# 客户端LUA规范

## 一,基本准则

### 1,添加注释

1. 添加可公开调用方法的注释
2. 类型文件名在头部或者类型简要描述处要进行足够的说明

### 2,代码文件格式

根据使用的插件而定，如无说明格式应为UTF8无BOM

### 3,require时使用斜杠

像这样: require(“xx/yy/zz”)，而不要像这样: require(“xx.yy.zz”)

## 二,优化部分

### 1, 严禁滥用匿名方法或者闭合

概述：能够使用对象方法或者静态方法时就一定要使用，如果需要捕捉额外的参数

可以自己定义一个中间类型进行回调传递,尤其是频繁调用的方法

错误方式：

You:Func(arg1,function(x)

self.youVar = x

end)

正确方式：

function You:CallbackFunc(x)

self.youVal = x

end

You:Func(arg1,You.CallbackFunc,You)

### 2, 合理使用局部变量让程序更快

概述：对于要反复调用的方法或者字段，应该在局部作用域内定义为局部变量

不合理方式：

function You:Func()

self.var1 = You.Func1(self.arg1.arg, other1)

self.var2 = You.Func1(self.arg1.arg, other2)

self.var3 = You.Func1(self.arg1.arg, other3)

self.var4 = You.Func1(self.arg1.arg, other4)

end

正确方式：

function You:Func()

local tf = You.Func1

local arg = self.arg1.arg

self.var1 = tf(sarg, other1)

self.var2 = tf(arg, other2)

self.var3 = tf(arg, other3)

self.var4 = tf(arg, other4)

end

### 3，使用对象池

概述：绝不允许出现频繁创建/销毁的情况

1. ，对于游戏对象既可以使用全局游戏对象池GbjPool,也可以根据情况在自己的结构类型中添加局部对象池
2. 自定义类型可以使用ObjPool，使用此对象池时，添加的lua类型必须具有如下字段
3. Name：字符串类型，用以确定类型的名称
4. New()：方法，用以创建此类型的实例
5. ，也不要出现如下情况

you.tbl = {}

在某一时刻直接：you.tbl = nil

这个时候应该构建自定义you.tbl的类型，通过对象池反复使用

### 4，table当数组使用时，移除尾部元素更快

对于列表这种数据结构，全世界支持面向oop的语言，不出意外都是在尾部移除最快

### 5，不要频繁的使用不可变字符串进行拼接

概述：lua中可以使用StrBuffer进行字符串拼接