

<<Lua游戏开发实践指南>>

图书基本信息

书名：<<Lua游戏开发实践指南>>

13位ISBN编号：9787111403357

10位ISBN编号：7111403355

出版时间：2013-1

出版时间：机械工业出版社

作者：Paul Schuytema, Mark Manyen

页数：246

译者：田剑

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>

<<Lua游戏开发实践指南>>

内容概要

本书是资深Lua游戏开发工程师10余年工作经验和智慧结晶，Lua语言创始人亲自作序推荐，是Lua游戏开发领域最具实战意义和代表性的著作之一。

它不仅详细讲解了在游戏开发中使用Lua的各种技术细节、方法技巧和最佳实践，而且讲解了如何使用Lua作为主要工具将游戏设计转化为代码实现的过程。

此外，它还重点阐述了Lua语言的核心要素。

最重要的是，本书包含大量精心设计的案例，并附赠了完整的源代码，可操作性极强。

全书一共15章：第1~3章简单地介绍了Lua语言的特性、授权，以及在游戏开发中的强大用途；第4~5章详细讲解了Lua语言的基本语法和核心要素；第6~7章讲解了Lua与C/C++程序的整合以及与C++的交互相关的技术细节；第8~9章介绍了开发前需要做的准备工作，以及如何设计Lua版本的实现；第10章讲解了如何使用Lua来处理游戏数据；第11章讲解了Lua驱动的GUI；第12章详细讲解了两个完整的游戏开发案例；第13章结合实例讲解了如何使用Lua定义和控制AI；第14章展示了Lua在图形绘制和图像处理方面的强大功能；第15章探讨了Lua与多媒体、Lua脚本的调试、Lua应用的资源管理以及Lua代码的发布等内容。

<<Lua游戏开发实践指南>>

作者简介

Paul

Schuytema 资深游戏开发专家兼Lua开发工程师，在游戏开发领域研究和实践10余年，积累了丰富的经验。

曾经在过去的10届全球游戏开发大会（GDC）中发表过重要演讲，在游戏开发领域颇有影响力。他设计并开发了20余款经典的游戏，发表了200多篇与游戏开发相关的专业文章，而且他还是多本书籍的作者或合著者。

Mark Manyen 资深游戏开发专家兼Lua开发工程师，Magic

Lantern的技术总监，在超过25款游戏中指导并使用了该技术。

他在Magic Lantern's J5

技术套件中引入了Lua，并且开发了所有与J5代码库相关联的“Lua Glue”函数。

<<Lua游戏开发实践指南>>

书籍目录

译者序

序

前言

致谢

第1章 游戏开发入门

1.1 越来越复杂的开发过程

1.2 更好的开发方式

1.3 为什么使用Lua

1.4 本章小结

第2章 脚本语言

2.1 脚本语言简介

2.2 Lua简介

2.2.1 Lua的历史

2.2.2 Lua授权

2.3 本章小结

第3章 游戏开发世界的Lua语言

3.1 脚本语言和游戏

3.2 游戏项目中的Lua

3.2.1 游戏界面

3.2.2 管理游戏数据

3.2.3 事件处理

3.2.4 保存和读取游戏状态

3.2.5 人工智能

3.2.6 快速构建原型

3.3 本章小结

第4章 Lua入门

4.1 使用Lua控制台

4.2 Lua基础

4.3 变量

4.3.1 nil

4.3.2 Boolean

4.3.3 string

4.3.4 Number

4.3.5 table

4.3.6 局部变量和全局变量

4.4 运算符

4.4.1 算术运算符

4.4.2 关系运算符

4.4.3 逻辑运算符

4.5 控制结构

4.5.1 if

4.5.2 while和repeat

4.5.3 for

4.5.4 break

4.6 本章小结

<<Lua游戏开发实践指南>>

第5章深入学习Lua

5.1 函数

5.1.1 单一参数

5.1.2 多个参数

5.1.3 返回值

5.2 标准库

5.2.1 assert(myValue())

5.2.2 dofile(filename)

5.2.3 math.floor()

5.2.4 math.random()

5.2.5 math.min()

5.3 字符处理

5.3.1 类型转换

5.3.2 string.char(n1, n2, ...)

5.3.3 string.len(myString)

5.3.4 string.sub(myString, start, end)

5.3.5 string.format()

5.3.6 string.find(sourceString, findString)

5.3.7 字符和格式

5.4 table数据结构

5.4.1 table.getn(myTable)

5.4.2 table.insert(myTable, position, value)

5.4.3 table.remove(myTable, position)

5.4.4 table引用

5.4.5 多维table

5.4.6 pairs()

5.5 I/O基础

5.6 本章小结

第6章Lua与C/C++程序的整合

6.1 初期设计要点

6.1.1 Lua环境

6.1.2 LuaGlue函数

6.2 基本实现方式

6.2.1 创建Lua运行环境

6.2.2 添加LuaGlue函数

6.2.3 命令处理

6.2.4 退出程序

6.2.5 cLua对象和LuaLib

6.2.6 使用cLua的例子

6.2.7 LuaGlue函数的优点

6.2.8 LuaGlue函数：参数和返回值

6.3 本章小结

第7章Lua与C++的交互

7.1 重新审视LuaGlue函数

7.2 C++代码和Lua的交互

7.3 事件驱动的编程

<<Lua游戏开发实践指南>>

- 7.3.1 示例事件
- 7.3.2 事件的参数
- 7.4 错误处理
- 7.5 本章小结
- 第8章 开发准备
- 8.1 Visual C++ 6.0 工作区
- 8.2 DirectX 基础
- 8.3 Lua GUI 简介
- 8.3.1 启动 GUI
- 8.3.2 界面
- 8.3.3 界面控件
- 8.3.4 事件
- 8.3.5 与 GUI 系统相关的 LuaGlue 函数
- 8.3.6 Shell 程序的扩展
- 8.4 调试窗口
- 8.5 Windows 注册表
- 8.6 本章小结
- 第9章 设计 Lua 版本的实现
- 9.1 游戏设计原则
- 9.1.1 什么是游戏
- 9.1.2 了解玩家的想法
- 9.2 基础库设定
- 9.3 设计文档
- 9.4 Lua 编程规范
- 9.5 本章小结
- 第10章 使用 Lua 处理游戏数据
- 10.1 简单的游戏数据
- 10.1.1 太空飞船的例子
- 10.1.2 《Take Away》的玩家飞船
- 10.1.3 敌舰数据
- 10.1.4 补给箱数据
- 10.2 大数据集
- 10.2.1 表单型数据
- 10.2.2 Lua 格式的数据文件
- 10.3 使用 Lua 保存游戏数据
- 10.3.1 案例1——《Frontrunner》
- 10.3.2 案例2——健身大亨
- 10.4 本章小结
- 第11章 Lua 驱动的 GUI
- 11.1 GUI 系统概要
- 11.2 GUI 的 C++ 类
- 11.2.1 GUI 控件：Sprite
- 11.2.2 GUI 控件：TextField
- 11.2.3 GUI 控件：Button
- 11.2.4 界面
- 11.2.5 GUI 管理器
- 11.3 GUI LuaGlue 函数

<<Lua游戏开发实践指南>>

- 11 4进一步的说明
- 11 5Lua游戏界面
 - 11 5 1界面设计原则
 - 11 5 2快速创建界面
 - 11 5 3载入界面
 - 11 5 4主菜单界面
 - 11 5 5Controls界面
 - 11 5 6InGame界面
- 11 6本章小结
- 第12章Lua游戏编程
 - 12 1游戏主循环
 - 12 2井字棋
 - 12 2 1游戏的初始化
 - 12 2 2游戏回合处理
 - 12 2 3模拟游戏回合
 - 12 3《Take Away》游戏的实现原理
 - 12 3 1InGame
 - 12 3 2使用计时器
 - 12 3 3玩家操作
 - 12 3 4子弹运动
 - 12 3 5飞船移动
 - 12 3 6绘制活动的物体
 - 12 4本章小结
- 第13章使用Lua定义和控制AI
 - 13 1智能的体现
 - 13 2 1点游戏
 - 13 3井字棋
 - 13 4《Take Away》游戏的实现
 - 13 4 1掠夺舰
 - 13 4 2攻击舰
 - 13 4 3冲击舰
 - 13 4 4混合舰
 - 13 4 5控制飞行方向
 - 13 4 6碰撞检测
 - 13 5其他AI的例子
 - 13 5 1静态追踪
 - 13 5 2近距离追踪
 - 13 5 3动态追踪
 - 13 5 4预判型追踪
 - 13 5 5炮塔攻击
 - 13 5 6躲避攻击
 - 13 5 7防御性射击
 - 13 5 8攻击伤害
 - 13 6有限状态机
 - 13 7路径寻找
 - 13 7 1算法概要
 - 13 7 2路径寻找示例

<<Lua游戏开发实践指南>>

- 13.7.3 Lua实现
- 13.8 本章小结
- 第14章 Lua和图像
- 14.1 运行绘图示例
- 14.1.1 指纹示例
- 14.1.2 爆炸示例
- 14.2 线性移动
- 14.2.1 GetCollisions函数
- 14.2.2 HitTest函数
- 14.2.3 进一步的说明
- 14.3 碰撞检测
- 14.3.1 LuaGlue函数SetTexture
- 14.3.2 坦克示例
- 14.4 2D粒子系统
- 14.5 本章小结
- 第15章 最后说明
- 15.1 添加音效和音乐
- 15.1.1 LuaGlue函数PlaySound
- 15.1.2 音乐
- 15.2 使用编辑器
- 15.3 调试Lua脚本
- 15.3.1 通用原则
- 15.3.2 调用DoFile函数
- 15.3.3 Lua 错误消息
- 15.3.4 使用实时调试窗口
- 15.3.5 使用文本框
- 15.3.6 使用文件输出
- 15.4 资源管理
- 15.4.1 资源的组织
- 15.4.2 运行时的文件夹
- 15.5 发布Lua代码
- 15.6 许可证
- 15.7 进一步的说明
- 15.8 本章小结

<<Lua游戏开发实践指南>>

章节摘录

版权页：插图： Lua在游戏开发中的一个很强大的特性是它能够作为保存、载人游戏数据的主要工具。

所有的游戏都和数据相关，在脚本层处理数据，意味着游戏设计师和脚本程序员可以独立于底层的核心代码，访问所有的游戏数据并且设计符合游戏需要的数据存储方式。

Lua还可以被用于保存游戏进度、游戏开始时的数据载入，以及玩家载人游戏进度。

在本章中，我们将要了解如何使用Lua变量和表来存放运行时数据，以及如何输出数据保存游戏进度。

10.1简单的游戏数据 几乎所有的游戏都有很重要的数据组件。

一些简单的经典游戏，如《space Invader》（太空侵略者）、《Asteroids》（爆破彗星），在运行时只有很少的数据，但也包含了基本的信息。

更多的现代游戏，如《Doom3》（毁灭战士3）或者《Neverwinter Nights》（无冬之夜），在运行时会有大量的数据产生。

有些数据在C++领域是十分清楚的。

想象一下3D第一人称射击游戏，在每一帧的图形绘制中，计算决定了屏幕上显示的场景以及用来表示动态物体的数据，如那些玩家射击的怪物和玩家自己控制的3D角色，这些物体都是由成百上千个点和三角形组成。

<<Lua游戏开发实践指南>>

编辑推荐

《Lua游戏开发实践指南》从最简单的HelloWorld到复杂的人工智能和路径搜索，使用了大量的例子为初学者详细解释了Lua语言的方方面面，并带领大家从游戏设计开始逐步实现游戏的快速原型，展示了完整的游戏开发流程。

<<Lua游戏开发实践指南>>

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>