

<<游戏编程精粹3>>

图书基本信息

书名：<<游戏编程精粹3>>

13位ISBN编号：9787115108708

10位ISBN编号：7115108706

出版时间：2003-10

出版时间：人民邮电出版社

作者：（美）特里格利亚 编，张磊 译

页数：565

字数：916000

译者：张磊

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>

<<游戏编程精粹3>>

内容概要

本书汇集了近70篇最新的游戏编程大师的技术文章。

这些文章都来自于实际经验的积累，各有独到之处，依其所属领域不同，全书划分为通用编程技术、数学技巧、人工智能、图形、网络和多玩家游戏、音频处理六章，覆盖了当今游戏开发中的所有关键技术领域。

本书适合游戏开发专业人员阅读。

对于入门级的读者，本书指出了您将要面临的各方面挑战，并提供大量的参考资料和资源助您提高专业知识和技术；对于专家级的读者，本书中实用的新思想与新技巧将帮助您节省大量游戏开发的宝贵时间。

作者简介

THOR ALEXANDER在过去的10年中致力于为游戏界创造更为可信的智能人物角色。最近，他创建了HARD CODED GAMES公司，为在线游戏创造具有艺术效果的人工智能和机器学习。在此之前，他在ELECTRONIC ARTS、微软还有XATRIX ENTERTAINMENT担任过资深人工智能开发和设计人员。

书籍目录

第1章 通用编程技术简介 2Kim Pallister1.1 调度游戏中的事件 4Michael Harvey, Carl S. Marshall1.1.1 调度器的组成 51.1.2 一个简单的调度器 81.1.3 高级概念 101.1.4 结论 111.1.5 参考文献 121.2 一个基于对象组合的游戏架构 13Scott Patterson1.2.1 游戏开发的各个阶段 131.2.2 游戏架构设计 141.2.3 游戏架构实现 171.2.4 源代码 201.2.5 参考文献 211.3 让C中的宏重现光辉 231.3.1 声明 231.3.2 第1个宏技巧:把枚举值转化为字符串 231.3.3 第2个宏技巧:利用二进制表达式得到编译期常量 251.3.4 第3个宏技巧:给标准断言添加描述性注释 261.3.5 第4个宏技巧:编译期断言 261.3.6 第5个宏技巧:得到一个数组里面的元素个数 271.3.7 第6个宏技巧:在一个字符串中间加入__LINE__ 271.3.8 第7个宏技巧:防止进入无限循环 281.3.9 第8个宏技巧:小型的特制语言 291.3.10 第9个宏技巧:简化类接口 301.3.11 结论 331.3.12 参考文献 331.4 平台无关的函数绑定代码生成器 34Allen Pouratian1.4.1 年轻与智慧 341.4.2 概要 351.4.3 细节 361.4.4 脚本 381.4.5 网络 381.4.6 结论 391.4.7 参考文献 391.5 基于句柄的智能指针 40Brian Hawkins1.5.1 用法 401.5.2 句柄 411.5.3 智能指针 421.5.4 结论 431.5.5 参考文献 431.6 定制STL分配器 44Pete Isensee1.6.1 一个范例 441.6.2 分配器的基础 451.6.3 分配器的要求 451.6.4 缺省的分配器对象 491.6.5 编写自己的分配器 491.6.6 潜在的用途 511.6.7 分配器状态数据 511.6.8 一些建议 521.6.9 实现细节 521.6.10 结论 521.6.11 参考文献 531.7 立即存盘 54Martin Brownlow1.7.1 为何如此困难 541.7.2 SAVEMGR类 551.7.3 SAVEOBJ类 551.7.4 数据类型与扩展 561.7.5 重载缺省函数 561.7.6 一个简单的例子 571.7.7 结论 581.8 自动列表设计模式 59Ben Board1.8.1 实现 591.8.2 实现时的注意事项 611.8.3 结论 631.9 浮点异常处理 64Soren Hannibal1.9.1 为什么要崩溃 641.9.2 你的程序处理浮点异常么 651.9.3 异常的类型 651.9.4 代码 651.9.5 调试浮点错误 661.9.6 结论 661.10 使用UML开发一个配合设计的游戏引擎 67Thomas Demachy1.10.1 对象就在游戏之中 671.10.2 动态的类--正如动态的棋子 701.10.3 协作与迭代 721.10.4 实现上的问题 731.10.5 结论 741.10.6 参考文献 751.11 使用Lex和Yacc分析自定义数据文件 76Paul Kelly1.11.1 Lex 771.11.2 Yacc 771.11.3 优点与缺陷 771.11.4 Yacc和Lex中的交互 781.11.5 针对游戏子系统的自定义数据文件 791.11.6 把数据输出工具与Lex和Yacc结合起来 801.11.7 一个完整的例子 801.11.8 结论 841.11.9 如何得到Flex和Bison 841.11.10 参考文献 841.12 为世界市场开发游戏 85Aaron Nicholls1.12.1 市场潜力 851.12.2 门面事,先处理--显示和输入 861.12.3 字符集 881.12.4 界面和设计方面的考虑 901.12.5 本地化 931.12.6 设计和规划中的考虑 941.12.7 测试 951.12.8 结论 981.12.9 参考文献 981.13 3D游戏中的实时输入和用户界面 99Greg Seegert1.13.1 实现用户界面 991.13.2 指定用户界面元素 1001.13.3 本地化问题 1011.13.4 输入系统 1021.13.5 鼠标与操纵杆 1031.13.6 在处理延迟方面用户界面的作用 1041.13.7 结论 1051.13.8 参考文献 1051.14 自然的选择:饼状菜单的演化 106Don Hopkins1.14.1 Feng GUI的饼状菜单 1061.14.2 对饼状菜单的研究与评估 1071.14.3 饼状菜单插件 1081.14.4 未来发展方向 1121.14.5 走进SimCity中的城镇 1121.14.6 Sims中的起居室 1141.14.7 结论 1151.14.8 参考文献 1151.15 轻量级的、基于规则的日志记录 117Brian Hawkins1.15.1 规则 1171.15.2 调试标志 1171.15.3 配置文件 1181.15.4 可配置的标志值 1191.15.5 日志记录 1191.15.6 用法 1211.15.7 结论 1221.15.8 参考文献 1221.16 日志服务 123Eric Robert1.16.1 管理信息 1231.16.2 系统层次 1241.16.3 Journal接口 1271.16.4 创建日志服务 1291.16.5 结论 1321.16.6 参考文献 1321.17 实时的层次化性能评测 133Greg Hjeistrom, Byon Garrabrant1.17.1 性能评测树 1341.17.2 用法 1341.17.3 实现 1361.17.4 结论 1391.17.5 参考文献 139第2章 数学技巧简介 141John Byrd2.1 对数与随机数生成的2基快速函数 143James McNeill2.1.1 整数的2基对数 1432.1.2 位掩码与随机数生成 1432.1.3 函数是如何工作的 1452.1.4 参考文献 1452.2 使用分数矢量得到更精确的几何图形 146Thomas Young2.2.1 问题 1462.2.2 一个解决方法:分数矢量 1492.2.3 使用分数矢量 1502.2.4 数字的范围 1512.2.5 实现上的细节 1522.2.6 结论 1532.2.7 参考文献 1532.3 三角函数的更多近似计算方法 154Robin Green2.3.1 衡量误差 1542.3.2 正弦与余弦函数 1552.3.3 多项式逼近 1622.3.4 有关收敛性的注意事项 1662.3.5 结论 1672.3.6 参考文献 1672.4 四元数的压缩 168Mark Zarb-Adami2.4.1 四元数 1682.4.2 三个最小数方法 1682.4.3 极点方法 1692.4.4 实现 1692.4.5 性能 1702.4.6 结论

1712.4.7 答谢 1712.4.8 参考文献 1712.5 受限的逆向运动学 172Jason Weber2.5.1 骨节层次
 1722.5.2 循环坐标推演 1732.5.3 旋转限制 1742.5.4 调整每个骨节,同时保持限制 1752.5.5 结论
 1772.5.6 参考文献 1772.6 针对物理建模的单元自动机 179Tom Forsyth2.6.1 CA基础 1792.6.2 八
 叉树 1822.6.3 实际的物理 1822.6.4 核心处理模型 1832.6.5 气体 1842.6.6 水流 1842.6.7 流速
 1852.6.8 热量 1862.6.9 火焰 1882.6.10 动态更新速率 1892.6.11 结论 1902.6.12 参考文献 1912.7
 在动态仿真中处理摩擦 192Miguel Gomez2.7.1 库仑摩擦力 1922.7.2 数值方法 1962.7.3 一个三维公
 式 1992.7.4 几何图形问题 2002.7.5 结论 2012.7.6 参考文献 201第3章 人工智能简介 203Steven
 Woodcock3.1 经GoCap优化过的机器学习 205Thor Alexander3.1.1 GoCap架构一览 2053.1.2 训练开
 车 2073.1.3 学习规则 2083.1.4 结论 2123.1.5 参考文献 2123.2 区域游览:对寻径模式的扩展
 213Ben Board, Mike Ducker3.2.1 辞旧 2143.2.2 迎新 2153.2.3 分而治之 2183.2.4 路径遍历
 2203.2.5 对此模式的扩展 2243.2.6 结论 2243.2.7 参考文献 2243.3 基于函数指针的内嵌式有限状
 态机 225Charles Farris3.3.1 什么是有限状态机 2253.3.2 FSM的实现 2263.3.3 实现CFSM 2283.3.4
 使用CFSM 2323.3.5 结论 2343.3.6 参考文献 2353.4 在RTS中的地形分析--一个隐藏的重要因
 素Daniel Higgins3.4.1 区域 2363.4.2 凸包 2403.4.3 重要的匹配器 2433.4.4 关隘 2463.4.5 进行地形
 分析 2493.4.6 结论 2503.4.7 参考文献 2503.5 一个针对AI代理、对象,以及任务的可扩展触发器系
 统 251Steve Rabin3.5.1 触发器系统简介 2513.5.2 对象自有的触发器系统 2523.5.3 定义条件
 2523.5.4 使用布尔逻辑组合条件 2523.5.5 定义响应 2543.5.6 求取触发器的值 2543.5.7 一次性触
 发与载入次数 2553.5.8 使用标志和计数器将触发器结合起来 2563.5.9 触发器系统与脚本语言的对比
 2573.5.10 局限性 2583.5.11 结论 2583.5.12 参考文献 2583.6 基于A*算法的战术式寻径
 259William van der Sterren3.6.1 有风险的A* 2603.6.2 对于有缺陷路径的战术式改良 2623.6.3 暴露时
 间与对敌人建模 2623.6.4 威胁并不仅仅是静态的 2643.6.5 更战术化的改进 2653.6.6 性能 2653.6.7
 有效的火力线以及视野的探测 2663.6.8 扩展的A*算法的代价 2673.6.9 ASE程序 2683.6.10 结论
 2683.6.11 参考文献 2693.7 快速游览网格的方法 270Stephen White, Christopher Christensen3.7.1 静
 态障碍与动态障碍 2703.7.2 游览网格 2703.7.3 门户 2723.7.4 建表 2743.7.5 其他的门户相关问题
 2753.7.6 表示生物 2763.7.7 动态障碍 2773.7.8 在静态障碍与动态障碍之间进行游览 2793.7.9 有
 关游览网格的其他想法 2793.7.10 结论 2803.8 在寻径与碰撞之间选择一种关系 281Thomas
 Young3.8.1 在碰撞控制下的运动 2813.8.2 对于寻径的碰撞模型 2813.8.3 方法1:具有容错性的AI
 2823.8.4 方法2:在无障碍空间一个子集内的寻径 2843.8.5 方法3:使用寻径器本身处理人物碰撞
 2863.8.6 实现沿路的运动 2883.8.7 结论 2893.8.8 参考文献 289第4章 图形简介 292Jeff Lander4.1
 消除T形连接与重新三角化 295Eric Lengyel4.1.1 T形连接的消除 2964.1.2 重新三角化 2974.1.3 实现
 2984.1.4 结论 2994.2 快速高程场法线的计算 300Jason Shankel4.2.1 一个任意网格上的法线
 3004.2.2 高程场法线 3014.2.3 结论 3034.2.4 例子程序 3034.2.5 参考文献 3044.3 快速计算面片
 法线 305Martin Brownlow4.3.1 定义 3054.3.2 传统方法 3054.3.3 相关问题 3064.3.4 一个更简单
 的方法 3064.3.5 其他的优点 3064.3.6 此方法的精确度有多大 3074.3.7 结论 3074.3.8 参考文献
 3074.4 快速、简单的遮蔽剪裁 308Wagner T. Corrêa, Princeton University4.4.1 可见性问题
 3084.4.2 PLP算法 3094.4.3 cPLP算法 3104.4.4 讨论 3104.4.5 实验结果 3114.4.6 结论 3124.4.7
 参考文献 3124.5 三角形条带的创建、优化以及渲染 314Carl S. Marshall4.5.1 三角形条带 3144.5.2
 三角形条带的创建 3154.5.3 优化 3184.5.4 渲染 3194.5.5 倾向于缓存的三角形条带 3194.5.6 连续
 分层细节的三角形条带 3194.5.7 结论 3204.5.8 参考文献 3204.6 针对复杂数据集计算优化阴影体
 321Alex Vlachos, Drew Card4.6.1 前期工作 3214.6.2 算法 3214.6.3 优化算法 3234.6.4 参考文献
 3244.7 针对人物运动的表面细分 325William Leeson4.7.1 各种细分模式 3254.7.2 骨节的层次化结构
 以及顶点积累缓冲 3294.7.3 优化 3304.7.4 系统集成 3324.7.5 源代码 3334.7.6 结论 3344.7.7 参
 考文献 3344.8 改良的骨节变换计算 335Jason Weber4.8.1 背景知识 3354.8.2 简单的方法 3364.8.3 添
 加骨节 3374.8.4 改变权重 3384.8.5 系统集成与优化 3404.8.6 结论 3424.8.7 参考文献 3424.9 针
 对真实人物运动的架构 343Thomas Young4.9.1 问题:针对任意目标的运动 3434.9.2 问题:运动之
 间的平滑过渡 3454.9.3 解决问题的一个架构:局部修正器与独立的插值系数 3464.9.4 应用:处理任
 意目标的运动 3474.9.5 位移修正器 3484.9.6 应用:变换 3494.9.7 其他细节 3494.9.8 结论 3504.9.9

参考文献 3504.10 可编程顶点着色器的编译器 351Adam Lake4.10.1 可编程顶点着色器 3514.10.2 编译器 3534.10.3 编译器的组成部分 3534.10.4 结论 3574.10.5 致谢 3574.10.6 参考文献 3574.11 画板光束 359Brian Hawkins4.11.1 矩阵 3594.11.2 顶点 3604.11.3 UV映射 3614.11.4 结论 3614.12 针对等测引擎的3D技术 362Greg Snook4.12.1 进入第三个维度 3634.12.2 方法1:画板越多,效果越好 3634.12.3 方法2:变换纹理 3644.12.4 方法3:垂直插值的纹理 3664.12.5 结论 3674.12.6 参考文献 3674.13 使用法向地图进行曲面模拟 368Oscar Blasco4.13.1 法向地图 3684.13.2 整个过程的纵览 3694.13.3 数据准备 3694.13.4 投影线 3704.13.5 得到细节信息 3714.13.6 后处理 3724.13.7 已知的问题 3724.13.8 其他方法 3734.13.9 结论 3734.13.10 致谢 3744.13.11 参考文献 3744.14 动态的、具有照片效果的地形光照 375Naty Hoffman, Kenny Mitchell4.14.1 背景知识 3754.14.2 解的分类 3774.14.3 日照:地平角、椭圆阴影以及PTM 3774.14.4 天空光照:辐射透过量的近似与分块 3794.14.5 活动的云层阴影 3804.14.6 基于视频的解决方案 3824.14.7 非地形对象 3834.14.8 结论 3834.14.9 参考文献 3834.15 立体图光照技术 385Kenneth L. Hurley4.15.1 立体图的物理属性 3854.15.2 如何与立体图进行数据交换 3864.15.3 使用立体图进行渲染 3864.15.4 对云层进行编码 3874.15.5 在一个立体图中对光源进行编码 3904.15.6 在立体图中渲染散射光照 3904.15.7 将日夜循环编码进立体图中 3914.15.8 结论 3914.15.9 参考文献 3914.16 程序纹理 392Mike Milliger4.16.1 参数与函数 3924.16.2 进入游戏世界 3934.16.3 硬件加速 3954.16.4 结论 3964.16.5 致谢 3974.16.6 参考文献 3974.17 独一无二的纹理 398Tom Forsyth4.17.1 程序纹理 3984.17.2 智能纹理缓存 3994.17.3 合成模型 3994.17.4 层的映射与变换 3994.17.5 层的源与过滤器 4004.17.6 合成方法 4004.17.7 对数字的控制 4014.17.8 动态纹理 4014.17.9 可扩展性 4024.17.10 使用CPU还是图形芯片进行合成运算 4034.17.11 演示程序 4034.17.12 结论 4044.17.13 参考文献 4044.18 使用纹理作为查找表进行逐像素光照计算 405Alex Vlachos, John Isidoro, Chris Oat4.18.1 不使用立体图进行h归一化(n.h/h.h映射)的镜面逐像素光照 4054.18.2 使用一个(n.h)k图的逐像素镜面指数 4074.18.3 色彩偏移的光晕 4094.18.4 拥有正确的逐像素衰减的逐像素点光照 4104.18.5 拥有正确的逐像素衰减的逐像素聚光灯与方向性光照 4114.18.6 结论 4134.18.7 参考文献 4134.19 使用手工制作的着色模型进行渲染 414Jan Kautz4.19.1 着色模型 4144.19.2 基于微表面的着色模型 4154.19.3 NDF着色 4154.19.4 使用NDF的凸凹贴图 4174.19.5 扩展 4184.19.6 结论 4184.19.7 参考文献 418第5章 网络和多玩家游戏简介 421Andrew Kirmse5.1 将实时策略游戏中的延迟最小化 422Jim Greer, EA.com, Zachary Booth Simpson, Mine Control5.1.1 帧锁定与事件锁定 4225.1.2 时间同步 4265.1.3 结论 4285.1.4 参考文献 4285.2 实时策略网络协议 429Jan Svarovsky5.2.1 其他的协议 4295.2.2 我们的协议 4305.2.3 精炼 4325.2.4 有用的模块 4345.2.5 在StarTopia中容易犯的错误 4355.2.6 示例游戏 4365.2.7 结论 4365.2.8 参考文献 4365.3 一个针对巨量多玩家游戏的灵活的仿真架构 437Thor Alexander5.3.1 架构一览 4375.3.2 支持类 4385.3.3 核心类 4405.3.4 管理器与工厂 4435.3.5 把它们都组合起来 4455.3.6 结论 4485.3.7 参考文献 4485.4 对多玩家游戏进行扩展 449Justin Randall5.4.1 改善游戏公平度的策略 4495.4.2 设计可扩展的服务器 4515.4.3 分布负载 4565.4.4 优化 4585.4.5 结论 4605.4.6 参考文献 4615.5 基于模板的对象序列化 462Jason Beardsley5.5.1 现存的解决方案 4625.5.2 可移植性 4645.5.3 Serializer类 4655.5.4 扩展与优化 4705.5.5 未来的工作 4725.5.6 结论 4725.5.7 参考文献 4725.6 安全套接字 473Pete Isensee5.6.1 IPsec 4735.6.2 警告 4745.6.3 安全连接 4745.6.4 包格式 4755.6.5 发送数据 4765.6.6 接收数据 4775.6.7 示例实现 4785.6.8 CryptoAPI 4805.6.9 性能 4805.6.10 安全性 4815.6.11 结论 4815.6.12 参考文献 4825.7 一个网络监控与模拟工具 483Andrew Kirmse5.7.1 界面 4835.7.2 网络监控 4845.7.3 TCP模拟 4845.7.4 UDP模拟 4845.7.5 主机带宽模拟 4855.7.6 结论 4855.8 使用DirectPlay8.1创建多玩家游戏 486Gabriel Rohweder5.8.1 DirectPlay内幕 4865.8.2 数据传输 4875.8.3 可重入的回调函数 4905.8.4 使用DirectPlay发送语音 4935.8.5 相关资源 4965.9 使用Java微型版开发无线游戏 497David Fox5.9.1 网络特性 4975.9.2 Java微型版 4985.9.3 J2ME网络精髓 4995.9.4 HTTP的限制 5005.9.5 优化数据包 5015.9.6 从服务器获取图像 5025.9.7 结论 5035.9.8 参考文献 504第6章 音频处理简介 506Scott Patterson6.1 使用Ogg Vorbis进行音频压缩 508Jack Moffitt6.1.1 心理声学压缩 5086.1.2 使用压缩的情况 5106.1.3 使用Ogg的代码示例 5116.1.4 结论 5146.1.5 参考文献 5146.2 创建一个美妙的3D音频

环境 515Garin Hiebert6.2.1 3D音频的核心概念 5156.2.2 有效地使用你的音频引擎 5166.2.3 实现
5176.2.4 结论 5186.2.5 参考文献 5186.3 使用轴对齐的边界框设置音障 520Carlo Vogelsang6.3.1 问
题 5206.3.2 解决方法 5216.3.3 实现 5246.3.4 结论 5246.3.5 参考文献 5246.4 使用双二次共振滤
波器 525Phil Burk6.4.1 数字滤波器的工作原理 5256.4.2 IIR与FIR滤波器 5266.4.3 双二次滤波器的实
现 5266.4.4 改变变量 5276.4.5 避免异常情况 5276.4.6 控制滤波器 5286.4.7 计算此滤波器的系数
5286.4.8 低通滤波器 5296.4.9 高通滤波器 5296.4.10 带通滤波器 5296.4.11 将滤波器进行串联
5306.4.12 将滤波器进行并联 5306.4.13 软件 5306.4.14 结论 5306.4.15 参考文献 5306.5 语音压
缩与音效的线性预测编码 531Eddie Edwards6.5.1 对语音建模 5326.5.2 软件仿真 5336.5.3 取代声带
5356.5.4 控制重新合成器 5366.5.5 增加扬声器的深度 5376.5.6 对数据编码 5376.5.7 速度
5386.5.8 实验 5386.5.9 参考文献 5386.6 复杂声音的随机合成方法 539Phil Burk6.6.1 线性同余算
法 5396.6.2 噪声种类 5406.6.3 软件示例 5416.6.4 软件 5456.6.5 结论 5456.6.6 参考文献 5456.7
针对游戏的实时模块化音频处理 546Frank Luchs6.7.1 模块化音频处理 5466.7.2 通过程序生成声音
5476.7.3 Sphinx MMOS系统 5476.7.4 处理器 5486.7.5 模块文件简介 5486.7.6 模块文件的应用
5506.7.7 源代码 5526.7.8 结论 5536.7.9 参考文献 553索引 554

<<游戏编程精粹3>>

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>