

文档: 教程: 基础教程: 基础教程六

出自 OGRE3D 开放资源地带

跳转到: [导航](#), [搜索](#)

目录

- [1 必备知识](#)
- [2 工程设置](#)
- [3 介绍](#)
- [4 开始](#)
- [5 CEGUI 是如何工作的](#)
- [6 添加退出按钮](#)
- [7 响应事件](#)
- [8 加载设置 \(Layout\)](#)
- [9 尝试](#)
- [10 如何在两个 GUI \(用户界面\) 之间转换 \(使用透明度\)](#)
- [11 结论](#)

必备知识

本教程是在假设你已经拥有 C++ 编程基础并能够成功建立和编译 OGRE 程序（如果你设置程序方面还存在问题，请参阅 [SettingUpAnApplication](#) 来获取详细信息）。本教程建立在之前的初学者教程的基础上，并且假设你已经学习了它们。

工程设置

下面的适用于下载源代码的用户：

```
添加 include 文件夹: $(OGRE_HOME)\Dependencies\include,  
                      $(OGRE_HOME)\Dependencies\include\CEGUI  
添加 lib 库路径:      $(OGRE_HOME)\OGREMain\Dependencies\Lib\Debug
```

确信已经链接 'CEGUI Base' 和 'OGRE GUI Render' 库，也就是说将下面一行添加进你的 Makefile 文件或 g++ 命令行：

```
-L/usr/local/lib -lCEGUIBase -lCEGUIOGRERender
```

下面的适用于 SDK 的用户：

```
添加 include 文件夹: $(OGRE_HOME)\include\CEGUI
```

确信已经在 debug 配置的中添加 'CEGUIBase_d.lib' 和 'OgreGUIRenderer_d.lib' 库（ 'CEGUIBase.lib' 和 'OgreGUIRenderer.lib' 在 release 配置中）。在 Visual C++ 中添加依赖，依次点击：项目 -> 属性 -> 配置属性 -> 链接。

CEGUIRender 源程序现在是从 Ogre CVS 下载代码中的一部分，一个示例工程，因此你必须将包含 OgreGUIRenderer 头文件和 lib 文件的文件夹路径添加到属性配置中。

另外，下面两个目录是必需的。尽管你在你的安装路径中的文件夹找不到。将其作为约定它就会起作用：

添加 Include 文件夹：

\$(OGRE_HOME)\Samples\Common\CEGUIRenderer\include

添加 Lib 路径： \$(OGRE_HOME)\Samples\Common\CEGUIRenderer\lib

介绍

[Crazy Eddies GUI](#) 系统是一个为不具备或缺乏用户界面制作功能的图形 API 或引擎提供免费用户界面支持的开源的库。这个使用 c++ 编写的库是针对那些想制作优秀的游戏却又没有 GUI（图形用户界面）子系统的专业游戏开发者。

开始

首先，你需要架构（skeleton）代码来创建具有 CEGUI 组件的 Ogre 程序。注意：如果你使用 <Windows.h>，你必须在 <Windows.h> 之前添加 #define NOMINMAX。

```
//mem probs without this next one
#include <OgreNoMemoryMacros.h>
#include <CEGUI/CEGUIImageset.h>
#include <CEGUI/CEGUISystem.h>
#include <CEGUI/CEGUILogger.h>
#include <CEGUI/CEGUISchemeManager.h>
#include <CEGUI/CEGUIWindowManager.h>
#include <CEGUI/CEGUIWindow.h>
#include "OgreCEGUIRenderer.h"
#include "OgreCEGUIResourceProvider.h"
//regular mem handler
#include <OgreMemoryMacros.h>
#include "ExampleApplication.h"
class GuiFrameListener : public ExampleFrameListener
{
private:
```

```

        CEGUI::Renderer* mGUIRenderer;
public:
    GuiFrameListener(RenderWindow* win, Camera* cam, CEGUI::Renderer*
renderer)
        : ExampleFrameListener(win, cam, false, false),
          mGUIRenderer(renderer)
    {
    }
};

```

仅仅是一个不做任何动作的空帧监听器，但在你按下“Esc”之前会一直循环。

```

class TutorialApplication : public ExampleApplication
{
private:
    CEGUI::OgreCEGUIRenderer* mGUIRenderer;
    CEGUI::System* mGUISystem;
    CEGUI::Window* mEditorGuiSheet;

```

这些是包含所有 CEGUI 数据的数据成员。我喜欢显式的调用 CEGUI 成员，一但你开始对 Ogre 成员添加调用，这将会明确的说明它们是来自 CEGUI。

```

public:
    TutorialApplication()
        : mGUIRenderer(0),
          mGUISystem(0),
          mEditorGuiSheet(0)
    {
    }
    ~TutorialApplication()
    {
        if(mEditorGuiSheet)
        {
            CEGUI::WindowManager::getSingleton().destroyWindow(mEditorGuiSheet);
        }
        if(mGUISystem)
        {
            delete mGUISystem;
            mGUISystem = 0;
        }
        if(mGUIRenderer)
        {
            delete mGUIRenderer;
            mGUIRenderer = 0;

```

```
    }
}
```

下面是你可以设置任意 Ogre 场景的地方，使用你在前五章教程学到的方法。在这个 Ogre 场景中，你仍要为其添加一个独立的相机（camera）和视窗（viewport）。

```
protected:
    void createScene(void)
    {
        // Set ambient light
        mSceneMgr->setAmbientLight(ColourValue(0.5, 0.5, 0.5));
```

下面是创建 CEGUI 日志的地方，一般都设置为 Informative 模式的。其具有四种模式：Standard, Errors, Informative 和 Insane。

```
        // Set up GUI system
        mGUIRenderer = new CEGUI::OgreCEGUIRenderer(mWindow,
            Ogre::RENDER_QUEUE_OVERLAY, false, 3000, mSceneMgr);
        mGUISystem = new CEGUI::System(mGUIRenderer);
```

```
CEGUI::Logger::getSingleton().setLogLevel(CEGUI::Informative);
```

创建一个新的 CEGUI 系统，使用“TaharezLook”来设置图（scheme）与鼠标指针，使用“BlueHighway-12”来设置字体。

```
CEGUI::SchemeManager::getSingleton().loadScheme((CEGUI::utf8*)"TaharezLookSkin.scheme");
    mGUISystem->setDefaultMouseCursor((CEGUI::utf8*)"TaharezLook", (CEGUI::utf8*)"MouseArrow");
```

```
CEGUI::MouseCursor::getSingleton().setImage("TaharezLook", "MouseMoveCursor");
    mGUISystem->setDefaultFont((CEGUI::utf8*)"BlueHighway-12");
```

```
mEditorGuiSheet=CEGUI::WindowManager::getSingleton().createWindow((CEGUI::utf8*)"DefaultWindow",
    (CEGUI::utf8*)"Sheet");
    mGUISystem->setGuiSheet(mEditorGuiSheet);
}
```

调用自定义的帧监听器，这样我们可以在需要时访问“mGUIRender”。

```

void createFrameLi stener(void)
{
    mFrameLi stener = new GuiFrameLi stener(mWindow, mCamera,
mGUIRenderer);
    mRoot->addFrameLi stener(mFrameLi stener);
}
};

```

下面是主函数也是程序的主循环，在本教程并不需要你修改这段代码。

```

    #if OGRE_PLATFORM == OGRE_PLATFORM_WIN32
#define WIN32_LEAN_AND_MEAN
#include "windows.h"
    INT WINAPI WinMain( HINSTANCE hInst, HINSTANCE, LPSTR strCmdLine,
INT )
    #else
    int main(int argc, char **argv)
    #endif
    {
        // Create application object
        TutorialApplication app;
        try {
            app.go();
        } catch( Exception& e ) {
            #if OGRE_PLATFORM == OGRE_PLATFORM_WIN32
                MessageBox( NULL, e.getFullDescription().c_str(), "An
exception has occurred!", MB_OK | MB_ICONERROR | MB_TASKMODAL);
            #else
                fprintf(stderr, "An exception has
occured: %s\n", e.getFullDescription().c_str());
            #endif
        }
        return 0;
    }
}

```

完成后，编译程序能得到一个空的窗口。请关掉程序，继续我们的学习。注意：如果实际操作中出现问题，你可以在应用程序所在文件夹中找到“CEGUI.log”文件分析查找错误

CEGUI 是如何工作的

本质上 CEGUI 是通过向窗口添加第二个场景，这个场景是在 OGRE 的基本渲染队列完成后才渲染的。这个场景仅仅是由一系列 3D 矩形对象组成的。（也就是两个多边形沿着其边压制到一起）。渲染矩阵是为消除矩形的突兀与歪斜而根据

他们的位置建立的。使用这些矩形，添加材质和响应就构成了用户界面（GUI）。一般情况下这是很不错的，因为一个 3D 的用户界面将会自动的缩放其元素来适应屏幕，并且使用硬件材质过滤。其将会比 C++ 标准的 2D 用户界面更加快速和漂亮。

“So in one sentence: CEGUI renders a 2D gui using 3D methods and hardware so you don't have to.” ——zeroskill

添加退出按钮

首先，我们需要为应用程序添加下面的头文件。本例中是 “Push Button”

```
#include <CEGUI/elements/CEGUIPushButton.h>
```

我们要在场景底部添加退出按钮。

```
CEGUI::PushButton* quitButton =  
(CEGUI::PushButton*)CEGUI::WindowManager::getSingleton().createWindow  
    ("TaharezLook/Button",  
(CEGUI::utf8*)"Quit");  
mEditorGuiSheet->addChildWindow(quitButton);  
quitButton->setPosition(CEGUI::Point(0.35f, 0.45f));  
quitButton->setSize(CEGUI::Size(0.3f, 0.1f));  
quitButton->setText("Quit");
```

完成后执行程序，一个漂亮的按钮将会出现在屏幕中。但请注意，程序此时仍然不做任何事情，因为我们并没有为其添加响应时间。

如果你编译时遇到了 `setPosition()` 和 `setSize()` 的调用错误：

```
'CEGUI::Window::setPosition' : cannot convert parameter 1 from  
'CEGUI::Vector2' to 'const CEGUI::UVector2 &'
```

将 `setPosition()` 和 `setSize()` 所在行分别用下面的代码替换：

```
quitButton->setPosition(CEGUI::UVector2(cegui_reldim(0.35f),  
cegui_reldim( 0.45f)) );  
quitButton->setSize(CEGUI::UVector2(cegui_reldim(0.35f),  
cegui_reldim( 0.1f)) );
```

响应事件

将下面函数添加到 Tutorial Application 的 `public:` 中

```
void setupEventHandlers(void)  
{
```

```

    CEGUI::WindowManager& wmgr =
CEGUI::WindowManager::getSingleton();
    wmgr.getWindow((CEGUI::utf8*)"Quit")-
>subscribeEvent(CEGUI::PushButton::EventClicked,
CEGUI::Event::Subscriber

(&TutorialApplication::handleQuit, this));
}
bool handleQuit(const CEGUI::EventArgs& e)
{
    static_cast<GuiFrameListener*>(mFrameListener)-
>requestShutdown();
    return true;
}

```

重写 GuiFrameListener 类来响应键盘和鼠标输入

```

class GuiFrameListener : public ExampleFrameListener, public
MouseMotionListener, public MouseListener
{
private:
    CEGUI::Renderer* mGUIRenderer;
    bool mShutdownRequested;
public:
    // NB using buffered input
    GuiFrameListener(RenderWindow* win, Camera* cam,
CEGUI::Renderer* renderer)
        : ExampleFrameListener(win, cam, true, true),
        mGUIRenderer(renderer),
        mShutdownRequested(false)
    {
        mEventProcessor->addMouseMotionListener(this);
        mEventProcessor->addMouseListener(this);
        mEventProcessor->addKeyListener(this);
    }
    // Tell the frame listener to exit at the end of the next frame
    void requestShutdown(void)
    {
        mShutdownRequested = true;
    }
    bool frameEnded(const FrameEvent& evt)
    {
        if (mShutdownRequested)
            return false;
        else

```

```

        return ExampleFrameListener::frameEnded(evt);
    }
    void mouseMoved (MouseEvent *e)
    {
        CEGUI::System::getSingleton().injectMouseMove(
            e->getRelX() * mGUIRenderer->getWidth(),
            e->getRelY() * mGUIRenderer->getHeight());
        e->consume();
    }
    void mouseDragged (MouseEvent *e)
    {
        mouseMoved(e);
    }
    void mousePressed (MouseEvent *e)
    {
        CEGUI::System::getSingleton().injectMouseButtonDown(
            convertOgreButtonToCegui(e->getButtonID()));
        e->consume();
    }
    void mouseReleased (MouseEvent *e)
    {
        CEGUI::System::getSingleton().injectMouseButtonUp(
            convertOgreButtonToCegui(e->getButtonID()));
        e->consume();
    }
    void mouseClicked(MouseEvent* e) {}
    void mouseEntered(MouseEvent* e) {}
    void mouseExited(MouseEvent* e) {}
    void keyPressed(KeyEvent* e)
    {
        if(e->getKey() == KC_ESCAPE)
        {
            mShutdownRequested = true;
            e->consume();
            return;
        }
        CEGUI::System::getSingleton().injectKeyDown(e->getKey());
        CEGUI::System::getSingleton().injectChar(e->getKeyChar());
        e->consume();
    }
    void keyReleased(KeyEvent* e)
    {
        CEGUI::System::getSingleton().injectKeyUp(e->getKey());
        e->consume();
    }

```



```

    }
    void keyClicked(KeyEvent* e)
    {
        // Do nothing
        e->consume();
    }
};

```

Ogre 1.4.0

如果你使用的是 Ogre 1.4.0 你将会要使用 OIS。在 Ogre3d 中使用 OIS 的更多细节, 请参阅[使用 OIS](#) 并且再看一看[基础教程 5](#):

```

class GuiFrameListener : public ExampleFrameListener, public
OIS::MouseListener, public OIS::KeyListener
{
private:
    CEGUI::Renderer* mGUIRenderer;
    bool mShutdownRequested;
public:
    // NB using buffered input
    GuiFrameListener(RenderWindow* win, Camera* cam, CEGUI::Renderer*
renderer)
        : ExampleFrameListener(win, cam, true, true),
        mGUIRenderer(renderer),
        mShutdownRequested(false)
    {
        mMouse->setEventCallback( this );
        mKeyboard->setEventCallback( this );
    }
    // Tell the frame listener to exit at the end of the next frame
    void requestShutdown(void)
    {
        mShutdownRequested = true;
    }
    bool frameEnded(const FrameEvent& evt)
    {
        if (mShutdownRequested)
            return false;
        else
            return ExampleFrameListener::frameEnded(evt);
    }
    bool mouseMoved( const OIS::MouseEvent &e )
    {
        using namespace OIS;

```

```

CEGUI::System::getSingleton().injectMouseMove(e.state.X.rel, e.state.Y
.rel);
    return true;
}
bool mousePressed (const OIS::MouseEvent &e, OIS::MouseButtonID
id)
{

CEGUI::System::getSingleton().injectMouseButtonDown(convertOgreButton
ToCegui(id));
    return true;
}
bool mouseReleased( const OIS::MouseEvent &e, OIS::MouseButtonID
id )
{

CEGUI::System::getSingleton().injectMouseButtonUp(convertOgreButtonTo
Cegui(id));
    return true;
}
bool keyPressed( const OIS::KeyEvent &e )
{
    if(e.key == OIS::KC_ESCAPE)
    {
        mShutdownRequested = true;
        return true;
    }
    CEGUI::System::getSingleton().injectKeyDown(e.key);
    CEGUI::System::getSingleton().injectChar(e.text);
    return true;
}
bool keyReleased( const OIS::KeyEvent &e )
{
    CEGUI::System::getSingleton().injectKeyUp(e.key);
    return true;
}
};

```

在 include 语句后 GuiFrameListener 声明前添加下面代码

```

CEGUI::MouseButton convertOgreButtonToCegui (int buttonID)
{
    switch (buttonID)

```

```

{
case MouseEvent::BUTTON0_MASK:
    return CEGUI::LeftButton;
case MouseEvent::BUTTON1_MASK:
    return CEGUI::RightButton;
case MouseEvent::BUTTON2_MASK:
    return CEGUI::MiddleButton;
case MouseEvent::BUTTON3_MASK:
    return CEGUI::X1Button;
default:
    return CEGUI::LeftButton;
}
}

```

Ogre 1.4.0

```

CEGUI::MouseButton convertOgreButtonToCegui (int buttonID)
{
    using namespace OIS;
    switch (buttonID)
    {
    case OIS::MB_Left:
        return CEGUI::LeftButton;
    case OIS::MB_Right:
        return CEGUI::RightButton;
    case OIS::MB_Middle:
        return CEGUI::MiddleButton;
    default:
        return CEGUI::LeftButton;
    }
}

```

将下面语句添加到创建场景方法（createscene）的末尾。

```
setupEventHandlers();
```

现在你可以编译并执行程序了。实现效果是点击按钮后退出。

加载设置（Layout）

CEGUI 使用 XML 格式来加载图形用户界面样式设置。复制下面 xml 代码到记事本，并将其以“Tutorial Gui.xml”命名另存在“\media\gui”文件夹下。

```

<?xml version="1.0" ?>
<GUI Layout>

```

```

    <Window Type="DefaultWindow" Name="Tutorial Gui">
        <Window Type="TaharezLook/Button" Name="Quit">
            <Property Name="AbsoluteRect" Value="l:224.000000
t:216.000000 r:416.000000 b:264.000000" />
            <Property Name="RelativeRect" Value="l:0.350000
t:0.450000 r:0.650000 b:0.550000" />
            <Property Name="Text" Value="Quit" />
        </Window>
    </Window>
</GUILayout>

```

Ogre 1.4.0

```

<?xml version="1.0" ?>
<GUILayout>
    <Window Type="DefaultWindow" Name="Tutorial Gui">
        <Window Type="TaharezLook/Button" Name="Quit">
            <Property Name="UnifiedPosition"
Value="{{0.35,0},{0.45,0}}" />
            <Property Name="UnifiedSize" Value="{{0.3,0},{0.1,0}}"
/>
            <Property Name="Text" Value="Quit" />
        </Window>
    </Window>
</GUILayout>

```

(补充说明:在复制上面的.xml 代码保存时,请手动删除行前的空格.否则会编译出错. edi tBy 自由骑士笃志 2008-04-25)

现在将程序中下列代码段注释掉

```

mEditorGuiSheet=
CEGUI::WindowManager::getSingleton().createWindow((CEGUI::utf8*)"DefaultWindow", (CEGUI::utf8*)"Sheet");
mGuiSystem->setGuiSheet(mEditorGuiSheet);
CEGUI::PushButton* quitButton =
(CEGUI::PushButton*)CEGUI::WindowManager::getSingleton().createWindow
("TaharezLook/Button",
(CEGUI::utf8*)"Quit");
mEditorGuiSheet->addChildWindow(quitButton);
quitButton->setPosition(CEGUI::Point(0.35f, 0.45f));
quitButton->setSize(CEGUI::Size(0.3f, 0.1f));
quitButton->setText("Quit");

```

在同样位置添加下列代码

```

    mEditorGuiSheet =
CEGUI::WindowManager::getSingleton().loadWindowLayout((CEGUI::utf8*)"
Tutorial Gui.xml");
    mGUISystem->setGuiSheet(mEditorGuiSheet);
    CEGUI::PushButton* quitButton=(CEGUI::PushButton*)CEGUI::

WindowManager::getSingleton().getWindow((CEGUI::utf8*)"Quit");

```

最后一行多余的，因为我们没有在以后使用指针，但是其说明了如何通过加载文件来进行访问。注意：我们在创建 xml 文件时要根据实际窗口进行设计。

完成后编译并执行，程序在外观上并没有变化。

尝试

- 视线相交和选取 Ogre mesh—当鼠标没有从 GUI 元素上移过。
- 定义一个在 GUI 根菜单上的鼠标点击动作。当你鼠标点击一个不在根窗口中的 GUI 元素时，它将会响应你的鼠标点击。如果你的鼠标并没有从一个 GUI 元素上滑过（也就是说当你的鼠标指针还在我们的 3D 场景中）则根窗口响应鼠标点击。
- 将鼠标点击转换到世界坐标系和视线相交（（Camera::getCamera）到 ViewportRay(mouseX, mouseY)）

```

// Start a new ray query
Ogre::Ray cameraRay = root::getSingleton().
    getCamera()->getCameraToViewportRay( mouseX, mouseY );
Ogre::RaySceneQuery *raySceneQuery = root::getSingleton().
    getSceneManager()->createRayQuery( cameraRay );
raySceneQuery->execute();
Ogre::RaySceneQueryResult result = raySceneQuery-
>getLastResults();
Ogre::MovableObject *closestObject = NULL;
real closestDistance = LONG_MAX;
std::list<Ogre::RaySceneQueryResultEntry>::iterator rayIterator;
for ( rayIterator = result.begin();
    rayIterator != result.end();
    rayIterator++ ) {
    if ( ( *rayIterator ).movable->getUserObject() != NULL )
    {
        if ( ( *rayIterator ).distance < closestDistance )
        {
            closestObject = ( *rayIterator ).movable;
            closestDistance = ( *rayIterator ).distance;
        }
    }
}

```

```

    }
    // No object clicked
    if ( closestObject == NULL ) {
        clickedObject = NULL;
        -----
        clickedObject is a class scoped variable
    } else {
        clickedObject = static_cast< object* >( closestObject-
>getUserObject( ) );
    }
    raySceneQuery->clearResults( );
    root::getSingleton( ).getSceneManager( )-
>destroyQuery( raySceneQuery )

```

如何在两个 GUI（用户界面）之间转换（使用透明度）

例如：如果你有一个登陆界面，在成功登陆后，进入了你的用户主界面。你会想在这两个界面间切换。 •第一步，加载登陆用户界面。

```

//First loading with this
mGUIRenderer = new CEGUI::OgreCEGUIRenderer(mWindow,
Ogre::RENDER_QUEUE_OVERLAY, false, 3000, mSceneMgr);
mGUISystem = new CEGUI::System(mGUIRenderer);
CEGUI::Logger::getSingleton().setLoggingLevel(CEGUI::Informative);

CEGUI::SchemeManager::getSingleton().loadScheme((CEGUI::utf8*)"WindowsLook.scheme");
mGUISystem->setDefaultMouseCursor((CEGUI::utf8*)"WindowsLook",
(CEGUI::utf8*)"MouseArrow");
CEGUI::Font *f =
CEGUI::FontManager::getSingleton().createFont("Commonwealth-10.font");
mGUISystem->setDefaultFont(f);
//End "first loading with this"
//Load a XML file
mEditorGuiSheet =
CEGUI::WindowManager::getSingleton().loadWindowLayout((CEGUI::utf8*)"Presentation.xml");
mGUISystem->setGuiSheet(mEditorGuiSheet);

```

•第二步，如果你想删除并重建一个 GUI，你需要做到以下：

```

if(mEditorGuiSheet)

CEGUI::WindowManager::getSingleton().destroyWindow(mEditorGuiSheet);

```

- 最后一步，加载其他的 GUI

```
mEditorGuiSheet =  
CEGUI::WindowManager::getSingleton().loadWindowLayout((CEGUI::utf8*)"Futura.xml");  
mGUISystem->setGuiSheet(mEditorGuiSheet);
```

重做第二步和最后一步来加载其他 GUI。

结论

这个教程为你展示了在 Ogre3D 下使用 CEGUI 的一些基本方法，你可以感受下使用 CUEGUI 编程的乐趣:)

[原文](#)

[上一章节：基础教程五](#)

[下一章节：基础教程七](#)

[目录](#)

取自

["http://ogre3d.cn/wiki/index.php?title=%E6%96%87%E6%A1%A3:%E6%95%99%E7%A8%8B:%E5%9F%BA%E7%A1%80%E6%95%99%E7%A8%8B:%E5%9F%BA%E7%A1%80%E6%95%99%E7%A8%8B%E5%85%AD"](http://ogre3d.cn/wiki/index.php?title=%E6%96%87%E6%A1%A3:%E6%95%99%E7%A8%8B:%E5%9F%BA%E7%A1%80%E6%95%99%E7%A8%8B:%E5%9F%BA%E7%A1%80%E6%95%99%E7%A8%8B%E5%85%AD)

查看

- [页面](#)
- [讨论](#)
- [源码](#)
- [历史](#)

个人工具


- [登录 / 创建账户](#)

导航


- [首页](#)
- [社区](#)
- [当前事件](#)
- [最近更改](#)
- [随机页面](#)

- [帮助](#)

搜索





Google 搜索





our search terms


search form




















工具箱

- [链入页面](#)
- [链出更改](#)
- [特殊页面](#)
- [可打印版](#)
- [永久链接](#)

Google Adsense



- 这页的最后修订在 2009 年 4 月 3 日（星期五）09: 27。
- 本页面已经被浏览 1, 592 次。
- [隐私政策](#)
- [关于 0gre3D 开放资源地带](#)
- [沪 ICP 备 09049564 号](#)