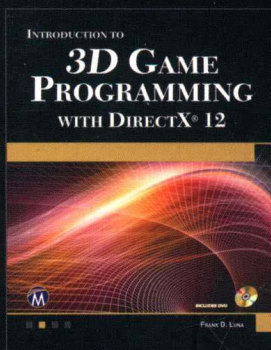




DirectX 12龙书，3D游戏开发宝典



DirectX 12 3D 游戏开发实战



[美] 弗兰克·D.卢娜 (Frank D. Luna) 著

王陈 译



DirectX 12 3D

游戏开发实战

[美] 弗兰克·D.卢娜 (Frank D. Luna) 著

王陈 译

人民邮电出版社

北京

图书在版编目 (C I P) 数据

DirectX 12 3D 游戏开发实战 / (美) 弗兰克·D. 卢娜 (Frank D. Luna) 著 ; 王陈译. -- 北京 : 人民邮电出版社, 2019. 1

ISBN 978-7-115-47921-1

I. ①D… II. ①弗… ②王… III. ①DirectX软件—程序设计②游戏程序—程序设计 IV. ①TP317

中国版本图书馆CIP数据核字(2018)第032934号

版权声明

Simplified Chinese language edition copyright ©2018 by Post & Telecom Press.

All rights reserved.

Introduction to 3D Game Programming with DirectX 12 by Frank D. Luna.

Copyright ©2016 Mercury Learning and Information, Inc.

本书中文简体版由 Mercury Learning and Information 公司授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式或任何手段复制和传播。

版权所有, 侵权必究。

-
- ◆ 著 [美] 弗兰克·D. 卢娜 (Frank D. Luna)
 - 译 王 陈
 - 责任编辑 罗子超
 - 责任印制 焦志炜

 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 三河市中晟雅豪印务有限公司印刷

 - ◆ 开本: 800×1000 1/16
 - 印张: 47.75
 - 字数: 1186 千字 2019 年 1 月第 1 版
 - 印数: 1-2 000 册 2019 年 1 月河北第 1 次印刷
- 著作权合同登记号 图字: 01-2016-3754 号
-

定价: 148.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广登字 20170147 号

内容提要

Direct3D 是微软公司 DirectX SDK 集成开发包中的重要组成部分,是编写高性能 3D 图形应用程序的渲染库,适用于多媒体、娱乐、即时 3D 动画等广泛和实用的 3D 图形计算领域。

本书围绕交互式计算机图形学这一主题展开,着重介绍 Direct3D 的基础知识和着色器编程的方法,并介绍了如何利用 Direct3D 来实现各种有趣的技术与特效,旨在为读者学习更高级的图形技术奠定坚实的基础。本书包括 3 部分内容。第一部分介绍必备的数学知识,涵盖向量代数、矩阵代数和变换等内容。这是贯穿全书的数学工具,是读者需要掌握的基础内容。第二部分重点介绍 Direct3D 的基础知识,展示用 Direct3D 来实现绘图任务的基本概念与技术,如渲染流水线、纹理贴图、混合、曲面细分等。第三部分则利用 Direct3D 来实现各种有趣的特效,如实例化与视锥体剔除、阴影贴图、环境光遮蔽等。

本书适合希望通过 Direct3D 来学习 3D 编程的 C++ 中级程序员阅读,也可供已对 Direct3D 有一定了解或具有非 DirectX API 使用经验的 3D 程序员参考。

谨以此书献给我的侄辈们——
Marrick、Hans、Max、Anna、Augustus、Presley 以及 Elyse

前言

Direct3D 12 是一款为运行在现代图形硬件上的各种 Windows 10 平台（Windows 桌面版、手机版和 Xbox One）编写高性能 3D 图形应用程序的渲染库。Direct3D 也是一种底层库，这也就意味着此种应用程序接口（API）与其下层控制的图形硬件模块关系更为紧密^①。Direct3D 的主要用户大多来自游戏产业，他们驾驭 Direct3D 来构建更加高端的渲染引擎。同时，它亦应用于如医药产业、科学可视化以及虚拟建筑漫游等行业，用来实现高性能的 3D 图形交互功能。另外，由于当今每一部新的个人电脑都已配备了现代图形设备，因此，非 3D 应用也开始逐步把计算密集型的工作移交至显卡来执行，以充分发挥其中 GPU（Graphics Processing Unit，图形处理器）的计算能力。这就是众所周知的 GPU 通用计算（general purpose GPU computing）技术。对此，Direct3D 也提供了用于编写 GPU 通用计算程序的计算着色器 API。尽管 Direct3D 12 程序通常以原生的 C++ 语言进行编写，但 SharpDX 团队正在致力于 .NET 包装器版的开发，因此，用户也可以从托管应用程序中来访问这一强大的 3D 图形 API。

本书围绕交互式计算机图形学这个主题展开，关注于通过 Direct3D 12 来进行游戏的开发。读者将从中学到 Direct3D 的基础知识以及着色器编程的方法。阅读完本书之后，读者就可以继续学习更加高级的图形技术了。本书共分为 3 个主要部分。第一部分讲解了本书后续要用到的数学知识。第二部分展示如何用 Direct3D 来实现基本绘图任务，例如初始化 Direct3D，定义 3D 几何图形，设置摄像机，光照，纹理，混合技术，模板技术，曲面细分技术，创建顶点、像素、几何图形以及计算着色器。第三部分则主要是利用 Direct3D 来实现各种有趣的技术与特效，例如动画角色网格、拾取技术、环境贴图、法线贴图、阴影贴图以及环境光遮蔽技术。

初学者最好按先后顺序通读全书。书中章节是按照由浅入深、逐步递进的顺序组织而成的。这样一来，读者便不会因过陡的学习曲线而如堕烟海。一般来讲，特定篇章中所用的技术与概念往往在之前的章节中有所交代。因此，读者最好在掌握了欲学习章节之前的所有内容后再继续前行。当然，有一定经验的读者可直接挑选感兴趣的部分进行阅读。

最后，部分读者可能会不禁琢磨：读完本书之后，究竟能够开发出何种类型的游戏来呢？这里对此给出的解释是：您最好亲自粗略地阅览此书，看看其中大概都在讲些什么内容。据此，基于本书所讲的技术知识再结合自己的聪明才智，至于能够开发出哪类游戏作品，想必这答案读者也就自会了然于胸了。

本书受众

本书主要适合以下 3 类读者：

1. 希望通过 Direct3D 最新版本来学习 3D 图形学编程的 C++ 中级程序员。

^① 尤其是到了 Direct3D 12，更像 Mantle 等 API 那样实现了前所未有的更底层的硬件抽象，削减驱动层的工作，转交给开发者负责，从而令图形的处理流程更加“智能”，使用起来犹如贴地飞行的“快感”。

2. 具有非 DirectX API（如 OpenGL）使用经验，并希望学习 DirectX3D 编程方面知识的 3D 程序员。
3. 具有一定的 DirectX3D 使用经验，并希望学习 DirectX3D 最新版本的程序员。

预备知识

需要强调的是，本书为重点介绍 DirectX3D 12、着色器编程以及 3D 游戏编程的读物，而并非是讨论一般计算机程序设计的读物。因此，读者需要具备下列预备知识：

1. 高中程度的数学知识，比如代数、三角学以及（数学）函数等。
2. Visual Studio 相关的使用技能，比如如何创建项目、为项目添加文件以及指定需要链接的外部库等。
3. 中级 C++ 编程技能以及数据结构知识，比如熟练地运用指针、数组、运算符重载、链表、继承、多态等。
4. 熟悉使用 Win32 API 进行 Windows 编程还是很有必要的，可谓是学习本书的基础。但这一条并非强制性要求，因为本书附录 A 中提供了 Win32 编程的相关入门知识。

需要配备的开发工具以及硬件环境

下面是进行 DirectX3D 12 编程的必备条件：

1. Windows 10 操作系统。
2. Visual Studio 2015 开发环境或其后续版本。
3. 一款支持 DirectX3D 12 的显卡（本书中的演示程序都已通过 Geforce GTX 760 平台的测试）。

使用 DirectX SDK 文档以及 SDK 示例

Direct3D 是一种规模庞大的 API，将其所有的细节都在一本书中体现是不切实际的。因此，为了获得更为深入的 API 信息，学习 DirectX SDK^① 文档的查阅方法势在必行。DirectX SDK 在 MSDN 上的最新文档为《Direct3D 12 Programming Guide》，即《Direct3D 12 编程指南》。

图 1 所示的是在线文档的截图。

DirectX 文档涵盖了 DirectX API 的方方面面，因此，它是一种不可或缺的参考资料。然而，由于此文档对预备知识的讲解并不深入且假设读者对此有一定认识，因而导致它无法成为初学者最佳的学习工

① DirectX 包罗系列与多媒体以及游戏开发有关的 API，因此 DirectX3D 只是 DirectX 的一个子集。详细信息请见《DirectX Graphics and Gaming》(ee663274)。本书则侧重 DirectX3D 的讲解。

具。但是，随着 DirectX 每个新版本的发布，该文档也在日益完善中。

换言之，这个文档主要还是用作参考。假设用户碰到一个与 DirectX 有关的数据类型或函数，如函数 `ID3D12Device::CreateCommittedResource`，并希望获取更多与之相关的信息，就可以方便地在该文档中搜索它，比如本示例中的函数（见图 2），以得到更为细致的描述。

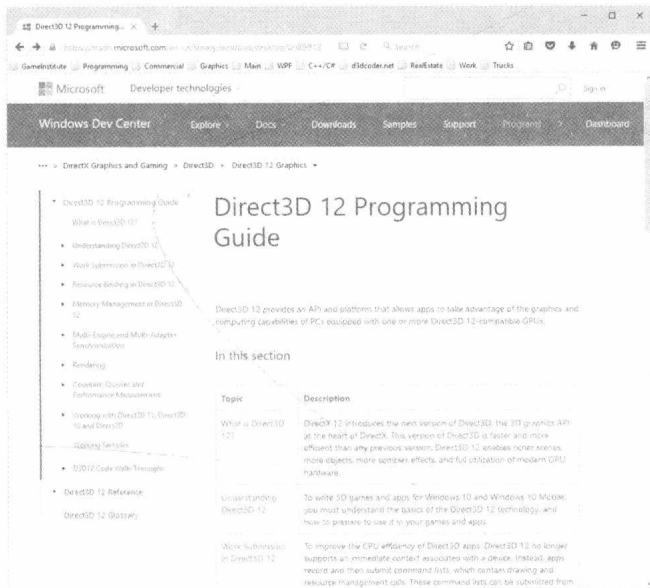


图 1 DirectX 文档中的《Direct3D 12 编程指南》

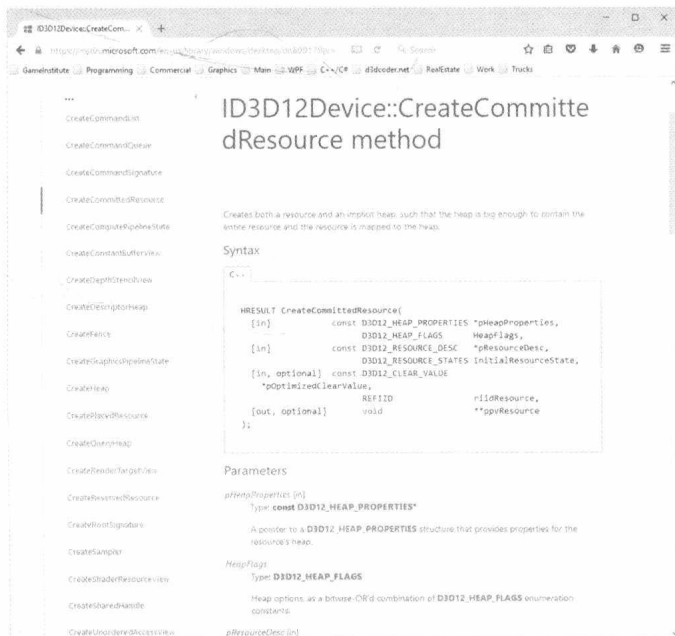


图 2 获取函数的相关文档

Note 在本书中，我们会不时地指导读者去阅览文档以获取更多的有关细节。

我们还建议读者研究一下官方提供的 Direct3D 12 演示程序。

微软官方可能还会在此陆续增添更多的例程。除此之外，读者还可以去 NVIDIA、AMD 以及 Intel 的官方网站上查找与 Direct3D 12 有关的示例。

明确学习目的

尽管我们努力遵循 Direct3D 12 的最佳实践，力图写出高效的代码，但本书中每个样例的主要目标还是为了阐述 Direct3D 中的基本概念以及演示图形编程技术。应当明确的是，写出最优代码并非本书最终目的，而且过分优化还可能导致原本意图明晰的代码变得含混不清，反而适得其反。希望读者将这一点铭记于心，尤其是在将书中例程代码合并到自己的项目中时，因此在此过程中，您可能为了追求程序更高的效率而重构代码。再者，为了把注意力集中在 Direct3D API 上，我们还在 Direct3D 之上构建了一层轻量级的框架。这就意味着我们很可能在源代码中，以硬编码的数值与定义其他内容的方式来令程序得以运行。类似地，在大型的 3D 应用程序中，可能要在 Direct3D 的基础之上实现一款渲染引擎。但本书的主旨却是 Direct3D API，而非设计渲染引擎。

例程与在线补充材料

读者可以登录本书的网站 (www.d3dcoder.net 和 www.merclearning.com)，以获取本书相关材料。在前者中，读者可以找到本书内所有例程的完整源代码以及项目文件。也可通过异步社区本书页面获取 (www.epubit.com)。在大多数情况下，DirectX 程序往往比较庞大，以至于不宜全部列入书中。因此，只得在书中嵌入与所讲内容密切相关的代码片段。为此，我们极力建议读者在学习相关的例程代码时去一睹它的全貌（为了便于读者学习，我们已将演示程序的规模尽量减小）。一般说来，在阅读过特定章节，并研究完所附演示代码后，读者应当能够自行独立地实现该章节中所述的例程。但事实上，一种更快捷的学习方法是在参考书籍和示例代码的同时，尝试着以自己的方式实现相关程序。

通过 Visual Studio 2015 安装演示项目^①

通过双击项目文件 (.vcxproj) 或解决方案文件 (.sln) 就可以方便地打开本书的演示程序。接下来，我们将详述如何通过 Visual Studio 2015 (VS15) 以本书的例程框架从头开始创建并构建一个项目。在此，

① 采用 Visual Studio 2017 的读者可以参考《Visual Studio 中的使用 C++ 的 DirectX 游戏开发》一文。

我们以第 6 章中的“Box”（立方体）演示程序为例。

下载本书的源代码

首先，读者需要下载本书所用的源代码并将其保存在硬盘的某个文件夹之中。为了便于讨论，假设这个文件夹的路径为 C:\d3d12book。在这里可以看到一系列文件夹，其中含有对应章节的例程项目。读者可能会注意到有个名为“Common”的文件夹，其中包含所有演示项目中都要复用的公共代码。现在便可以在源代码文件夹中新建一个文件夹，用来存放我们自己的例程，例如 C:\d3d12book\MyDemos。随后，我们将基于本书中的例程框架在该文件夹中创建一个新的项目。

注意

Note

事实上，读者自己设置的目录结构大可不必如此，这只不过是本书例程的结构而已。如果读者希望按自己的意愿来设置源代码文件，可以将演示项目放在任何地方，只要使 Visual Studio 能找到 Common 目录中的源代码即可。

创建一个 Win32 项目

首先运行 VS15，接着在主菜单中依次选择 **File**（文件）→**New**（新建）→**Project**（项目），如图 3 所示。

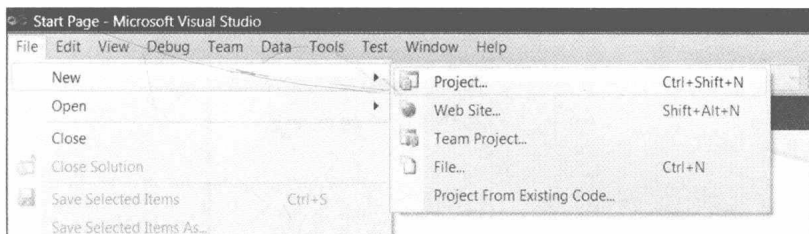


图 3 创建一个新项目

在弹出的 **New Project**（新项目）对话框（如图 4 所示）左侧 **Visual C++** 项目类型的树形控件中选择 **Visual C++**→**Win32**，再于右侧选择 **Win32 Project**（Win32 项目）。接下来，给项目起个名称，并指定项目文件夹的保存位置。别忘了取消默认选中的 **Create directory for solution**（为解决方案创建目录）复选框。随后单击 **OK**（确定）按钮。

接着，又会弹出一个新的对话框。其左侧有 **Overview**（概述）和 **Application Settings**（应用程序设置）两个选项。选择 **Application Settings**，便会出现如图 5 所示的对话框。在这里，需要确保选择 **Windows application**（Windows 应用程序）选项和 **Empty project**（空项目）复选框，之后再单击 **Finish**（完成）按钮。至此，我们已成功创建了一个空的 Win32 项目，但在构建 DirectX 项目例程之前，我们还有一些事情需要做。

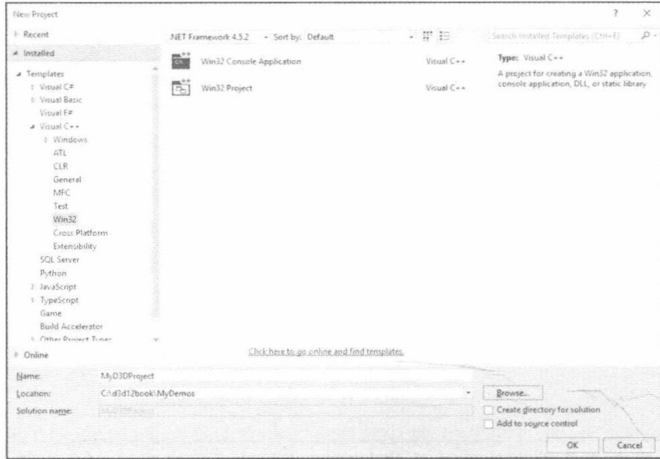


图 4 新项目的相关设置

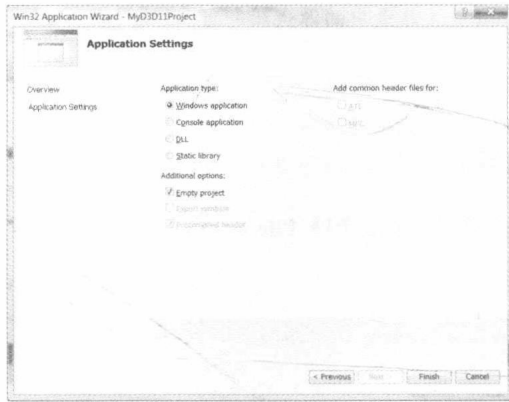


图 5 应用程序的相关设置

链接 DirectX 库

通过在源代码文件 `Common/d3dApp.h` 中使用 `#pragma` 预处理指令来链接所需的库文件，如：

```
// 链接所需的 d3d12 库
#pragma comment(lib, "d3dcompiler.lib")
#pragma comment(lib, "D3D12.lib")
#pragma comment(lib, "dxgi.lib")
```

对于创建演示程序而言，该预处理指令使我们免于打开项目属性页面并在连接器配置项下指定附加依赖库。

添加源代码并构建项目

至此，项目已经配置完成。现在来为它添加源代码并对其进行构建。首先，将“Box”演示程序的

源代码 `BoxApp.cpp` 以及 `Shaders` 文件夹（位于 `d3d12book\Chapter 6 Drawing in Direct3D\Box`）复制到工程目录之中。

待复制完上述文件之后，我们以下列步骤来将源代码添加到当前的项目之中。

1. 右键单击解决方案资源管理器下的项目名称，在弹出的下拉菜单中依次选择 **Add(添加)**→**Existing Item**（现有项），将文件 `BoxApp.cpp` 添加到项目中。
2. 右键单击解决方案资源管理器下的项目名称，在弹出的下拉菜单中逐步选择 **Add**→**Existing Item**，前往读者放置本书 **Common** 文件夹的位置，并将此文件夹中所有的 `.h/.cpp` 文件都添加到项目之中。现在，方案资源管理器看起来应当与图 6 相同。
3. 再次右键单击解决方案资源管理器下的项目名称，从菜单中选择 **Properties**（属性）。再从 **Configuration Properties**（配置属性）→**General**（常规）选项卡下，将 **Target Platform Version**（目标平台版本）设置为版本 **10.x**^①，以令目标平台为 Windows 10。接着单击 **Apply**（应用）按钮。
4. 大功告成！源代码文件现都已位于项目之中，读者可以在主菜单中选择 **Debug**（调试）→**Start Debugging**（开始调试）进行编译、链接以及执行该演示程序。应用程序的执行效果应当与图 7 所示的一致。

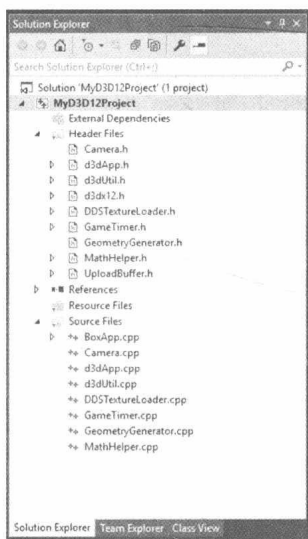


图 6 添加“Box”例程所需源代码之后的解决方案资源管理器

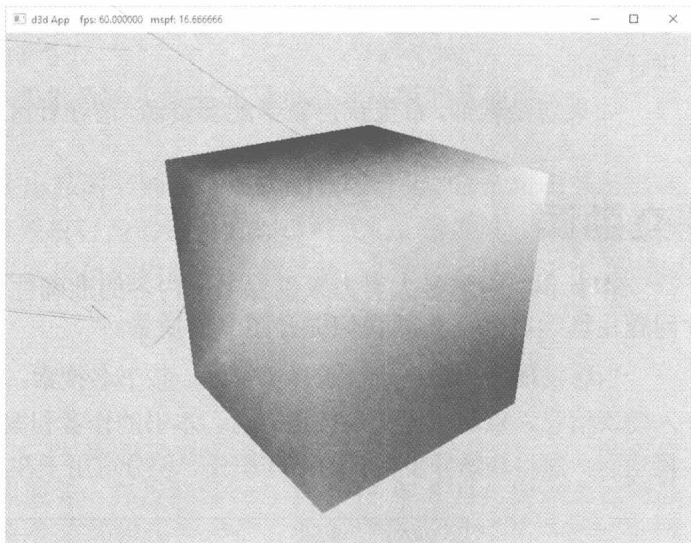


图 7 “Box”演示程序的效果

注意

Note

`Common` 目录下的大量代码都是构建本书例程的基石。所以，建议读者先不必忙于查看这些代码。待读到了本书中与之相关的章节后，再研究它们也不迟。

① 其中的“x”对应于构建项目时所采用的具体 SDK 版本。

资源与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供相关资源和后续服务。

配套资源

本书提供如下资源：

- 本书配套源代码；
- 练习媒体素材；
- 部分练习题解答；
- 书中图片资源。

要获得以上配套资源，请在异步社区本书页面中点击 **配套资源**，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

如果您是教师，希望获得教学配套资源，请在社区本书页面中直接联系本书的责任编辑。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，点击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。

The screenshot shows a web form titled "提交勘误" (Submit勘误) with three tabs: "详细信息" (Detailed Information), "写书评" (Write a Review), and "提交勘误" (Submit勘误). The form contains three input fields: "页码:" (Page Number), "页内位置 (行数):" (Page Position (Line Number)), and "勘误内容:" (勘误内容:). Below the input fields is a rich text editor with icons for bold (B), italic (I), underline (U), strikethrough (ABC), list (三), link (S), and unlink (S). In the bottom right corner, there is a "字数统计" (Word Count) label and a "提交" (Submit) button.

扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术 etc。



异步社区



微信服务号

致谢

在此，我要对审阅本书早期版本的 Rod Lopez、Jim Leiterman、Hanley Leung、Rick Falck、Tybon Wu、Tuomas Sandroos、Eric Sandegren、Jay Tennant 与 William Goschnick 表示感谢。向现存于本书网站上、运用于演示程序中的 3D 模型以及纹理的制作者 Tyler Drinkard 致以谢意。我还要感谢 Dale E. La Force、Adam Hoult、Gary Simmons、James Lambers 以及 William Chin，他们曾给予我极大的帮助。另外，亦感激为我提供了 DirectX 12 beta 版本的 Matt Sandy，以及耐心解答用户们在使用 beta 版时遇到的种种问题的 DirectX 团队。最后，我还要感谢 Mercury Learning and Information 出版社的全体工作人员，尤其是出版人 David Pallai 以及项目经理 Jennifer Blaney，是他们直接促成了本书顺利的出版发行。

目录

第一部分 必备的数学知识

第 1 章 向量代数	3	2.4 单位矩阵	38
1.1 向量	3	2.5 矩阵的行列式	39
1.1.1 向量与坐标系	4	2.5.1 余子阵	40
1.1.2 左手坐标系与右手坐标系	5	2.5.2 行列式的定义	40
1.1.3 向量的基本运算	6	2.6 伴随矩阵	41
1.2 长度和单位向量	8	2.7 逆矩阵	42
1.3 点积	9	2.8 用 DirectXMath 库处理矩阵	43
1.4 叉积	12	2.8.1 矩阵类型	44
1.4.1 2D 向量的伪叉积	13	2.8.2 矩阵函数	46
1.4.2 通过叉积来进行正交化处理	13	2.8.3 DirectXMath 矩阵示例程序	47
1.5 点	14	2.9 小结	49
1.6 利用 DirectXMath 库进行向量运算	15	2.10 练习	49
1.6.1 向量类型	16	第 3 章 变换	52
1.6.2 加载方法和存储方法	17	3.1 线性变换	52
1.6.3 参数的传递	18	3.1.1 定义	52
1.6.4 常向量	20	3.1.2 矩阵表示法	53
1.6.5 重载运算符	21	3.1.3 缩放	53
1.6.6 杂项	21	3.1.4 旋转	55
1.6.7 Setter 函数	21	3.2 仿射变换	58
1.6.8 向量函数	23	3.2.1 齐次坐标	58
1.6.9 浮点数误差	26	3.2.2 仿射变换的定义及其矩阵表示	58
1.7 小结	28	3.2.3 平移	59
1.8 练习	29	3.2.4 缩放和旋转的仿射矩阵	61
第 2 章 矩阵代数	34	3.2.5 仿射变换矩阵的几何意义	61
2.1 矩阵的定义	34	3.3 变换的复合	62
2.2 矩阵乘法	36	3.4 坐标变换	63
2.2.1 定义	36	3.4.1 向量的坐标变换	64
2.2.2 向量与矩阵的乘法	37	3.4.2 点的坐标变换	65
2.2.3 结合律	37	3.4.3 坐标变换的矩阵表示	66
2.3 转置矩阵	38		

3.4.4 坐标变换矩阵及其结合律	66	3.6 DirectXMath 库提供的变换函数	69
3.4.5 坐标变换矩阵及其逆矩阵	67	3.7 小结	70
3.5 变换矩阵与坐标变换矩阵	68	3.8 练习	71

第二部分 Direct3D 基础

第 4 章 Direct3D 的初始化	77	4.3.9 设置视口	114
4.1 预备知识	77	4.3.10 设置裁剪矩形	115
4.1.1 Direct3D 12 概述	77	4.4 计时与动画	116
4.1.2 组件对象模型	78	4.4.1 性能计时器	116
4.1.3 纹理格式	79	4.4.2 游戏计时器类	117
4.1.4 交换链和页面翻转	80	4.4.3 帧与帧之间的时间间隔	118
4.1.5 深度缓冲	81	4.4.4 总时间	120
4.1.6 资源与描述符	83	4.5 应用程序框架示例	123
4.1.7 多重采样技术的原理	85	4.5.1 D3DApp 类	123
4.1.8 利用 Direct3D 进行多重采样	87	4.5.2 非框架方法	126
4.1.9 功能级别	88	4.5.3 框架方法	127
4.1.10 DirectX 图形基础结构	89	4.5.4 帧的统计信息	128
4.1.11 功能支持的检测	92	4.5.5 消息处理函数	130
4.1.12 资源驻留	93	4.5.6 “初始化 Direct3D 演示” 程序	131
4.2 CPU 与 GPU 间的交互	94	4.6 调试 Direct3D 应用程序	135
4.2.1 命令队列和命令列表	94	4.7 小结	137
4.2.2 CPU 与 GPU 间的同步	98	第 5 章 渲染流水线	139
4.2.3 资源转换	100	5.1 3D 视觉即错觉?	140
4.2.4 命令与多线程	101	5.2 模型的代表	141
4.3 初始化 Direct3D	102	5.3 计算机色彩基础	142
4.3.1 创建设备	102	5.3.1 颜色运算	143
4.3.2 创建围栏并获取描述符的 大小	104	5.3.2 128 位颜色	143
4.3.3 检测对 4X MSAA 质量级别的 支持	104	5.3.3 32 位颜色	144
4.3.4 创建命令队列和命令列表	105	5.4 渲染流水线概述	145
4.3.5 描述并创建交换链	105	5.5 输入装配器阶段	147
4.3.6 创建描述符堆	107	5.5.1 顶点	147
4.3.7 创建渲染目标视图	108	5.5.2 图元拓扑	147
4.3.8 创建深度/模板缓冲区及其 视图	110	5.5.3 索引	151
		5.6 顶点着色器阶段	152
		5.6.1 局部空间和世界空间	153