

<<游戏编程全接触>>

图书基本信息

书名：<<游戏编程全接触>>

13位ISBN编号：9787115132192

10位ISBN编号：7115132194

出版时间：2006-1

出版单位：人民邮电出版社

作者：（美）Mike McShaffry

页数：387

字数：684000

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>

<<游戏编程全接触>>

内容概要

本书的作者以其对游戏行业的独到见解，讲述了游戏开发的方方面面。从游戏是怎样一个行业到游戏业中形形色色的人；从琐的技巧和算法，到游戏的结构和架构；从调试技巧到进度里程碑的创建和项目进度的管理。

本书共15章，分成四大部分：第一部分“游戏编程基础”，介绍了游戏的主要组成部分及其如何相互作用，并展示了一些游戏编程工具箱中的必备工具；第二部分“让游戏跑起来”，向读者展示第一个充满内容的游戏代码示例，包括用户界面代码、2D图形代码和主循环；第三部分“构建你的游戏”，进一步讲述更复杂的代码，如3D代码、Windows游戏的特殊代码和一些优秀的debug工具；第四部分“专业游戏制作”，展示了实际的开发游戏过程，从进度安排到测试，到最终的游戏交付。

无论对于游戏项目经理，游戏程序员，还是一名对游戏充满好奇心的狂热爱好者，本书都将让读者受益匪浅。

<<游戏编程全接触>>

作者简介

Mike McShaffry在刚会敲键盘时就开始了游戏编程。
高中毕业不久他就进入了电脑游戏业的训练营——Origin Systems公司。
他参与开发了Martian Dreams、《创世纪VII：黑门》、《创世纪VIII：异教徒》、《创世纪IX：阿森松》和《网络创世纪》。
在那里工作了7年后，Mike辞职了，并于1997年创建了他的第一个公司——车库里出来的Tornado Alley。
它的目标是为孩子制作一款“成年人禁止入内”的超多用户世界。
随后他在Glass Eye Entertainment公司就职，为Monty Kerr工作。
他在那里开发了Microsoft Casino。
短短10个月之后，Monty要求Mike和他新组建的团队开创自己的公司，称为Compulsive Development，它专门和微软合作开发棋牌类游戏项目。
Mike目前是Glass Eye Entertainment公司软件开发的领导。

<<游戏编程全接触>>

书籍目录

- 第一部分 游戏编程基础第1章 游戏编程很奇怪, 因为..... 31.1 好的方面 31.1.1 工作 41.1.2 人 41.1.3 工具——软件开发包(SDK) 51.1.4 硬件 51.1.5 平台 61.1.6 会展 71.2 坏的地方 81.2.1 游戏编程难得离谱 81.2.2 其他零碎 91.2.3 那不是个bug——那是个特性 91.2.4 工具 101.3 恶心之处 101.3.1 命中移动目标 111.3.2 冲刺模式(和冲刺大餐) 111.3.3 哼, 骗子 121.3.4 操作系统地狱 121.3.5 雇员流动的天然特质 131.4 这一切都值得, 对吗?
- 13第2章 游戏中有什么 152.1 显示技术: 快速概述 152.1.1 分辨率和位深 152.1.2 核心显示技术: 2D或者3D 172.2 混合2D和3D技术 232.2.1 Grim Fandango 232.2.2 视频驱动程序和性能问题 242.3 我必须用DirectX吗 252.3.1 DirectX的设计逻辑体系 252.3.2 Direct3D或者OpenGL 262.3.3 DirectSound或者RAD游戏工具的Miles Sound 262.3.4 DirectInput或者自己实现 272.3.5 DirectPlay、TCP或者UDP 272.3.6 DirectShow或Bink 272.3.7 DirectMusic或自己实现 282.4 用户界面编码 282.5 资源缓冲 282.6 主循环 292.7 其他 30第3章 所有游戏程序员都应该了解的“简单东西” 313.1 不可或缺的设计实践 313.1.1 避免隐藏代码和非简单操作 323.1.2 类结构: 保持简单 323.1.3 继承和包含 333.1.4 变坏的虚函数 333.1.5 使用接口类 343.1.6 考虑使用工厂 343.1.7 从游戏逻辑中分离用户界面 353.1.8 实现流构造函数 353.2 智能指针和裸指针 363.2.1 引用计数 363.2.2 智能指针 373.3 正确使用内存 413.3.1 了解不同类型的内存 413.3.2 优化内存访问 443.3.3 内存对齐 453.3.4 虚拟内存 463.3.5 使用内存映射文件 463.3.6 编写自己的内存管理器 483.4 游戏脚本语言 493.4.1 使用脚本来处理文本 503.4.2 事件脚本 513.4.3 解释与编译 513.4.4 用Lex和Yacc开发自己的语言 523.4.5 Python和Lua 563.5 Mike一背包的好东西 573.5.1 一个很棒的随机数发生器 573.5.2 用Optional支持可选变量 593.5.3 集合的伪随机遍历 643.6 这并不是那么简单 67第二部分 让游戏跑起来第4章 build游戏 714.1 一点动机 714.2 创建项目 724.2.1 创建坚不可摧的目录结构 724.2.2 游戏引擎放在哪里? 754.2.3 Visual Studio build选项配置 754.2.4 build配置 764.3 源代码仓库和版本控制 764.3.1 NXN的AlienBrain 774.3.2 微软的Visual SourceSafe 784.3.3 免费的东西: CVS、RCS、..... 784.3.4 Perforce软件的Perforce 794.3.5 非常昂贵的工具: StarTeam和ClearCase 794.3.6 使用源代码控制分支 794.4 build游戏: 一门黑色艺术? 814.4.1 自动化build 824.4.2 build机器 824.4.3 自动build脚本 824.5 创建build脚本 834.5.1 标准build 834.5.2 里程碑build 844.6 多个项目和共享代码 854.7 最后的建议 86第5章 用户界面编程和输入设备 875.1 获取设备状态 875.2 使用鼠标(和游戏杆) 905.2.1 捕获鼠标 915.2.2 使用鼠标拖放 925.3 使用键盘 945.3.1 Mike的键盘窥探器 945.3.2 GetAsyncKeyState()和其他函数 975.3.3 在Windows下处理Alt键 985.4 用户界面组件 985.4.1 屏幕 985.4.2 对话框: 模态与非模态 1015.4.3 控件 1035.4.4 控件判别 1055.4.5 命中测试和焦点顺序 1065.4.6 控件状态 1075.5 更多控件属性 1075.5.1 热键 1085.5.2 工具提示 1085.5.3 上下文敏感帮助 1085.5.4 拖动 1085.5.5 声音和动画 1095.6 最后的用户界面提示 109第6章 每一个游戏开发者都应该掌握的2D知识 1116.1 2D绘制和DirectX 1126.1.1 像素和视频硬件 1126.1.2 视频硬件和缓存 1136.1.3 视频表面和内存表面 1156.2 基本2D绘制概念 1166.2.1 Windows位图和其他GDI的小玩意 1176.2.2 颜色键和色度键 1186.2.3 复制表面 1206.2.4 使用Alpha通道复制表面 1216.3 绘制文本 1276.4 使用精灵(sprite) 1306.4.1 一个基本的精灵类 1306.4.2 排序顺序和位置 1316.4.3 绘制和动画 1326.4.4 初始化精灵 1356.4.5 Restore() 1366.4.6 脏矩形绘制 1366.5 图形文件格式 1376.6 结论 138第7章 初始化和主循环 1397.1 初始化101 1397.2 C++初始化中的一些缺陷 1407.3 初始化游戏 1447.3.1 检查系统资源 1447.3.2 计算CPU速度 1457.3.3 估计VRAM 1497.3.4 加载游戏调试选项 1507.3.5 你拥有的是个垃圾袋吗 1517.3.6 初始化资源缓存 1517.3.7 使用CreateWindow来创建窗口 1527.3.8 初始化声音系统 1547.3.9 加载用户可设置的游戏选项 1547.3.10 创建绘图表面 1557.3.11 初始化游戏对象 1567.4 主循环 1567.4.1 渲染并呈现屏幕 1597.4.2 更新游戏状态 1617.4.3 一个简单的协同多任务调度器

<<游戏编程全接触>>

1637.4.4 从CProcess派生的类示例 1697.4.5 CProcess派生类的更多使用 1707.5 平稳着陆：干净漂亮地退出 1707.5.1 我该如何从这里逃脱 1717.5.2 关闭游戏 1727.6 我现在就可以做游戏了吗

173第8章 载入并缓冲资源 1758.1 图形和音乐的格式 1758.1.1 位图和纹理 1768.1.2 哪一个更好：24位、16位还是8位的图形 1768.1.3 使用有损压缩 1778.1.4 声音和音乐 1778.1.5 视频和过场电影 1788.2 资源文件 1798.2.1 打包资源到一个文件中 1808.2.2 打包资源的其他一些优点 1818.3 数据压缩 1818.4 iPac：一个资源文件生成器 1828.4.1 iPac概观 1828.4.2 iPac数据文件 1858.4.3 生成头文件 1878.4.4 其他与管理资源有关的特性 1888.5 资源高速缓存 1888.6 世界设计和缓存预测 1938.7 我的缓存不够了 196第三部分 构建你的游戏第9章 所有游戏程序员必须掌握的3D图形学知识 1999.1 3D图形流水线 1999.2 你的DirectX 9试练场 2009.3 3D数学101 2009.3.1 坐标和坐标系统 2019.3.2 矢量数学 2029.3.3 矩阵数学 2069.3.4 四元组数学 2119.3.5 视变换 2139.3.6 投影变换 2149.4 我受够数学了——请停下来 2159.4.1 三角形 2159.4.2 光照、法线和颜色 2169.4.3 贴有纹理的顶点 2189.4.4 其他顶点数据 2199.4.5 三角形网格 2199.4.6 索引的三角形网格 2219.4.7 材质 2239.4.8 纹理 2249.4.9 二次抽样(subsampling) 2259.4.10 mip-mapping 2269.5 3D图形——它那么容易 226第10章 3D引擎 22910.1 建立项目 22910.2 使用场景图 23010.2.1 场景图结点 23410.2.2 建造场景 23910.2.3 一个实用的摄像机控制器 24210.2.4 把SceneGraph放进DirectX游乐场 24710.3 遗漏了什么？ 24910.4 3D中间件回顾 25010.4.1 Renderware Graphics 25110.4.2 Intrinsic Alchemy 25110.4.3 NDL出品的NetImmerse 25210.4.4 虚幻引擎 25210.5 开发自己的3D引擎 25210.6 物理引擎 25310.7 还没满足？

254第11章 开发Windows游戏需要特殊考虑的事项 25511.1 Microsoft Foundation Class(MFC)到底怎么样 25611.1.1 MFC——你一定疯了!!？ @! 25711.1.2 MFC——你一定要使用它 25711.1.3 对MFC的最后裁决 25811.2 窗口模式和全屏模式 25911.2.1 丢失或者不兼容的表面 26011.2.2 坏窗口 26211.2.3 GDI对话框和页面翻转(Flipping) 26311.2.4 需要处理的消息 26311.2.5 WM_ACTIVATE 26411.2.6 WM_SYSCOMMAND 26511.2.7 WM_MOVE 26511.2.8 WM_DEVICECHANGE 26611.2.9 WM_POWERBROADCAST 26711.2.10 WM_DISPLAYCHANGE 26711.2.11 WM_ENTERSIZEMOVE、WM_EXITSIZEMOVE 26711.2.12 WM_GETMINMAXINFO 26711.3 操作系统相关的内容 26811.3.1 API兼容性和UNICODE 26811.3.2 游戏的注册表键 26911.3.3 Windows 95 26911.3.4 Windows 98和Windows ME 26911.3.5 Windows NT 27011.3.6 Windows 2000 27011.3.7 Windows XP 27011.4 “专为Windows设计(Designed for Windows)”标志认证程序 27011.5 结论 285第12章 调试游戏 28712.1 处理失败的艺术 28812.2 调试基础 28912.2.1 使用调试器 29112.2.2 安装Windows符号文件 29212.2.3 调试全屏游戏 29412.2.4 远程调试 29412.2.5 调试微型转储(dump) 29612.3 调试技巧 29912.3.1 调试是一个实验 30012.3.2 重现bug 30112.3.3 减小复杂度 30212.3.4 设置下一语句 30212.3.5 汇编级别调试 30312.3.6 给代码添加调料 30512.3.7 提取调试信息 30612.3.8 Lint和其他代码分析器 30612.3.9 BoundsChecker和实时分析器 30712.3.10 消失的bug 30712.3.11 调整数值 30712.3.12 caveman调试 30812.3.13 当一切办法都失败了 30812.4 各种各样的bug 30912.4.1 内存泄漏和堆溢出 30912.4.2 游戏数据损坏 31212.4.3 栈破坏 31312.4.4 剪切和粘贴的bug 31412.4.5 空间不足 31412.4.6 只有在release模式才出现的bug 31512.4.7 惹事生非的多线程 31512.4.8 奇怪的那些 31512.5 结束小思 317第四部分 专业游戏制作第13章 进度计划的艺术 32113.1 好进度、坏进度 32113.2 所有进度计划的关键：里程碑 32213.3 做计划前需要懂得的事情 33013.3.1 使用微软Project的技巧与诀窍 33013.3.2 电脑游戏中的关键编程任务 33113.3.3 基于物件的进度安排 33213.3.4 编写用户手册和cluebook 33313.3.5 了解队伍，了解游戏类型 33313.4 创建进度计划 33313.4.1 收集任务仓库 33413.4.2 关于预计时间的一些说明 33513.4.3 分配任务和平衡进度 33513.4.4 产品黑洞 33713.4.5 把进度切分成里程碑 33813.4.6 撰写项目里程碑文档 33913.4.7 编写个人里程碑文档 33913.4.8 在放出猎犬前还有一件你应该知道的事情 34013.5 把事情做对 340第14章 测试须知 34314.1 游戏为什么有bug 34314.1.1 理由1：开发组从没有见到bug重现 34314.1.2 理由2：项目

<<游戏编程全接触>>

组决定把bug置之不理 34414.2 测试计划 34514.2.1 功能测试 34514.2.2 压力测试 34814.2.3
可玩性测试 35014.2.4 可用性测试 35014.2.5 配置测试 35114.3 安排测试 35214.4 自动化的
测试 35214.5 bug数据库 35914.6 哪些bug得到修正 36514.6.1 绝不要修复这些bug 36714.6.2
bug花费高昂 36714.7 对bug数据库的静态分析 36814.8 测试组 37014.9 beta公测 37114.10
结束语 372第15章 驶向结束 37315.1 结束时的问题 37415.1.1 质量 37415.1.2 代码
37515.1.3 内容 37815.2 应付大麻烦 37915.2.1 严重滞后于计划的项目 37915.2.2 人事相关问题
38215.2.3 竞争对手会置你于死地 38415.2.4 没有出路,是吗?
38415.2.5 最后说一句:不要慌张 38515.3 光明就在前方——毕竟不是一场训练 38515.3.1 测
试存档 38515.3.2 补丁build或者产品演示 38615.3.3 事后分析 38615.3.4 如何利用你的时间
387

<<游戏编程全接触>>

媒体关注与评论

大师的倾情奉献，商业级电脑游戏开发过程的全面演绎！

作为Origin System公司（《银河飞将》和《创世纪》系列的开发商）和微软的主程序员，从第一页开始，MIKE MCSHAFFRY就向你展示了如何创建和编写最棒的交互游戏。

本书含有大量的技巧和代码示例，再加上对关键非代码因素的广泛叙述，它成为了创作优美的BUG高品质游戏代码的终极指南。

这些游戏能够通过如电子艺界和微软这样的顶级游戏发行商的体验。

本书涵盖了编程、团队合作、设计合适的游戏架构、代码优化等等游戏开发的全部关键方面，也包含了在《网络创世纪》这样的前沿游戏项目中磨练成长的程序员们所使用的大量秘密技巧和窍门。

所有具体编程概念的介绍都使用C++语言。

此书必将成为所有游戏开发者手中的经典！

<<游戏编程全接触>>

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>