

《Lua实战：iOS游戏开发》

图书基本信息

书名：《Lua实战：iOS游戏开发》

13位ISBN编号：9787115331755

10位ISBN编号：7115331758

出版时间：2014-1

出版社：人民邮电出版社

作者：(澳)Jayant Varma

页数：358

译者：麦秆创智

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《Lua实战：iOS游戏开发》

内容概要

lua实战：ios游戏开发》程语言的相关知识，并着重展现了lua在ios平台开发过程中的编程方法。《lua实战：ios游戏开发》从lua语言基础开始，详细地讲解了lua中的变量、运算符、基础函数、数学函数、字符串函数、线程等基本知识，总结了lua开发中的特殊之处。此外，还介绍了corona sdk、gideros studio、moai、l·ve和codea这几个框架，为lua语言在ios游戏开发过程中的使用进行了极佳的铺垫。最后，书中介绍了一些与lua有关的库和第三方应用程序，并通过一个复杂的实例全面地总结了书中的知识。

《lua实战：ios游戏开发》适合对使用lua ios下进行游戏开发感兴趣的读者，也适用于有一定编程基础，想要深入学习lua的开发人员。

《Lua实战：iOS游戏开发》

作者简介

jayant varma是oz apps的执行董事，一家专注于为本地和国际企业提供移动产品增值策略和移动开发及培训方面专业服务的澳大利亚咨询公司。在mac app store上好几款应用都得益于他的功劳。

jayant是麻省理工学院的工商管理学硕士，同时也是一位资深的it人士，在多个国家的自动化和高等教育等领域有超过20年的丰富经验。遵循个人对教育事业的热爱，他在詹姆斯库克大学授课和制订教学大纲。

他的移动开发之路从2000年的一台microsoft pocket pc(compaq ipaq)一直到现在的苹果ios。他对技术和游戏的痴迷可以追溯到他在第一台计算机上编写的游戏——zx spectrum。

jayant曾是acs(澳大利亚计算机学会)的活跃分子，以及当地分会执行董事会的成员。他是首个当地yit(青年it)分会的创始人之一。他创办了auc(苹果大学联盟)的ios开发学习班，非常热心地指导和帮助年轻的开发者们。他还在博客上发布相关技术话题来帮助其他的开发者。

你可以关注jayant的twitter@ozapps，博客<http://howto.oz-apps.com>和<http://reivewme.oz-apps.com>，以及lua博客<http://dev.oz-apps.com>和@learnlua。

书籍目录

《lua实战：ios游戏开发》

第1章 lua入门	1
1.1 什么是lua	1
1.2 lua历史	1
1.3 lua大事记	2
1.4 上手lua	3
1.5 配置lua	4
1.5.1 线上lua shell	4
1.5.2 windows、max os x和*nix	4
1.5.3 ios	4
1.6 lua特性	4
1.6.1 变量	4
1.6.2 用变量实现hello world	5
1.6.3 字符串	6
1.6.4 数字	7
1.6.5 值和类型	8
1.6.6 块和作用域	9
1.7 lua运算符	10
1.7.1 算术运算符	10
1.7.2 关系运算符	10
1.7.3 逻辑运算符	10
1.7.4 其他运算符	11
1.8 总结	11
第2章 lua库	12
2.1 基础函数	12
2.1.1 assert(v[,message])	12
2.1.2 collectgarbage([opt[,arg]])	12
2.1.3 dofile([filename])	13
2.1.4 error(message[,level])	13
2.1.5 _g	13
2.1.6 getfenv([f])	13
2.1.7 getmetatable(object)	13
2.1.8 ipairs(t)	13
2.1.9 load(func[,chunkname])	14
2.1.10 loadstring(string[,chunkname])	14
2.1.11 next(table[,index])	14
2.1.12 pairs(t)	14
2.1.13 pcall(f,arg1,...)	14
2.1.14 print(...)	14
2.1.15 rawequal(v1,v2)	15
2.1.16 rawget(table,index)	15
2.1.17 rawest(table,index,value)	15
2.1.18 select(index,...)	15
2.1.19 setfenv(f,table)	15
2.1.20 setmetatable(table,metatable)	15
2.1.21 tonumber(e[,base])	15
2.1.22 tostring(e)	15

2.1.23	type(v)	16
2.1.24	unpack(list[,i[,j]])	16
2.1.25	_version	16
2.1.26	xpcall(f,err)	16
2.2	系统函数库	17
2.2.1	表函数	17
2.2.2	系统函数	19
2.3	表(tables)概述	21
2.3.1	数组表	21
2.3.2	关联数组表	21
2.4	函数：进阶	21
2.4.1	对象表	22
2.4.2	.和:的区别	23
2.5	小结	25
第3章	文件操作	26
3.1	隐式函数	26
3.1.1	io.close([file])	26
3.1.2	io.flush()	26
3.1.3	io.input([file])	27
3.1.4	io.lines([filename])	27
3.1.5	io.open(filename[,mode])	27
3.1.6	io.output([file])	27
3.1.7	io.read(...)	28
3.1.8	io.tmpfile()	28
3.1.9	io.type(obj)	28
3.1.10	io.write(...)	28
3.2	显式函数	28
3.2.1	file:close()	28
3.2.2	file:flush()	29
3.2.3	file:lines()	29
3.2.4	file.read([format])	29
3.2.5	file.seek([whence][,offset])	29
3.2.6	file.setvbuf(mode[,size])	30
3.2.7	file.write(...)	30
3.3	在游戏中使用文件i/o	30
3.4	保存变量	31
3.5	获取数据	31
3.5.1	代码分析	32
3.5.2	潜在问题	32
3.6	变量存储数据	33
3.6.1	代码分析	33
3.6.2	潜在问题	34
3.7	数据写入文件	35
3.8	存储表	35
3.9	动态变量	36
3.10	小结	37
第4章	lua数学函数	38
4.1	lua数学函数简介	38
4.1.1	math.abs(x)	38

- 4.1.2 math.acos(x) 38
- 4.1.3 math.asin(x) 38
- 4.1.4 math.atan(x) 39
- 4.1.5 math.atan2(y,x) 39
- 4.1.6 math.ceil(x) 39
- 4.1.7 math.cos(x) 39
- 4.1.8 math.cosh(x) 39
- 4.1.9 math.deg(x) 39
- 4.1.10 math.exp(x) 39
- 4.1.11 math.floor(x) 40
- 4.1.12 math.fmod(x,y) 40
- 4.1.13 math.frexp(x) 40
- 4.1.14 math.huge 40
- 4.1.15 math.ldexp(m,e) 40
- 4.1.16 math.log(x) 40
- 4.1.17 math.log10(x) 41
- 4.1.18 math.max(x,...) 41
- 4.1.19 math.min(x,...) 41
- 4.1.20 math.modf(x) 41
- 4.1.21 math.pi 41
- 4.1.22 math.pow(x,y) 41
- 4.1.23 math.rad(x) 41
- 4.1.24 math.random([m[,n]]) 41
- 4.1.25 math.randomseed(x) 42
- 4.1.26 math.sin(x) 42
- 4.1.27 math.sinh(x) 42
- 4.1.28 math.sqrt(x) 42
- 4.1.29 math.tan(x) 42
- 4.1.30 math.tanh(x) 42
- 4.2 在游戏中应用数学函数的实例 42
 - 4.2.1 赋值 43
 - 4.2.2 递增与递减 43
 - 4.2.3 引入坐标点 44
 - 4.2.4 条件分支 44
 - 4.2.5 抛硬币 45
 - 4.2.6 掷骰子 45
 - 4.2.7 使用标记位(flag) 46
 - 4.2.8 多种标记位 47
 - 4.2.9 利用数学函数实现循环 48
 - 4.2.10 在游戏中引入网格 48
 - 4.2.11 移动角色 56
- 4.3 逻辑运算函数 59
- 4.4 小结 61
- 第5章 字符串 62
 - 5.1 什么是字符串 62
 - 5.1.1 string.byte(s[,i[,j]]) 62
 - 5.1.2 string.char(...) 63
 - 5.1.3 string.dump(funtion) 63
 - 5.1.4 string.find(s,pattern[,init[,plain]]) 63

- 5.1.5 string.format(formatstring,...) 63
- 5.1.6 string.gmatch(s,pattern) 64
- 5.1.7 string.gsub(s,pattern,repl[,n]) 64
- 5.1.8 string.len(s) 64
- 5.1.9 string.lower(s) 64
- 5.1.10 string.match(s,patterns[,init]) 64
- 5.1.11 string.rep(s,n) 64
- 5.1.12 string.reverse(s) 65
- 5.1.13 string.sub(s,i[,j]) 65
- 5.1.14 string.upper(s) 65
- 5.2 模式 65
 - 5.2.1 字符类 65
 - 5.2.2 模式项 66
 - 5.2.3 模式 66
 - 5.2.4 捕获 66
- 5.3 使用字符串函数 66
 - 5.3.1 将字符串转化为大写 67
 - 5.3.2 将字符串转化为小写 67
 - 5.3.3 将字符串转化为首字母大写 67
 - 5.3.4 填充字符串 67
 - 5.3.5 csv功能 68
 - 5.3.6 用千位分隔符规范数字的格式 69
 - 5.3.7 字母频率 69
 - 5.3.8 判断回文字符串 70
 - 5.3.9 分割字符串 70
- 5.4 关卡管理 70
- 5.5 小结 72
- 第6章 线程 73
 - 6.1 协同线程 73
 - 6.1.1 协同线程函数 73
 - 6.1.2 创建协同线程 74
 - 6.2 表 77
 - 6.2.1 自定义元表 78
 - 6.2.2 实际例子 83
 - 6.3 lua面向对象 83
 - 6.4 什么是对象 87
 - 6.5 小结 90
- 第7章 技巧提示 91
 - 7.1 通用lua函数 91
 - 7.1.1 使用printf函数 91
 - 7.1.2 计算表中元素的数量 92
 - 7.1.3 使用isempty 92
 - 7.1.4 获取元素索引 92
 - 7.1.5 判断一个表是否是数组 93
 - 7.1.6 设置默认值 93
 - 7.1.7 复制表 94
 - 7.1.8 深度复制 94
 - 7.1.9 复制数组部分 95
 - 7.1.10 复制非数组部分 95

- 7.1.11 合并两个表 95
- 7.1.12 判断表是否包含特定值 95
- 7.1.13 找出table之间的差异 96
- 7.1.14 表作为返回值 96
- 7.1.15 表内元素排序 96
- 7.1.16 判断表中特定项的频度 97
- 7.1.17 转换数字为罗马数字 97
- 7.1.18 创建链表 98
- 7.1.19 扩展标记变量 99
- 7.1.20 字符串补零 100
- 7.1.21 获取字符类型的月份 100
- 7.1.22 像数组一样访问字符串 101
- 7.1.23 测量一维或二维中两点间距 101
- 7.1.24 判断两点的夹角 101
- 7.1.25 限定数字范围 101
- 7.1.26 线性插值法 102
- 7.1.27 获取值的标识符 102
- 7.2 碰撞 103
 - 7.2.1 使用ispointInrect 103
 - 7.2.2 使用pointIncircle 103
 - 7.2.3 判断一个矩形是否包含另一个矩形 104
 - 7.2.4 判断一个圆形是否包含另一个圆形 104
 - 7.2.5 识别重叠矩形 104
 - 7.2.6 识别重叠圆形 105
 - 7.2.7 判断一个圆形和一个矩形是否重叠 105
 - 7.2.8 使用pointIntriangle 106
 - 7.2.9 使用pointInpolygon 106
- 7.3 其他通用函数 107
 - 7.3.1 布尔值比较 107
 - 7.3.2 将c/java循环代码转换为lua 107
 - 7.3.3 应用物体摩擦力 108
 - 7.3.4 模拟一个玩偶盒 108
 - 7.3.5 正弦滚动条 109
 - 7.3.6 在面板的单行或单列放置唯一图块 109
 - 7.3.7 利用数组输出大规模模板文本 110
- 7.4 参数处理 111
 - 7.4.1 固定参数 111
 - 7.4.2 动态参数 112
 - 7.4.3 动态名称参数 112
 - 7.4.4 使用可变参数 112
 - 7.4.5 解析传入参数列表 113
 - 7.4.6 生成只读table 115
 - 7.4.7 实现堆栈 116
 - 7.4.8 参数和表之间的转换 117
- 7.5 二维矢量 118
- 7.6 小结 120
- 第8章 corona sdk 121
 - 8.1 配置corona sdk 121
 - 8.2 corona sdk工作原理 121

- 8.3 corona sdk helloworld 122
 - 8.3.1 图像版本 123
 - 8.3.2 hello world 124
 - 8.3.3 设备显示hello world 125
- 8.4 不仅仅是hello world：在屏幕上创建一个矩形 126
 - 8.4.1 群组 128
 - 8.4.2 图像 130
 - 8.4.3 事件驱动 130
 - 8.4.4 触屏输入 131
 - 8.4.5 物理特性 132
- 8.5 一切都是计时 134
 - 8.5.1 帧 135
 - 8.5.2 实现生命条 136
- 8.6 利用enterframe实现动画 138
 - 8.6.1 再说生命条 139
 - 8.6.2 使用transition函数 140
 - 8.6.3 移除屏幕中的对象 145
- 8.7 制作音效 145
 - 8.7.1 发出声音 145
 - 8.7.2 加工声音 146
- 8.8 视频处理 148
- 8.9 实现一个电梯 149
 - 8.9.1 放大视图 150
 - 8.9.2 其他事件 153
 - 8.9.3 使用地图 155
- 8.10 浏览器 156
- 8.11 对象回收 157
- 8.12 企业版本 158
- 8.13 小结 158
- 第9章 gideros studio 159
 - 9.1 许可 159
 - 9.2 安装gideros studio 159
 - 9.2.1 包含内容 160
 - 9.2.2 设置许可 160
 - 9.2.3 第一步 161
 - 9.2.4 运行代码 162
 - 9.2.5 配置项目 163
 - 9.3 架构 165
 - 9.4 gideros引擎 165
 - 9.4.1 hello小鸟 166
 - 9.4.2 对齐图像 166
 - 9.4.3 群组 167
 - 9.4.4 显示文字 168
 - 9.4.5 绘制形状 170
 - 9.4.6 应用程序对象 174
 - 9.4.7 添加音效 176
 - 9.4.8 事件 176
 - 9.4.9 动画 181
 - 9.4.10 局域网和网络 182

- 9.4.11 gps和指南针 183
- 9.4.12 加速计 184
- 9.4.13 陀螺仪 184
- 9.4.14 物理效果 185
- 9.5 插件 186
- 9.6 小结 188
- 第10章 moai 190
 - 10.1 什么是moai 190
 - 10.2 获取moai 190
 - 10.3 moai sdk 191
 - 10.4 hello moai 191
 - 10.5 显示文字 194
 - 10.5.1 truetype字体 194
 - 10.5.2 位图字体 195
 - 10.5.3 文本属性 196
 - 10.6 绘制矢量图元 197
 - 10.6.1 绘制直线 197
 - 10.6.2 绘制矩形 197
 - 10.6.3 绘制填充矩形 199
 - 10.6.4 绘制圆 200
 - 10.6.5 绘制填充圆 200
 - 10.6.6 绘制椭圆 200
 - 10.6.7 绘制填充椭圆 200
 - 10.6.8 绘制多边形 201
 - 10.6.9 绘制点 201
 - 10.6.10 绘制属性 202
 - 10.7 绘制图像 202
 - 10.7.1 绘制自定义图像 203
 - 10.7.2 载入图像 203
 - 10.7.3 copying images 204
 - 10.7.4 保存图像 204
 - 10.7.5 修改图像尺寸 204
 - 10.7.6 图像的像素提取 204
 - 10.8 动画 205
 - 10.9 栅格deck 206
 - 10.10 线程 209
 - 10.11 组 210
 - 10.12 处理输入 212
 - 10.12.1 键盘事件 212
 - 10.12.2 鼠标事件 213
 - 10.12.3 触控事件 216
 - 10.13 声音 217
 - 10.14 显示对话框 217
 - 10.15 播放视频 218
 - 10.16 设备旋转 218
 - 10.17 通知 218
 - 10.18 网络 219
 - 10.19 使用json 221
 - 10.20 使用base64 221

- 10.21 压缩数据 222
- 10.22 物理特性 223
 - 10.22.1 box2d物理功能 223
 - 10.22.2 chipmunk物理功能 226
- 10.23 moai云 228
 - 10.23.1 创建web服务 228
 - 10.23.2 使用web服务 230
- 10.24 小结 231
- 第11章 I?ve 232
 - 11.1 架构 232
 - 11.2 安装I?ve 233
 - 11.3 运行I?ve 233
 - 11.4 回调函数 234
 - 11.4.1 love.load() 234
 - 11.4.2 love.update(dt) 234
 - 11.4.3 love.draw() 234
 - 11.4.4 love.mousepressed(x,y,button) 234
 - 11.4.5 love.mousereleased(x,y,button) 234
 - 11.4.6 love.keypressed(key,unicode) 234
 - 11.4.7 love.keyreleased(key) 235
 - 11.4.8 love.focus() 235
 - 11.4.9 love.quit() 235
 - 11.5 I?ve命名空间 235
 - 11.5.1 love.audio 235
 - 11.5.2 love.event 235
 - 11.5.3 love.filesystem 236
 - 11.5.4 love.font 236
 - 11.5.5 love.graphics 236
 - 11.5.6 love.image 236
 - 11.5.7 love.joystick 236
 - 11.5.8 love.mouse 237
 - 11.5.9 love.physics 237
 - 11.5.10 love.sound 237
 - 11.5.11 love.thread 237
 - 11.5.12 love.timer 237
 - 11.6 图形模块 237
 - 11.6.1 图像 237
 - 11.6.2 移动 240
 - 11.6.3 活动窗口 241
 - 11.6.4 使用键盘控制移动 241
 - 11.6.5 旋转对象 245
 - 11.6.6 绘制图元 246
 - 11.7 应用程序设置——conf.lua 251
 - 11.8 创建特效 252
 - 11.9 物理 254
 - 11.10 显示文本 256
 - 11.11 着色器 257
 - 11.12 制造声音 259
 - 11.13 示例游戏代码 260

11.14	小结	265
第12章	codea	266
12.1	获取codea	266
12.2	codea的架构	267
12.3	hello world	267
12.4	参数	268
12.5	使用codea进行绘制	269
12.5.1	设备旋转	269
12.5.2	系统键盘	270
12.5.3	绘制模式	270
12.5.4	背景颜色	270
12.5.5	画笔颜色	271
12.5.6	填充色	271
12.5.7	线宽	271
12.5.8	绘制直线	271
12.5.9	抗锯齿	271
12.5.10	绘制圆和椭圆	272
12.5.11	绘制矩形	273
12.5.12	绘制文本	273
12.6	显示图片	274
12.6.1	屏幕外绘制	275
12.6.2	在图片中载入精灵	275
12.6.3	保存图片	275
12.6.4	保存数据	275
12.6.5	“无色”绘制	276
12.6.6	变换函数	276
12.7	录制视频	277
12.8	触控	277
12.8.1	使用触控进行绘制	278
12.8.2	多点触控	279
12.9	制造声音	280
12.10	ios硬件	281
12.11	下载数据	283
12.12	显示网页	284
12.13	物理特性	284
12.14	矢量计算	286
12.14.1	anglebetween	286
12.14.2	cross	286
12.14.3	dist	286
12.14.4	distsqr	286
12.14.5	dot	286
12.14.6	len	286
12.14.7	lensqr	287
12.14.8	normalize	287
12.14.9	rotate	287
12.14.10	rotate90	287
12.15	小结	287
第13章	库	288
13.1	particle candy	288

示例代码	289
13.2 text candy	290
示例代码	291
13.3 widget candy	291
示例代码	292
13.4 corona autolan	293
13.5 corona profiler	293
示例代码	294
13.6 director	294
示例代码	295
13.7 lime	295
示例代码	295
13.8 rapanui	295
示例代码	296
13.9 gideros illustrator(svg library)	296
示例代码	297
13.10 tnt particles library	297
示例代码	297
13.11 busted	297
示例代码	298
13.12 moses	298
示例代码	298
13.13 allen	299
示例代码	299
13.14 bhwx	299
示例代码	299
13.15 小结	301
第14章 第三方应用程序	302
14.1 ide	302
14.1.1 glider	302
14.1.2 coronacomplete	303
14.1.3 lua studio	304
14.1.4 zerobrane studio	305
14.2 文本编辑器	306
14.2.1 notepad++	306
14.2.2 textmate	307
14.2.3 textwrangler	308
14.2.4 bbedit	308
14.2.5 sublime text 2	309
14.2.6 textastic代码编辑器	310
14.3 片段收集器	311
14.3.1 codebox	312
14.3.2 code collector pro	312
14.4 版本控制	313
14.5 位图编辑器	314
14.5.1 pixen	314
14.5.2 graphicsgale	315
14.5.3 pixelmator	316
14.5.4 axialis iconworkshop	317

- 14.5.5 inkscape 317
- 14.5.6 paint.net 318
- 14.5.7 iconify 319
- 14.6 精灵动画和层级创建工具 320
 - 14.6.1 spritehelper 320
 - 14.6.2 levelhelper 320
 - 14.6.3 tiled 321
 - 14.6.4 texturepacker 322
 - 14.6.5 physicseditor 323
 - 14.6.6 particle designer 324
 - 14.6.7 glyph designer 325
 - 14.6.8 spriteloq 326
 - 14.6.9 zo? 327
 - 14.6.10 tnt animator studio 328
- 14.7 音频工具 328
 - 14.7.1 audacity 328
 - 14.7.2 bfxr 329
 - 14.7.3 nodebeat 330
- 14.8 小结 331
- 第15章 示例代码 332
 - 15.1 图形 332
 - 15.2 搭建环境 333
 - 15.3 让直升机飞起来 334
 - 15.4 使用加速度仪 336
 - 15.5 移动直升机 336
 - 15.6 发出声音 338
 - 15.7 开枪 339
 - 15.8 让子弹飞 340
 - 15.9 敌军降临 341
 - 15.10 移动场景物品 344
 - 15.11 挂了 345
 - 15.12 上点颜色 347
 - 15.13 显示信息 348
 - 15.14 game over , 再来一遍 350
 - 15.15 碰撞 352
 - 15.15.1 射击飞机和气球 354
 - 15.15.2 营救与加油 356
 - 15.16 放到一起 357
 - 15.17 小结 358

《Lua实战：iOS游戏开发》

精彩短评

- 1、真的很不错的书,对于想学习lua尝试用lua开发ios游戏的同学可以看看
- 2、不管本书好坏，中文版的lua+oc编程只有这一本书，想入门就买吧

精彩书评

1、有一行判断小球是否超出屏幕范围的判断是if newX > dimensions[3] or newX < dimensions[0] then speed.x = - speed.x ;endif newY > dimensions[4] or newY < dimensions[1] then speed.y = - speed.y ;end这里的屏幕是把坐标放在一个数组里local dimensions = {0,0,320,480};这样来做的 作者一定是 C++出身 因为C++中才是index以0开始但是Lua中都是table 都是以1开始的所以if newX > dimensions[3] or newX < dimensions[1] then speed.x = - speed.x ;endif newY > dimensions[4] or newY < dimensions[2] then speed.y = - speed.y ;end估计是手误 希望初学者不要对此产生疑惑

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：www.tushu000.com