

# 武炼巅峰客户端编程指南

---

## 一、软件操作

为统一软件使用习惯，解决开发过程中出现的一些不必要的痛点，现整理以下几点要求，希望各位记得和做到。

### 1. SmartGit

- `commit` 必须注明详细的有效信息，如：新增，删除，修改（修复），注释某功能（方法）；避免使用意义不明确的信息，如：1，修改，删除。
- 优先使用 `pull --rebase`，可以有效降低代码冲突。
- 如遇冲突，优先本地解决后再上传；如无法自行解决，请与冲突方联系商议解决；或保存本地代码后撤销更改，再拉取远端代码，最后覆盖本地代码回去。

### 2. CocoStudio

- 资源和`csd`文件必须上传到`cocostudio`目录，确保每个开发人员的资源是一致的（至少保证没有资源残缺）。
- 工程文件可以不上传，如上传，请确保是完整的工程包。

### 3. TexturePacker

- `plist`请使用公用机的`TexturePacker`打包，避免因为不同机子使用的软件版本不同而造成打包出来的文件可能不一致的问题。

### 4. TiledMap

- `TiledMap`地图文件交给策划人员去修改，客户端人员尽量不要修改。
- 如需要修改地图层面的设计，请先知会相关技术人员或负责人，协定方案后再进行。

---

## 二、代码风格

由于各人有各人的编程习惯，前期工程又没有做好该方面的规划，因此造成目前工程中的代码风格各异，给技术人员在阅读他人代码时制造不少阅读难度或差异化，因此之后请各位尽量统一风格。现总结以下必须遵从的重要几点，其他可选的可参见**doc/2.代码风格指南.md**。

1. 顶部文件注释，需注明该文件的职能简要，创建人以及备注信息，如：

```
-----  
-- 说明：热更新工具  
-- 作者：jiangliang.zqgame  
-- 备注：仅用于热更新，请勿添加其他无关方法  
-----
```

2. 缩进，留白与对齐

- 缩进

- 设置**TAB**为**4**个空格
- 严格对齐左端缩进，不要严重影响阅读体验，如：
  - 错误

```
local a = 1  
  local b = 22  
local c = 333  
local d = {  
  k1 = 0,  
  k2 = 1,  
  k3 = {  
    kk1 = 0,  
    kk2 = 0,
```

```
    },  
}
```

■ 正确

```
local a = 1  
local b = 22  
local c = 333  
local d = {  
    k1 = 0,  
    k2 = 1,  
    k3 = {  
        kk1 = 0,  
        kk2 = 0,  
    },  
}
```

○ 留白

■ 符号之间留白，如：

```
local right = a + b -- 推荐  
local wrong = a+b   -- 不推荐
```

■ 方法与方法之间留白，如：

```
function A()  
    -- TODO A...  
end  
  
function B()  
    -- TODO B...  
end
```

- 方法块内有明显分块的留白，如：

```
function A()  
    local a = 0  
    local b = 2  
    -- 此行为留白  
    local function assistFunc1(...)  
        -- TODO assistFunc1  
    end  
    --此行为留白  
    local function assistFunc2(...)  
        -- TODO assistFunc2  
    end  
    -- 此行为留白  
    assistFunc1(a)  
    assistFunc2(a, b)  
end
```

- 对齐

- 按左端对齐，除此之外，其他部分不严格要求对齐，看个人喜好，如下两种方式都是可以的：

```
local varA = 1  
local varAA = 2  
local varAAA = 3
```

```
local varA    = 1  
local varAA   = 2  
local varAAA  = 3
```

### 3. 注释

- 顶部文件注释，见第1点。
- 方法注释，重要方法或者公用方法请务必填写注释，注释需注明方法职能、参数信息及意义以及创建人，可选附带示例。如：

```
-----  
-- 功能：获取装备战斗力  
-- 参数：  
--   eid - 装备id  
-- 示例：  
--   local power = getEquipFightPower(10011)  
-- 作者：jiangliang.zqgame  
--  
function Common:getEquipFightPower(eid)  
    -- TODO getEquipFightPower  
end
```

- 重要标记注释，如有待办，待修复，警告，有错误，废弃的代码，请使用以下这些标签作为标记：

```
-- TODO 待办说明  
local function todoCase()  
end  
  
-- FIXME 待修复说明  
local function fixmeCase()  
end  
  
-- NOTE 警告说明  
local function noteCase()  
end  
  
-- ERROR 错误说明，可能发生的错误
```

```
local function errorCase()  
end  
  
-- DEPRECATED 废弃说明，注明替代方案或替代接口  
local function deprecatedCase()  
end
```

---

## 三、代码质量

虽然项目没有规定有代码审查这个步骤，但良好的代码应该至少具备可阅读性和可维护性（暂且不论健壮性），也为了方便他人接手或修改你的代码，因此很有必要在编程的过程中，要求个人提高对项目的全局观意识，和保持优良代码的设计与习惯，切勿局限于功能的快速实现，而忽略了代码的质量。

保持代码质量和优良设计有几点好处。一，虽然功能并不一定按预期运行，但至少能够看清个人的思路和表达，为预期问题的所在提供极大便利；二，代码易于维护和扩展，出了问题，他人也能快速理解并进行修改，对后续的扩展也比较简单；三来，养成编写良好代码的习惯，对个人今后的成长来说颇有裨益。

最后推荐几本书目，希望大家抽空翻阅，激励共勉。

- [《重构-改善既有代码的设计》](#) 力荐
- [《编写可读代码的艺术》](#) 力荐
- [《程序员的职业素养》](#) 力荐
- [《重构与模式》](#) 推荐
- [《写给大家看的面向对象编程书》](#) 可选
- [《代码整洁之道》](#) 可选