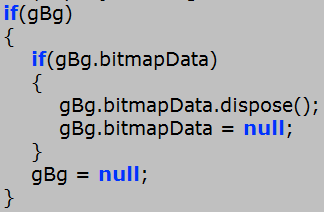
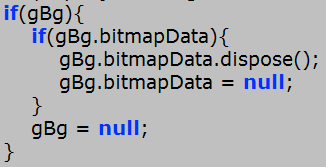
## ActionScript代码规范

**宗旨：代码简洁、易懂、BUG少**

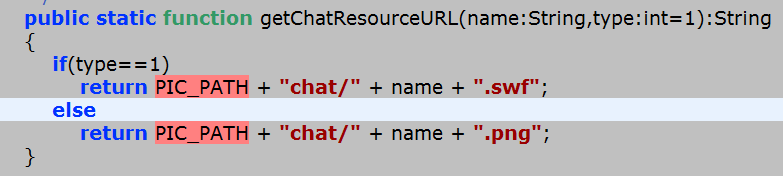
一．格式

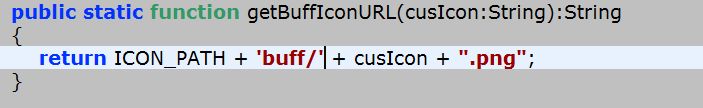
1. 缩进：用Tab缩进而不是用空格
2. 大括号使用：

统一使用齐头式：

不使用结尾式：

1. 引号使用：

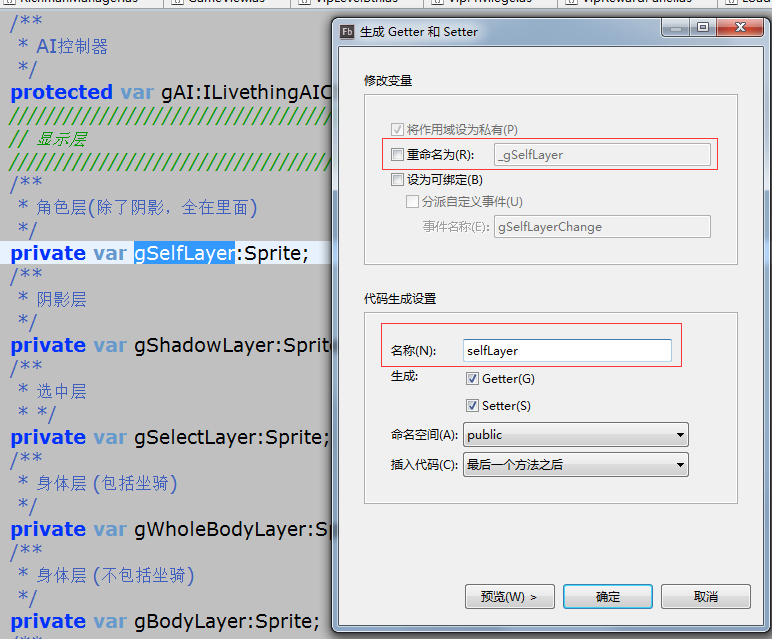
字符串用双引号

不要用单引号或单双混用

1. 命名
2. 私有全局变量命名：gXxxYyy



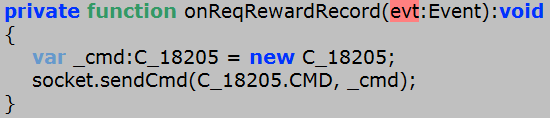
快速生成set/get方法时要改为xxxYyy：



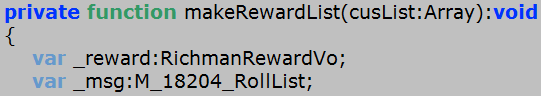
1. 公有全局变量命名：xxxYyy



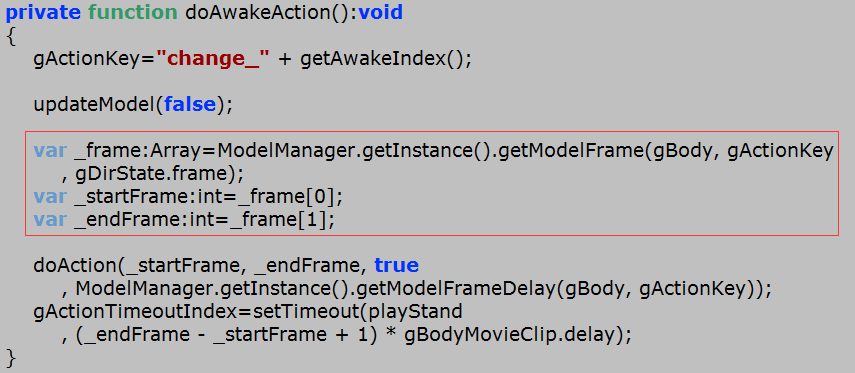
1. 参数变量命名：xxxYyy



或者cusXxxYyy



1. 局部变量命名：\_xxxYyy

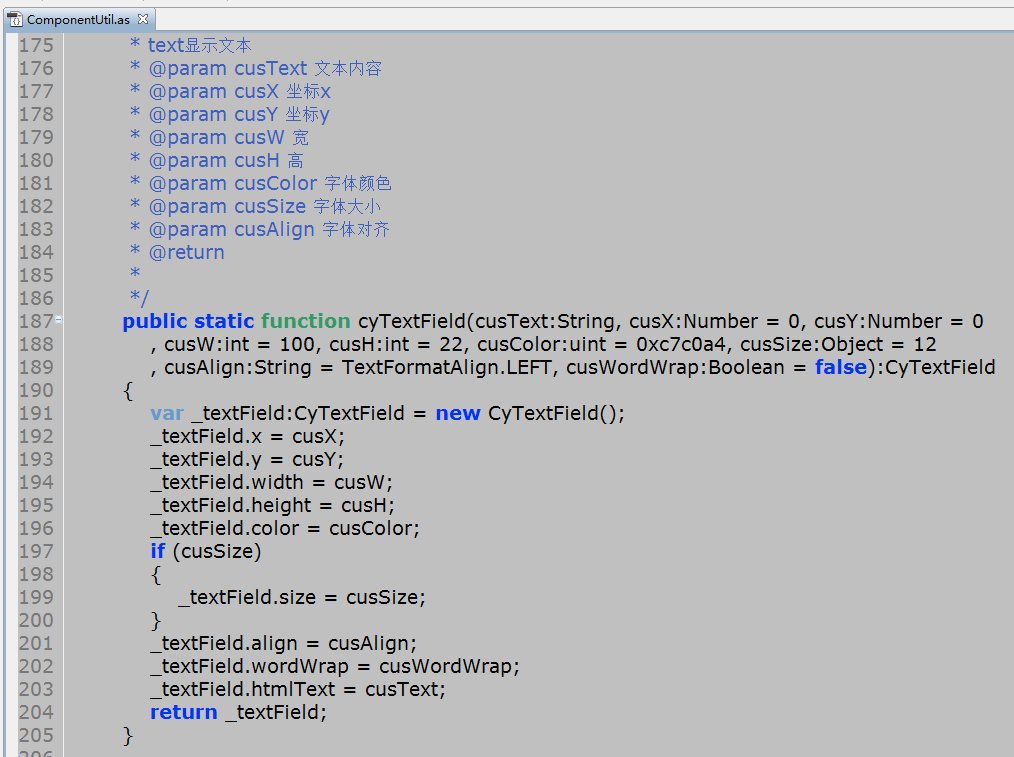


1. 命名尽量不要使用拼音，更不要使用难懂的简写，而是使用英文

如：红包数量不要使用var hongbaoNum:int;或var hbsl:int;

而应该var redPacketNum:int;

1. 函数调用
2. 生成ui控件类使用工厂创建：



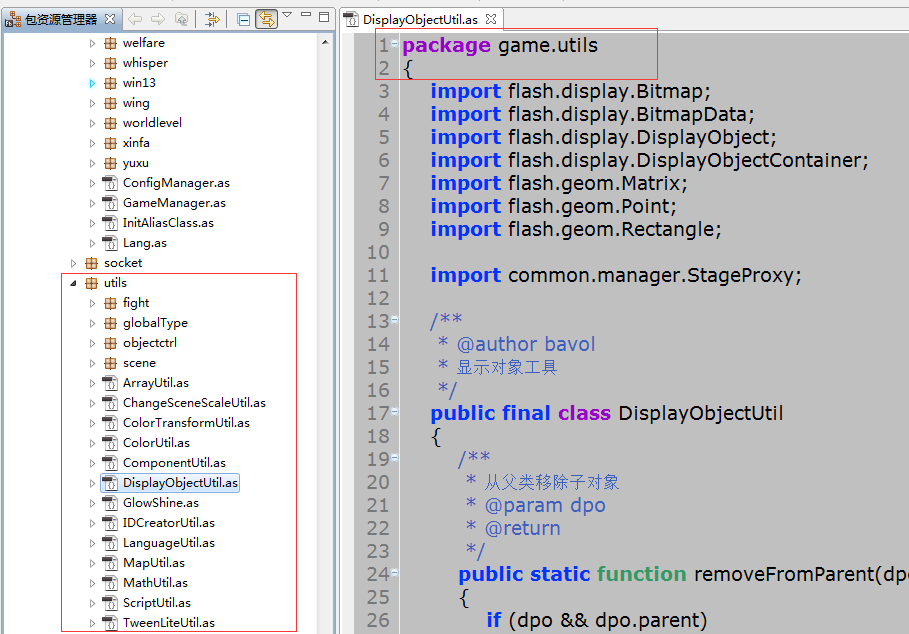
而不是直接new Button()

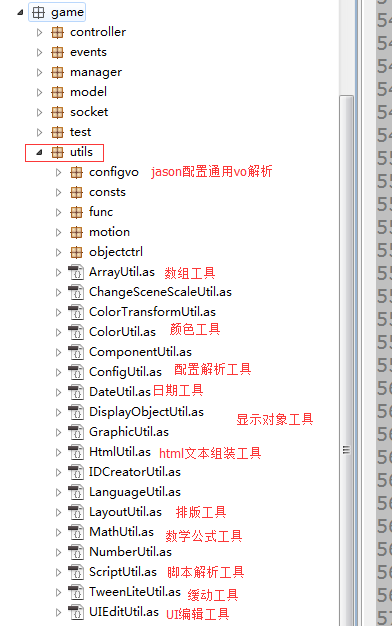
1. 经常使用的对象使用池创建：



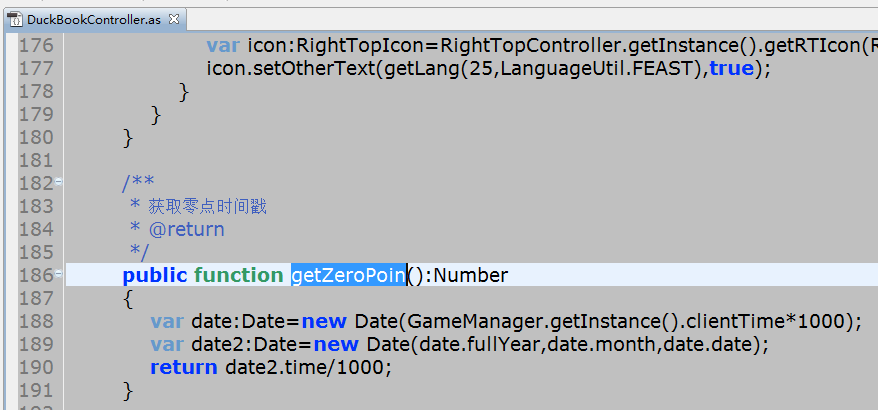
而不是直接new SimpleEffect

3.对时间格式化，文本加颜色，加下划线等使用工具类





而不是在各自类里面加各自的方法



4.所有 功能/系统/界面 都应该在UINameManager里面定义--例如：

定义界面 – UINameManager.Test

public static const Test:int = 90000;

12000 –前3位120表示<界面号>，后2位00表示<子界面号>

public static const Test1:int = 90001;

public static const Test2:int = 90002;

注册界面:(和以前一样，只注册00子界面就可以了)

registerData(UINameManager.Test, TestPop, null, “”, 774, 539, 1. true);

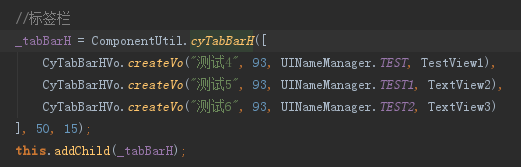
打开界面:

OpenView.open(UINameManager.Test); --打开测试界面，并切到Test分页

OpenView.open(UINameManager.Test1); --打开测试界面，并切到Test1分页

OpenView.open(UINameManager.Test2; --打开测试界面，并切到Test2分页

标签栏(注册子界面)



1. 游戏公共用法规范

---不要自己手动添加冗余代码，看代码注释说明或其他人用法

1.Alert弹窗

2.提示飘字

3.右下角弹窗

4.文本组装

5.右下角历史记录

6.带链接文本(鼠标移上去会偏移)

7.寻路，挂机等使用

8.飞鞋使用

9.上线请求协议

10.客户端语言显示（模块类禁止出现中文夹杂在代码中），见doc/data/language文件夹（convert命令：language）

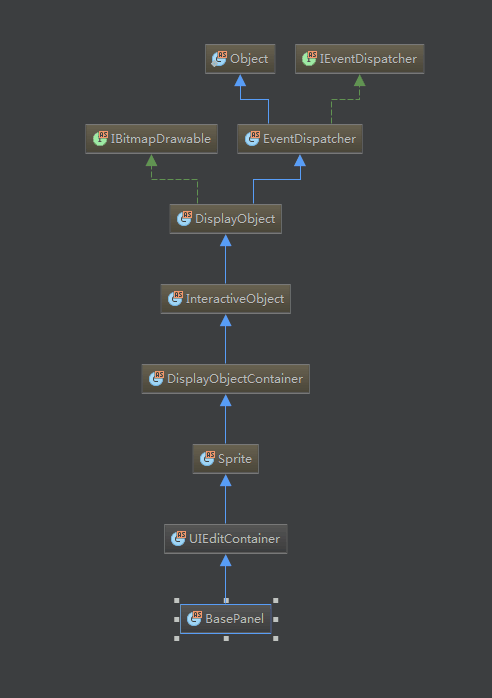
11.颜色统一使用ColorUtil中定义的常量，不要单独写颜色代码，如\_txt.color=0xff4e00应该写成\_txt.color=ColorUtil.RED;

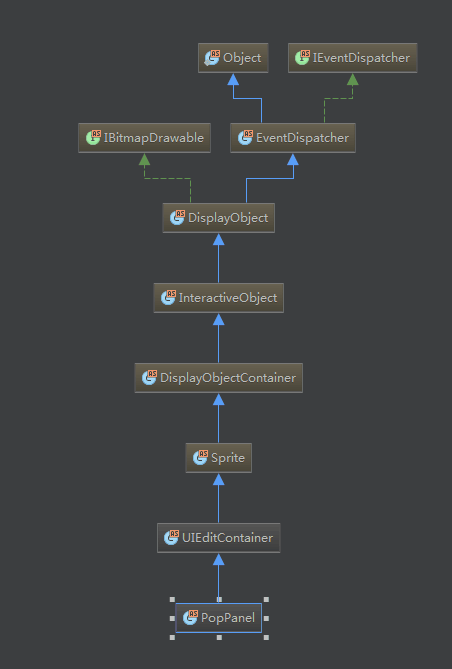
12.常用对象池ObjectManager和自定义对象池ObjectPoolManager的使用

1. UI编辑器使用规范

支持UI编辑功能的容器有3个：UIEditContainer，BasePanel，PopPanel

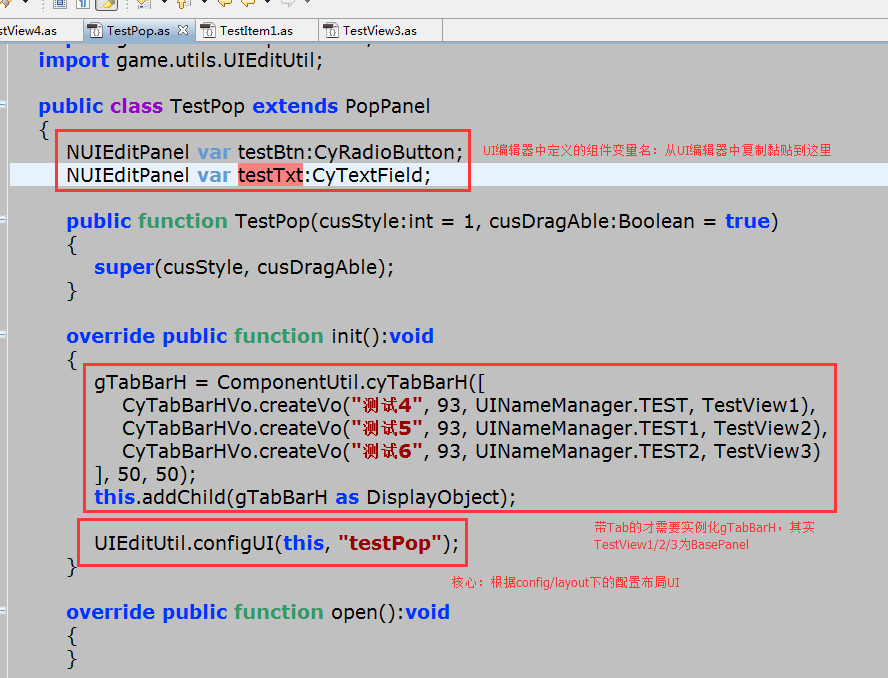
继承结构如下：

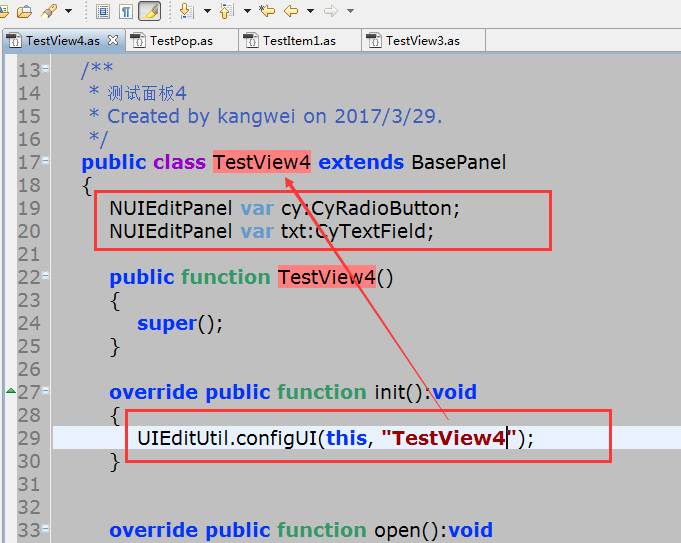




其中，PopPanel是窗体，BasePanel/BasePanelOnRes是窗体中各类面板的基类，UIEditContainer是最底层支持UI编辑功能的容器。

代码使用范例：





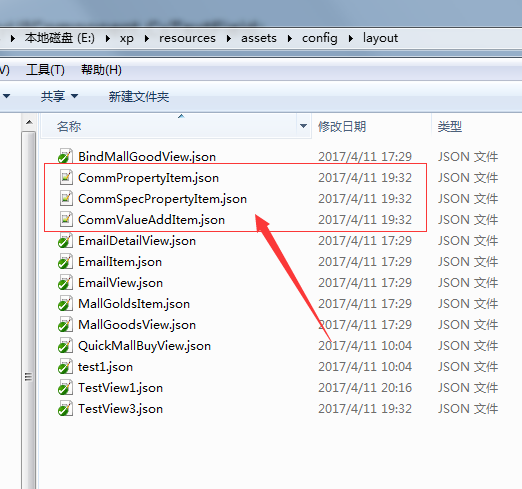
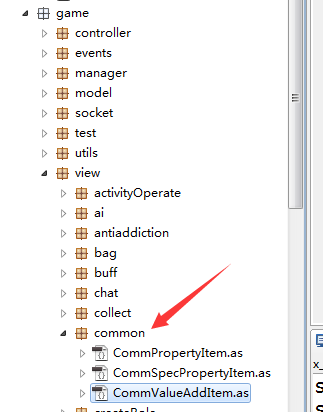


即把原先手写组件的过程变成一行代码即可。

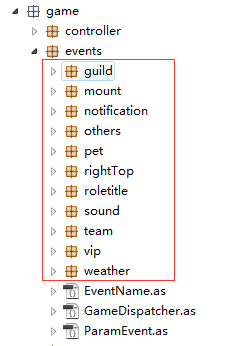
注：所有UI编辑的组件访问控制都定义为NUIEditPanel，由UI编辑器复制而来，不要手动修改！！

具体UI编辑器如何使用见“UI编辑器操作手册.doc”

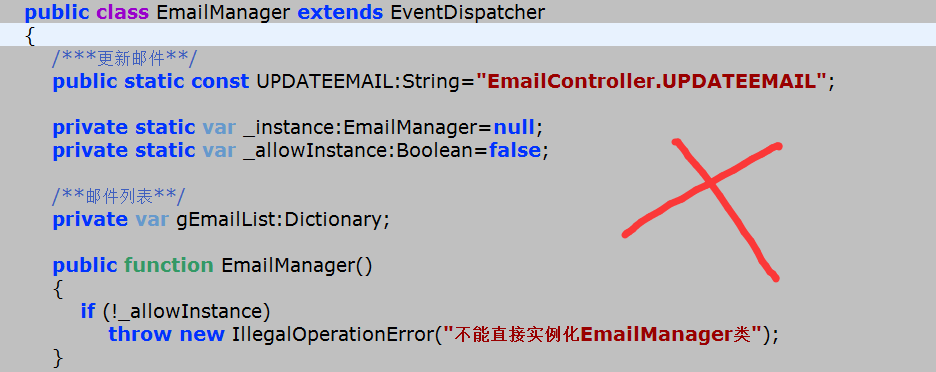
5.2.公用编辑组件

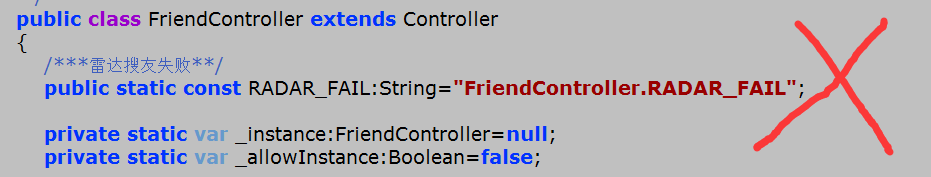
在game.view.common下存放公共编辑组件，对应config/layout下的布局配置，在进行界面编辑时可选用这些公共组件，避免冗余。

1. 事件侦听
2. 各模块在game.event包下添加事件文件

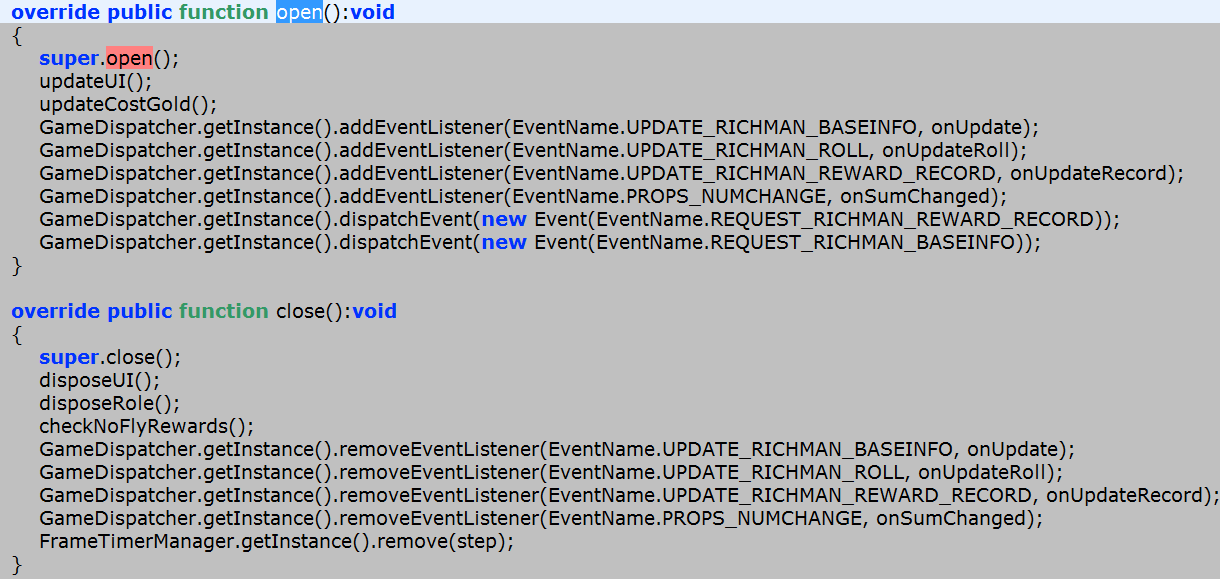


不要在Manager或Controller里加事件名常量





2.在需要的事件增加事件监听，在不需要继续使用的时候，务必移除对事件的监听



1. 清理界面注意

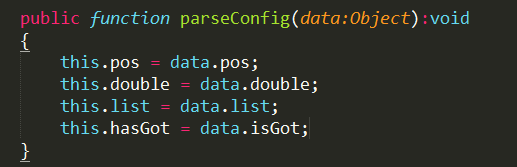
用到UI编辑器创建的CyPropsGrid在super.dispose之后需要设置为null，不然无法清理成功。

1. 配置解析

1.Json配置读进来后用new XxVo(data)解析，属性一律用public且要求加注释



也可以new XxVo()后用方法解析：



1. 解析工具game.utils.ConfigUtil

配置表常用的配置格式都可以在ConfigUtil中有方法解析

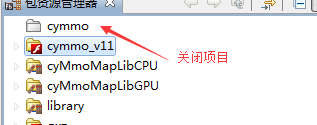


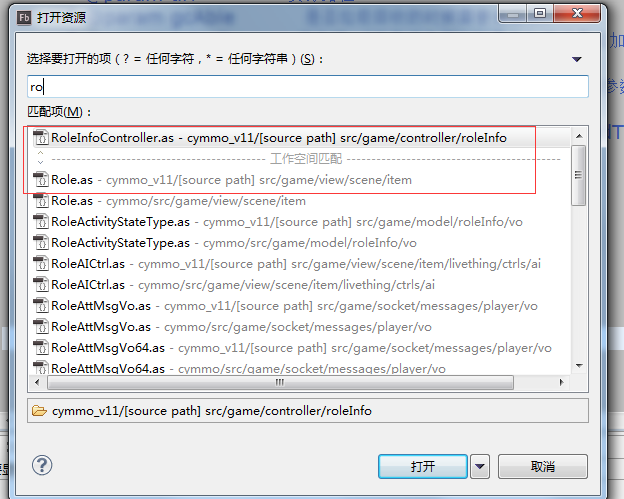
有其他新增的通用配置格式，也可在此添加。不要重复添加解析方法！！！

1. 重要类不要擅自随意修改

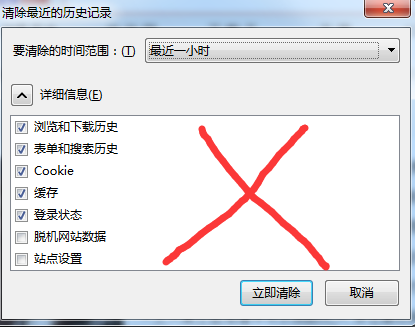
场景带Scene开头的类（如Scene.as等）和场景显示单位（如Livething.as等）不要擅自随意修改，修改前请跟海滨或守东商量，修改时遵守代码规范，修改后再跟海滨或守东说。

1. 编译
2. 考虑到更好的测试性能问题，统一用GPU版本编译



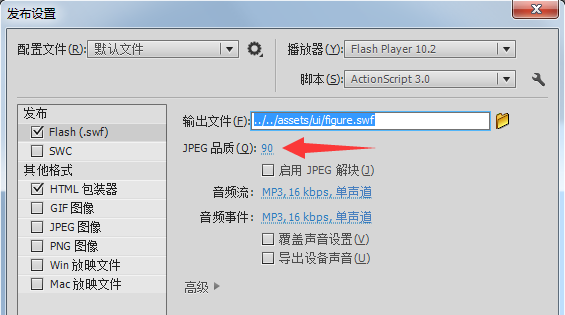


编辑时选中v11的as文件，按F11自动编译GPU版本

1. 除了地图资源外，私服编译后一律不需要清缓存。
2. 

七、fla资源规范：默认80%压缩失真太严重，采用90%压缩，公共资源如common.Bg\_1用无损格式；【ps切png图用“存储为web所用格式”png24选项（**不要用“存储为”保存NPG**）】

已经创建的fla文件都需要重新修改发布设置！

Fla文件发布设置为90%压缩，如图：  


八、切图命名规范：（海外版翻译方便查找中文资源）

带中文资源的切图命名统一加前缀(translate首二字母)：tr\_XXX，如tr\_图标背景；