1. 使用C++等其他语言作为底层逻辑框架，使用类似LUA语言作为功能实现逻辑表述。
2. 脚本亦有入口，使用入口加载所有的脚本文件。
3. 调用机制？该底层框架是如何调用这些LUA的实现函数的？
4. LUA函数又是怎么不像C++那样，需要添加头文件即可调用底层逻辑框架。
5. 我们这游戏的消息处理方式是怎样的？就是客户端发送消息过来后，底层是怎么处理的，然后再怎么调用脚本？

学习记录：

1. 在分析功能的时候，注意一点是最终数据一定有读入和读出的接口或者地方，那么这个地方是在哪里呢？找到读入读出的地方和方式，即可以根据读取的脉络找到功能的使用方法。
2. 功能的实现寻找方法有两种，一是根据客户端的调用信息，查找服务端接口，根据接口的调用脉络找到整个操作方法。二是根据服务端的调试输出文件，逆向寻找到调用的最初函数（对于脚本调试而言，逆向寻找是最困难的，因为一个函数可能被上百个其他函数调用。要分辨和初始调试），正向找到最后的实现函数。
3. 使用结果分析法，根据结果快速清楚某一函数是否有用。
4. 一般使用到数据库，都是在服务器建立时创建数据库，然后建立各个数据表，再就是根据对各个表格进行初始化；应该留有一个在后面服务器已经启动时，可以创建新表和为新表添加列的接口。
5. 在定义新属性时注意写入数据库的类型；如果类型不同，数据可能无法写入。可以查看错误提示
6. 要注意在lua中尾调用和闭包的区别；闭包是在函数内定义新的完整函数（可以是匿名函数），该函数也可以放置在return 上，作为返回。如果

function f(x)

return function (x)

return x+1

end

end

c1=f(x)

则c1为指向匿名函数的指针。

如果是

function a()

content

return x

end

function b()

content

return a()

end

c1= b()

这种结构则是属于尾调用,c1等于a()计算的最终结果。