

代码复审核查表

Team:

Larghetto

Member:

- 2106050218_徐诚鸿(Euphocc)
- 2106050209_王雨乐(王不会)
- 2106050223_杨祁元(没嘴见人的朽木君)
- 2106050203_余成远(Boléro)
- 2106050101_徐欣怡(五亿个小铃铛)

1. 概要部分

(1) 代码能符合需求和规格说明么？

符合需求与规格说明。

(2) 代码设计是否有周全的考虑？

设计考虑周全,设计文档经过反复审核与修改,符合OOP现代程序设计方法

(3) 代码可读性如何？

代码具有较强可读性。

(4) 代码容易维护么？

容易维护，各个功能分模块划分，每个模块低耦合高内聚，模块合并时功能正常,目前尚未发现问题；各部分代码都符合规范,后期易维护。

(5) 代码的每一行都执行并检查过了吗？

对于可读的代码，我们进行了详细审核与测试；
事实证明，程序是没有明显问题的，但是一些潜在的问题，未被发现的问题，我们可能并未发现。

2. 设计规范部分

(1) 设计是否遵从已知的设计模式或项目中常用的模式？

设计遵从已知的设计模式或项目中常用的模式：自底向上、面向对象。

(2) 有没有硬编码或字符串/数字等存在？

有硬编码或字符串/数字等存在。

(3) 代码有没有依赖于某一平台，是否会影响将来的移植？

代码没有依赖于某一平台，不会影响将来移植。

(4) 开发者新写的代码能否用已有的Library/SDK/Framework中的功能实现？在本项目中是否存在类似的功能可以调用而不用全部重新实现？

新写的代码使用已有的Library/SDK/Framework中的功能实现，在本项目中不存在类似的功能可以调用而不用全部重新实现。

(5) 有没有无用的代码可以清除？（很多人想保留尽可能多的代码，因为以后可能会用上，这样导致程序文件中有很多注释掉的代码，这些代码都可以删除，因为源代码控制已经保存了原来的老代码。）

有，但大部分代码都已经清除，仅留下少部分注释。

3. 代码规范部分

(1) 修改的部分符合代码标准和风格么（详细条文略）？

代码风格规范：简明，易读，无二义性。在一定程度上，修改的部分符合。

4. 具体代码部分

(1) 有没有对错误进行处理？对于调用的外部函数，是否检查了返回值或处理了异常？

我们对错误进行了处理，检查了返回值和异常处理。

(2) 参数传递有无错误，字符串的长度是字节的长度还是字符（可能是单/双字节）的长度，是以0开始计数还是以1开始计数？

参数没有错误，字符串的长度是字符长度，是以0开始计数。

(3) 边界条件是如何处理的？Switch语句的Default是如何处理的？循环有没有可能出现死循环？

边界条件采取特判形式进行处理；不含Switch语句；没有出现死循环。

(4) 有没有使用断言（Assert）来保证我们认为不变的条件真的满足？

没有使用断言（Assert）来保证我们认为不变的条件真的满足。

(5) 对资源的利用，是在哪里申请，在哪里释放的？有没有可能导致资源泄露（内存、文件、各种GUI资源、数据库访问的连接，等等）？有没有可能优化？

JavaScript与现代浏览器自带GC(Garbage Collection)机制进行回收与释放,不会导致资源泄漏
可能在部分循环层面（如碰撞检测标记）进行优化。

(6) 数据结构中是否有无用的元素？

数据结构中没有无用的元素。

5. 效能

(1) 代码的效能（Performance）如何？最坏的情况是怎样的？

代码可以直接完成功能，效能良好；在极端情况下(如同时有上千上万人访问页面)，不影响主程序执行的模块（数据库模块）会缓慢甚至崩溃；

(2) 代码中，特别是循环中是否有明显可优化的部分（C++中反复创建类，Java中 string 的操作是否能用StringBuilder 来优化）？

存在可优化的部分,但不明显。

(3) 对于系统和网络调用是否会超时？如何处理？

在极端情况下可能超时,超时时,重新开始进行程序执行.

6. 可读性

代码可读性如何? 有没有足够的注释?

代码具有一定的可读性, 有足够的注释.

7. 可测试性

代码是否需要更新或创建新的单元测试?

代码存在可进一步优化的地方(如按钮不够美观)
若使用增量模型对代码进行进一步优化升级,则增加部分应当进行单元测试.