

<<Windows游戏程序设计基础>>

图书基本信息

书名：<<Windows游戏程序设计基础>>

13位ISBN编号：9787121145872

10位ISBN编号：7121145871

出版时间：2011-10

出版时间：电子工业出版社

作者：秦海玉 编

页数：292

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游戏程序设计基础>>

内容概要

本书讲述了如何使用visual c++来开发windows平台游戏的基础知识，主要针对想学习游戏设计以及希望进行windows程序设计的初学者。

全书分为4篇共14章。

第1篇为windows编程基础，除了基本的概念和工具的使用外，还介绍了窗口与消息机制、unicode编程、gdi编程等内容；第2篇为游戏编程基础，介绍了库的编程技术、设计模式、帧的基本概念，并搭建了一个windows下游戏编程的框架；第3篇的射击游戏综合运用前面两篇的所有知识，完成了一款射击游戏，并讲述了xml的基本编程以及关卡设计；第4篇为进一步学习打下基础，讲解了最基础的windows核心编程，包括进程管理、多线程编程、互斥访问和同步，以及direct x应用于2d编程的内容。

本书介绍了在windows环境下进行游戏编程所用到的各种基础知识，而并非局限于windows api本身的编程。

本书每个章节都以简单的实例辅助对知识点的理解，力求深入浅出，并且逐渐递增难度。

本书面向本科院校、高职院校的数字媒体技术专业、软件工程、软件技术等相关专业以及数码艺术设计爱好者。

<<Windows游戏程序设计基础>>

书籍目录

第1篇 windows编程基础

第1章 引言

- 1.1 电子游戏产业
 - 1.1.1 概述
 - 1.1.2 游戏产业价值链
 - 1.1.3 电子游戏的历史
 - 1.1.4 电子游戏产业的游戏分类
- 1.2 windows的发展历史
- 1.3 windows应用程序开发
- 1.4 一些说明
- 1.5 习题

第2章 学习使用visual c++

- 2.1 visual c++安装和使用
 - 2.1.1 visual c++的版本信息
 - 2.1.2 创建win32应用程序
- 2.2 win32控制台应用程序设计
 - 2.2.1 预编译头文件
 - 2.2.2 unicode编码
 - 2.2.3 tchar
 - 2.2.4 debug和release
 - 2.2.5 基本的调试方法
- 2.3 良好的编程习惯
 - 2.3.1 使用正确的代码格式
 - 2.3.2 采用匈牙利命名法
- 2.4 win32 api中的常见数据类型
- 2.5 习题

第3章 创建窗口与消息响应

- 3.1 windows应用程序的基本概念
 - 3.1.1 窗口
 - 3.1.2 入口函数
 - 3.1.3 消息和窗口函数
 - 3.1.4 进队消息与不进队消息
 - 3.1.5 使用msdn来学习窗口消息
- 3.2 创建窗口
 - 3.2.1 包含头文件
 - 3.2.2 定义类名和标题字符串
 - 3.2.3 注册窗口
 - 3.2.4 窗口类的风格
 - 3.2.5 创建窗口
 - 3.2.6 窗口风格
 - 3.2.7 显示窗口
 - 3.2.8 更新窗口
 - 3.2.9 消息循环
 - 3.2.10 窗口函数
 - 3.2.11 应用程序的退出

<<Windows游戏程序设计基础>>

- 3.3 关键在于应用
- 3.4 习题
- 第4章 windows程序设计基础
 - 4.1 gdi的各种概念
 - 4.1.1 什么是gdi
 - 4.1.2 图形设备上下文
 - 4.1.3 wm_paint消息
 - 4.2 gdi版本的hello, world!
 - 4.2.1 beginpaint和endpoint
 - 4.2.2 输出文本
 - 4.3 打字程序
 - 4.3.1 按键消息
 - 4.3.2 字符消息
 - 4.3.3 强制刷新
 - 4.3.4 简陋的打字程序
 - 4.4 涂鸦程序
 - 4.4.1 gdi函数
 - 4.4.2 涂鸦程序1.0
 - 4.4.3 鼠标消息
 - 4.4.4 getdc与releasedc
 - 4.4.5 设置字符串
 - 4.4.6 画线与填充
 - 4.4.7 涂鸦程序2.0
 - 4.5 时钟
 - 4.5.1 获取系统时间
 - 4.5.2 生成和销毁定时器
 - 4.5.3 获取按键的输入状态
 - 4.5.4 wm_nchittest
 - 4.5.5 使用右键菜单
 - 4.6 使用资源
 - 4.6.1 使用图标资源
 - 4.6.2 使用鼠标指针
 - 4.6.3 使用菜单
 - 4.6.4 加速键
 - 4.7 文件内核对象
 - 4.7.1 createfile
 - 4.7.2 文件大小和文件位置
 - 4.7.3 读/写文件
 - 4.7.4 与文件相关的通用对话框
 - 4.7.5 实现涂鸦文件读/写
 - 4.8 习题
- 第5章 进一步提高windows编程
 - 5.1 图像文件的格式
 - 5.2 位图
 - 5.2.1 位图的概念
 - 5.2.2 加载位图
 - 5.2.3 设备dc与内存dc

<<Windows游戏程序设计基础>>

- 5.2.4 bitblt
- 5.2.5 绘制位图文件的基本流程
- 5.2.6 完整的代码
- 5.2.7 改变窗口的位置与尺寸
- 5.3 内存dc的应用
- 5.3.1 镂空图
- 5.3.2 一个小程序
- 5.3.3 为什么会有屏闪
- 5.3.4 背景滚动
- 5.4 俄罗斯方块
- 5.4.1 设计思路
- 5.4.2 设计block
- 5.4.3 cgame的设计
- 5.4.4 cgame的实现
- 5.5 习题

第2篇 游戏编程基础

第6章 库与cximage库

- 6.1 库的基本概念
- 6.2 编译、链接、执行
- 6.3 静态库的基本编程
- 6.4 动态库编程
- 6.4.1 动态库基本编程过程
- 6.4.2 使用统一的头文件
- 6.4.3 再次审视编译链接执行
- 6.5 使用cximage库
- 6.5.1 生成cximage
- 6.5.2 通过dll使用cximage
- 6.5.3 使用cximage显示图片
- 6.5.4 旋转图片
- 6.5.5 封装cximage
- 6.6 习题

第7章 常用设计模式的介绍

- 7.1 面向对象程序设计
- 7.1.1 面向对象程序设计的三大机制
- 7.1.2 面向对象程序设计的原则
- 7.2 设计模式概论
- 7.3 模板方法模式
- 7.4 策略模式
- 7.5 习题

第8章 搭建windows游戏程序框架

- 8.1 分析win32项目的基本代码
- 8.2 游戏循环
- 8.3 “糟糕”的消息机制
- 8.3.1 同步函数与异步函数
- 8.3.2 实现游戏循环
- 8.4 利用模板方法模式搭建windows游戏程序框架
- 8.5 使用并完善框架类

<<Windows游戏程序设计基础>>

- 8.5.1 使用capplication
- 8.5.2 实现双缓冲技术避免屏闪
- 8.5.3 计算并显示帧率
- 8.6 查询按键及退出游戏的实现
 - 8.6.1 获取按键状态
 - 8.6.2 escape退出
- 8.7 游戏暂停功能的实现
 - 8.7.1 单击空格键暂停游戏
 - 8.7.2 是否当前窗口判断游戏暂停
- 8.8 六字真言：输入、计算、输出
- 8.9 习题

第3篇 射 击 游 戏

第9章 图像显示与背景

- 9.1 必要的数学运算
 - 9.1.1 坐标和角度运算
 - 9.1.2 敌机的旋转角度
 - 9.1.3 边界检测
- 9.2 管理器的概念
 - 9.2.1 怪物
 - 9.2.2 怪物管理器
 - 9.2.3 怪物仿真程序的逻辑
- 9.3 图片资源的管理
- 9.4 实现背景的滚动
 - 9.4.1 背景类cbackground的声明
 - 9.4.2 背景类cbackground的实现
 - 9.4.3 背景类cbackground的使用
- 9.5 可移动对象接口类imovable
 - 9.5.1 可移动对象
 - 9.5.2 可移动对象管理器
- 9.6 实现云彩
- 9.7 习题

第10章 飞机和子弹

- 10.1 总体设计
- 10.2 飞机
- 10.3 实现战机
- 10.4 敌机
 - 10.4.1 整体设计
 - 10.4.2 敌机类cenemy
 - 10.4.3 敌机管理器
 - 10.4.4 敌机与战机的碰撞
 - 10.4.5 使用敌机管理器类
- 10.5 敌机飞起来
 - 10.5.1 敌机与策略
 - 10.5.2 匀速策略类
 - 10.5.3 策略管理器
 - 10.5.4 直线型策略
 - 10.5.5 让敌机具备跟踪功能

<<Windows游戏程序设计基础>>

- 10.6 子弹
 - 10.6.1 总体思路
 - 10.6.2 子弹管理器
 - 10.6.3 碰撞检测
 - 10.6.4 发射
- 10.7 习题
- 第11章 实现关卡
 - 11.1 xml编程基础
 - 11.1.1 xml的基本概念
 - 11.1.2 xml文件示例
 - 11.1.3 使用xml notepad
 - 11.2 通过xml实现关卡
 - 11.2.1 使用tinyxml
 - 11.2.2 xml实现敌机类型
 - 11.2.3 xml实现子弹类型
 - 11.2.4 xml实现战机子弹
 - 11.2.5 xml实现关卡脚本
 - 11.3 习题
- 第4篇 知识拓展
- 第12章 windows体系结构
 - 12.1 多任务的操作系统
 - 12.2 windows的体系结构
 - 12.2.1 核心态
 - 12.2.2 用户态
 - 12.3 抢先式多任务工作方式
 - 12.4 windows内存管理机制
 - 12.5 多线程
 - 12.6 习题
- 第13章 windows高级编程
 - 13.1 内核对象与内核对象句柄
 - 13.1.1 内核对象
 - 13.1.2 创建内核对象
 - 13.1.3 使用计数
 - 13.1.4 进程间共享内核对象
 - 13.1.5 触发/未触发状态
 - 13.2 进程
 - 13.2.1 进程与线程
 - 13.2.2 通过任务管理器查看程序的信息
 - 13.2.3 创建子进程
 - 13.2.4 终止进程
 - 13.3 多线程编程
 - 13.3.1 进程与线程
 - 13.3.2 引入线程的好处
 - 13.3.3 如何使用多线程
 - 13.3.4 创建线程
 - 13.3.5 多个线程使用同一个线程函数
 - 13.3.6 线程的调度

<<Windows游戏程序设计基础>>

- 13.4 互斥访问共享资源
 - 13.4.1 提出问题
 - 13.4.2 互锁函数
 - 13.4.3 关键段操作
 - 13.4.4 使用内核对象保证互斥访问
- 13.5 线程同步
 - 13.5.1 单向的同步关系
 - 13.5.2 循环的同步关系
- 13.6 习题
- 第14章 使用directx完成二维游戏绘制
 - 14.1 directx概述
 - 14.2 初始化direct3d
 - 14.2.1 获取idirect3d9接口
 - 14.2.2 构造d3dpresent_parameters结构体
 - 14.2.3 创建idirect3ddevice9对象
 - 14.2.4 创建direct3d窗口
 - 14.3 显示图形
 - 14.3.1 页面切换
 - 14.3.2 在响应wm_paint消息的代码中进行绘制
 - 14.3.3 顶点与fvf (灵活顶点格式)
 - 14.3.4 三角形
 - 14.3.5 绘制二维三角形
 - 14.4 使用directgraphics进行2d图像处理
 - 14.4.1 纹理
 - 14.4.2 创建并使用纹理
 - 14.4.3 常用渲染状态
 - 14.5 把direct3d融入框架中
 - 14.5.1 纹理类
 - 14.5.2 渲染系统类
 - 14.5.3 使用渲染系统
 - 14.6 习题
- 附录a visual c++编译常见错误
- 附录b ascii码表
- 附录c 标准模板库的使用
 - c.1 stl概述
 - c.1.1 要点
 - c.1.2 头文件
 - c.1.3 名字空间
 - c.1.4 迭代器
 - c.2 常用数据容器与数据结构
 - c.3 vector
 - c.3.1 插入数据
 - c.3.2 遍历
 - c.4 list
 - c.4.1 list的迭代器
 - c.4.2 遍历
 - c.4.3 删除元素

<<Windows游戏程序设计基础>>

c.5 map

c.5.1 数据的插入和修改

c.5.2 数据的遍历

c.5.3 数据的查找

c.5.4 数据的清空与判空

c.6 使用标准模板库 (stl)

参考文献

<<Windows游戏程序设计基础>>

章节摘录

版权页：插图：自1990年代中期以来，电子游戏产业的最新做法是，游戏玩家的崛起成为游戏内容的开发者。

游戏玩家的崛起成为了游戏内容的第四方开发者，使得更多开放源代码模型的游戏设计开发和工程出现了。

玩家创建用户修改的游戏（MOD），在某些情况下与原游戏一样流行，甚至比原游戏更受欢迎。这方面的一个例子是游戏《反恐精英》，它在最初是电子游戏《半条命》一个的MOD，并最终成为一个非常成功的发行游戏。

虽然这种“修改者共享”可能只增加特定的游戏用户群的约1%，但数量的增加将提供更多的修改游戏的机会（例如发放源代码），并伴随国际玩家群体的上升而增加。

据Ben Sawyer推测，到2012年将存在多达600000的在线游戏公共开发者。

这将有效地为游戏产业价值链增加一个新的组成部分，并且如果继续走向成熟，它将融入整个行业。早期的开发成本最低，电子游戏应该是相当有利可图的。

每个游戏可由一个程序员开发，或由一个程序员和美工师组成的小团队开发，可以出售数10万份。

这些游戏中的许多只花了几个月的时间制作，所以开发商每年可以发布几个题目。

因此，发行商往往可以大有收益，如游戏销售的版税。

许多早期的游戏发行商是从这个经济气候起步的，如Origin Systems、雪乐山、卡普空、美国动视和美国艺电。

随着计算和图形处理能力的增加，开发团队的规模也随之扩大，需要大量的工作人员来解决日益复杂的图形和编程。

现在，即使是使用中间层和预先构建的游戏引擎，游戏预算也可以轻易达到数百万美元。

最专业的游戏需要一至三年的开发，这进一步提高了预算的压力。

一些开发商正在转向新的制作和发行渠道，如在线销售，以降低成本。

今天，通过主要系统和游戏等的销售对电子游戏产业的经济产生了重大的影响，例如，《侠盗猎车手》上市第一周销售额超过了5亿美元。

该游戏的收入超过了《蜘蛛侠3》的上市周末和原来的销量冠军《光环3》的总和。

许多人也受益于电子游戏经济上的成功，其中包括任天堂的前董事长、日本首富山内溥。

电子游戏产业目前面临着资金紧张，因为它既要继续盈利，又要给人才加薪。

结果是，传统的新游戏来源——游戏开发商基本上是正在消亡或纳入大发行商。

游戏产业目前正经历一个巩固的阶段和由成本升级带来的纵向一体化。

这种气候也孕育了充满活力的独立游戏开发商组成的小公司，这些小公司尝试使用互联网，而不是传统的零售渠道与玩家沟通。

电子游戏的分类标准五花八门，新的游戏类型不断涌现。

对于某一款游戏，很有可能它是若干种游戏类型的合成。

根据电子游戏内容、游戏的目的，大致可以分为以下几种：· RPG= Role - Playing Game：角色扮演游戏，由玩家扮演游戏中的一个或数个角色，有完整故事情节的游戏。

RPG游戏是最能引起玩家共鸣的游戏类型。

<<Windows游戏程序设计基础>>

编辑推荐

《Windows游戏程序设计基础》：数码艺术设计丛书之一。

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>