项目制作之随笔

1、Lua的Debug调试输出





使用xpcall可以解决这个问题, 比pcall多了一个参数。使用debug.traceback可以将traceback的信息存储到msg变量。

xpcall 接受两个参数：调用函数、错误处理函数。当错误发生时，Lua会在栈释放以前调用错误处理函数，因此可以使用debug库收集错误相关信息。

常用的debug处理函数：debug.debug和debug.traceback，前者给出Lua的提示符，你可以自己动手察看错误发生时的情况；后者通过traceback创建更多的错误信息，也是控制台解释器用来构建错误信息的函数。

你可以在任何时候调用debug.traceback获取当前运行的traceback信息。

cocos.init文件根据配置信息初始化Cocos2d-lua框架（require加载lua模块时必须使用”.”来代替路径中的”/”符号）

根据项目需求，可以在cocos.init文件中注释掉不使用的模块，比如physics3d、spine、cocosbuilder等。

2、日志颜色区别输出

引入LoggerHelper.h 文件，并放于Win-Class中，在AppDelegate.cpp中引用





之后，便可在lua文件中引用logger\_info("logger\_info")、logger\_debug("logger\_debug")、logger\_warning("logger\_warning")、logger\_error("logger\_error")、logger\_fatal("logger\_fatal")等带有颜色的log输出。

由于只在Win平台下引入了颜色日志，故在Lua中需要对log进行封装，使得在不同平台调用不同方法。

3、调试的debug控制台



4、cocos2d-lua 3.5以后的全局变量声明



Cocos2d-x 3.5的lua项目生成后，变成了MVC模式，并且，加入了一个全局变量的检测功能。也就是说，你不小心用了全局变量，他会提示你出错！

所以，C2d加入了一个全局命名空间的检测，通过setmetatable的方式。该函数在src/[**cocos**](http://lib.csdn.net/base/cocos)/framework/init.lua



那么如果我一定要用到全局变量才行呢？这个问题有若干种解决办法

1、最简单粗暴的就是直接注释掉着一块。

2、将CC\_DISABLE\_GLOBAL设置为False

3、在mian.lua的  require "cocos.init"之前 require 所需的库和全局的变量

但是如果是开发者自己需要使用全局变量。而不是其他库需要用到全局变量的话！请使用使用官方推荐的方法cc.exports.\*  来充当全局变量！

5、

6、cocos2d-Lua继承关系



如果子类GameLayer重新定义了派生类，则执行结果如左。如果子类GameLayer没有重定义派生类，执行结果如右。注意，如果直接调用.new()或:new()方法则会直接调用ctor()而不再调用create()。

