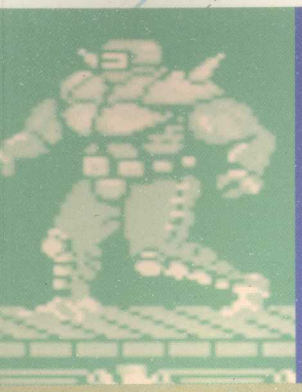
教育部 文化部
高等学校动漫类规划教材

数字游戏中的图形编程技术



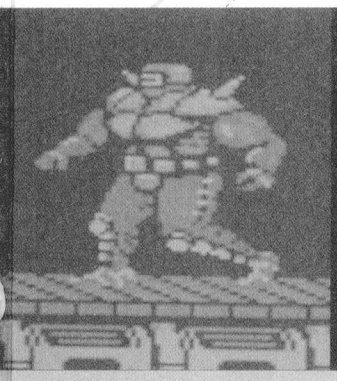
丁刚毅 主编 王崇文 副主编

高等教育出版社
HIGHER EDUCATION PRESS

教育部 文化部
高等学校动漫类规划教材

数字游戏中的图形编程技术

SHUZI YOUXI ZHONG DE TUXING BIANCHENG JISHU



丁刚毅 主编 王崇文 副主编

高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容简介

本书是教育部、文化部高等学校动漫类规划教材，强调 OpenGL 在数字游戏编程中的应用，本书通过大量的实例，深入浅出地介绍了 OpenGL 编程技术。通过本书的学习，读者可以掌握利用 OpenGL API 编写三维图形及数字游戏应用开发的方法。全书共有 11 章，主要包括 OpenGL 简介、绘制几何物体、视图变换、颜色与光照的使用、混合模式、抗锯齿和雾、图像与位图的操作、纹理映射，以及 OpenGL 在移动平台的应用等内容。书中的实例代码都是从大量实际应用中精心筛选出来的，并经过适当地修改、完善和严格测试。

本书可作为高等学校数字媒体专业或数字艺术专业高年级本科生教材，也可作为图形学、游戏程序设计课程的辅助参考资料，还可供欲进入游戏开发、影视特效、虚拟现实与增强现实、移动图形应用等领域的初、中级程序员、高校与科研机构的相关研究人员学习参考。

图书在版编目（CIP）数据

数字游戏中的图形编程技术 / 丁刚毅主编. -- 北京：高等教育出版社，2012.12
教育部、文化部高等学校动漫类规划教材
ISBN 978-7-04-033662-7

Ⅰ. ①数… Ⅱ. ①丁… Ⅲ. ①图形软件－程序设计－高等学校－教材 Ⅳ. ①TP391.41

中国版本图书馆CIP数据核字(2012)第214623号

> 数字游戏中的图形编程技术

丁刚毅 主编　王崇文 副主编

策划编辑 …… 马天魁
责任编辑 …… 郭亚嫘
书籍设计 …… 张申申
责任校对 …… 殷　然
责任印制 …… 朱学忠

出版发行 / 高等教育出版社
社　　址 / 北京市西城区德外大街 4 号
邮政编码 / 100120
印　　刷 / 北京信彩瑞禾印刷厂
开　　本 / 787mm×1092mm 1/16
印　　张 / 14.25
字　　数 / 280 千字
购书热线 / 010-58581118

咨询电话 / 400-810-0598
网　　址 / http://www.hep.edu.cn
　　http://www.hep.com.cn
网上订购 / http://www.landraco.com
　　http://www.landraco.com.cn
版　　次 / 2012 年12月第 1 版
印　　次 / 2012 年12月第 1 次印刷
定　　价 / 43.40 元（含光盘）

物 料 号　33662-00

文化是一个民族的灵魂，而动漫这种特殊的文化载体，以其视听传播的直观性，更容易跨越文化、民族的边界而产生长远的影响。好的动漫作品、动漫形象，伴随一代又一代人的成长，历久而弥新。

进入 21 世纪以来，我国动漫、新媒体产业迅速发展，成为文化产业最重要的组成部分之一。当今国际综合国力竞争中文化的地位和作用更加凸显，文化产业成为加快经济增长方式转变的重要增长点。文化产业繁荣发展的根本是创新，而创新则要求我们建设一支适应时代要求、富有开拓精神、善于创新创造的文化人才队伍。

为了进一步推动我国动漫人才建设，文化部、教育部于 2009 年成立了高等学校动漫类教材建设专家委员会，旨在进一步加强高校动画、新媒体学科理论研究和人才培养，组织高水平教材的编写工作。本套系列教材即是过去两年来的重要工作成果之一。

今年是“十二五”规划的开局之年，也是我国文化改革发展加速推进的关键一年。这套教材在这个关键时期推出，将进一步规范和提高国内高等院校的动漫类专业教学，从而对我国动漫产业的人才储备和持续发展产生积极影响。

国以才兴，业以才立。中国动漫、新媒体产业的希望和未来，寄托在本套教材的读者，也就是全国高校动漫类专业学生身上。我们希望，这套教材能对你们的成长有所裨益，我们也期待，你们能够创作更多更好的优秀中国动漫作品。

是为序。

文化部党组副书记、副部长

扶持动漫产业发展部际联席会议成员、办公室主任

欧阳坚

2011 年 3 月

我们迎来了国内游戏业的春天。游戏业的价值终于得到了认可，并已得到政府的重点支持，政府已将网络游戏通用引擎研究以及示范产品开发列入国家“863”计划，大量风险资金投入到游戏研发和运营之中，不少相关的公司正逐渐成长起来。然而目前在游戏开发技术方面的普及工作还相当不足，在高校里比较难以学到最新的计算机图形学知识，同时市场上关于游戏研发的图形编程技术教材也不多见，这既是我们面临的问题，也是在向我们的挑战。

本书即是献给数字媒体专业方向的有志于从事游戏设计和开发的年轻朋友们。本书从实际应用的角度出发，全书以 OpenGL 在实际应用中频繁出现的技术重点和难点为主要内容，完全以对实例的精心讲解贯穿全书，并在各个实例中穿插 OpenGL 和 3D 图形学的相关原理和概念，舍弃 OpenGL 中与实际 3D 图形应用开发关联不大的琐碎细节，以一种全新的方式引导读者快速掌握实际开发中所必须掌握的最重要、最实用的概念、原理和编程技巧，事半功倍地进入相关开发领域。本书中的实例代码都是从大量实际应用中精心筛选出来的，并经过适当的修改、完善和严格测试。

全书共分 11 章，各章节课时安排如下：第 0 章 2 个学时，主要介绍数字游戏的相关概念和常见的图形编程接口。第 1 章 4 个学时，主要介绍 OpenGL 的发展历程及其基本特点、体系架构和相关函数库，要求学生能够配置 OpenGL 的编程开发环境。第 2 章 3 个学时，主要讲述如何清除屏幕，以及如何绘制几何物体（包括点、直线和平面多边形）。第 3 章 6 个学时，主要介绍如何确定模型在场景中显示的结果，以及如何选择一个观察点来查看场景。第 4 章 3 个学时，主要讲述计算机颜色的概念以及 OpenGL 的颜色模式、颜色定义及两种模式应用场合等内容。第 5 章 5 个学时，着重讲述光照模型、光源设置、材质定义以及有关的计算机图形学概念等内容。介绍如何运用 OpenGL 函数来绘制真实感的图形。第 6 章 4 个学时，主要讲述 OpenGL 的三个特殊效果处理：混合模式，抗锯齿和雾，如何运用这些特殊效果实现对真实现象的模拟。第 7 章 3 个学时，主要讲解 OpenGL 中的两个重要数据类：位图和图像。第 8 章 6 个学时，详细介绍 OpenGL 纹理映射有关的内容：基本步骤、纹理定义、纹理控制、映射方式和纹理坐标等。第 9 章 4 个学时，主要讲解 OpenGL 在现在的主流移动平台之一 Android 上面的应用。第 10 章 8 个学时，通过实现一个简单的 3D 小游戏，实现对知识的整合，用来进一步加深对 OpenGL 的认识和理解。需要注意的是，

这里建议的课时数，都是包含了课堂实验的，望使用本教材时根据具体情况进一步调整。

虽然本书的定位非常适合没有 OpenGL 编程经验的初学者，但这并不意味着没有相关基础就能轻松阅读本书，因为图形编程不像某些领域那样学几句语句就能操刀上阵，而是需要花费更多的时间学习其他相关内容才能成为高手。因此，要成为高手就需要学习使用 C++ 或 Java，熟悉计算机图形学一些基本知识，最好还能具备一些线性代数方面的基础。

在本书的创作过程中，笔者得到了很多朋友的热心帮助，他们是刘利凯、黄宏旺、孙红、张龙泽、田堃、黄金等，同时高等教育出版社各级领导和相关编辑的精心组织、细心审阅和修改保证了本书高质量地如期出版。书中还参考了大量国内外专著、教材和图片，在此一并表示衷心的感谢！

由于笔者的教学和科研水平有限，书中难免有不当之处，敬请同行专家和读者不吝指正。同行专家、读者的宝贵意见将给我更大的激励，并有助于提高以后面世的新版本的质量。

第 6 章
混合模式，抗锯齿和雾

/097/

第 7 章
位图和图像

/121/

第 8 章
纹理映射

/139/

第 9 章
OpenGL 在 Android 移动平台上的应用

/161/

第 10 章

小试牛刀

/179/

参考文献

预备知识

0.1 什么是数字游戏

数字游戏指的是以数字技术为手段设计开发，并运行在数字化设备平台的游戏。相对于传统游戏，数字游戏的特点是具有跨媒介特性和历史发展性等优势。数字游戏的称谓具有兼容性，是许多种不同游戏媒介的集合。数字游戏相对于“电子游戏”、“计算机游戏”、“视频游戏”、“交互游戏”等称谓而言，更具有延展性和本质性。

“数字游戏”一词具备一定的延展性，即历史发展性。也就是说，无论游戏发展到何种境地，只要继续采用数字化的手段，就可称之为“数字游戏”。而“视频游戏”的界定则指“通过终端屏幕呈现出文字或图像画面的游戏方式”，将游戏限定于凭借视频画面进行展示的类别。随着技术的发展，数字化的游戏将逐渐超越视频的范畴，朝向更为广阔的现实物理空间和赛伯空间（Cyberspace）发展。同样，“计算机游戏”一词也将概念限定到一个较小的范畴，单指计算机平台上的游戏，而其他例如基于手机、PS2、Xbox、PSP、街机等平台的游戏均具有类似的设计特性和技术手段，却被划出圈外。而“电子游戏”作为通俗的称谓在国内普遍流传。由于历史的机缘，数字游戏引入我国之始正值 20 世纪 80 年代中期，正是电子技术方兴未艾，数字概念尚未萌动的年代。因此，“电子游戏”便一直沿用至今。时至今日，“电子游戏”更倾向于指代基于传统电子技术下的老式游戏（尤见于西方），而较少用来指代网络游戏、虚拟现实游戏等较新型的游戏 。

“数字游戏”一词可以涵盖电脑游戏、网络游戏、电视游戏、街机游戏、手机游戏等各种基于数字平台的游戏，从本质层面概括出了该类游戏的共性。这些游戏虽然彼此面目迥异，但是却有着类似的原理——即在基本层面均采用以信息运算为基础的数字化技术。以电视游戏为例，这一类型的游戏虽然以电视为屏幕，但其主机仍然可被看成是某种计算机。如 20 世纪 80 年代任天堂的 FC 游戏机，采用 8 位处理器，52 色的图像控制器和兼容 5 声道的声音发生器；而 2006 年微软推出的 Xbox360 则包括了 3 个 3.2 GHz 的 Xenon 处理器，支持 DirectX 9.0 的视频处理芯片和 512 MB 的内存。二者的计算速度相差 1 800 多倍，但其结构都符合冯 · 诺伊曼的计算机模型。因而，这些基于数字技术的游戏可以从一个平台

移植到另一平台，并维持原作的基本风格和面貌。而且，同一款游戏也往往同时推出不同平台上的版本。例如，2004 年由 Treyarch Studios 开发的《蜘蛛侠 2》就同时发售了基于 PC、PS2、XBOX、NGC、GBA 等五个平台的版本，其剧情、画面、音效、关卡都基本一致。这也从另一个侧面说明了不同类别游戏的本质同一性，即数字化性。

0.2.1 什么是图形编程接口

要理解什么是图形编程接口，首先要理解什么是编程接口。API（Application Programming Interface, 应用程序编程接口）是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件的以访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。以此类推，图形编程接口就是指一组可以驱动图形硬件的函数库，基于这些函数库，开发人员能够方便地使用计算机绘制图形而无需理解底层的硬件是如何实现的。目前最常见的图形编程接口包括 DirectX 显示部分以及 OpenGL。

0.2.2 DirectX

Microsoft DirectX 是这样一组技术：它们旨在使基于 Windows 的计算机成为运行和显示具有丰富多媒体元素（例如全色图形、视频、3D 动画和丰富音频）的应用程序的理想

平台。DirectX 是由很多 API 组成的，按照性质分类，可以分为四大部分，显示部分、声音部分、输入部分和网络部分。

显示部分担任图形处理的关键，分为 DirectDraw（DDraw）和 Direct3D（D3D），前者主要负责 2D 图像加速。它包括很多方面：我们播放 mpg、DVD 电影、看图、玩小游戏等都是使用 DDraw，你可以把它理解成所有画线的部分都是使用 DDraw。后者则主要负责 3D 效果的显示，比如 CS 中的场景和人物、FIFA 中的人物等，都是使用了 DirectX 的 Direct3D。

0.2.3 OpenGL

OpenGL 是行业领域中最为广泛接纳的 2D/3D 图形 API，其自诞生至今已催生了各种计算机平台及设备上的数千个优秀应用程序。OpenGL 是独立于视窗操作系统或其他操作系统的，亦是网络透明的。在包含 CAD、内容创作、能源、娱乐、游戏开发、制造业、制药业及虚拟现实等行业领域中，OpenGL 可帮助程序员实现在 PC、工作站、超级计算机等硬件设备上的高性能、极具冲击力的高视觉表现力图形处理软件的开发。

OpenGL 定义了一个跨编程语言、跨平台的编程接口的规范，可用于二维和三维图像（二维的亦可）。OpenGL 提供了功能强大，调用方便的底层图形库，OpenGL 的前身是 SGI 公司为其图形工作站开发的 IRIS GL。IRIS GL 是一个工业标准的 3D 图形软件接口，功能虽然强大但是移植性不好，于是 SGI 公司便在 IRIS GL 的基础上开发了 OpenGL。OpenGL 的英文全称是“Open Graphics Library”，顾名思义，OpenGL 便是“开放的图形程序接口”。虽然 DirectX 在家用市场全面领先，但在专业高端绘图领域，OpenGL 是不能被取代的主角。

0.3 游戏引擎

0.3.1 什么是游戏引擎

我们可以把游戏的引擎比作赛车的引擎，大家知道，引擎是赛车的心脏，决定着赛车的性能，赛车的速度、操纵感这些直接与车手相关的指标都是建立在引擎的基础上的。游戏也是如此，玩家所体验到的剧情、关卡、美工、音乐、操作等内容都是由游戏的引擎直接控制的，它扮演着中场发动机的角色，把游戏中的所有元素捆绑在一起，在后台指挥它们同时、有序地工作。简单地说，引擎就是“用于控制所有游戏功能的主程序，从计算碰撞、物理系统和物体的相对位置，到接受玩家的输入，以及按照正确的音量输出声音等”。可见，引擎并不是什么玄乎的东西，无论是 2D 游戏还是 3D 游戏，无论是角色扮演游戏、即时策略游戏、冒险解谜游戏或是动作射击游戏，哪怕是一个只有 1 MB 的小游戏，都有这样一段起控制作用的代码。经过不断地改进，如今的游戏引擎已经发展为一套由多个子系统共同构成的复杂系统，从建模、动画到光影、粒子特效，从物理系统、碰撞检测到文件管理、网络特性，还有专业的编辑工具和插件，几乎涵盖了开发过程中的所有重要环节。以下对引擎的一些关键部件作一个简单的介绍。首先是光影效果，即场景中的光源对处于其中的人和物的影响方式。游戏的光影效果完全是由引擎控制的，折射、反射等基本的光学效果以及动态光源、彩色光源等高级效果都是通过引擎的不同编程技术实现的。其次是动画，目前游戏所采用的动画系统可以分为两种：一是骨骼动画系统，一是模型动画系统，前者用内置的骨骼带动物体产生运动，比较常见，后者则是在模型的基础上直接进行变形。引擎把这两种动画系统预先植入游戏，方便动画师为角色设计丰富的动作造型。引擎的另一重要功能是提供物理系统，这可以使物体的运动遵循固定的规律，例如，当角色跳起的时候，系统内定的重力值将决定他能跳多高，以及他下落的速度有多快，子弹的飞行轨迹、车辆的颠簸方式也都是由物理系统决定的。碰撞探测是物理系统的核心部分，它可以探测游戏中各物体的物理边缘。当两个 3D 物体撞在一起的时候，这种技术可以防止它们相互穿过，这就确保了当你撞在墙上的时候，不会穿墙而过，也不会把墙撞倒，因为碰撞探测会根据你和墙之间的

特性确定两者的位置和相互的作用关系。渲染是引擎最重要的功能之一，当 3D 模型制作完毕之后，美工会按照不同的面把材质贴图赋予模型，这相当于为骨骼蒙上皮肤，最后再通过渲染引擎把模型、动画、光影、特效等所有效果实时计算出来并展示在屏幕上。渲染引擎在引擎的所有部件当中是最复杂的，它的强大与否直接决定着最终的输出质量。引擎还有一个重要的职责就是负责玩家与电脑之间的沟通，处理来自键盘、鼠标、摇杆和其他外设的信号。如果游戏支持联网特性的话，网络代码也会被集成在引擎中，用于管理客户端与服务器之间的通信。通过上面这些枯燥的介绍我们至少可以了解到一点：引擎相当于游戏的框架，框架打好后，关卡设计师、建模师、动画师只要往里填充内容就可以了。因此，在 3D 游戏的开发过程中，引擎的制作往往会占用非常多的时间，《马克思·佩恩》的 MAX-FX 引擎从最初的雏形 Final Reality 到最终的成品共花了四年多时间，LithTech 引擎的开发共花了整整五年时间，耗资 700 万美元，Monolith 公司（LithTech 引擎的开发者）的老板詹森·霍尔甚至不无懊悔地说："如果当初意识到制作自己的引擎要付出这么大的代价的话，我们根本就不可能去做这种傻事。没有人会预料得到五年后的市场究竟是怎样的。"正是出于节约成本、缩短周期和降低风险这三方面的考虑，越来越多的开发者倾向于使用第三方的现成引擎制作自己的游戏，一个庞大的引擎授权市场已经形成。其中收益最大的是各大网络游戏公司，通过第三方引擎开发的网络游戏获益巨大。但随着市场急剧变化，用第三方引擎开发网络游戏的成本也越来越高。于是游戏引擎开发商们开始绞尽脑汁设计一种可以大量节约开发成本和周期的引擎。直到 2010 年 zerodin 引擎开发的巨作 Dragona 出品引起了各大游戏业巨头关注，因为此时方才发现了巨作也可以用这么少的成本，这么短的时间开发而成。

0.3.2 主流游戏引擎

3D 游戏引擎应该是包括 3D 图形的各种算法整合起来，提供便捷的 SDK 接口以方便别人在这个基础上开发游戏的模块。优秀的 3D 游戏引擎，会把复杂的图形算法都稳定高效地封装在模块内部，对外则提供简捷、有效的 SDK 接口，人们可以非常轻松地学会使用这些 SDK，并且通过这些简单的 SDK，就可以完全满足各种复杂的 3D 游戏功能需求。优秀的 3D 游戏引擎，一般会提供功能强大的编辑器，包括引擎的场景编辑、模型编辑、动画编辑、粒子编辑等功能，游戏开发中的美术师可以借助于这些工具，大幅度提高工作效率、工作品质。优秀的 3D 游戏引擎，都会配套地提供第三方插件，如 3DS Max、Maya 的导出插件。当前市面上 3D 游戏引擎，还会同时提供网路、数据库、脚本等方面的功能。随着人们对图像画

质的需求越来越高，3D 游戏引擎也越来越复杂，相信以后，3D 游戏引擎会逐步成为一个独立的产业，更大程度地促进社会的发展，提高人们的生活水平。

目前主流的游戏引擎包括：BigWorld 公司的 BigWorld 引擎，Emergent 公司的 Gamebryo，Epic 公司的 unreal2、unreal3，Crytek 公司的 CryEngine1、CryEngine2 和 CryEngine3，Garage Games 公司的 Torque 3D 引擎，Hero 公司的 Hero Engine 等。