1．概要部分

（1）代码能符合需求和规格说明么？

说明代码基本上没有出现多少错误，但是不代表代码就是完全正确无误的，其中还有其他条件的限制，包括执行后是否能够达到目的。

（2）代码设计是否有周全的考虑？

代码的设计是比较周全的，但不是绝对的，我们的能力也许有限，并不能考虑到所有的方面，若有一定的疏漏还需关注。

（3）代码可读性如何？

代码具有一定的可读性。

（4）代码容易维护么？

由于在程序中存在未知漏洞，由于使用产生的积累问题，为了补充新功能而对程序的代码进行修改维护。我们还未做到这点，我们没发现程序中的未知漏洞。

（5）代码的每一行都执行并检查过了吗？

对于可读的代码，我们进行了一步步的检查；除此之外，我们通过执行程序，来检查代码中是否还存在问题。事实证明，程序是没有问题的，但是一些潜在的问题，未被发现的问题，我们可能并没有发现，所以这类问题，没有进行检查。

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。

（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？

游戏基于windows开发，但不影响移植，本游戏基于unity引擎开发，unity有成熟的跨平台移植管理办法，可移植到多平台。

3．代码规范部分

（1）修改的部分符合代码标准和风格么）?

代码风格规范：简明，易读，无二义性。在一定基础上，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？

没有边界条件。没有Switch语句的Default，循环没有出现死循环。

（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。

（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，所以说效能还是不错的，但是速度的快，之类的，我们真达不到这种水平，这样的程序，也应该不必要效能测试，最坏的情况无不是代码无法完成需求，出现错误，速度慢。

（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，C#中 string 的操作是否能用StringBuilder 来优化）？

代码中的循环并不多，所以没必要再进行所谓的优化。也没有明显的优化部分。

（3）对于系统和网络调用是否会超时？如何处理?

没有网络的调用

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。