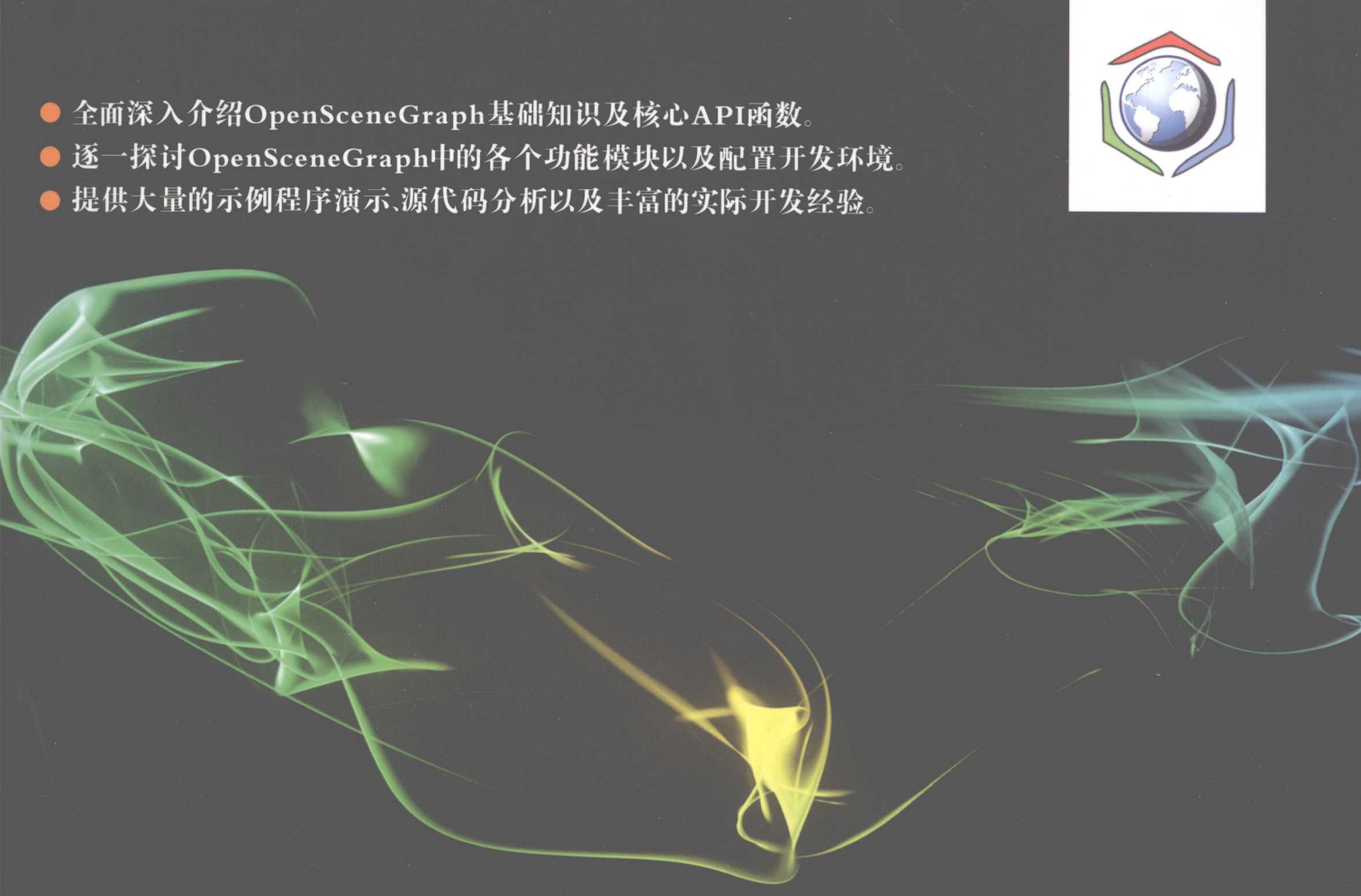

- 全面深入介绍OpenSceneGraph基础知识及核心API函数。
- 逐一探讨OpenSceneGraph中的各个功能模块以及配置开发环境。
- 提供大量的示例程序演示、源代码分析以及丰富的实际开发经验。

OpenSceneGraph
三维渲染引擎编程指南

肖 鹏 刘更代 徐明亮 编著

- 读者将具备开发一款基于OpenSceneGraph的虚拟现实系统的能力。
- 拥有丰富的社区资源和强大的网络支持,以方便读者进一步的学习和交流。

清华大学出版社

OpenSceneGraph

OpenSceneGraph 三维渲染引擎编程指南

肖　鹏　刘更代　徐明亮　编著

清华大学出版社
北　京

内 容 简 介

本书是一本全面深入介绍 OpenSceneGraph（OSG）基础及核心 API 函数的入门教程。OpenSceneGraph（OSG）是一个基于工业标准 OpenGL 跨平台的三维开源场景图形系统应用程序开发接口（API）。作为一个高性能的图形开发引擎，它在 3D 程序开发中扮演着重要的角色。

本书按照 OSG 的设计结构体系，逐一深入讨论 OSG 的各个功能模块。首先介绍了 OSG 的历史和开源组织，以及配置开发环境；然后深入探讨 OSG 的核心库、NodeKits 工具库、OSG 插件库、互操作库及扩展库等，重点分析了如何将 OSG 集成到用户应用开发程序的核心功能及各种应用技术，主要包括场景组织和管理、场景数据优化、交互操作及数据实时动态更新等技术；最后探讨了关于 OSG 地形与地理信息的应用。

本书要求读者有比较好的 C++基础知识和一些 3D 数学基础知识，适合所有对 OpenGL 和 OSG 编程感兴趣的读者。

图书在版编目（CIP）数据

OpenSceneGraph 三维渲染引擎编程指南/肖鹏，刘更代，徐明亮编著. —北京：清华大学出版社，2010.1

ISBN 978-7-302-21303-1

I. O…　II. ①肖…　②刘…　③徐…　III. 计算机图形学　IV. TP391.41

中国版本图书馆 CIP 数据核字（2009）第 181375 号

责任编辑：熊　健　朱　俊
封面设计：刘　超
版式设计：魏　远
责任校对：王　云
责任印制：孟凡玉
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京市清华园胶印厂
装 订 者：三河市溧源装订厂
经　　销：全国新华书店
开　　本：203×260　**印　张**：27.25　**彩　插**：1　**字　数**：624 千字
版　　次：2010 年 1 月第 1 版　　**印　　次**：2010 年 1 月第1 次印刷
印　　数：1～4000
定　　价：49.80 元

前　言

Preface

OpenSceneGraph（OSG）场景图形系统是一个基于工业标准 OpenGL 的软件接口，它让程序员能够更加快速、便捷地创建高性能、跨平台的交互式图形程序。本书是一本基于 OSG 2.8 版本的编程入门指南书籍，也通用于 OSG 2.x 系列及以后版本的开发。

从 OSG 正式发布以来，OSG 可以阅读的文档资料只有源代码。值得庆幸的是，OSG 的源代码中提供了一些相关的程序示例，通过它们，用户可以学习如何使用 OSG 编程实现虚拟系统软件的功能。分析示例程序很艰辛，并非一朝一夕的事情。笔者通过对 OSG 大量示例程序和源代码的分析及丰富的实际应用经验，逐渐对 OSG 的开发流程及核心有了一定的了解。随着 OSG 的不断发展，越来越多的人开始关注、学习 OSG 和应用 OSG 开发，但由于 OSG 本身文档资料缺乏，作为初学者，很难在短时间内通过分析示例程序和源代码得到提高。笔者长期置身于 OSG 开发，作为 OsgChina（http://www.OsgChina.org）中国官方网站管理员，同时也是 OSG 国内各大论坛 OSG 版面的版主，在与开发者交流的过程中，发现初学者普遍存在入门难、应用开发难等问题，于是写了《OpenSceneGraph 三维渲染引擎编程指南》一书。

本书首先介绍了 OSG 的历史和开源组织、如何获取和正确安装 OSG 以及一些简单示例程序的运行；然后深入探讨了一些 OSG 的数学基础、内部管理机制和实用技术，主要包括坐标系统、内存管理、场景树结构、图形节点的概念和分类、OSG 的状态属性和模式控制、纹理映射、光照和材质、I/O 接口、场景渲染控制、更新回调、文字、动画阴影及粒子系统添加等功能的具体介绍；最后探讨了关于 OSG 地形与地理信息的应用，主要包括 VirtualPlanetBuilder 和 OSGGIS 的使用方法，还提出了关于大规模地形渲染的解决方案。

通过本书，读者不但可以学习到 OSG 在虚拟现实开发中最重要、最实用的知识和技能，还可学习到一些开源软件使用的基本技法。在学习完本书后，读者将会对 OSG 有一个深入全面的了解，同时也将具备开发一个基于 OSG 的虚拟现实系统的能力。

本书参阅了国内外大量的最新图形开发和虚拟现实实用技术的资料，所讲解的内容都是基于图形开发的最新方法和技术。书中提供了大量丰富、新颖、实用的原创示例程序，并经过严格的测试，可以完全应用到实际工程或者系统开发中。由于作者能力有限，书中难免存在错误和不足之处，恳请广大读者批评指正。同时，欢迎到 OsgChina 中国官方讨论区（http://bbs.OsgChina.org）发帖提问，笔者会尽量在第一时间回复相关的问题。

本书能够顺利完成并出版，得到了这些人大量的帮助：

感谢编辑，他对本书倾注了大量的心血，给了我巨大的帮助和指导。

感谢我的师兄刘更代和徐明亮，没有他们，难以想象这本书什么时候才能完成并出版。

感谢我的师兄杨石兴和曹明亮，他们引导我进入 OSG 的开发领域。

感谢我的爸爸、妈妈和姐姐，他们给了我生活自主的空间，让我能够尽情地飞翔在自己的天空。

感谢我的女友雄英，她每天都给我很多鼓励和帮助。

希望本书能够帮助您进入 OSG 开发的殿堂，能够让您的生活因为 OSG 而更加美好。

肖　鹏

目 录

Contents

第 1 章

OpenSceneGraph 概述

欢迎开始学习 OpenSceneGraph（OSG）的旅程。通过本章，读者将了解到 OSG 的历史、发展和组成模块，并将学习如何获取、安装及编译 OSG。本章将对本书中使用的一些规范加以说明，还会讲解如何使用 OSG 的常用工具以及如何设置开发环境。

1.1 OpenSceneGraph 简介

1.1.1 什么是 OpenSceneGraph

OSG 是一个开源的场景图形管理开发库，主要为图形图像应用程序的开发提供场景管理和图形渲染优化功能。它使用可移植的 ANSI C++编写，并使用已成为工业标准的 OpenGL 底层渲染 API。因此，OSG 具备跨平台性，可以运行在 Windows、Mac OS X 和大多数类型的 UNIX 和 Linux 操作系统上。在 OSG 中，大部分的操作可以独立于本地 GUI，但是 OSG 也包含了针对某些视窗系统特有功能的支持代码，这主要是源于 OpenGL 本身的特性。

OSG 是公开源代码的，它的用户许可方式为修改过的 GNU 宽通用公共许可证（GNU Lesser General Public License，LGPL）。

1.1.2 OpenSceneGraph 的历史和发展

早在 1997 年，Don Burns 便作为软件设计顾问受雇于 Silicon Graphics（SGI），他在业余时间还喜欢滑翔运动。正因为对计算机图形和滑翔机同样的热衷及对尖端渲染设备的了解，他使用 Performer 场景图形（SGI 专有）系统设计了一套基于 SGI Onyx 的滑翔仿真软件。

由于受到其他滑翔爱好者的鼓励，Don 开始尝试使用 Linux 上的 Mesa3D 和 3dfx 的 Voodoo 设备，以开发基于更多硬件平台的仿真软件。当这套软件开始支持 OpenGL 时，场景图形的概念还未能应用于 Linux。为了填补这一空缺，Don 开始编写一套简单的、类似于 Performer 的场景图形系统，名为 SG。SG 的开发强调朴素且易用，它满足了当时人们对于场景图形系统的需求，也使 Don 的滑翔仿真软件能够运行于低成本的 Linux 系统。

到了 1998 年，Don 在滑翔爱好者的邮件组中遇到了 Robert Osfield。那时 Robert 在 Midland Valley Exploration 工作，那是一个来自苏格兰格拉斯哥的油气公司。Robert 同样对计算机图形学和可视化技术有着浓厚的兴趣。两人开始合作，对仿真软件进行改善。Robert 倡导开源，并提议将 SG 作为独立的开源场景图形项目继续开发，并由自己担任项目主导。项目的名称改为 OpenSceneGraph，当时共有 9 人加入了 OSG 的用户邮件列表。

2000 年底，Brede Johansen 为 OpenSceneGraph 作出了第一份贡献，即添加了 OSG 的 OpenFlight 模块。当时他在挪威孔斯贝格的 Kongsberg Maritime 船舶仿真公司工作，该公司后来设计了基于 OSG 的 SeaView R5 视觉系统。

同样在 2000 年，Robert 离开了原来的工作单位，作为 OpenSceneGraph 的专业服务商开始全职进行 OSG 的开发工作。在这段时间，他设计并实现了今天的 OSG 所使用的许多核心功能，并且是在完全没有客户和薪酬的情况下完成的。Don 到了 Keyhole 数字地图公司（现在是 Google 的 Google Earth 部门），于 2001 年辞职，他也组建了自己的公司——Andes ComputerEngineering，位于加利福尼亚州的圣何塞，公司成立后继续进行 OSG 的开发工作。第一届 OpenSceneGraph“同好”会在 SIGGRAPH 2001 举行，只有 12 个人参加，听众包括了来自 Magic Earth 的代表，其目的是寻求一个开源的场景图形库来支持油气

相关软件的开发。他们与 Don 和 Robert 讨论了开发的事宜，并成为了 OSG 的第一个收费用户[1]。

随着这几年开源的不断发展，OSG 的模块和第三方附加库不断完善，OSG 已具备对高性能渲染、海量地形数据库、地理信息及多通道等的支持。

1.1.3　OSG 中国

于 2008 年初成立的 OsgChina——OSG 中国官方网（http://www.OsgChina.org），作为国内目前最大的专注于 OpenSceneGraph（OSG）发展和研究的网站和论坛，以及 OSG 英文官方站点（http://www.openscenegraph.org）的唯一中文镜像站，一直致力于为国内 OSG 爱好者和开发者们提供一个交流和相互学习的无障碍平台，并且不断地收集、整理、归纳、创新，形成了愈加完备的 OSG 学习者资源集散中心，有数以千计的注册会员作为坚实后盾。

OsgChina 日常工作人员由国内资深开发者 FreeSouth、FlySky、array 和 Hesicong 等组成，他们为 OSG 在中国发展提供强大的技术支持。同时，OsgChina 将举办定期或不定期的 OSG 相关会议和研讨活动，邀请国内外的开发者和企业代表参加，与会者可以在会上展出自己的作品，宣传自己的品牌和理念。OsgChina 还向与会者提供工作和进修机会，并开展各种专题讲座和学习会活动。

如果读者在学习的过程中有很多疑问，欢迎注册 OSG 中国官方讨论社区（http://bbs.OsgChina.org）发帖提问，共同提高。

1.1.4　OSG 组成模块

通过前面的介绍已经了解，OpenSceneGraph 及其扩展位于系统的 API 一级，即系统的底层绘图硬件和相应的软件驱动程序之上封装了 OpenGL，并对其余的底层图形显示方式予以支持，利用 OpenSceneGraph 可以轻松地开发其上层的应用程序。OSG 层次结构图如图 1-1 所示。

在进一步学习 OSG 之前，有必要学习 OSG 的组成模块，从而有利于从总体上把握学习 OSG 及应用 OSG 开发的方向。

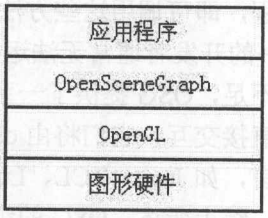

图 1-1　OSG 层次结构图

OSG 主要包括 4 个库，下面分别进行介绍。

（1）OSG 核心库（Core Library）

核心库是 OSG 的核心，也是其存在且不断得到发展的根本原因。它的主要功能就是实现最核心的场景数据库的组织和管理、对场景图形的操作以及为外部数据库的导入提供接口等。它主要包括以下 4 个库。

- ☑ osg 库：基本数据类，负责提供基本场景图类，构建场景图形节点，如节点类、状态类、绘制类、向量和矩阵数学计算以及一般的数据类。同时，它包含一些程序所需要的特定功能类，如命令行解析和错误调试信息等。
- ☑ osgUtil 库：工具类库，提供通用的公用类，用于操作场景图形及内容，如更新、裁减、遍历、数据统计及场景图的一些优化。
- ☑ osgDB 库：数据的读写库，负责提供场景中数据的读写工作，提供了一个文件工具类。注意，OSG 中场景图管理是通过遍历场景图层次结构来完成大部分的数据处理工作的。
- ☑ osgViewer 库：是在 OSG 2.0 后逐步发展稳定的一个视窗管理库，可以集中各种窗体系统，提

供 OSG 与各种 GUI 的结合。因此，它是跨平台的 3D 管理窗口库。

（2）OSG 工具库（NodeKit）

OSG 工具库主要是对 OSG 核心库中 osg 库的扩充，是对 OSG 核心库的一个补充，它实现了一些特定的功能。它主要包括以下 6 个库。

- ☑ osgFX 库：特殊效果节点工具，用于渲染特效节点，包括异性光照特效（osgFX::Anisotropic Lighting）、凹凸贴图特效（osgFX::BumpMapping）、卡通渲染特效（osgFX::Cartoon）、刻线特效（osgFX::Scribe）和立方图镜面高光特效（osgFX::SpecularHighlights）等。
- ☑ osgParticle 库：粒子系统的节点工具，用于模拟各种天气或者自然现象效果，如雨效、雪效和爆炸模拟等。
- ☑ osgSim 库：虚拟仿真效果的节点工具，用于特殊渲染，如地形高程图、光点节点和 DOF 变换节点等。
- ☑ osgTerrain 库：生成地形数据的节点工具，用于渲染高程数据，如 TIF、IMAGE 和 DEM 等各种高程数据格式。注意，OSG 通过一个开源库 GDAL 读取这些高程数据。
- ☑ osgText 库：文字节点工具，用于向场景中添加文字信息，它完全支持 TrueType 字体。
- ☑ osgShadow 库：阴影节点工具，用于向场景中添加实时阴影，提高场景渲染的真实性。

（3）OSG 插件库

OSG 插件库是 OSG 的一个非常重要的特点。通过各种第三方库的支持，OSG 能够直接或间接地导入 3D 模型或图片等场景数据，可以省去大量绘制图形的工作，从而极大地方便了开发者。具体支持的各种数据格式可参见第 5 章。

（4）OSG 内省库（osgIntrospection）

OSG 内省库提供了一个与语言无关的运行程序接口，确保了 OSG 可以在更多的环境下运行。osgIntrospection 库允许软件系统使用反射式和自省式的编程范例与 OSG 交互。应用程序和软件可以使用 osgIntrospection 库和方法迭代 OSG 的类型、枚举量和方法，并且不需要了解 OSG 编译和链接时的过程，即可调用这些方法。Smalltalk 和 Objective-C 等语言包括了内建的反射式和自省式支持，但是 C++的开发者通常无法运用这些特性，因为 C++并未保留必要的元数（metadata）。为了弥补 C++的这一不足，OSG 提供了一系列自动生成的、从 OSG 源代码创建的封装库。用户程序不需要与 OSG 的封装直接交互，它们将由 osgIntrospection 整体进行管理。由于 osgIntrospection 及其封装的结果，许多的语言，如 Java、TCL、Lua 和 Python，都可以与 OSG 进行交互。

综上所述，OSG 的组成模块如图 1-2 所示。

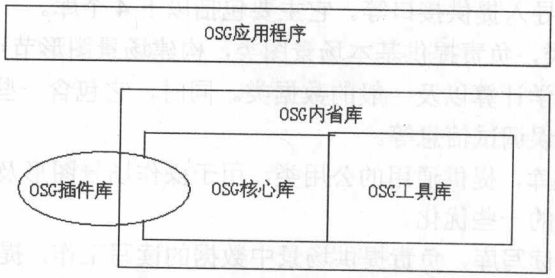

图 1-2　OSG 组成模块图

1.2　开发的预备知识

本书为准备学习，并使用 OSG 进行程序开发的开发者编写。本书主要介绍实用 OSG 函数及其应用，在阅读此书前，读者应该有一定的 C++开发经验。OSG 是一个 C++的 API 库，同时也是对 OpenGL 的底层封装，所以读者还需先了解 OpenGL 的渲染流程。

OSG 对于标准模板库（STL）的运用十分广泛，因此，读者在阅读本书前应当对 STL 容器，特别是列表（list）、向量组（vector）和映射（map）有较深的理解。如果读者对于设计模式（design patterns）也有一定的认识和理解，将对 OSG 的学习大有裨益，不过并非是必要的。

在学习使用 OSG 编程之前，读者有必要首先熟悉 3D 图形学。对于本书而言，读者需要熟悉 OpenGL 编程接口，而 OpenGL 是标准的跨平台底层 3D 图形 API 函数库。对于 3D 图形学，希望读者能够理解对不同类型的坐标空间熟练应用笛卡儿三维坐标系来表示三维几何数据。同时，读者需要了解一些基本的图形术语概念，不过并不需要对底层图形硬件的实现有很深入的认识和理解。

读者最好还要具备一定的线性代数知识，熟悉采用向量表示三维坐标的方法以及渲染过程中采用矩阵变换进行图形系统的相关操作，因此，还有必要了解矩阵运算的有关知识。

每个人的编程风格都是不一样的，下面简单介绍一下本书中的一些编程规范。

1．命名规范

本书使用标准的 C++命名规则，编写一个子模块或派生类时，遵循其基类或整体模块的命名风格，保持命名风格在整个模块中的统一性。标识符采用英文单词或其组合，直观且可以拼读，望文知意。

2．编程风格

OSG 中提供的智能指针可以方便管理内存，防止内存的泄露。但建议初学时不要写成指针的方式，以避免不必要的错误或异常发生。例如：

```
osg::Geode* geode = new osg::Geode();              //指针的写法
osg::ref_ptr<Geode> geode = new osg::Geode();      //建议读者的写法
```

当然，如果读者是熟练的 C++开发人员，可以根据自己的爱好选择。注意，在一个大型系统的开发过程中，使用指针或不使用应当统一，以避免带来不必要的麻烦。

笔者还喜欢对头文件进行分类处理，以方便对头文件进行管理及操作。例如：

```
#include <osg/Node>
#include <osg/Geode>
#include <osg/Group>

#include <osgDB/ReadFile>
#include <osgDB/WriteFile>
```

这种写法不是固定的，只是个人偏好，但这种写法将程序开发用到的库放在一起，可以做到一目了然，不容易造成混乱。读者可以根据自己的编程风格自由选择。

1.3 OSG 的安装及编译

接下来读者将正式开始接触并使用 OSG。下面首先向读者介绍一些 OSG 安装及编译的一些知识，这些也是开源引擎使用的一些基本技法。在以后的开发过程中，读者可能会用到很多其他的基于 OSG 开发的开源引擎，这些基本技法可以帮助读者学习、提高，同时，它将贯彻读者学习的整个过程。

1.3.1 OSG 的获取及安装

OpenSceneGraph 源代码的获取方式比较简单，只需要读者打开 IE 浏览器，在地址栏输入“http://www.openscenegraph.org”，然后单击 Downloads 下的 Current Release（最新稳定版本）或者 Developer Releases（开发版）超链接，在打开的网页中单击相应的超链接进行下载即可。如果读者机器上没有浏览器或者读者懒得上网找的话，需要下载一个签出源文件的工具 tortoisewin32svn.exe，正确且完全安装以后，新建一个空的文件夹，右击，在弹出的快捷菜单中选择 SVN CheckOut 命令（这样签出的源代码可以保持随时更新），打开 Checkout 对话框，如图 1-3 所示，在 URL of repository 下拉列表框中输入 SVN 地址“http://www.openscenegraph.org/svn/osg/OpenSceneGraph/tags/OpenSceneGraph-2.7.4”，然后单击 OK 按钮，读者将立刻得到 OSG 的最新源文件。

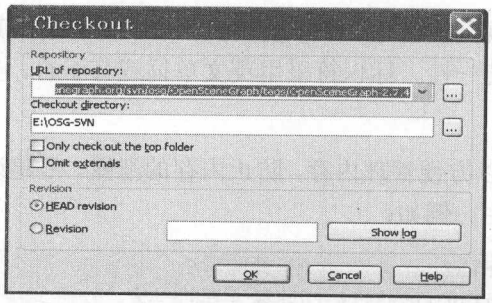

图 1-3 SVN 源码签出

获取了源代码以后，还有一项非常艰巨的工程，即编译 OSG。对于 OSG 初学者来说，编译 OSG 是一件比较痛苦的事情，但当学习完 1.3.2 节的内容后，读者会觉得编译 OSG 是一件非常有趣且有挑战性的任务。

OSG 的安装比较简单。OSG 中国官方网站（http://www.OsgChina.org）会在官方发布最新的 OSG 稳定版本后第一时间制作一个安装包，该安装包包含 OSG 常用的插件库（基于 VS2005SP1 或者 VS2008，详细的信息见安装包说明）。OSG 2.2 以前的安装包都由 flmn 制作，OSG 2.3x 系列的安装包由 FreeSouth 制作，OSG 2.4 系列以后的安装包由笔者制作并发布。读者可登录 OSG 中国官方讨论区（http://bbs.OsgChina.org）下载 OSG 安装包。

在论坛可以下载以往所有的安装包，下载完毕后，读者可以直接进行安装，安装完毕后，在 cmd 中输入如下命令：

```
osgVersion      //显示当前版本信息
OpenSceneGraph Library 2.6.0      //输出信息
```

然后输入“osglogo”，会出现如图 1-4 所示的渲染效果，即表示安装成功。

图 1-4　osglogo 运行截图

1.3.2　编译 OSG

编译 OSG 确实是一件非常繁琐的工作，所以，读者需要静下心，耐心地一步一步慢慢来。作为一个 OSG 开发者，编译 OSG 也是“家常便饭”。

编译 OSG 需要使用第 1.3.1 节中获取的源码，同时需要使用一个新工具——CMake。CMake 是一个跨平台的安装（编译）工具，可以用简单的语句来描述所有平台的安装（编译过程），它能够输出各种格式的 makefile 或者 project 文件，能测试编译器所支持的 C++特性，类似 UNIX 下的 auotconf（这里介绍这么多就足够了，如果想了解更多，请到百度搜索学习）。读者还需要下载一些插件库，如果读者使用的是 VS2005SP1，官方提供了编译好的插件库；如果不是的话，不幸地告诉读者，您还得从网上把插件一个一个下载下来，慢慢编译。下面讲解直接使用官方的插件库编译 OSG 的方法，具体操作步骤如下：

（1）解压源文件，其中有一个 CMakeLists.txt 文件，如图 1-5 所示。

（2）打开 CMake，把 CMakeLists.txt 文件拖到 CMake 中，会出现如图 1-6 所示的界面。单击 Configure 按钮，选择开发环境，它会搜索、检测出当前读者的开发环境。

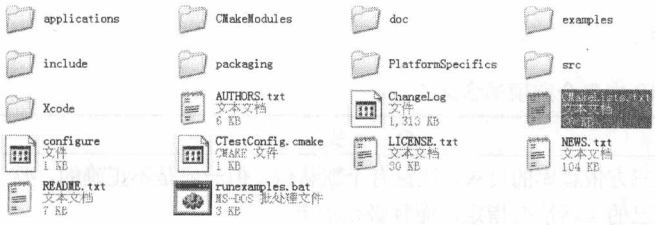

图 1-5　源文件

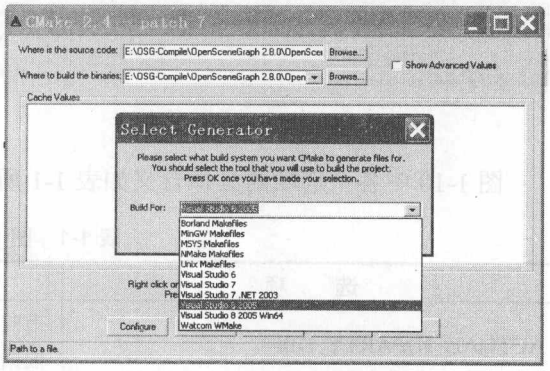

图 1-6　选择开发环境

（3）待 CMake 自动配置完成后，出现如图 1-7 所示的配置选项设置窗口。

（4）手动来配置。需要的手动配置的选项有下面 3 项。

☑　第三方插件库 ACTUAL_3DPARTY_DIR，选择相应的目录即可，如图 1-8 所示。

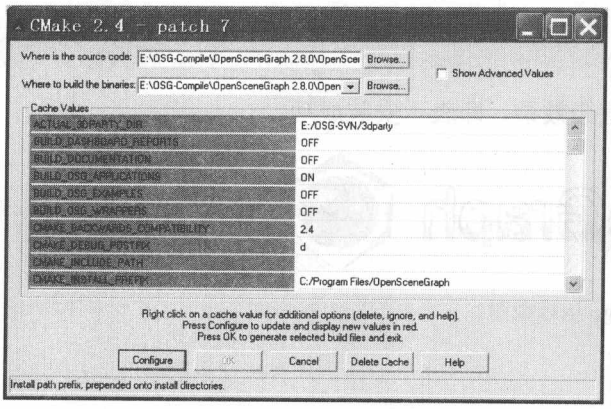

图 1-7　配置选项设置

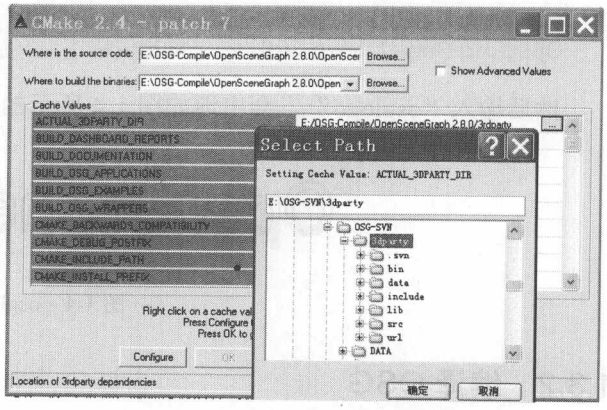

图 1-8　选择第三方库目录

☑ 编译 OSG 的例子，BUILD_OSG_EXAMPLES 由 OFF 设置成 ON，如图 1-9 所示。

☑ 一些插件的头文件，如字体的 fbuild2.h，可能在编译时找不到头文件，因此需要手动指定。待配置完成以后，单击 Configure 按钮，看到不再有红色的选项后，单击 OK 按钮，如图 1-10 所示。CMake 的配置告一段落，随即出现 C++解决方案。

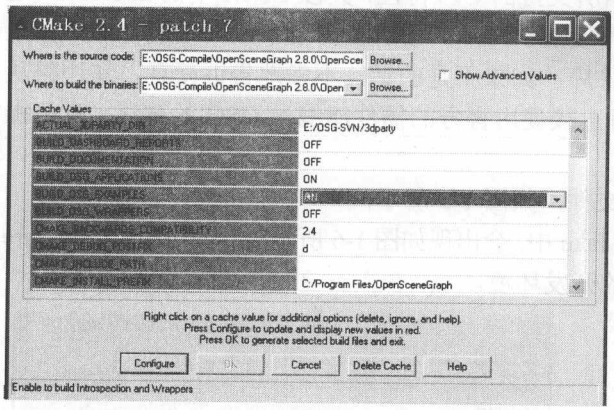

图 1-9　选择例子编译

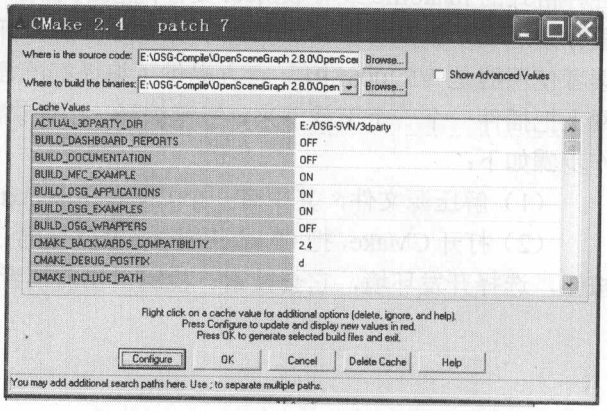

图 1-10　正确配置各种选项

图 1-10 中各个选项的具体含义如表 1-1 所示。

表 1-1　图 1-10 中各个选项的含义

选　项	含　义
ACTUAL_3DPARTY_DIR	指向第三方依赖库的目录，注意有个默认值，但一般是不正确的，要根据自己的实际情况指定，而且必须指定
BUILD_OSG_APPLICATIONS	是否编译 OSG 的可执行文件，里面包含了 4 个极其重要的程序，分别是 osgarchive、osgconv、osgversion 和 osgviewer，在后面会讲解它们的使用方法
BUILD_OSG_EXAMPLES	是否编译 OSG 的默认例子，一定要选择 ON，因为例子很重要
BUILD_OSG_PLUGINS	是否编译 OSG 的插件，要选择 ON
BUILD_OSG_WRAPPERS	是否编译 OSG 的内省库