SAMS

• 游戏开发经典丛书 •

(美) Tom Miller 著
敖富江 译

3D游戏编程入门经典

Beginning 3D Game Programming

GAME Beginning Programming

清华大学出版社

3D 游戏编程入门经典

(美) Tom Miller 著

敖富江 译

清华大学出版社

北 京

EISBN: 0-672-32661-2

北京市版权局著作权合同登记号 图字：01-2005-1074

图书在版编目(CIP)数据

3D 游戏编程入门经典/(美)米勒(Miller, T.)著；敖富江译. —北京：清华大学出版社，2006.3
书名原书：Beginning 3D Game Programming
ISBN 7-302-12167-2

Ⅰ.3… Ⅱ.①米… ②敖… Ⅲ.三维—动画—游戏软件开发 Ⅳ.TP311.5

中国版本图书馆 CIP 数据核字(2005)第 139698 号

出 版 者：清华大学出版社 **地 址**：北京清华大学学研大厦
http://www.tup.com.cn **邮 编**：100084
社 总 机：010-62770175 **客户服务**：010-62776969
组稿编辑：曹 康
文稿编辑：王 黎
封面设计：康 博
版式设计：康 博
印 刷 者：北京密云胶印厂
装 订 者：北京市密云县京文制本装订厂
发 行 者：新华书店总店北京发行所
开 本：185×260 **印张**：23.5 **字数**：487 千字
版 次：2006 年 3 月第 1 版 2006 年 3 月第1 次印刷
书 号：ISBN 7-302-12167-2/TP·7845
印 数：1～4000
定 价：48.00 元

前　言

成为游戏开发人员的条件

我所遇到的每一个开发人员都至少在某一段时间想成为一个游戏开发人员。对于很多人来说，视频游戏不只是空闲时的一种爱好，他们完全被游戏所吸引。人们沉浸在这些虚拟世界中，常常梦想由自己创建如此神奇的天地。

不要被美丽的图形、奇妙的故事情节和感人悦耳的音乐所愚弄，编写游戏是非常困难的工作，只有某些特定的开发人员才能够获得成功。除了所需要的技术天赋之外，一个优秀的游戏开发人员应当拥有其他一些技能，例如，您首先是一个游戏爱好者。如果不是游戏爱好者，则不可能编写出伟大的游戏，这一条件使得游戏开发工作更具有挑战性。

成为游戏开发人员确实不是一件容易的事情。如果没有经验，则不会被游戏开发公司聘用，并且当没有公司聘用时，将更难以获得经验。当前只有少量的课程以及某些学校专注于讲授游戏的开发。但是，入门的最好方式是制作一个样片(demo reel)。它能向您未来的老板展示您的能力和处理事情的方式。

通过本书，您将制作出一个引人注目的样片。

读者对象

我常常被问到："为什么每个人都想使用.NET Framework 编写游戏程序？"。其他的问题包括："NET Framework 不是只用于 Web 服务器应用程序的吗？"，"它不是很慢吗？"，等等。对于游戏开发人员(或者未来的游戏开发人员)来说，这些都是很重要的问题，但他们误解了.NET Framework。

.NET Framework 不是最新的 Web 服务器版本，也不是任何服务器组件的扩展。当然能够使用.NET Framework 创建强大的 Web 服务器应用程序，但是，这并不是它们的全部功能。.NET Framework 包括一种功能强大的客户端应用程序编程接口(API)以及 Managed DirectX，实际上.NET 开发人员需要掌握整个 DirectX API。利用它能够编写很多新的应用程序，包括游戏。如果认为.NET Framework 只能够编写服务器应用程序，则显得有一点单纯。您也可以利用它创建复杂的客户端应用程序。

关于.NET 的性能问题仍然存在，这些问题也很难简单描述。当引入一门"新的"语言或者运行库时，开发人员在采用之前，通常比较犹豫。不久以前，很多游戏仍

然是采用汇编语言编写的，因为游戏开发人员不相信 C 或者 C++语言足够快。.NET Framework 也遵循这个规则。在证明.NET Framework 的性能之前，游戏开发人员都会以一种怀疑的眼光看待它。在本书中，大量的游戏是使用.NET 运行库开发的。事实胜于雄辩，与其喋喋不休地讨论.NET 运行库的性能有多好，还不如让本书中一些真实的游戏来证明这一点。

为什么使用.NET Framework

任何曾经编写过 Windows 程序的人都(无论是否使用.NET Framework)能够认识到，即使使用 Win32 API 编写简单的 Windows 应用程序都比较困难。设计.NET Framework 的目的是提供一种较简单的方式来执行在 Windows 程序中一些常见的事情，并且为普通的开发人员自动处理很多问题，例如内存的管理。

利用.NET Framework，开发人员可以省去很多麻烦的任务，例如花费三天的时间寻找一个内存泄漏 bug，他们可以将一些功能直接添加到正在编写的游戏里面。很多情况下，从游戏中删除某些好的特性是因为需要花费大量的时间解决问题，而这些问题.NET Framework 已经为您解决。

.NET Framework 的另外一个令人感兴趣的特性是语言的中立性。只要所使用的功能兼容公共语言规范(Common Language Specification，CLS)，则可以采用任何能够使用 CLS 兼容功能的语言。过去，Visual Basic 开发人员转向使用严格的 C++进行编码时，可能比较困难。现在，Visual Basic .NET 开发人员能够较容易的转向 C#，因为这两种语言之间仅存在较小的语法差别。本书中的代码是使用 C#编写的。

本书的目的

编写本书的目的是为了满足游戏开发群体的需求，他们缺少关于本书主题方面的信息。由于不能找到所需要的信息，太多可能成为优秀游戏开发人员的程序员选择了退出。目前市场上的多数游戏开发书籍是关于 2D 图形的，它们是比较好的起点，但现在编写的多数游戏几乎全是 3D 图形的，并且现在的游戏爱好者也希望如此。并不是不需要继续编写 2D 游戏，但是如果那是游戏开发人员所能够做的所有，则他所做的可能毫无用处。

本书并不教导您如何去编写一个价值数百万美元的游戏。而是给出所有的工具和信息，让读者自己学会如何开发 3D 游戏。在阅读本书的过程中，将实现两个完整的 3D 游戏，阅读完本书后，读者将能够设计并实现自己的 3D 游戏，以使自己成为一名优秀的游戏开发人员。本书中最后的游戏将留给读者作为一个练习。

作者简介

Tom Miller 是 Managed DirectX API 的设计师和首席开发人员。自 1997 年开始，他工作于 Microsoft 公司。他最初在 Visual Basic 组工作，后来他喜爱上了游戏和游戏编程，因此进入了 DirectX 组。自 1999 年末，他在 DirectX 组工作，并且已经使得 DirectX API(和通常的游戏编程)为更多人所接受。他也编写了到目前为止最具权威的关于 Managed DirectX 库的书籍。

欢迎您的宝贵意见!

作为本书的读者，您同时还是本书最重要的批评家和评论员。我尊重您的意见，想知道我们在什么地方做得不错，在哪些地方还能够做得更好，您最想看到我们提供哪些方面的内容，以及您希望传递给我们的任何明智的言语。

我们非常欢迎您的评论。您可以给我们写电子邮件或者直接给我写信，阐明您喜欢或不喜欢本书的哪些方面——以及我们能够在哪些方面改进本书。

请您注意的是，我们不能在与本书主题相关的技术问题方面帮助您。但是，我们拥有一个用户服务小组，与本书相关的特定技术问题将转寄给他们。

信中务必包含本书的书名和作者，以及您的姓名、电子邮件地址和电话号码。我们将仔细阅读您的评论，并将它们与本书的作者和编辑共享。

我的电子邮件地址是：feedback@samspublishing.com

信件地址是：

Michael Stephens
Associate Publisher
Sams Publishing
800 East 96th Street
Indianapolis, IN 46240 USA

读者服务

关于本书和 Sams 出版社的其他更多信息，请访问我们的网站 http://www.samspublishing.com。在搜索框中输入 ISBN 号(不包括连字号)或者书名，就可以查到您想看的书。

也可访问 http://www.tupwk.com.cn/downpage/，下载相关代码。

目　录

第 I 部分　Microsoft .NET 简介

第 II 部分　图形和游戏 1 的介绍

第 III 部分　基本的数学规则

第 IV 部分　间接图形、对等网、游戏 2

第 VI 部分 附 录

第 I 部分

Microsoft .NET 简介

第 1 章　游戏开发和托管代码

第 1 章　游戏开发和托管代码

如果熟悉了如何利用 CLR(公共语言运行库)编写代码后，在面临选择开发语言时，您可能已经知道了您的选择。在 Visual Studio .NET 产品的最新版本中，当编写托管代码时，可以使用 4 种语言：C#、Visual Basic .NET、Managed C++和 J#。此外还可以使用从 VisualStudio .NET 产品之外的第三方销售商处获得的其他语言，例如 COBOL 或者 FORTRAN。

尽管本书中将讨论的概念可以很容易地移植到任何完全兼容 CLS(通用语言规范)的语言，但实际的代码将仅包含所提到的前两种语言：即 C#和 Visual Basic. NET。本书中将仅使用 C#代码。您可以从 http://www.tupwk.com.cn/downpage 中下载本书配套的安装文件，获取相关代码。

在本章中，您将学习到：

- 定义.NET
- 托管代码
- 使用 Visual Studio .NET IDE
- 在命令行中编译托管代码
- 开发人员
- 游戏开发过程
- 工具

1.1　什么是.NET?

自从 Microsoft 公司宣布并发行.NET 之后，人们一直在尝试指出这种新“事物”到底是什么。根据 Microsoft 公司的市场活动，人们知道它将对计算产生革命性作用。这是一个很宏远的目标，现在断言它是否能够完成目标还太早。但是，它正在一步步地向此目标努力。

当人们讨论.NET 时，无法确定他们正在讨论.NET 的哪个部分。Microsoft 公司发行的其他“产品”或“思想”都不具有如此多的不同形式。紧随.NET 名字的是众多的产品、服务，甚至是概念，因此指出.NET 实际上是什么，是非常困难的。

当本书中讨论.NET 时，它指从.NET FrameworkSDK 中可获得的新的开发语言和运行库。该 SDK 包含.NET 运行库。而.NET 运行库包含运行为.NET 环境编写的

应用程序所需要的所有东西。可以认为.NET 运行库由几部分组成。CLR 的部件驻留在 GAC(Global Assembly Cache)中。也包括 Microsoft .NET 语言的编译器(C#、VB .NET、VJ#等等)。可以在图 1-1 中看到 GAC。

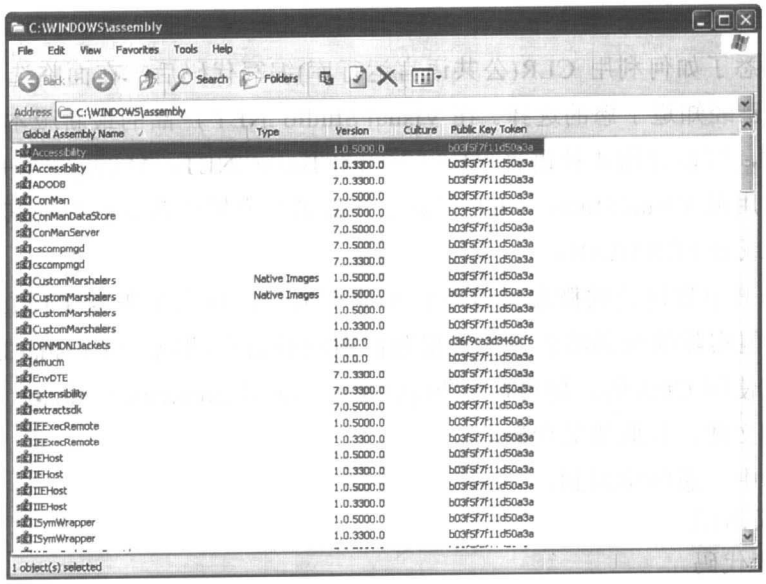

图 1-1 GAC

人们对运行.NET 代码的最常见误解之一是，代码是“解释执行的”，像 Java 代码或者老的 Visual Basic 运行库一样。事实上，为.NET 编写的代码在执行前首先被编译。当编译.NET 应用程序时，它被编译为一种中间语言(IL，intermediary language)。这种 IL 实际上存储在可执行文件中或者已经创建的库中。

IL 可能在两个位置中的某一处被编译为本机代码(native code)。在安装代码时，可以执行一个称为 ngen(native generation，即本机生成器)的进程。它将 IL 直接编译为本机代码，并将所编译的本机代码存储在 GAC 中的特定位置——本机程序集缓存(native assembly cache)中。假设在安装时没有编译代码，则代码在第一次执行前必须被编译。在应用程序启动期间，.NET 运行库中一种称为 JIT(Just In Time)编译器的特殊功能在后台执行编译工作。

在后一种情形中，因为发生在后台的编译工作，应用程序的启动时间将受到影响。当启动时间对应用程序非常重要时(例如正在编写游戏时)，确保在安装阶段包含 ngen 步骤是比较明智的。但是，在这期间无法进行某些优化，而如果利用 JIT 编译代码，则可以进行这些优化，因此如果启动时间不是很重要，则可以让.NET 运行库处理它所能够做的工作。

1.2 什么是托管代码

在本书中将经常提到托管代码。在全书中使用的 API 被称为 Managed 的 DirectX，.NET 语言常被称为托管语言。术语“托管”来源于.NET 运行库具有一个内置的内存管理器这一事实。

在“过去”(只是几年前)，使用 C 和 C++编写代码的开发人员不得不自己进行内存管理。当不再需要已分配的内存空间时，必须将其释放，除非希望该内存被“泄漏”，内存泄漏将带来严重的性能问题。更糟糕的是因为直接处理指针，而它很容易破坏项目正在使用的内存。在很多情况下，这将导致很长时间的故障调试，因为通常实际看到出错的地方并不是内存初始被破坏的地方。

人们认为 C 和 C++语言难于掌握，主要是因为具有很多这种类型的问题。许多开发人员不愿意尝试 C 和 C++，也是因为这个原因，他们尝试使用其他没有这些令人头痛问题的高级语言，例如 Visual Basic。尽管这些新语言具有易用易学的优点，但也具有一些缺点。它们的性能无法与 C 和 C++语言相比，在大多数情况下显得特别慢。另外，因为底层操作系统是使用 C++开发的，所以这些语言难以实现 C++的所有功能。尽管可以使用它们处理很多非常好的工作，但是如果想要获得操作系统的所有性能和优势，只能依靠自己。

与.NET 运行库的第一个版本相比，.NET 的大多数内容都已经改变了。Microsoft 公司几乎完全重新设计了一种新的 API，竭力确保开发人员关心的问题都会被解决。这种新的运行库必须易学易用，快速高效，并且不存在令人头痛的内存管理问题。在本书中，将看到.NET 在这些方面的好处。

1.3 使用 Microsoft Visual Studio .NET 2003 IDE 编写代码

提示：

*本书设定在 Visual Studio .NET IDE 中编写代码。这不是使用.NET 编写游戏的需求，也不是使用.NET 本身的需求，它是本书所选择的 IDE。*图 1-2 显示了 Visual Studio .NET 2003 IDE。

该 IDE 提供了编写.NET 应用程序所需要的所有工具。它不仅包括编写代码所需要的编辑器，而且还有其他大量功能，使得.NET 应用程序的开发变得容易。它的设计使得您能够方便地创建丰富的内容，如 Windows 应用程序。它也具有一个内置的编译器和调试器，并且无缝集成了所有功能。本书设定使用这种 IDE 进行开发。