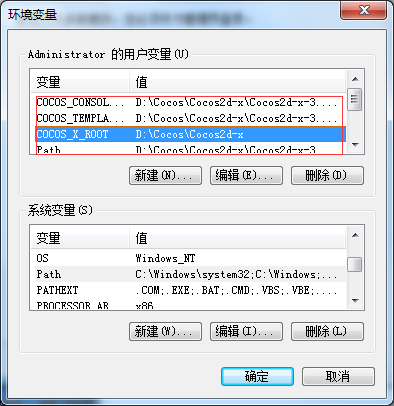
为path环境变量，追加新值:

%ANT\_HOME%/bin;%JAVA\_HOME%/bin;

如果上一步骤中，没有为Cocos添加相应的环境变量,可以手动添加:

例如:

Cocos 2D-X v3.10的版本的解压路径为: D:\Cocos\Cocos2d-x\Cocos2d-x-3.10\



为Path追加值:

D:\Cocos\Cocos2d-x\Cocos2d-x-3.10\templates;D:\Cocos\Cocos2d-x\Cocos2d-x-3.10\tools\cocos2d-console\bin

COCOS\_X\_ROOT D:\Cocos\Cocos2d-x

COCOS\_TEMPLATES\_ROOT D:\Cocos\Cocos2d-x\Cocos2d-x-3.10\templates

COCOS\_CONSOLE\_ROOT D:\Cocos\Cocos2d-x\Cocos2d-x-3.10\tools\cocos2d-console\bin

1. 创建一个新项目:

在配置环境变量的步骤中，已经配置了一个Cocos 2D-X v3.10版本的引擎，如果使用这个引擎，创建新项目，可以在任意位置，

Shift+右键🡪在此处打开命令窗口🡪输入下列命令

在E:\MyProjects\Cocos2D-JS目录下，创建一个js的新项目，项目名为Hello

cocos new E:\MyProjects\Cocos2D-JS\Hello -l js --no-native //最简

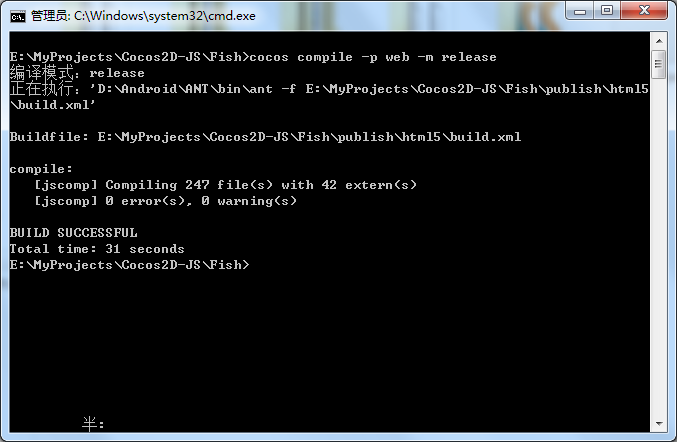
cocos new E:\MyProjects\Cocos2D-JS\Hello -l js //完整

如果不想使用配置的引擎，创建新项目，可以在对应引擎的目录下【\tools\cocos2d-console\bin】输入上述语句，创建对应的引擎版本的项目。

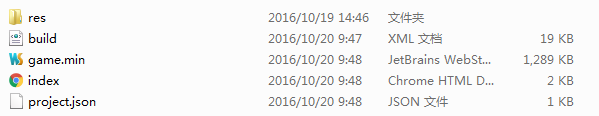
1. 打包发布:

在对应项目的根目录中，Shift+右击🡪在此处打开命令窗口🡪输入

cocos compile –p web –m release



运行之后，创建了文件夹publish,在publish\html5目录下，就是打包发布的项目:



1. Cocos 2D-JS左下角的三行数据意义:

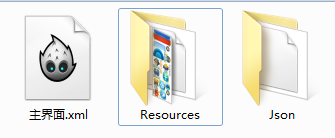


从上至下，依次为:

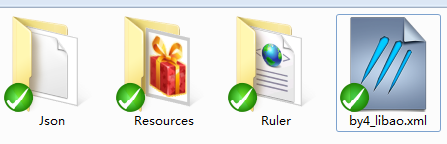
* 当前创建的渲染批次，简单理解为需要渲染多少个贴图出来;
* 渲染一帧需要的时间;
* FPS

1. 在Cocos 2D-JS项目中，引入Cocos Studio导出的资源:
   1. 安装Cocos Studio V1.6.0.0版本,使用UI编辑器、动画编辑器，打开美工传过来的UI工程文件(\*.xml.ui),美工传过来的

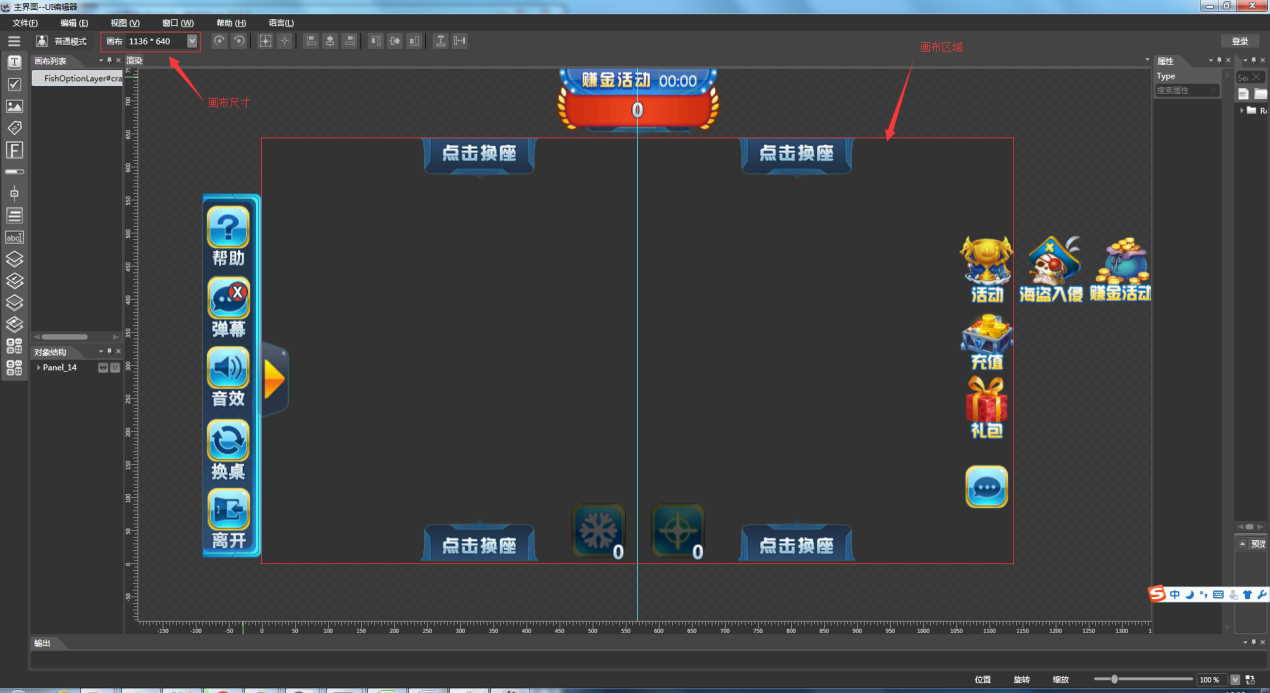
UI工程中的文件如下:



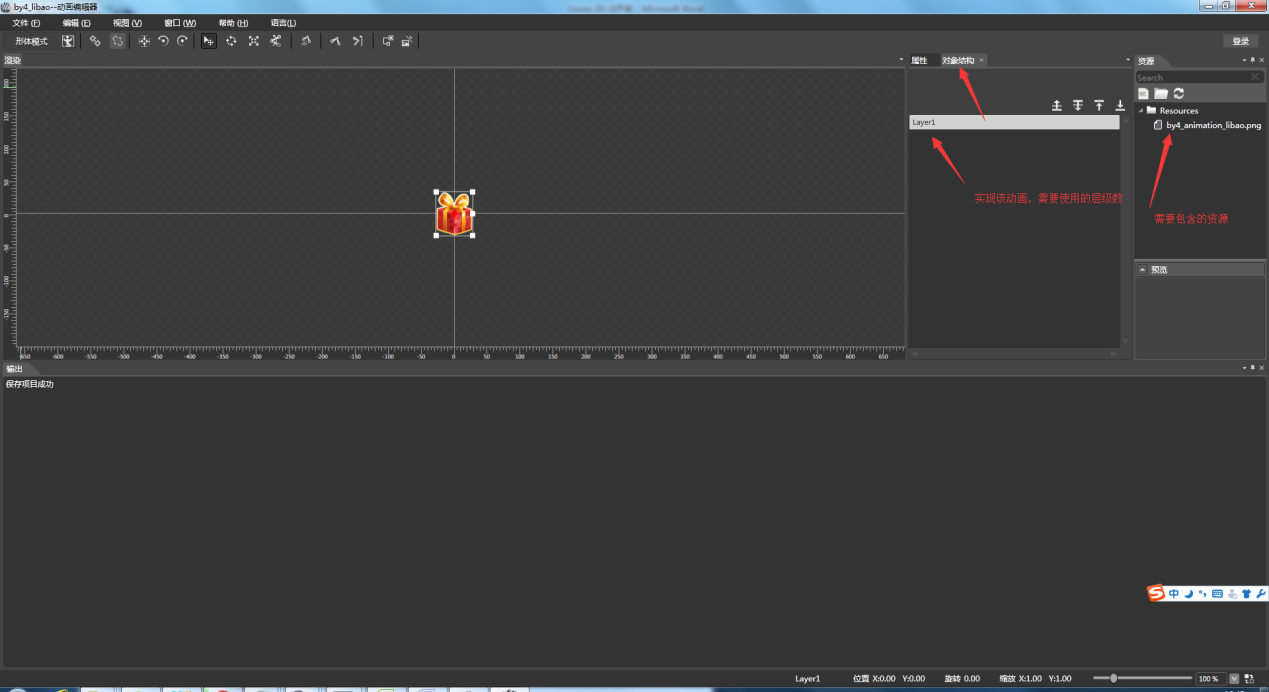
动画中的文件如下:



* 1. 打开之后的UI工程、动画工程文件如下:



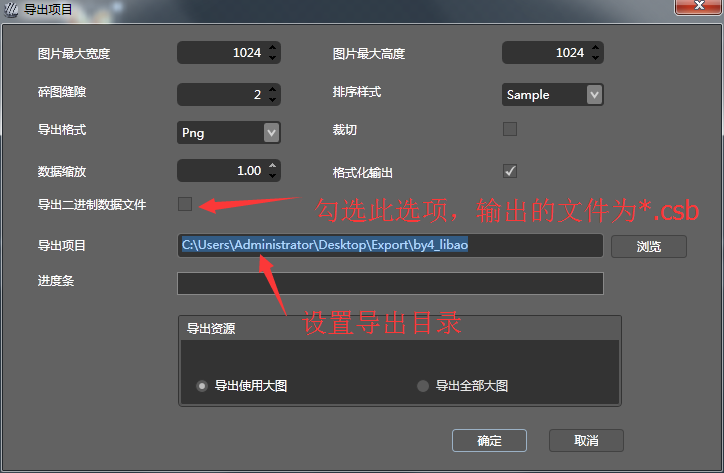
在UI编辑器中，可以使用画布尺寸，作为分辨率适配方案的canva尺寸。



* 1. 导出项目: 【文件】🡪【导出项目】或者快捷键【Ctrl+E】

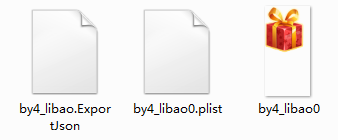


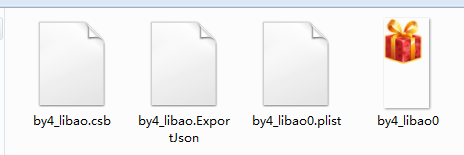
* 1. 动画编辑器中，在导出弹窗中，输入配置参数:

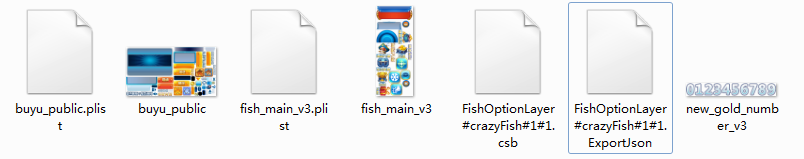


勾选“导出二进制数据文件”，导出文件文件的扩展名为”.csb”和”\*. ExportJson”，否则，扩展名为“\*.ExportJson”。

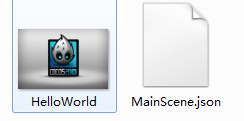
* 1. 导出后的资源列表如下:







* 1. 使用UI工程导出的资源，在Cocos 2D-JS项目中使用:



//在Resources.js引入上述资源:

**var *res*** = {  
 **MainScene\_png**:**"res/HelloWorld.png"**,  
 **MainScene\_json**:**"res/MainScene.json"**};

* 1. 在Cocos 2D-JS项目的project.json配置文件的modules列表中，追加“cocostudio”:



* 1. 引入:

//方法一:

**var** m\_menuScene= ***ccs***.load(***res***.**MainScene\_json**).node;  
**this**.addChild(m\_menuScene);

**其中，m\_menuScene在Cocos 2D-JS中，就可以当成一个节点使用了。**

**//方法二:**

**不同版本，所使用的加载方式不同**

**var rootNode;**

**if (version[0] == 1) {**

**rootNode = ccs.uiReader.widgetFromJsonFile(jsonFile);**

**} else if (version[0] == 2){**

**rootNode = ccs.csLoader.createNode(jsonFile);**

**}**

**获取widget**

**ccs.uiReader.widgetFromJsonFile();**

**或者ccui.helper.seekWidgetByName();**

* 报错:ccs is not defined(…)
  + 原因:

在project.json中，没有添加Cocos Studio组件;

* + 解决:

在project.json配置文件的modules列表中，追加(“cocostudio”)括号里的内容。



* 报错:Uncaught TypeError:Cannot read property ‘style’of null(…) CCBoot.js:729

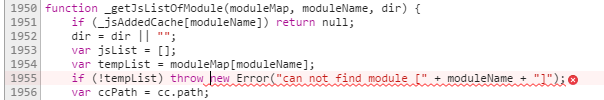


* + 原因:

在project.json配置文件的modules列表中，配置参数时，格式不正确。后面多加了一个逗号。

* 报错:Uncaught Error:Can not find module [cocosstudio](…)





* + 原因:

在project.json配置文件，添加的参数“cocosstudio”不正确，应该是“cocostudio”

* 报错:Uncaught Error:child alreadyadded. It can’t be added again(…)



* + 环境:

**var** m\_menuScene= ***ccs***.load(***res***.**MainScene\_json**);  
**this**.addChild(m\_menuScene);

* + 原因:

虽然ccs.load()方法的返回值为结构体

*{*

*node: cc.Node,*

*action: cc.Action*

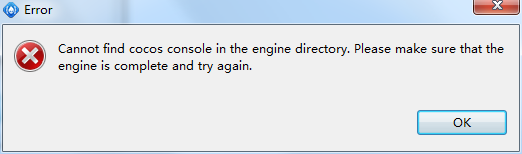
*}*

不能直接添加到Layer中。

* + 解决:

**this**.addChild(m\_menuScene.node);

1. Cocos Code IDE使用时遇到的坑:



在菜单栏Window🡪Perferences，弹出的对话框中Cocos🡪JavaScript🡪JavaScript Frameworks一栏中，输入要使用的Cocos 2D-JS引擎路径根目录时，使用的引擎不完整。

1. Cocos Studio使用时遇到的坑:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | E:\user\7cocos2d\test\myhibao>cocos compile -p web -m release  Running command: compile  Building mode: release  Traceback (most recent call last):    File "E:\user\7cocos2d\cocos2d-js-v3.3\cocos2d-js-v3.3\tools\cocos2d-console\b  in\/cocos.py", line 719, in <module>      run\_plugin(command, argv, plugins)    File "E:\user\7cocos2d\cocos2d-js-v3.3\cocos2d-js-v3.3\tools\cocos2d-console\b  in\/cocos.py", line 682, in run\_plugin      plugin.run(argv, dependencies\_objects)    File "E:\user\7cocos2d\cocos2d-js-v3.3\cocos2d-js-v3.3\tools\cocos2d-console\p  lugins\project\_compile\project\_compile.py", line 1435, in run      self.build\_web()    File "E:\user\7cocos2d\cocos2d-js-v3.3\cocos2d-js-v3.3\tools\cocos2d-console\p  lugins\project\_compile\project\_compile.py", line 1150, in build\_web      build\_web.gen\_buildxml(project\_dir, project\_json, publish\_dir, buildOpt)    File "E:\user\7cocos2d\cocos2d-js-v3.3\cocos2d-js-v3.3\tools\cocos2d-console\p  lugins\project\_compile\build\_web\\_\_init\_\_.py", line 80, in gen\_buildxml      jdk\_version = check\_jdk\_version()    File "E:\user\7cocos2d\cocos2d-js-v3.3\cocos2d-js-v3.3\tools\cocos2d-console\p  lugins\project\_compile\build\_web\\_\_init\_\_.py", line 16, in check\_jdk\_version      child = subprocess.Popen(commands, stderr=subprocess.PIPE)    File "C:\Python27\lib\subprocess.py", line 710, in \_\_init\_\_      errread, errwrite)    File "C:\Python27\lib\subprocess.py", line 958, in \_execute\_child      startupinfo)  WindowsError: [Error 2] |

* 环境:

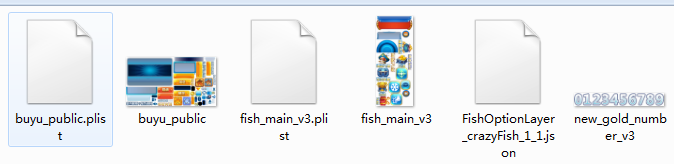
使用Cocos Studio打包时，报上述错误。

* 原因:

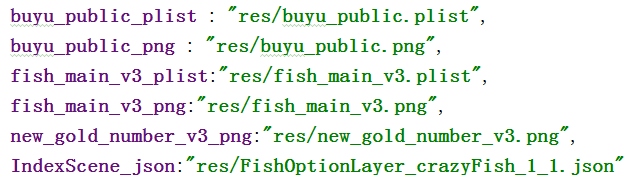
没有安装JDK，配置对应的JDK的环境变量。

1. 操作Cocos Studio导出的资源中的节点

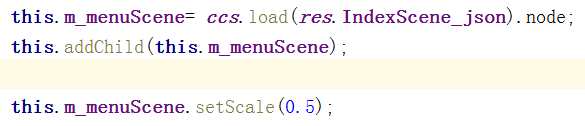
* 资源如下:



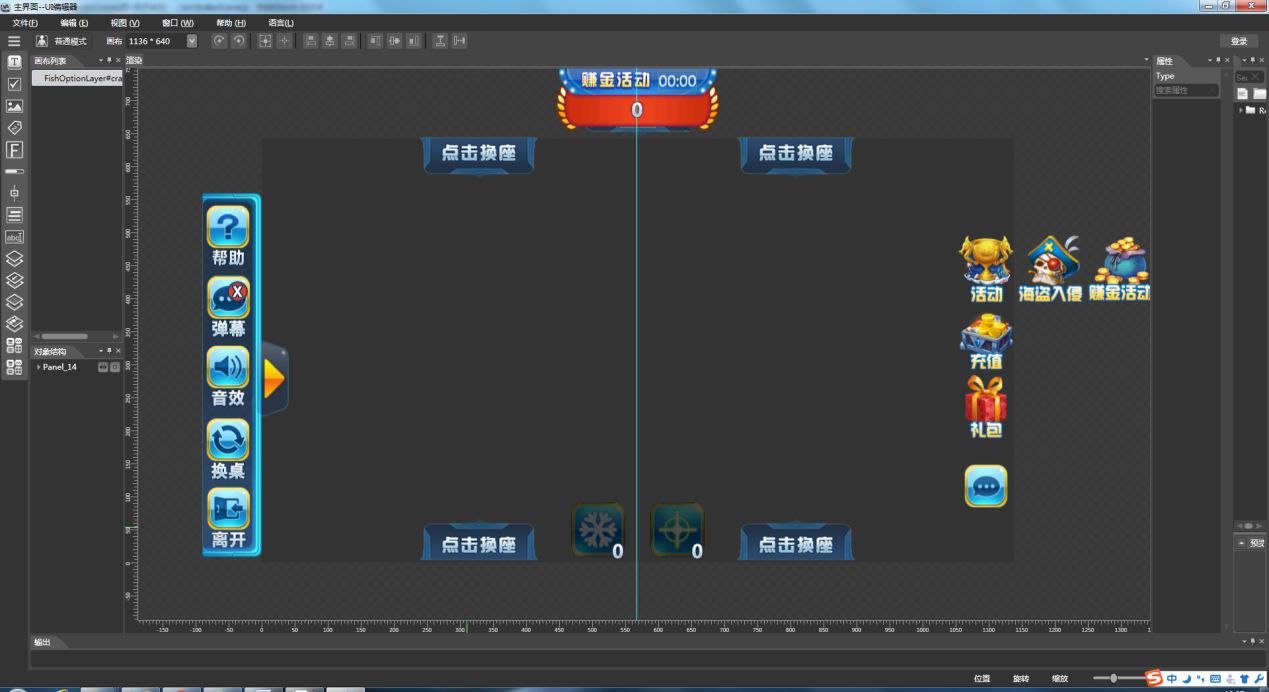
* 在Resources.js文件中，加载资源:



* 加载到游戏中:



* 美工使用CocosUIEditer做的UI界面效果:



* 加载后的游戏效果:



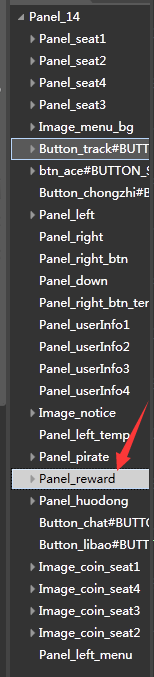
* 目标效果:这一个Panel,从上至下运动



其中，UI编辑器中，名称为Panel\_reward就是想要操作的Panel的名称。根节点Panel\_14就是添加到Layer中的this.m\_menuScenej节点。

代码实现:

**var** box= **this**.**m\_menuScene**.getChildByName(**"Panel\_reward"**);  
**var** moveBy= ***cc***.moveBy(1, ***cc***.p(0, ***cc***.**winSize**.**height**\* (-0.35)));  
box.runAction(moveBy);



* 设置Panel面板中文本数值:



获取、操纵某个节点,必须从根节点🡪父节点🡪子节点，逐步遍历，直到找到该节点为止:

**this**.**m\_labelTimer**= box.getChildByName(**"Label\_reward\_daojishi"**);  
**this**.**m\_labelTimer**.setString(**"20:00"**);

同理，设置数字标签的取值:



*//金币数***var** gold= box.getChildByName(**"AtlasLabel\_rewardpool"**);  
gold.setString(**"2"**);

* 为Button类型，添加单击响应事件:



**//获取节点**

**this**.**m\_btn\_huodong** = ***ccui***.**helper**.seekWidgetByName(**this**.**m\_menuScene**, **"Button\_huodong#BUTTON\_SOUND\_CLICK#BUTTON\_ANIMATION\_ZOOM\_OUT"**);

**//添加响应事件**  
**this**.**m\_btn\_huodong**.addTouchEventListener(**this**.onButtonTouch);*//活动*

**//按钮响应事件**

**onButtonTouch:function(sender, type){  
 *//type的类型有以下四种:  
 /\*  
 ccui.Widget.TOUCH\_BEGAN = 0;//开始点击  
 ccui.Widget.TOUCH\_MOVED = 1;//移动  
 ccui.Widget.TOUCH\_ENDED = 2;//结束  
 ccui.Widget.TOUCH\_CANCELED = 3;//触摸取消  
 \*/* var indexLayer= *cc*.director.getRunningScene().getChildByTag(1);  
 *//仅仅响应TouchEnd事件* switch (type){  
 case *ccui*.Widget.TOUCH\_ENDED:  
 {  
 switch (sender.getName()){  
 case "Button\_chongzhi#BUTTON\_SOUND\_CLICK#BUTTON\_ANIMATION\_ZOOM\_OUT":*//充值* {  
 console.log("充值！");  
 break;  
 }  
 case "Button\_huodong#BUTTON\_SOUND\_CLICK#BUTTON\_ANIMATION\_ZOOM\_OUT":*//活动* {  
 indexLayer.controlPanel\_HuoDong();  
 break;  
 }  
 case "Button\_libao#BUTTON\_SOUND\_CLICK#BUTTON\_ANIMATION\_ZOOM\_OUT":*//礼包* {  
 console.log("礼包！");  
 break;  
 }  
 case "Button\_chat#BUTTON\_SOUND\_CLICK#BUTTON\_ANIMATION\_ZOOM\_OUT":*//聊天* {  
 console.log("聊天！");  
 break;  
 }  
 case "Panel\_menu\_btn":*//展开左边的箭头* {  
 indexLayer.controlPanel\_Left();  
 break;  
 }  
 default :{break;}  
 }  
 break;  
 }  
 default :{break;}  
 }  
}**

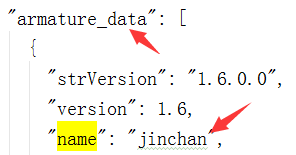
**说明:**

**在按钮的响应事件函数中，this指针不是本类，此处需要注意。**

* 使用美工制作的动画文件资源(\*.ExportJson、\*.png和\*.plist),在游戏中播放动画:

*//加载动画资源文件****ccs***.**armatureDataManager**.addArmatureFileInfo(***res***.**testEasing\_ExportJson**);  
**var** armature = ***ccs***.**Armature**.create(**"jinchan"**);*//获取动画节点  
//配置参数*armature.**scale** = 0.2;  
armature.**x** = ***cc***.**winSize**.**width** \* **Math**.random()\* 0.5;  
armature.**y** = ***cc***.**winSize**.**height** \* **Math**.random()\* 0.5;  
**this**.addChild(armature);

**其中，”jinchan”是\*.ExportJson文件下的armature\_data/name**



* 上述步骤，仅仅只是在游戏中，添加了该节点，并没有实现动画功能:

//方法一:利用项目ID索引播放

armature.getAnimation().playWithIndex(0);

//方法二:利用动画名播放

armature.getAnimation().play(**"dead"**, -1 , **true**);

动画名的路径:animation\_data🡪mov\_data🡪name;

动画的总数量是由animation\_data数组的长度决定。

* 报错:加载资源失败



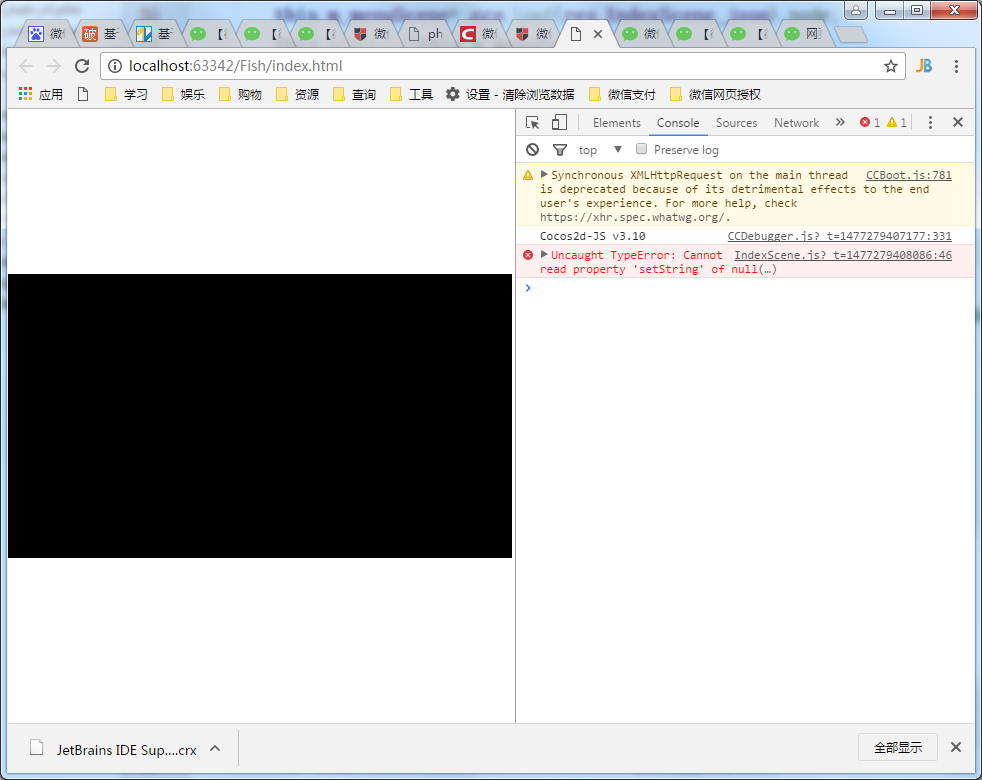
* + 环境:

加载资源文件名为: **FishOptionLayer#crazyFish#1#1.json**

* + 原因:

在JS中，“#”是字符串的特殊字符，自动发生截取。

* + 解决:使用别的字符 例如“\_”代替“#”。
* 报错:Uncaught TypeError:Cannot read propery ‘setString’ of null(…)



* + 环境:

1. 优化:使用Cocos Studio导出的资源，加载到游戏中，渲染批次的值 太大

思路:

获取暂时不显示的父节点，逐步遍历，隐藏子节点:

*//控制左边面板中某些节点的隐藏和显示*controlLeftPanel:**function**(bShow){  
 *//左边面板不显示* **for**(**var** i=0; i< **this**.**m\_leftPanel**.getChildrenCount(); ++i)  
 {  
 **var** child= **this**.**m\_leftPanel**.getChildren()[i];

//根据名字，特殊处理  
 **if**((**"Panel\_menu\_btn"**!= child.getName())&&  
 (**"Image\_89"**!= child.getName())){  
 child.setVisible(bShow== **undefined**?**false**:bShow);  
 }  
 }  
}

1. TexturePacket加密文件的使用:

\*.pvr.ccz加密文件的使用，和普通图片(\*.png、\*.jpg)的使用方法一致，唯一不同之处在于:

在Resources.js文件中，不必加载该文件，否则会报错。