# FLASH TEAM 代码规范白皮书 version 1.2

**为了增加开发效率，我们需要一套比较严格的代码规范，一直想写这样一套规范。**

**该规范必须遵循以下机制：**

1. 项目如果为新项目，我们必须严格按照规范中的要求。
2. 项目如果是移植项目，在不改变原来代码的基础上，我们要求重构的类和增加修改的代码符合该规范要求。
3. 大家相互监督，如果发现项目中（移植项目源代码除外）有违该代码规范地地方，应及时指出，并立即让相关责任人做出整改。
4. 项目中，我们不会像research项目对代码管理那么不重视，各位要严格要求自己，这样可以帮助我们更有效率地开发和日后的debug。

**以下为规范细则：**

1. 属性的规范：
2. 在类里面定义变量或属性统一在变量前加“\_”如：private var \_id:int;
3. 方法的参数统一增加“$”

如：public function BuildingItem3D($id:int,$i:int,$j:int,$isShadow:Boolean=false)

1. 常量用大写 如：private const USELESS\_FUNCTION\_INFO:String = "0";
2. Static属性前加s\_
3. 成员属性前加m\_

2． 类名的规范：

1）两个单词的首字母统一大写 如：Class BuildingItem3D

2）如果该类为新定义的，原来项目不存在的，需要增加作者信息，日期用/\*\*/注释（最好能附上简单的类描述）

如：

/\*\*

\*

\*author :Jeff

\*Date:2012.7.2

\*/

Public Class Test

1. 方法名的规范：
2. 第一个单词首字母小写，后面几个单词的首字母大写

如：private function initializeGraphicsAPI():void

1. 如果该方法在原来类中不存在，需要增加作者信息，及简单的方法描述

用//注释

如：

//Make this Starling instance the <code>current</code> one.

public function makeCurrent():void

{

sCurrent = this;

}

1. 注意方法{}位置对应，保证代码可读性

4． 其他：

1）for 语句定义规范（在for语句外定义length）

Var i:uint=0;

Var \_len:uint=Array.length；

for(;i<\_len;i++){

}

这样提高效率。

2）能用public 方法去调用执行的，尽量用public方法，尽量少用侦听器写自定义方法。

3）不用的对象勤置null，不用的侦听勤remove，在线程比较空闲的时候或者重新开始游戏的时候GC()，多用AIR3 新提供的removechildren（）等方法。

1. 关于Review：
2. 每次上传一个版本，需要review，review的责任人为该类的作者，或者leader。
3. Review 需要注意有没有removelistener，有没有硬代码，是否符合代码规范，及上传的版本类中的代码会不会违背该类的作用等。
4. Commit规范：需要注明reviewer及日期。
5. 如果由于review做的不到位，产生的硬代码，bug或者内存泄漏等问题，需要reviewer和相关程序员共同解决，如果会引起项目的拖延等风险，需要程序员和reviewer优先解决review不到位的问题，并按时按量完成task。
6. 每周进行一次周review，由leader来关注项目是否有明显内存泄漏，及明显bug，并督促相关程序员修改完善，如果会引起项目的拖延等风险，需要程序员和leader优先解决review不到位的问题，并按时按量完成task。

张弛

2013.3.1