

触控未来官方教材

触控未来 CEO 李志远作序

Cocos 引擎创始人王哲作序



Cocos2d-x

学习笔记



完全掌握 Lua API 与游戏项目开发

Cocos2d-x Learning Notes: the Definitive Guide to Lua API and Game Projects Development

赵志荣 关东升 著

Zhao Zhirong Guan Dongsheng

清华大学出版社





Cocos2d-x

学习笔记



完全掌握 Lua API 与游戏项目开发

Cocos2d-x Learning Notes: the Definitive Guide to Lua API and Game Projects Development

赵志荣 关东升 著

Zhao Zhirong Guan Dongsheng

清华大学出版社

北 京

内 容 简 介

本书系统论述了 Cocos2d-x Lua API 游戏编程和开发技术,内容涵盖 Cocos2d-x 中的核心类、瓦片地图、物理引擎和 AudioEngine 音频引擎等知识。全书分为 16 章:准备开始、Lua 语言基础、Cocos2d-x Lua API 开发环境搭建、Cocos2d-x 引擎与 Lua 绑定、游戏中的文字、菜单、精灵、场景与层、动作和动画、用户事件、AudioEngine 音频引擎、粒子系统、瓦片地图、物理引擎、Cocos2d-x 多分辨率屏幕适配,以及手机游戏项目实战。

本书适合作为普通高校计算机、动漫设计、数字媒体等相关专业的游戏开发课程的教材,也适合作为手机游戏开发培训机构的培训教材及广大手机游戏开发者的自学参考用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Cocos2d-x 学习笔记:完全掌握 Lua API 与游戏项目开发/赵志荣,关东升著.--北京:清华大学出版社,2016

(未来书库)

ISBN 978-7-302-43204-3

I. ①C… II. ①赵… ②关… III. ①移动电话机—游戏程序—程序设计 ②便携式计算机—游戏程序—程序设计 IV. ①TN929.53 ②TP311.5 ③TP368.32

中国版本图书馆 CIP 数据核字(2016)第 037584 号

责任编辑:盛东亮

封面设计:范华明

责任校对:胡伟民

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:清华大学印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:186mm×240mm 印 张:19.25 插 页:1 字 数:490 千字

版 次:2016 年 5 月第 1 版 印 次:2016 年 5 月第 1 次印刷

印 数:1~2500

定 价:49.00 元

未来书库 编审委员会

(按姓氏拼音顺序)

主	编	李志远		
编	审	委	员	
		陈	纾	关东升
		李	宁	刘克男
		王	铮	王瑞锦
丛	书	责	任	编
编	辑	盛	东	亮

丛书序

FOREWORD

在策划“未来书库”这套丛书时,我有一个愿望,就是希望在泛 IT 行业知识快速更新的时代,可以为读者提供一套规范、专业、系统的学习资料。我们应该都有这样的体验,在互联网时代,各类学习资料可以用海量来形容,而且更新变化十分快,随之而来的体验是遴选困难、无所适从。为此,我们邀请并组织了业内各个领域的专业人士共同策划出版这套图书。

“未来书库”定位于泛 IT 行业的新技术。力求为广大读者,包括广大高校师生提供一套适应时代需求、符合技术发展潮流的专业书籍。这套书将涵盖计算机技术、互联网技术、移动开发技术、游戏开发技术……从零基础到进阶提升,都有所涉及,适合读者系统学习,并适合作为高校相关课程的教材。

在此要感谢“未来书库”的各位作者,是各位作者的鼎力支持,促成了这套图书的出版。各位作者将自己多年的工作经验和专业知

识进行了系统整理,并无私分享。这些作者,很多也都是多年的朋友和同事,在繁忙的工作之余,对于丛书的编写和审校付出了大量的心血,确保了丛书中每一本图书的优良品质。

当然,也要感谢清华大学出版社的各位朋友,正是你们和作者的共同努力,才使得“未来书库”得以持续完成,并以一流的装帧和出版水平呈现给读者。泛 IT 行业的技术和知识更新很快,除了我们持续进行内容的增加和版本更新以外,我们也非常希望国内外专家、从业者和广大的读者为这套图书建言献策,开发出更多优秀图书产品。同时,也欢迎更多的朋友和同行加入我们的丛书作者行列,共同为广大读者分享开发经验、分享新技术,传道授业。



触控未来 CEO

序

FOREWORD

经过短短数年的发展,在刚刚过去的 2015,中国移动游戏市场实际销售收入已经达到了 514.6 亿,这个看似疯狂的数字又一次验证了移动游戏市场的飞速增长。虽说这个市场已经变得愈发复杂多样,但不断涌现的现象级产品依然屡次刷新着整个行业的认知。

从 2010 年发布第一个版本开始,Cocos2d-x 很幸运地成为了手游行业飞速发展的见证者,也是一直坚守在行业底层技术领域的支持者。

在过去数年间,基于 Cocos 引擎诞生了大量的标杆产品,直到现在——App Annie 畅销排行前 10 的产品中,Cocos 一直占据着 50% 的份额。市场和开发者的认可,是对 Cocos 引擎团队多年来的坚持和专注最好的回馈。

一直以来,Cocos 引擎团队始终秉承开源的精神,致力于为开发者提供更优质的开发体验,带来更高的开发效率。在不断推出的新功能、新工具中,也为众多 Cocos 开发者带来了回报,帮助整个行业产出了更大的价值。

关东升与赵志荣老师合作的这套“Cocos2d-x 学习笔记”系列图书,非常详细地介绍了 Cocos2d-x 所支持的三个语言 C++、Lua 和 JavaScript 的 API 使用,覆盖了核心类、数据结构、物理引擎、内存管理等各个方面,从编程语言准备、环境搭建开始,到精灵、对象、场景、层等细节以及动作、特效、动画等动态特性的完善和处理,都有很详尽的解释,并通过实战项目进行验证,具有非常高的实用性和参考价值,非常值得阅读。

希望本系列图书可以很好地完成它们的使命,使得教育和游戏产业的结合更加紧密,帮助越来越多的想要加入 Cocos 开发者阵营的新人步入游戏开发者的行列,也为已经深耕在这个行业的众多开发者提供更有价值的参考。



Cocos 引擎创始人

前言

PREFACE

手机游戏产业的发展越来越火热,很多公司推出了自己的游戏引擎,北京触控科技有限公司的 Cocos2d-x 游戏引擎就是其中的佼佼者,它的优势在于实现了“一个平台开发、多个平台发布”。截止到 2016 年初,Cocos2d-x 游戏引擎在全球 199 个国家和地区有超过 40 万开发者使用,已经成为全球使用率最高的手机游戏引擎之一。

为了推动 Cocos2d-x 游戏引擎在我国高校与相关行业的应用与普及,提高相关专业人才培养及游戏产业发展的水平,进一步提高普通高校计算机、动漫设计、数字媒体等专业游戏类课程的教学质量,满足各大高校不断增长的人才培养需求、教学改革与课程改革要求,北京触控科技有限公司与清华大学出版社立项开展了《未来书库》的建设工作。我有幸受触控教育部门之邀,专门为广大高校师生、培训机构及有志于从事 Cocos2d-x 游戏开发的读者撰写此书。

Cocos2d-x 游戏引擎提供了三种语言(C++/JavaScript/Lua)的 API,开发者可以根据自己的技术背景选择不同语言。本书专门论述 Lua 语言 API,并以游戏项目开发的实例,抛砖引玉,教会读者动手实践。

关于源代码

为了更好地为广大读者提供服务,我们专门为本书建立了一个网站 <http://www.51work6.com/book/cocos3.php>,大家可以查看相关出版进度,并对书中内容发表评论,提出宝贵意见。

勘误与支持

我们在网站 <http://www.51work6.com/book/cocos3.php> 中建立了一个勘误专区,及时地把书中的问题、失误和纠正反馈给广大读者。如果您在阅读本书过程中,发现了任何问题,可以在网上留言,可以发送电子邮件到 eorient@sina.com,也可以在新浪微博中与我们联系: @tony_关东升。我们会在第一时间回复您。

本书主要由赵志荣、关东升执笔撰写。此外,智捷课堂的赵浩丞、赵大羽、关锦华也参与了部分内容的编写工作。在此感谢触控教育部门的李志远和清华大学出版社的盛东亮给我们提供了宝贵的意见。感谢我的家人容忍我的忙碌,以及对我的关心和照顾,使我能抽出这么多时间,投入全部精力地专心编写此书。

由于时间仓促,书中难免存在不妥之处,请读者谅解,并提出宝贵意见。

赵志荣 关东升

2016 年 1 月于北京

目录

CONTENTS

丛书序	III
序	V
前言	VII
第 1 章 准备开始	1
1.1 使用实例代码	1
1.2 图示的约定	1
1.2.1 图中的箭头	1
1.2.2 图中的手势	2
1.2.3 图中的圈框	2
1.2.4 类图	2
第 2 章 Lua 语言基础	4
2.1 Lua 开发环境搭建	4
2.1.1 安装 LDT 工具	4
2.1.2 HelloLua 实例测试	8
2.2 标识符和保留字	12
2.2.1 标识符	13
2.2.2 保留字	13
2.3 常量和变量	13
2.3.1 常量	13
2.3.2 变量	14
2.3.3 命名规范	14
2.4 注释	15
2.5 Lua 数据类型	16
2.5.1 数据类型	16

2.5.2	type 函数	17
2.5.3	数据类型转换	17
2.6	运算符	18
2.6.1	算术运算符	18
2.6.2	关系运算符	20
2.6.3	逻辑运算符	21
2.6.4	运算优先级	22
2.7	控制语句	22
2.7.1	分支语句	22
2.7.2	循环语句	24
2.7.3	跳转语句	27
2.8	table 类型	27
2.8.1	字典	28
2.8.2	数组	29
2.9	字符串类型	30
2.9.1	字符串截取	30
2.9.2	字符串转换	31
2.9.3	字符串查询	31
2.9.4	字符串格式化	32
2.10	函数	33
2.10.1	使用函数	33
2.10.2	变量作用域	34
2.10.3	多重返回值	34
2.11	闭包函数	35
2.11.1	嵌套函数	35
2.11.2	返回函数	36
2.11.3	使用闭包表达式	37
2.12	Lua 中的面向对象	38
2.12.1	Lua 中的对象	38
2.12.2	类的实现	39
	本章小结	40
第 3 章	Cocos2d-x Lua API 开发环境搭建	41
3.1	搭建环境	41
3.1.1	Cocos 引擎下载安装	41
3.1.2	Cocos Framework 下载和安装	41

3.2 Cocos Code IDE 开发工具	44
3.2.1 安装 Cocos Code IDE	44
3.2.2 配置 Cocos Code IDE	46
3.2.3 使用 Cocos Code IDE 调试功能	48
本章小结	51
第 4 章 Cocos2d-x 引擎与 Lua 绑定	52
4.1 Cocos2d 家谱	52
4.1.1 Cocos2d-x 引擎	53
4.1.2 Cocos2d-x 绑定 Lua	54
4.2 第一个 Cocos2d-x Lua 绑定游戏	54
4.2.1 创建工程	54
4.2.2 Cocos Code IDE 中运行	57
4.2.3 工程文件结构	57
4.2.4 重构 HelloLua 案例	58
4.3 Cocos2d-x 核心概念	60
4.3.1 导演	60
4.3.2 场景	61
4.3.3 层	61
4.4 Node 与 Node 层级架构	62
4.4.1 Node 中重要的操作	63
4.4.2 Node 中重要的属性	64
4.4.3 游戏循环与调度	67
4.5 Cocos2d-x 坐标系	68
4.5.1 UI 坐标	68
4.5.2 OpenGL 坐标	69
4.5.3 世界坐标和模型坐标	70
本章小结	74
第 5 章 游戏中的文字	75
5.1 使用标签	75
5.1.1 Label 类	76
5.1.2 Label 类实例	77
5.2 位图字体制作	79
5.2.1 Glyph Designer 工具	80
5.2.2 使用 Glyph Designer 制作位图字体	80

本章小结	84
第 6 章 菜单	85
6.1 菜单相关类	85
6.2 文本菜单	86
6.3 精灵菜单和图片菜单	87
6.4 开关菜单	91
本章小结	93
第 7 章 精灵	94
7.1 Sprite 精灵类	94
7.1.1 创建 Sprite 精灵对象	94
7.1.2 实例：使用纹理对象创建 Sprite 对象	95
7.2 精灵的性能优化	97
7.2.1 使用纹理图集	97
7.2.2 使用精灵帧缓存	100
7.3 纹理图集制作	102
7.3.1 TexturePacker 工具	102
7.3.2 使用 TexturePacker 制作纹理图集	104
本章小结	107
第 8 章 场景与层	108
8.1 场景与层的关系	108
8.2 场景切换	108
8.2.1 场景切换相关函数	108
8.2.2 场景过渡动画	113
8.3 场景的生命周期	114
8.3.1 生命周期函数	114
8.3.2 多场景切换生命周期	117
本章小结	118
第 9 章 动作和动画	119
9.1 动作	119
9.1.1 瞬时动作	120
9.1.2 间隔动作	124
9.1.3 组合动作	129

9.1.4 动作速度控制	133
9.1.5 函数调用	136
9.2 特效	138
9.2.1 网格动作	139
9.2.2 实例：特效演示	140
9.3 动画	142
9.3.1 帧动画	142
9.3.2 实例：帧动画使用	143
本章小结	145
第 10 章 用户事件	146
10.1 事件处理机制	146
10.1.1 事件分发器	147
10.1.2 触摸事件	148
10.1.3 实例：单点触摸事件	150
10.1.4 实例：多点触摸事件	153
10.1.5 键盘事件	156
10.2 加速度计与加速度计事件	157
10.2.1 加速度计	157
10.2.2 加速度计事件	158
10.2.3 实例：运动的小球	159
本章小结	160
第 11 章 AudioEngine 音频引擎	161
11.1 Cocos2d-x 中音频文件	161
11.1.1 音频文件介绍	161
11.1.2 Cocos2d-x 跨平台音频支持	162
11.2 使用 AudioEngine 引擎	163
11.2.1 音频文件的预处理	163
11.2.2 播放背景音乐	164
11.2.3 停止播放背景音乐	166
11.3 实例：设置背景音乐与音效	168
11.3.1 GameScene 场景实现	168
11.3.2 SettingScene 场景实现	171
本章小结	173

第 12 章 粒子系统	174
12.1 问题的提出	174
12.2 粒子系统基本概念	175
12.2.1 实例：打火机	175
12.2.2 粒子发射模式	177
12.2.3 粒子系统属性	177
12.3 Cocos2d-x 内置粒子系统	180
12.3.1 内置粒子系统	180
12.3.2 实例：内置粒子系统	180
12.4 自定义粒子系统	183
12.4.1 代码创建	183
12.4.2 plist 文件创建	186
本章小结	189
第 13 章 瓦片地图	190
13.1 地图性能问题	190
13.2 Cocos2d-x Lua 中瓦片地图 API	191
13.3 使用 Tiled 地图编辑器	193
13.3.1 新建地图	197
13.3.2 导入瓦片集	199
13.3.3 创建层	200
13.3.4 在普通层上绘制地图	201
13.3.5 在对象层上添加对象	204
13.4 实例：忍者无敌	206
13.4.1 设计地图	207
13.4.2 程序中加载地图	208
13.4.3 移动精灵	209
13.4.4 检测碰撞	211
13.4.5 滚动地图	215
本章小结	218
第 14 章 物理引擎	219
14.1 使用物理引擎	220
14.1.1 物理引擎核心概念	220
14.1.2 物理引擎与精灵关系	221

14.2	Cocos2d-x 中物理引擎	221
14.2.1	Cocos2d-x 物理引擎 Lua API	221
14.2.2	实例: HelloPhysicsWorld	225
14.2.3	实例: 接触与碰撞检测	228
14.2.4	实例: 使用关节	232
	本章小结	233
第 15 章	多分辨率屏幕适配	234
15.1	屏幕适配问题的提出	234
15.2	Cocos2d-x 屏幕适配	235
15.2.1	三种分辨率	235
15.2.2	分辨率适配策略	238
15.2.3	纹理图集资源适配	241
15.2.4	瓦片地图资源适配	242
	本章小结	242
第 16 章	敏捷开发项目实战——迷失航线手机游戏	243
16.1	迷失航线游戏分析与设计	243
16.1.1	迷失航线故事背景	243
16.1.2	需求分析	243
16.1.3	原型设计	244
16.1.4	游戏脚本	245
16.2	任务 1: 游戏工程的创建与初始化	246
16.2.1	迭代 1.1: 创建工程	246
16.2.2	迭代 1.2: 添加资源文件	247
16.2.3	迭代 1.3: 添加常量文件 SystemConst.lua	248
16.2.4	迭代 1.4: 多分辨率支持	251
16.3	任务 2: 创建 Loading 场景	253
16.3.1	迭代 2.1: 添加场景和层	253
16.3.2	迭代 2.2: Loading 动画	255
16.3.3	迭代 2.3: 异步加载纹理缓存	256
16.4	任务 3: 创建 Home 场景	257
16.4.1	迭代 3.1: 添加场景和层	257
16.4.2	迭代 3.2: 添加菜单	259
16.5	任务 4: 创建设置场景	260
16.6	任务 5: 创建帮助场景	262

16.7	任务 6: 游戏场景实现	263
16.7.1	迭代 6.1: 创建敌人精灵	263
16.7.2	迭代 6.2: 创建玩家飞机精灵	269
16.7.3	迭代 6.3: 创建炮弹精灵	271
16.7.4	迭代 6.4: 初始化游戏场景	272
16.7.5	迭代 6.5: 游戏场景菜单实现	277
16.7.6	迭代 6.6: 玩家飞机发射炮弹	280
16.7.7	迭代 6.7: 炮弹与敌人的接触检测	281
16.7.8	迭代 6.8: 玩家飞机与敌人的接触检测	285
16.7.9	迭代 6.9: 玩家飞机生命值显示	287
16.7.10	迭代 6.10: 显示玩家得分情况	288
16.8	任务 7: 游戏结束场景	288
	本章小结	291

第 1 章



准备开始

当你拿到这本书的时候,你会说:“哇!这么厚!我应该怎么开始呢?”本章我们不讨论技术,而是告诉读者本书的结构,书中的一些约定,以及如何使用本书的案例。

1.1 使用实例代码

本书作为一本介绍编程技术的书,书中有很多实例代码。读者下载本书代码并解压缩,可以看到下面的目录结构:

```
ch2
└─ ch4
   └─ 4.2
      └─ 4.4.2
         └─ 4.4.3
            └─ HelloLua
               └─ 4.5.3
                  └─ ch5
                     └─ ch6
                        └─ ch7
                           └─ ch8
                              └─ ch9
                                 └─ ch10
                                    └─ ch11
                                       └─ ch12
                                          └─ ch13
                                             └─ ch14
                                                └─ ch15
                                                   └─ ch16
                                                      └─ LostRoutes
                                                         └─ 设计资料
                                                            └─ LuaTemplate
```

图 1-1 实例代码目录结构

图中的 ch4 表示第 4 章代码,在 ch 目录下一般是各个小节的内容,例如:4.4.3 表示第 4.4.3 节,在小节目录下 HelloLua 是该节的工程代码,需要使用 Cocos Code IDE 工具打开。LuaTemplate 目录是本书使用的工程模板。

1.2 图示的约定

为了更形象地说明知识点或描述操作,本书添加了很多图示,下面简要说明图示中一些符号的含义。

1.2.1 图中的箭头

如图 1-2 所示,箭头用于说明用户的动作,一般箭尾是动作开始的地方,箭头指向动作

结束的地方。

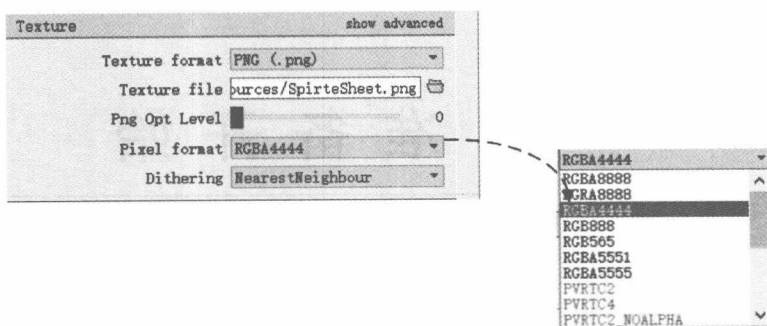


图 1-2 图中箭头

1.2.2 图中的手势

为了描述操作,我们在图中放置了^①等手势符号,这说明单击了该处的按钮,如图 1-3 所示。

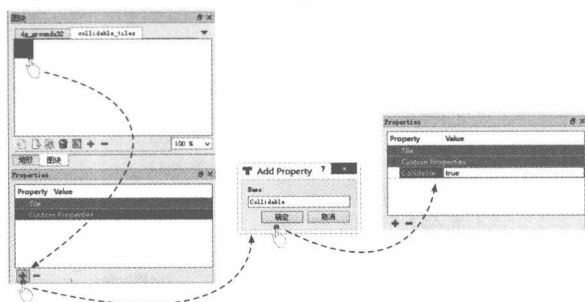


图 1-3 图中手势

1.2.3 图中的圈框

有时读者会看到如图 1-4 所示的圈框,其中的内容是选中的内容或重点要说明的内容。

1.2.4 类图

为了剖析和解释代码,书中还是用到了一些 UML^① 图,其中使用最多的是类图,它描述

① Unified Modeling Language (UML) 又称统一建模语言或标准建模语言,是始于 1997 年的一个 OMG 标准。它是一个支持模型化和软件系统开发的图形化语言,为软件开发的所有阶段提供模型化和可视化支持,包括由需求分析到规格,到构造和配置。面向对象的分析与设计(OOA&D, OOAD)方法的发展在 20 世纪 80 年代末至 90 年代中出现了—个高潮, UML 是这个高潮的产物。它不仅统一了 Booch、Rumbaugh 和 Jacobson 的表示方法,而且对其作了进一步的发展,并最终统一为大众所接受的标准建模语言。——引自于百度百科 <http://baike.baidu.com/view/174909.htm>