

CSDN

[首页](#)[博客](#)[专栏课程](#)[下载](#)[问答](#)[社区](#)[插件](#)[认证](#)

class

Q

[登录/注册](#)[会员中心](#)[收藏](#)

toLua:简洁的使用说明

原创

hsz8250

2006-05-25 17:41:00

10000

收藏

版权

分类专栏：

脚本语言

 文章标签：

lua


function

class

null

api

types

 **脚本语言** 专栏收录该内容

0 订阅

7 篇文章

订阅专栏

tolua的最新版本是5.0,下载地址:http://www.tecgraf.puc-rio.br/~celes/tolua/
以下是简单的使用说明:

功能内容: 可以在LUA脚本中使用C++的方便对象,包含创建, 释放, 调用成员函数
文件包括:Main.cpp,tClass.cpp,tClass.h,tClass.pkg,tClassLua.cpp(自动生成)

tClass.h(定义提供给LUA使用的类)

```
#ifndef TCLASS_H
#define TCLASS_H
```

```
// 简单的类
class A
{
public:
    int a;
    A( int nTemp );
    int Get();
};

#endif
```

tClass.cpp(类的简单实现)

```
#include "tClass.h"

A::A( int nTemp )
{
    a = nTemp;
}

int A::Get()
{
    return a;
}
```

tClass.pkg (重要的一环)
tolua会根据tClass.pkg来自动生产tClassLua.cpp,tClass.pkg是手动编写的,而已格式也有规定

,以下是tClass.pkg内容:
\$#include "tClass.h"

```
class A
{
    int a;
    A( int nTemp );
    int Get();
};
```

.pkg的书写格式与C++的格式差不多, 是只没有大部分的关键字,例如:public等,注意,需要包函

tClass.h的头文件,格式是\$#include "tClass.h",

举报

不存在,所以生成代码的时候可能会出现权限访问错误,例如在int a是私有的情况,可能生产的代

码会出问题。所以设计类的时候需要注意.

生成代码:在命令行中输入:tolua -o tClassLua.cpp,tClass.pkg

生成的代码

```
/*
** Lua binding: tClass
** Generated automatically by tolua 5.0a on 05/25/06 17:00:17.
*/

#ifdef __cplusplus
#include "stdlib.h"
#endif
#include "string.h"

#include "tolua.h"

/* Exported function */
TOLUA_API int tolua_tClass_open (lua_State* tolua_S);

#include "tClass.h"

/* function to register type */
static void tolua_reg_types (lua_State* tolua_S)
{
    tolua_usertype(tolua_S,"A");
}

/* get function: a of class  A */
static int tolua_get_A_A_a(lua_State* tolua_S)
{
    A* self = (A*) tolua_tousertype(tolua_S,1,0);
#ifdef TOLUA_RELEASE
    if (!self) tolua_error(tolua_S,"invalid 'self' in accessing variable 'a'",NULL);
#endif
    tolua_pushnumber(tolua_S,(lua_Number)self->a);
    return 1;
}

/* set function: a of class  A */
static int tolua_set_A_A_a(lua_State* tolua_S)
{
    A* self = (A*) tolua_tousertype(tolua_S,1,0);
#ifdef TOLUA_RELEASE
    tolua_Error tolua_err;
    if (!self) tolua_error(tolua_S,"invalid 'self' in accessing variable 'a'",NULL);
    if (!tolua_isnumber(tolua_S,2,0,&tolua_err))
        tolua_error(tolua_S,"#vinvalid type in variable assignment.",&tolua_err);
#endif
    self->a = ((int) tolua_tonumber(tolua_S,2,0));
    return 0;
}

/* method: new of class  A */
static int tolua_tClass_A_new00(lua_State* tolua_S)
{
#ifdef TOLUA_RELEASE
    tolua_Error tolua_err;
    if (
        !tolua_isusertable(tolua_S,1,"A",0,&tolua_err) ||
        !tolua_isnumber(tolua_S,2,0,&tolua_err) ||
        !tolua_isnoobj(tolua_S,3,&tolua_err)
    )
        return -1;
#endif
    int argc = tolua_getparamcount(tolua_S,1,&tolua_err);
    tolua_new(tolua_S,argc,tClass_A);
    return 1;
}
```

```
)
goto tolua_error;
else
#endif
{
    int nTemp = ((int) tolua_tonumber(tolua_S,2,0));
    {
        A* tolua_ret = (A*) new A(nTemp);
        tolua_pushusertype(tolua_S,(void*)tolua_ret,"A");
    }
}
return 1;
#ifdef TOLUA_RELEASE
tolua_error:
tolua_error(tolua_S,"#error in function 'new'.",&tolua_err);
return 0;
#endif
}

/* method: Get of class A */
static int tolua_tClass_A_Get00(lua_State* tolua_S)
{
#ifdef TOLUA_RELEASE
    tolua_Error tolua_err;
    if (
        !tolua_isusertype(tolua_S,1,"A",0,&tolua_err) ||
        !tolua_isnoobj(tolua_S,2,&tolua_err)
    )
        goto tolua_error;
    else
#endif
    {
        A* self = (A*) tolua_tousertype(tolua_S,1,0);
#ifdef TOLUA_RELEASE
        if (!self) tolua_error(tolua_S,"invalid 'self' in function 'Get'",NULL);
#endif
        {
            int tolua_ret = (int) self->Get();
            tolua_pushnumber(tolua_S,(lua_Number)tolua_ret);
        }
    }
    return 1;
#ifdef TOLUA_RELEASE
    tolua_error:
    tolua_error(tolua_S,"#error in function 'Get'.",&tolua_err);
    return 0;
#endif
}

/* Open function */
TOLUA_API int tolua_tClass_open (lua_State* tolua_S)
{
    tolua_open(tolua_S);
    tolua_reg_types(tolua_S);
    tolua_module(tolua_S,NULL,0);
    tolua_beginmodule(tolua_S,NULL);
    tolua_cclass(tolua_S,"A","A","",NULL);
    tolua_beginmodule(tolua_S,"A");
    tolua_variable(tolua_S,"a",tolua_get_A_A_a,tolua_set_A_A_a);
    tolua_function(tolua_S,"new",tolua_tClass_A_new00);
}
```

```
tolua_function(tolua_S,"Get",tolua_tClass_A_Get00);

tolua_endmodule(tolua_S);

tolua_endmodule(tolua_S);

return 1;

}
```

如果生成成功就没有任何的提示.现在可以把tClassLua.cpp加到我们的工程中;
到最后的使用过程
注意:由于生成的过程中, 根据包的文件生成的一个函数,这里我们需要在Main中定义一下!
TOLUA_API int tolua_tClass_open (lua_State* tolua_S);
然后需要调用它, 让它把生成的代码内嵌到LUA中.

```
Main.cpp
extern "C" {
#include "lua.h"
#include "lualib.h"
#include "lauxlib.h"
}

#include "tolua.h"
#include "tClass.h"
#include <string>

TOLUA_API int tolua_tClass_open (lua_State* tolua_S);

int main ()
{
    lua_State* L = lua_open();
    luaopen_base(L);

    tolua_tClass_open( L );

    // 调用脚本过程
    // 在脚本中就可以随意的使用这个类型
    std::string str = "obj = A:new(2); print(obj:Get() )";

    lua_dostring( L,str.c_str() );

    lua_close(L);

    return 0;
}
```

tolua还有比较多的功能, 都是能够将C/C++的类型,函数,结构等等, 方便的内嵌入LUA中调用!

打赏

文章很值，打赏犒劳作者一下

全面了解傲腾技术，提升数据中心性能 广告关闭
创新的傲腾技术填补了内存与存储之间的空白，将高性能持久化及大容量低延迟完美结合，优化工作负载，深...

优质评论可以帮助作者获得更高权重



相关推荐

ToLua原理和使用教程_Wik1229_tlua 6-21

ToLua原理和使用教程 ToLua是用来实现C++程序中的类、变量、函数等绑定到lua程序中。绑定后lua代码可以直

tolua++ 编译 及使用 简单介绍 - boshuzhang的专栏 - CSDN博客 11-8
toLua:简洁的使用说明 2013年09月18日/综合/共 3926字/字号 小中大/评论关闭 tolua的最新版本是5.0,下载地

ToLua原理和使用教程_Wik1229_tlua 6-21

ToLua原理和使用教程 ToLua是用来实现C++程序中的类、变量、函数等绑定到lua程序中。绑定后lua代码可以直

tolua++ 编译 及使用 简单介绍 - boshuzhang的专栏 - CSDN博客 11-8
toLua:简洁的使用说明 2013年09月18日/综合/共 3926字/字号 小中大/评论关闭 tolua的最新版本是5.0,下载地

tolua++ 编译及使用 简单介绍 标签: tolua++ luaC++_bo... 6-19

tolua++ 是用来把 C++代码 生成 在Lua中使用的接口的工具,编译步骤比较繁琐。 1、下载lua5.1.4,解压后,新建空

tolua笔记《02》- lua脚本的简单调用_7-SEVENS 7-14

ToLua原理和使用教程_Wik1229_tlua 6-21

ToLua原理和使用教程 ToLua是用来实现C++程序中的类、变量、函数等绑定到lua程序中。绑定后lua代码可以直

tolua++ 编译 及使用 简单介绍 - boshuzhang的专栏 - CSDN博客 11-8
toLua:简洁的使用说明 2013年09月18日/综合/共 3926字/字号 小中大/评论关闭 tolua的最新版本是5.0,下载地

tolua++ 编译及使用 简单介绍 标签: tolua++ luaC++_bo... 6-19

tolua++ 是用来把 C++代码 生成 在Lua中使用的接口的工具,编译步骤比较繁琐。 1、下载lua5.1.4,解压后,新建空

tolua笔记《02》- lua脚本的简单调用_7-SEVENS 7-14

tolua#-LuaFramework 07-21

tolua#-LuaFramework 带UGUI NGUI两个版本

tolua++ 1.0.92 11-12

tolua++-1.0.92 源码 tolua++ is an extended version of tolua, a tool to integrate C/C++ code with Lua. tolua++ in

tolua++ 编译及使用 简单介绍 标签: tolua++ luaC++_bo... 6-19

tolua++ 是用来把 C++代码 生成 在Lua中使用的接口的工具,编译步骤比较繁琐。 1、下载lua5.1.4,解压后,新建空

tolua笔记《02》- lua脚本的简单调用_7-SEVENS 7-14

tolua++的使用 u013790419的专栏  1885

如约，今天加班加点搞定了tolua++的使用，算是对上一篇的承接，也是跟自己的约定的开始。下面是正文！首先

tolua++使用 u010314872的专栏 1041

在用cocos2d-x的时候已经用过tolua++了，今天有心情就下了源码包来试一试。tolua++的编译可以很容易在vs

tolua 新版本 05-24
unity tolua 相关源码 可供使用。

[Unity热更新]tolua# & LuaFramework(八): 更新下载(上) 宏哥的博客 7716

效果图: 未更新前的: 重新启动后: 更新流程分析: 1.找到GameManager.cs, 这个类包含了解包和更新下载

tolua++编译器绿色版 12-03

tolua++ 编译器 win版本 tolua++是一种第三方的软件包来自绿盟xdowns,可以为Lua提供面向对象的特性,这样我

tolua++中文文档整理 zgl390963305的博客 1854

tolua++ 参考手册中文版 作者: Mrzhu007 日期: 2018-04-15 博客地址: 金色世界 之前看了tolua++如何将C++

有关lua,luCI的介绍 user_920的专栏 1万+

一颗璀璨的月光宝石——Lua Lua语言简介 1993 年在巴西里约热内卢天主教大学(Pontifical Catholic University of Rio de Janeiro) 的计算机科学系教授Roberto Ierusalimsky 和Roberto

tolua 从lua传值到c++ 女武神的骑行啊的博客 421

一般都是使用tolua将c++部分的函数和参数，在lua部分调用，现在有个需求，要从lua部分，获取参数，传递到c

文档都是乱七八糟，很多还是过时的，没有从0开始，真是各种恶心。 **LuaManager**: 启动**Lua**，开始Main入口 **lu**

tolua

框架整理

u013341672的博客

4825

文/知乎 @罗培羽

第一篇：代码热更新

这些日子在找Unity3D的现成框架，希望能给后续项目开发带来

tolua++1.0.93编译器（win版）

12-03

编译好的tolua++ 编译器基于1.0.93版编译，可以直接使用

【木头Cocos2d-x 007】在Lua中使用自定义类——tolua++工具使用（上集）

mmdev 90

在Lua中使用自定义类——tolua++工具使用（上集） 笨木头花心贡献，啥？花心？不呢，是用心~转载请注明，

在Lua中使用自定义类——tolua++工具使用（下集）

尘墨 697

转载请注明，原文地址： http://blog.csdn.net/musicvs/article/details/8166655 正文： 上回说到，把LuaCocos2

Unity3d的热更新过程（ToLua）

04-24

请将ToLua框架和PhpStudy下载下来，并且要安装PhpStudy（weburl这个路径的话看图）

©2020 CSDN 皮肤主题: 大白 设计师:CSDN官方博客 返回首页

关于我们 招贤纳士 广告服务 开发助手

400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息
北京互联网违法和不良信息举报中心 网络110报警服务 中国互联网举报中心 家长监护 Chrome商店下载
©1999-2021北京创新乐知网络技术有限公司 版权与免责声明 版权申诉 出版物许可证 营业执照

关于我们 招贤纳士 广告服务 开发助手

400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息
北京互联网违法和不良信息举报中心 网络110报警服务 中国互联网举报中心 家长监护 Chrome商店下载
©1999-2021北京创新乐知网络技术有限公司 版权与免责声明 版权申诉 出版物许可证 营业执照

hsz8250

码龄19年 暂无认证

16

24万+

8万+

11万+

原创

周排名

总排名

访问

等级

1377

24

1

35

5

积分

粉丝

获赞

评论

收藏

私信

关注

搜博主文章

热门文章

BugFree使用说明

15271

想尽快进入游戏开发行业的必经之路!

14889

toLua:简洁的使用说明

9997

一个实而不华的LuaTinker(1)

8963

Doxygen代码注释规定和生产文档说明

6265

分类专栏

C/C++

4篇

CEGUI

Direct3D

Linux

OGRE

5篇

STL

最新评论

BugFree使用说明

逗逗哥: 大佬 我想知道怎么把项目创建好后再创建模块

想尽快进入游戏开发行业的必经之路!

ton365 回复 hsz8250: 对现在C++游戏开发新人有什么要求,

Rational Purify 使用手册(2)

ytfrdfiw: 写得不错，谢谢!

一个实而不华的LuaTinker(1)

jackbjy: 你好，lua_tinker可以像luabind的那样注册函数时注册到某个模块中，比如...

Ogre中设置渲染性能参数

starmorning2011: 什么是渲染,解释下撒!

最新文章

如何学好游戏3D引擎编程

谈一谈网络编程学习经验

分享平时工作中那些给力的shell命令

2011年 12篇

2007年 3篇

2006年 22篇