

<<Windows游戏编程大师技巧>>

图书基本信息

书名：<<Windows游戏编程大师技巧>>

13位ISBN编号：9787115292483

10位ISBN编号：7115292485

出版时间：2012-11

出版时间：人民邮电出版社

作者：拉莫斯

页数：734

字数：1362000

译者：沙鹰

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Windows游戏编程大师技巧>>

内容概要

《Windows游戏编程大师技巧(第2版)》是著名游戏程序设计大师André LaMothe的代表作。

全书分为4个部分，共计15章和6个附录。

作者循循善诱地从程序设计的角度介绍了在Windows环境下进行游戏开发所需的全部知识，包括Win32编程以及DirectX中所有主要组件(包括DirectDraw、DirectSound、DirectInput和DirectMusic)。书中还用单独的章节详细讲授了2D图形学和光栅化技术、游戏算法、多线程编程、文本游戏和解析、人工智能(包括模糊逻辑、神经网络和遗传算法)、物理建模(完全碰撞反应、动量传递和正反向运动学)及实时模拟等游戏程序开发中的关键技术。

附录部分介绍了本书光盘的内容，如何安装DirectX，回顾了数学和三角学的基础知识、C++编程的基础知识，还给出了游戏编程资源以及ASCII表。

《Windows游戏编程大师技巧(第2版)》所附光盘上带有本书中所有程序的源代码、关于Direct3D和General

3D的文章和在线书籍以及众多免费的素材。

《Windows游戏编程大师技巧(第2版)》适合想要学习Windows游戏编程的人员阅读，对于有一定经验的专业游戏开发人员，也具有较高的参考价值。

<<Windows游戏编程大师技巧>>

作者简介

本书作者Andre LaMothe有25年的计算行业从业经验，拥有数学、计算机科学和电子工程等学位，是20岁时就在NASA做研究工作的少数几人之一。作者的著作以及参与策划的套系，都成为游戏设计和开发领域的经典。

书籍目录

第一部分 Windows编程基础

第1章 学海无涯

- 1.1 历史一瞥
- 1.2 设计游戏
- 1.3 游戏类型
- 1.4 集思广益
- 1.5 设计文档和情节串联图板
- 1.6 使游戏具有趣味性
- 1.7 游戏的构成
- 1.8 常规游戏编程指导规范
- 1.9 使用工具
- 1.10 从准备到完成——使用编译器
- 1.11 示例：FreakOut
- 1.12 小结

第2章 Windows编程模型

- 2.1 Windows的起源
 - 2.1.1 早期版本的Windows
 - 2.1.2 Windows 3.x
 - 2.1.3 Windows 95
 - 2.1.4 Windows 98
 - 2.1.5 Windows ME
 - 2.1.6 Windows XP
 - 2.1.7 Windows NT/2000
 - 2.1.8 Windows 基本架构：Win9X/NT
- 2.2 多任务和多线程
 - 2.2.1 获取线程的信息
 - 2.2.2 事件模型
- 2.3 按照微软风格编程：匈牙利符号表示法
 - 2.3.1 变量的命名
 - 2.3.2 函数的命名
 - 2.3.3 类型和常量的命名
 - 2.3.4 类的命名
 - 2.3.5 参数的命名
- 2.4 世界上最简单的Windows程序
 - 2.4.1 总是从WinMain()开始
 - 2.4.2 程序剖析
 - 2.4.3 选择一个信息框
- 2.5 现实中的Windows应用程序
- 2.6 Windows类
- 2.7 注册Windows类
- 2.8 创建窗口
- 2.9 事件处理程序
- 2.10 主事件循环
- 2.11 产生一个实时事件循环
- 2.12 打开多个窗口

<<Windows游戏编程大师技巧>>

2.13 小结

第3章 高级Windows编程

3.1 使用资源

3.1.1 整合资源

3.1.2 使用图标资源

3.1.3 使用光标资源

3.1.4 创建字符串表资源

3.1.5 使用.WAV声音资源

3.1.6 使用编译器创建.RC文件

3.2 使用菜单

3.2.1 创建菜单

3.2.2 装载菜单

3.2.3 响应菜单事件消息

3.3 GDI(图形设备接口)简介

3.3.1 重拾WM_PAINT信息

3.3.2 视频显示基础和色彩(Video Display Basics and Color)

3.3.3 RGB和调色板模式

3.3.4 基本文本显示

3.4 处理重要事件

3.4.1 Windows操作

3.4.2 处理键盘事件

3.4.3 处理鼠标事件

3.5 自行发送消息

3.6 小结

第4章 Windows GDI、控件和灵感

4.1 高级GDI绘图

4.1.1 掀开图形设备描述表的盖头来

4.1.2 颜色、画笔和画刷

4.1.3 使用画笔

4.1.4 使用画刷

4.2 点、线、平面多边形和圆

4.2.1 绘制点

4.2.2 绘制线段

4.2.3 绘制矩形

4.2.4 绘制圆

4.2.5 绘制多边形

4.3 深入文本和字体

4.4 定时高于一切

4.4.1 WM_TIMER消息

4.4.2 低层定时操作

4.5 使用控件

4.5.1 按钮

4.5.2 向子控件发送消息

4.6 获取信息

4.7 T3D游戏控制台程序

4.8 小结

第二部分 DirectX和2D基础

<<Windows游戏编程大师技巧>>

第5章 DirectX基础知识和令人生畏的COM

- 5.1 DirectX基础
 - 5.1.1 HEL和HAL
 - 5.1.2 更多的DirectX基础类
- 5.2 COM：是微软的杰作，还是魔鬼的杰作
 - 5.2.1 COM对象究竟是什么
 - 5.2.2 接口标识符和GUID的详细内容
 - 5.2.3 创建一个类COM对象
 - 5.2.4 COM的简要回顾
 - 5.2.5 可运行的COM程序
- 5.3 应用DirectX COM对象
 - 5.3.1 COM和函数指针
 - 5.3.2 创建和使用DirectX界面
 - 5.3.3 接口查询
- 5.4 COM的前景
- 5.5 小结

第6章 初次邂逅DirectDraw

- 6.1 DirectDraw的接口
 - 6.1.1 接口的特性
 - 6.1.2 组合使用接口
- 6.2 创建DirectDraw对象
 - 6.2.1 对DirectDraw进行错误处理
 - 6.2.2 顺便提一下接口
- 6.3 与Windows协作
- 6.4 设置模式
- 6.5 色彩的奥秘
- 6.6 创建显示表面
 - 6.6.1 创建一个主显示表面
 - 6.6.2 关联调色板
 - 6.6.3 绘制像素
 - 6.6.4 清理资源
- 6.7 小结

第7章 高级DirectDraw和位图图形

- 7.1 使用高彩模式
 - 7.1.1 16位高彩模式
 - 7.1.2 获取像素格式
 - 7.1.3 24/32位真彩色模式
- 7.2 双缓冲
- 7.3 表面动态
- 7.4 页面切换
- 7.5 显存块移动单元(Blitter)
 - 7.5.1 使用Blitter进行内存填充
 - 7.5.2 从一个表面向另一个表面复制位图
- 7.6 基础裁剪知识
 - 7.6.1 将像素按视口裁剪
 - 7.6.2 位图裁剪技巧
 - 7.6.3 使用IDirectDrawClipper进行DirectDraw裁剪

<<Windows游戏编程大师技巧>>

- 7.7 使用位图
 - 7.7.1 载入.BMP文件
 - 7.7.2 使用位图
 - 7.7.3 载入8位位图
 - 7.7.4 载入16位位图
 - 7.7.5 载入24位位图
 - 7.7.6 总结位图
 - 7.8 离屏表面
 - 7.8.1 创建离屏表面
 - 7.8.2 在离屏表面上进行Blitting
 - 7.8.3 设置Blitter
 - 7.8.4 色彩键
 - 7.8.5 源色彩键
 - 7.8.6 目标色彩键
 - 7.8.7 使用Blitter(终于！
)
 - 7.9 位图的旋转和缩放
 - 7.10 离散采样理论
 - 7.11 色彩效果
 - 7.11.1 256色模式下的色彩动画
 - 7.11.2 256色模式下的色彩旋转
 - 7.11.3 使用RGB模式的技巧
 - 7.12 手动色彩变换及查询表
 - 7.13 新的DirectX色彩和Gamma控制接口
 - 7.14 将GDI和DirectX联用
 - 7.15 DirectDraw的庐山真面目
 - 7.15.1 主DirectDraw对象
 - 7.15.2 关于表面
 - 7.15.3 使用调色板
 - 7.16 在窗口模式下使用DirectDraw
 - 7.17 小结
- 第8章 矢量光栅化及2D变换
- 8.1 绘制线条
 - 8.1.1 Bresenham算法
 - 8.1.2 算法的速度优化
 - 8.2 基本2D图形裁剪
 - 8.2.1 利用点斜式计算两条直线的交点
 - 8.2.2 利用一般式计算两条直线的交点
 - 8.2.3 利用矩阵式计算两条直线的交点
 - 8.2.4 裁剪直线
 - 8.2.5 Cohen-Sutherland算法
 - 8.3 线框多边形
 - 8.3.1 多边形数据结构
 - 8.3.2 多边形的绘制及裁剪
 - 8.4 2D平面里的变换
 - 8.4.1 平移
 - 8.4.2 旋转

<<Windows游戏编程大师技巧>>

- 8.4.3 缩放
- 8.5 矩阵引论
 - 8.5.1 单位矩阵
 - 8.5.2 矩阵加法
 - 8.5.3 矩阵乘法
 - 8.5.4 使用矩阵进行变换
- 8.6 平移
- 8.7 缩放
- 8.8 旋转
- 8.9 填充实心多边形
 - 8.9.1 三角形和四边形类型
 - 8.9.2 绘制三角形和四边形
 - 8.9.3 三角形解构详述
 - 8.9.4 四边形光栅化的一般性讨论
 - 8.9.5 多边形的三角化
- 8.10 多边形碰撞检测
 - 8.10.1 接近度、边界球/圆
 - 8.10.2 边界盒
 - 8.10.3 点包含
- 8.11 深入定时和同步
- 8.12 卷轴和摇镜头
 - 8.12.1 页面卷轴引擎
 - 8.12.2 均匀平铺(Tile)显示引擎
 - 8.12.3 稀疏位图平铺显示引擎
- 8.13 伪3D等轴测引擎
 - 8.13.1 方法1：基于单元，全二维
 - 8.13.2 方法2：基于全屏，具有一些二维或三维的碰撞网络
 - 8.13.3 方法3：采用全三维数学运算，使用一个固定的相机视角
- 8.14 T3DLIB1函数库
 - 8.14.1 引擎架构
 - 8.14.2 新的游戏编程控制台程序
 - 8.14.3 基本定义
 - 8.14.4 可用的宏
 - 8.14.5 数据类型和结构
 - 8.14.6 全局定义
 - 8.14.7 DirectDraw接口
 - 8.14.8 2D多边形函数
 - 8.14.9 2D基本图元函数
 - 8.14.10 数学和误差函数
 - 8.14.11 位图函数
 - 8.14.12 调色板函数
 - 8.14.13 实用工具函数
- 8.15 BOB引擎
- 8.16 小结
- 第9章 DirectInput输入和力反馈
 - 9.1 输入循环
 - 9.2 DirectInput序曲

<<Windows游戏编程大师技巧>>

- 9.2.1 DirectInput组件
- 9.2.2 设置DirectInput的一般步骤
- 9.2.3 数据采集模式
- 9.2.4 创建主DirectInput对象
- 9.2.5 101键盘
- 9.2.6 读数据过程中的问题：重获取
- 9.2.7 捕捉鼠标
- 9.2.8 使用游戏杆
- 9.2.9 将输入消息化
- 9.3 力反馈
 - 9.3.1 力反馈的物理原理
 - 9.3.2 设置力反馈
 - 9.3.3 力反馈演示程序
- 9.4 编写一个广泛适用的输入系统：T3DLIB2.CPP
- 9.5 小结
- 第10章 用DirectSound和DirectMusic演奏乐曲
 - 10.1 在PC上对声音编程
 - 10.2 从此有了声音...
 - 10.3 数码声音与MIDI比较——音质好且存储省
 - 10.3.1 数码声音——从“位”开始
 - 10.3.2 合成声音与MIDI
 - 10.3.3 MIDI概述
 - 10.4 发声硬件
 - 10.4.1 波表合成
 - 10.4.2 波导合成
 - 10.5 数码录音：设备和技术
 - 10.5.1 录制声音
 - 10.5.2 处理声音
 - 10.6 DirectSound中的麦克风
 - 10.7 初始化DirectSound
 - 10.7.1 理解协作等级
 - 10.7.2 设定协作等级
 - 10.8 主声音缓冲区与辅助声音缓冲区
 - 10.8.1 使用辅助缓冲区
 - 10.8.2 创建辅助声音缓冲区
 - 10.8.3 把数据写入辅助声音缓冲区
 - 10.9 渲染声音
 - 10.9.1 播放声音
 - 10.9.2 停止播放
 - 10.9.3 控制音量
 - 10.9.4 调整频率
 - 10.9.5 调整声道平衡
 - 10.10 用DirectSound反馈信息
 - 10.11 从磁盘中读取声音数据
 - 10.11.1 .WAV文件格式
 - 10.11.2 读取.WAV文件
 - 10.12 DirectMusic：伟大的试验

<<Windows游戏编程大师技巧>>

- 10.13 DirectMusic的体系结构
 - 10.14 初始化DirectMusic
 - 10.14.1 初始化COM
 - 10.14.2 创建一个演奏对象
 - 10.14.3 给演奏对象增加端口
 - 10.15 加载一个MIDI段
 - 10.15.1 创建一个加载程序
 - 10.15.2 加载MIDI文件
 - 10.16 操作MIDI段
 - 10.16.1 播放一个MIDI段
 - 10.16.2 停止一个MIDI段
 - 10.16.3 检查MIDI段的状态
 - 10.16.4 释放一个MIDI段
 - 10.16.5 关闭DirectMusic
 - 10.16.6 一些DirectMusic的实例
 - 10.17 T3DLIB3声音和乐曲库
 - 10.17.1 头文件
 - 10.17.2 类型
 - 10.17.3 全局变量
 - 10.17.4 DirectSound API封装
 - 10.17.5 DirectMusic API封装
 - 10.18 小结
- 第三部分 核心游戏编程
- 第11章 算法、数据结构、内存管理和多线程
- 11.1 数据结构
 - 11.1.1 静态结构和数组
 - 11.1.2 链表
 - 11.2 算法分析
 - 11.3 递归
 - 11.4 树结构
 - 11.4.1 建立二分查找树(BST)
 - 11.4.2 搜索BST
 - 11.5 优化理论
 - 11.5.1 运用你的头脑
 - 11.5.2 数学技巧
 - 11.5.3 定点运算
 - 11.5.4 循环体展开
 - 11.5.5 查找表
 - 11.5.6 汇编语言
 - 11.6 制作演示
 - 11.6.1 预先记录的演示
 - 11.6.2 由人工智能控制的演示
 - 11.7 保存游戏的手段
 - 11.8 实现多人游戏
 - 11.8.1 轮流
 - 11.8.2 分屏
 - 11.9 多线程编程技术

<<Windows游戏编程大师技巧>>

- 11.9.1 多线程编程的术语
- 11.9.2 为何要在游戏中使用线程
- 11.9.3 取得一个线程
- 11.9.4 线程间的消息传递
- 11.9.5 等待合适时机
- 11.9.6 多线程和DirectX
- 11.9.7 高级多线程编程
- 10.10 小结
- 第12章 人工智能
- 12.1 人工智能初步
- 12.2 确定性AI算法
 - 12.2.1 随机运动
 - 12.2.2 跟踪算法
 - 12.2.3 反跟踪：闪避算法
- 12.3 模式以及基本控制脚本的编写
 - 12.3.1 基本模式
 - 12.3.2 具备条件逻辑处理的模式
- 12.4 行为状态系统建模
 - 12.4.1 基本状态机
 - 12.4.2 加入更多表现个性的行为
- 12.5 用软件对记忆和学习进行建模
- 12.6 计划树和决策树
 - 12.6.1 计划编程
 - 12.6.2 实现真正的计划生成器
- 12.7 寻路
 - 12.7.1 试探法(Trial and Error)
 - 12.7.2 轮廓跟踪
 - 12.7.3 碰撞规避跟踪
 - 12.7.4 路点寻路
 - 12.7.5 一个赛车例子
 - 12.7.6 可靠的寻路
- 12.8 高级AI脚本
 - 12.8.1 设计脚本语言
 - 12.8.2 使用C/C++编译器
- 12.9 人工神经网络
- 12.10 遗传算法
- 12.11 模糊逻辑
 - 12.11.1 普通集合论
 - 12.11.2 模糊集合理论
 - 12.11.3 模糊语言变量及其规则
 - 12.11.4 模糊流形与成员关系
 - 12.11.5 模糊关联矩阵
 - 12.11.6 用模糊化的输入处理FAM
 - 12.11.7 暖融融
- 12.12 为游戏创造真正的AI
- 12.13 小结
- 第13章 基本物理建模

<<Windows游戏编程大师技巧>>

- 13.1 基本物理学定律
 - 13.1.1 质量(m)
 - 13.1.2 时间(t)
 - 13.1.3 位置(s)
 - 13.1.4 速率(v)
 - 13.1.5 加速度(a)
 - 13.1.6 力(F)
 - 13.1.7 多维空间中的力
 - 13.1.8 动量(P)
- 13.2 线性动量的物理性质：守恒与传递
- 13.3 模拟万有引力效果
 - 13.3.1 模拟重力井
 - 13.3.2 模拟炮弹弹道
- 13.4 讨厌的摩擦力
 - 13.4.1 摩擦基本概念
 - 13.4.2 斜面上的摩擦力(高级)
- 13.5 基本的特殊碰撞反应
 - 13.5.1 简单的x,y反弹物理
 - 13.5.2 计算任意方向上的平面碰撞反应
 - 13.5.3 矢量反射示例
 - 13.5.4 线段的交点
- 13.6 实际2D物体间的精确碰撞响应(高级)
- 13.7 解决n-t坐标系统
- 13.8 简单运动学
 - 13.8.1 求解正向运动学问题
 - 13.8.2 解决反向运动学问题
- 13.9 粒子系统
 - 13.9.1 每颗粒子都需要的东西
 - 13.9.2 设计粒子引擎
 - 13.9.3 粒子引擎软件
 - 13.9.4 产生初始条件
 - 13.9.5 整合微粒系统
- 13.10 创建游戏的物理模型
 - 13.10.1 物理建模的数据结构
 - 13.10.2 基于帧的模拟与基于时间的模拟
- 13.11 小结
- 第14章 文字时代
 - 14.1 什么是文字游戏
 - 14.2 文字游戏如何工作
 - 14.3 从外部世界获得输入
 - 14.4 语言分析和解析
 - 14.4.1 词法分析
 - 14.4.2 句法分析
 - 14.4.3 语义分析
 - 14.5 组成游戏世界
 - 14.5.1 表示场景
 - 14.5.2 放置物体

<<Windows游戏编程大师技巧>>

- 14.5.3 让事情发生
- 14.5.4 自由移动
- 14.5.5 物品系统
- 14.6 实现视觉、听觉和嗅觉
 - 14.6.1 听觉
 - 14.6.2 嗅觉
 - 14.6.3 视觉
- 14.7 实时响应
- 14.8 错误处理
- 14.9 造访Shadow Land
- 14.10 Shadow Land中使用的语言
- 14.11 编译和运行Shadow Land
- 14.12 小结
- 第15章 综合运用
 - 15.1 Outpost的设计初稿
 - 15.1.1 游戏背景
 - 15.1.2 设计游戏玩法
 - 15.2 用于编写游戏的工具
 - 15.3 游戏场景：在太空中卷动
 - 15.4 玩家控制的飞船：鬼怪号
 - 15.5 小行星带
 - 15.6 敌人
 - 15.6.1 边哨(Outpost)
 - 15.6.2 捕食者太空雷
 - 15.6.3 战舰
 - 15.7 “宝物”
 - 15.8 HUD
 - 15.9 粒子系统
 - 15.10 玩游戏
 - 15.11 编译Outpost
 - 15.12 结束语
- 第四部分 附录
 - 附录A 光盘内容简介
 - 附录B 安装DirectX和使用C/C++编译器
 - 附录C 数学和三角学回顾
 - 附录D C++入门
 - 附录E 游戏编程资源
 - 附录F ASCII表

章节摘录

版权页：插图：在DirectX 8.0版本里，Microsoft决定将DirectDraw和Direct3D紧密地集成到一起，总称为DirectXGraphics。

结果就是DirectDraw在8.0版本里被移除了，不过，你仍然可以使用DirectDraw，在DirectX8.0版本中并没有将它升级。

另外，DirectSound和DirectMusic也被紧密地集成，从而成为现在的DirectXAudi0。

最后，DirectShow（来自以前的DirectMedia）现在也已经被集成到DirectX里。

Microsoft那些家伙可真是有够忙活的！

上面说的这些有点过于庞杂和令人费解了，但是DirectX有一点非常好，那就是通过COM（我们很快会熟悉它的），使用者实际可以按照需要来决定究竟使用DirectX 3.0、5.0、6.0或是其他任何版本。

对于本书而言，使用7.0版本和8.0版本就已经绰绰有余了。

此外，如果你了解DirectX的其中一个版本的话，你就能一通百通地了解其他所有版本。

尽管语法上可能有些微小的变化，界面也有些细微差别，但是总的来说，它们还是一样的。

只有Direct3D是惟一有很大改动的部分，我们在本书中不做该方面的阐述，实际上，我们将重点讨论的是游戏编程的内容。

虽然，在CD里有两本关于3D的书，其中有一本就是讲Direct3D的，但是在本书中，我们不会涉及太多DirectX的内容，所以你只要学习有关游戏制作的那部分就足够了。

但是，毕竟你的整个游戏编程生涯并不会与DirectX绑在一起，如果你使用的是其他的API（应用编程接口），你仍然可以理解游戏编程的基础技术，这才是本书的终极目标。

5.1.1 HEL和HAL 在图5—2里，你会发现在DirectX的下面有两个层分别叫做HEL（Hardware Emulation Layer，硬件仿真层）和HAL（Hardware Abstraction Layer，硬件抽象层）。

原理是这样的：DirectX是一种具有前瞻性的设计思路，它假定那些高级功能将由硬件实现。

但是，如果硬件并不支持其中某些功能特性，那又该如何处理呢？

这就是HAL和HEL双重模式的基本设计思路。

HAL，也就是硬件抽象层，是直接和硬件对话的一层。

HAL是一种设备驱动程序，通常由设备生产商提供，你可以通过常规DirectX调用直接和它进行通信。

条件是，只有当硬件能够直接支持你所要求执行的功能的时候HAL才被使用，从而达到加速的效果。

例如，当你要求绘制一个位图的时候，硬件数据块复制器（Blitter）能迅速完成这个任务，比软件循环要高效得多。

HEL，也就是硬件仿真层，运用于当硬件不支持你所要求的性能的时候。

比方说，当你要求视频卡完成位图旋转，如果硬件不支持旋转这个功能，HEL就会加入，通过软件运算来完成该项任务。

显然，这么处理速度较慢，但是关键是，这样就不会因为硬件不支持而影响你的应用程序。

程序依然可行，只是慢一点而已。

另外，HAL和HEL之间的切换对用户而言是透明的。

<<Windows游戏编程大师技巧>>

媒体关注与评论

《Windows游戏编程大师技巧（第2版）》堪称是众多游戏开发人员的启蒙经典。

作者André LaMothe是把复杂问题简单化讲解方面的大师。

时至今日，书中的游戏基础知识和理论、开发技巧，涉及到游戏开发方方面面的核心问题等，仍然很有阅读参考的价值。

沙鹰的译本也很准确可靠，得到了读者的认同。

如果你想在游戏开发方面打下扎实的基本功，推荐您读读这本书。

——腾讯游戏琳琅天上工作室总经理 姚晓光(NPC6)这是我见过的Windows游戏编程方面最好的一本书。

如果你连游戏编程什么都不知道，还想要了解如何开发游戏，甚至想要从事游戏开发工作，请一定不要错过这本书。

本书对于理论和实践的把握都很到位！

——BigWorld高级软件工程师 史晓明（polyrandom）

<<Windows游戏编程大师技巧>>

编辑推荐

《3D游戏开发大师技巧》是游戏界传奇人物大师Andre LaMothe力作，Windows游戏程序员的入行必读书，由中国较早一代游戏开发者中的代表人物之一沙鹰翻译，由姚晓光、史晓明等专业人士重磅推荐。

<<Windows游戏编程大师技巧>>

名人推荐

《Windows游戏编程大师技巧（第2版）》堪称是众多游戏开发人员的启蒙经典。

作者André LaMothe是把复杂问题简单化讲解方面的大师。

时至今日，书中的游戏基础知识和理论、开发技巧，涉及游戏开发方方面面的核心问题等，仍然很有阅读参考的价值。

沙鹰的译本也很准确、可靠，得到了读者的认同。

如果你想在游戏开发方面打下扎实的基本功，推荐你读读这本书。

——姚晓光（NPC6），腾讯游戏琳琅天上工作室总经理 这是我见过的Windows游戏编程方面最好的一本书。

如果你连游戏编程是什么都不知道，还想要了解如何开发游戏，甚至想要从事游戏开发工作，请一定不要错过这本书。

本书对于理论和实践的把握都很到位！

——史晓明（polyrandom），BigWorld高级软件工程师

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>